UNIVERSIDAD POLITÉCNICA SALESIANA

FACULTAD DE INGENIERÍAS

SEDE QUITO – CAMPUS SUR

CARRERA DE INGENIERÍA DE SISTEMAS

MENCIÓN TELEMÁTICA

ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SOFTWARE PROTOTIPO DE FIREWALL Y SERVIDOR PROXY MULTIPLATAFORMA CON TECNOLOGÍA JAVA.

TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE INGENIERO DE SISTEMAS

LUCÍA GUADALUPE ARÉVALO GALÁRRAGA

GUSTAVO ALEXANDER VACA TELLO

DIRECTOR: ING. RAFAEL JAYA

Quito, junio 2010

DECLARACIÓN

Nosotros, Lucía Guadalupe Arévalo Galárraga y Gustavo Alexander Vaca Tello, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración, cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Universidad Politécnica Salesiana, según lo establecido por la Ley de Propiedad Intelectual, por su reglamento y por la normativa institucional vigente.

_____ _____

Lucía Arévalo Galárraga Gustavo Vaca Tello

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Lucía Guadalupe Arévalo Galárraga y Gustavo Alexander Vaca Tello, bajo mi dirección.

_____

Ing. Rafael Jaya

Director de Tesis

DEDICATORIA

A la luz de mi vida que desde el cielo me brindo la fortaleza para seguir adelante y que siempre quiso estar conmigo guiando mis pasos para ti mi querida abuelita Paulina que mientras estuviste conmigo siempre me querías ver triunfar.

Como olvidar a mis queridos padres que con mucho sacrificio me brindaron su eterno apoyo y comprensión y me convirtieron en el hombre que ahora soy.

Siempre existirá una luz en mi vida que se impregnara en mi corazón y esa luz se llamara eternamente, Nancy, Gustavo, Paulina y Juan Genaro para ustedes con mucho amor.

Gustavo

A mis amados padres, quienes con su amor, esfuerzo y dedicación han sido el soporte fundamental de todas y cada una de las etapas de mi vida. A mi amado padre, que es el ejemplo más significativo de dedicación y constancia, porque él ha sido mi inspiración y mi modelo a seguir. A mi adorada madre, quien siempre ha estado junto a mí en todo momento y con sus cuidados y enseñanzas; me ha formado como una mujer integra y de valores.

A mis queridos hermanos, que son mi motivación y han compartido junto a mí el amor, el cariño, los sueños; pero sobre todo el calor de una verdadera familia.

A mis abuelitos, que han sido mis segundos padres y que han fomentado en mí el ejemplo de lucha, esfuerzo y trabajo duro para alcanzar los objetivos anhelados.

Lucía

AGRADECIMIENTOS

A mi Hermana, mis sobrinos que en tiempos difíciles fueron mi fortaleza, a mis queridos maestros que con sabiduría me guiaron por la senda del aprendizaje.

A mis amigos que durante el largo trayecto hicieron de las tempestades pequeños aguaceros; a todos aquellos quienes confiaron en mí y aportaron de alguna manera para mi desarrollo humano e intelectual.

No quiero terminar, sin antes agradecer a mi Tutor quien con ahincó y paciencia ayudó a la culminación exitosa de este Proyecto.

Gustavo

A Dios y a la Virgen por darme salud, fortaleza y bendiciones para terminar esta etapa del camino.

A mis padres por ser el apoyo incondicional en cada momento, por todas sus enseñanzas y por el esfuerzo que han hecho para forjarme como profesional.

A mí estimado director de tesis Ing. Rafael Jaya, por confiar en mis capacidades y ser un excelente guía en mi vida universitaria.

A mi amigo y compañero Gustavo, gracias por caminar junto a mí durante todos estos años de estudio, por compartir conmigo un sueño y porque ahora lo estamos haciendo realidad.

En fin, son muchas las personas especiales a las que me gustaría agradecer su amistad, apoyo, ánimo y compañía en las diferentes etapas de mi vida. Algunas están aquí conmigo y otras en mis recuerdos pero sobre todo en mi corazón. Sin importar en dónde estén quiero darles las gracias por formar parte de mí, por todo lo que me han brindado siempre ocuparán un lugar en mis pensamientos.

Lucía

CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE ESPACIOS DE CÓDIGO

ÍNDICE DE CUADROS

RESUMEN

Una de las principales preocupaciones al momento de implementar o hacer uso de una red de comunicaciones y de los equipos que la conforman, es el nivel de seguridad que tendrá la información que se almacena en dichas máquinas.

El desarrollo de un software que permita otorgar una mejor seguridad a las redes de datos, es en la actualidad el objetivo que todos los usuarios de un sistema esperan obtener; es así que el conocer como es el funcionamiento de los principales sistemas de seguridad que se maneja en un equipo permitirá identificar las principales vulnerabilidades y trabajar sobre las debilidades encontradas, de manera que se pueda obtener un sistema que proteja la información que se maneja en un host.

A través de este trabajo, se pretende realizar el análisis, diseño e implementación de un software prototipo que permita conocer el funcionamiento de dos sistemas empleados para la seguridad y control de equipos dentro de una red, estos son, el Sistema Firewall y el Servidor Proxy.

El desarrollo de este trabajo consta de cinco capítulos, en donde se registra paso a paso la ingeniería de software aplicada para la obtención del sistema propuesto.

Es así como en el capítulo I, denominado “Antecedentes”, se identifica la problemática a tratar, que será la base y permitirá dar a conocer el propósito de desarrollo de este trabajo, así como los objetivos que se pretende alcanzar, los lineamientos que se desea conocer y una descripción general del software a desarrollar.

En el capítulo II, denominado “Sustento Teórico del proyecto”, en dicho capítulo, se englobará las principales teorías que serán la base fundamental del desarrollo

de este trabajo, porque serán las fuentes necesarias que permitirán conocer que es lo que se desea lograr.

En el capítulo III, denominado “Análisis del Prototipo”, se realizará el análisis de factibilidad para conocer que tan viable es la creación del software, a nivel operativo, técnico y económico, así como de las herramientas necesarias para la creación del mismo y el análisis de las plataformas operativas de ejecución del software desarrollado.

El capítulo IV, denominado “Diseño y Construcción del Software Prototipo”, se genera una serie de diagramas UML que permitirán conocer la estructura que tendrá el sistema prototipo que se desea obtener, así como también los principales métodos y clases desarrolladas en lenguaje de programación Java, que permitan la construcción de dicho software.

El capítulo V, denominado “Implementación y Pruebas”, permitirá conocer como se debe implementar el sistema, y las diferentes ejecuciones realizadas al mismo, para verificar el funcionamiento esperado.

De esta manera, se obtendrán las respectivas conclusiones y recomendaciones según el desarrollo del sistema en cada fase de su construcción.

PRESENTACIÓN

El desarrollo del presente trabajo tiene como finalidad el diseño y la aplicación de un sistema prototipo de firewall y servidor proxy, a través del cual se conocerá como es la estructura lógica de funcionamiento de estas herramientas que son parte activa dentro de la seguridad de las redes de comunicaciones.

La construcción del software propuesto, permitirá que el usuario pueda manejar la seguridad de su equipo, independientemente de la plataforma operativa que se emplee para el trabajo cotidiano, porque la principal característica que tiene este sistema es ser multiplataforma.

La creación del sistema antes mencionado, permitirá personalizar según las necesidades de usuario los servicios que se desean controlar en un host determinado y de esta manera comprender cuales son los eventos que se desencadenan en conjunto para proporcionar la adecuada protección a un equipo, a través de una interfaz gráfica accesible al usuario.

Es así como este trabajo es el resultado de la investigación realizada y de los conocimientos adquiridos que en complemento, permitirán obtener los resultados deseados en el producto final.

CAPÍTULO I

1. ANTECEDENTES

En el capítulo I, se hará referencia al problema identificado, el cual será la base de la construcción del presente proyecto y al que se pretende dar solución a través del establecimiento de objetivos claros, que permitirán justificar el trabajo que se realizará, así como un panorama general del software que se desarrollará a lo largo de la investigación mencionada.

La seguridad es en la actualidad, un tema de especial relevancia para los servicios de una red de comunicaciones que permiten el desarrollo de las actividades, sean estas personales o de carácter empresarial.

Los ataques a los sistemas de información, cada vez son más frecuentes, con el fin de obtener, suprimir o modificar información de gran valor para los usuarios de dichos sistemas; esta es la razón por la que los sistemas de defensa están en constante evolución para enfrentar nuevos retos en términos de seguridad.

En un equipo de cómputo, es de vital importancia contar con software antivirus para la protección de la información frente a programas dañinos que desean acceder a través de la red a la que está conectado.

La implementación de una barrera de seguridad conocida, impide que agentes externos, sean estos usuarios, programas, o equipos, se conecten directamente a la máquina determinada, reduciendo en gran parte el riesgo de manipular la información almacenada.

Es por esta razón que es de gran utilidad el conocer el funcionamiento de las herramientas que proporcionan seguridad para salvaguardar la información manejada y sobre la cual se desarrollan todas las actividades de cualquier organización.

1.1. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, la mayoría de las instituciones dependen cada vez más de redes de información y un problema que las afecte, por mínimo que sea, puede llegar a comprometer la continuidad de sus operaciones y los servicios que estas proporcionan dentro del entorno de dicha organización, limitando sus capacidades y volviéndolas vulnerables al fácil acceso de intrusos.

La falta de medidas de seguridad en las redes es un problema que está en constante crecimiento. Cada vez es mayor el número de atacantes y existe un nivel de organización más alto, con el cual intentan infringir el entorno corporativo de una red de datos por lo que van adquiriendo día a día habilidades más especializadas que les permiten obtener mayores beneficios.

La compleja distribución de los equipos que conforman una red es una de las principales dificultades para detectar y corregir los múltiples y variados problemas de seguridad que van apareciendo al realizar controles y verificaciones de las vulnerabilidades operativas que pueden surgir en el desempeño funcional de la red.

Por tal motivo se han ido incrementando acciones que violan e irrespetan la privacidad y la propiedad de recursos y sistemas, esta situación ha permitido generar sistemas de información mucho más confiables y seguros que manejen altos niveles de desempeño, fiabilidad y autenticidad; promoviendo un entorno controlado que salvaguarde la integridad de la información.

1.2. OBJETIVOS DE LA INVESTIGACIÓN

1.2.1. OBJETIVO GENERAL

Construir un software prototipo, que permita detectar irrupciones de intrusos que violen la seguridad de un host dentro de una red, utilizando tecnología Java.

1.2.2. OBJETIVOS ESPECÍFICOS

- Definir políticas de seguridad que impidan el acceso de agentes no deseados a la información manejada, mediante la creación de un firewall.

- Fomentar la utilización de diferentes plataformas para el desarrollo y pruebas de prototipos de software dentro de un entorno de red.

- Generar conexiones seguras mediante la construcción de un servidor proxy que permita el control, el anonimato y filtrado de peticiones realizadas por el usuario al intentar conectarse a la web.

- Analizar y rastrear los puertos que se desean habilitar para el acceso seguro al manejo de la información.

1.3. JUSTIFICACIÓN DEL PROBLEMA

El estudio de las vulnerabilidades informáticas, permite definir políticas que gestionen la operatividad del acceso restringido a los recursos de la red para los usuarios que interactúan con cada uno de los servicios que provee la infraestructura de comunicación que tienen a su disposición.

De acuerdo a este análisis, se pueden desarrollar prototipos que controlen y administren las intromisiones que intenten perjudicar la autenticidad y la confiablidad de la información que es generada en la red de una determinada institución.

La construcción de este prototipo, persigue una finalidad práctica, puesto que con su desarrollo se beneficiará notablemente el control y seguridad de la información que manejan los gestionadores y administradores de red para mantener un entorno de trabajo confiable y eficaz.

1.4. ALCANCE DEL PROYECTO

El desarrollo de este software prototipo implicará:

- ✓ El control de acceso a programas que intenten conectarse al internet.
- ✓ Verificará cada una de las peticiones de servicio que se realice al momento de acceder a los recursos proporcionados por la red informática y que pueden afectar el funcionamiento normal del sistema.
- ✓ Este software podrá ser aplicado en plataformas operativas como Linux en distribuciones como Ubuntu y en Windows versión XP.
- ✓ Se manejará MYSQL 5.0 para el almacenamiento de datos, el cual también proveerá la información para los reportes requeridos.
- ✓ Proveerá un reporte estadístico de intromisiones en el sistema.
- ✓ Gestionará las peticiones de servicio para salir al internet mediante el Proxy.

- ✓ Para la creación de este prototipo, se utilizará Lenguaje Java para entornos web; el prototipo permitirá la configuración de los servicios de Firewall así como la activación del servicio proxy.

1.5. DESCRIPCIÓN GENERAL DEL SISTEMA

El proyecto a desarrollar, consiste en analizar, diseñar, construir e implementar a través de máquina virtuales, un software prototipo que permita detectar y controlar vulnerabilidades que se presentan en la red de datos por influencia de intrusos, que desean acceder de forma no autorizada a la información, ya sea para manipularla o utilizarla en su favor.

Se debe recalcar que el software que se desarrollará es un prototipo, entendiéndose de forma tal como "una representación, demostración o simulación limitada del diseño de un software planificado incluyendo su interfaz y su funcionalidad, que permite a las partes responsables de su creación experimentar, probarlo en situaciones reales y explorar su uso, no lleva a cabo la totalidad de las funciones necesarias del software final."[1]

Las características que presentan los Prototipos son:

- Tienen una aplicación funcional
- Evolucionan a través de un proceso iterativo
- Tienen un costo bajo de desarrollo.

Por lo tanto, el diseño de esta herramienta, permitirá conocer las funciones que desempeña un firewall, es decir se conocerá como se desencadenan los procesos

[1] Lacalle Alberto, Definición de Prototipo

para controlar la accesibilidad de agentes externos a las máquinas en donde se instalará el programa.

De la misma manera, se conocerá también como trabaja el servidor proxy, en su función de interceptar la navegación por páginas web, controlando la seguridad, el rendimiento y el anonimato de los usuarios que hacen uso de este servicio.

CAPÍTULO II

2. SUSTENTO TEÓRICO DEL PROYECTO

En el capítulo II, se citarán las bases teóricas que fundamentan la investigación y que a su vez presentarán de forma general una visión del trabajo que se pretende realizar, a través de los principales conceptos relacionados a la seguridad de los equipos y redes de comunicaciones, así como de las tecnologías y principales plataformas que serán parte de la ejecución de dicho proyecto.

2.1. SEGURIDAD INFORMÁTICA

La seguridad informática, consiste en asegurar que los recursos del sistema de información, sean estos materiales, informáticos o programas, sean utilizados de manera correcta y que el acceso a la información allí contenida, así como su modificación, sólo sea posible a las personas que se encuentren acreditadas y autorizadas para su manipulación.[2]

Para conocer que se pretende proteger mediante la seguridad informática, se debe conocer a que hacen referencia los términos más utilizados al abordar el tema, entre los cuales están:

- Dato: es la unidad mínima con la que se compone la información.

- Información: es un conjunto de datos que tiene un significado y un sentido particular según como y quien la procese.[3]

[2] s/a, Seguridad informática, www.wikipedia.org

[3] ALDEGANI Gustavo, Seguridad Informática. MP Ediciones. Argentina. 1997. Página 22.

Establecer el valor de la información es algo relativo, pues esta constituye un recurso intangible y frágil. Existe Información de tipo pública, en donde el acceso no tiene restricción; y otra de tipo privada, que solo puede ser visualizada por un grupo determinado de personas.

2.1.1. OBJETIVOS DE LA SEGURIDAD INFORMÁTICA

Los sistemas de información están conformados por los datos, el material y los recursos de software que permiten su almacenamiento y accesibilidad según los requerimientos de los usuarios que la manejen, por lo que es de vital importancia su protección ante la presencia de agentes externos no autorizados.

La seguridad informática consiste en garantizar que el material y los recursos de software se usen únicamente para los propósitos para los que fueron creados y dentro del marco previsto.

Por tal razón, la seguridad informática se resume en cinco objetivos principales:

- **Integridad:**

La información sólo puede ser modificada por quien está autorizado y de manera controlada, de tal forma, que si existe alguna alteración, se pueda verificar su procedencia.

Una falla de integridad puede estar dada por anomalías en el hardware, software, virus informáticos y/o modificación por personas que se infiltran en el sistema.

- **Confidencialidad:**

Asegurar que sólo los individuos autorizados tengan acceso a los recursos que se intercambian, de tal manera que sea intangible para aquellos usuarios ajenos al sistema.

En casos de falta de confidencialidad, la Información puede provocar severos daños al usuario o volverse obsoleta y perder su valor.

- **Disponibilidad:**

Garantizar que la información este disponible siempre que sea necesaria.Esto requiere que la misma se mantenga correctamente almacenada, que el hardware y el software funcionen perfectamente y que se respeten los formatos para su recuperación en forma satisfactoria.

- **Autenticación:**

Consiste en asegurar que solo los agentes autorizados tengan acceso a la información, a través de la confirmación de la identidad. Esta propiedad también permite asegurar el origen de la información, validando el emisor de la misma, para evitar suplantación de identidades.

2.1.2. AMENAZAS CONTRA LA SEGURIDAD INFORMÁTICA

Es considerada como amenaza, a cualquier elemento que comprometa al sistema. Las amenazas pueden ser analizadas en tres momentos: antes, durante y después del ataque; estos mecanismos conforman políticas que garantizarán la seguridad del sistema informático.

- **Antes:** La Prevención: mecanismos que aumentan la seguridad (o fiabilidad) de un sistema durante su funcionamiento normal.

- **Durante:** La Detección: mecanismos orientados a revelar violaciones a la seguridad. Generalmente son programas de auditoría.

- **Después:** La Recuperación: mecanismos que se aplican, cuando la violación del sistema ya se ha detectado, para volver a la función normal.

Las amenazas al sistema a menudo son imprevisibles o inevitables, de modo que la única protección posible es la redundancia (en el caso de los datos) y la descentralización a través de una estructura de redes de comunicaciones.

Estos fenómenos pueden ser causados por:

Amenazas internas:

Generalmente estas amenazas pueden ser más serias que las externas por varias razones como son:

- El usuario: causa del mayor problema ligado a la seguridad de un sistema informático por el conocimiento de la estructura del mismo.

- Programas maliciosos: programas destinados a perjudicar o a hacer un uso ilícito de los recursos del sistema. Es instalado en el computador abriendo una puerta a intrusos o bien modificando los datos.

- Un intruso: persona que consigue acceder a los datos o programas de los cuales no tiene acceso permitido.

Amenazas externas:

Son aquellas amenazas que se originan de afuera de la red.

Al no tener información certera de un origen, un atacante tiene que realizar ciertos pasos para poder conocer qué es lo que hay en ella y buscar la manera de atacarla. La ventaja que se tiene en este caso es que el administrador de la red puede prevenir una buena parte de los ataques externos.

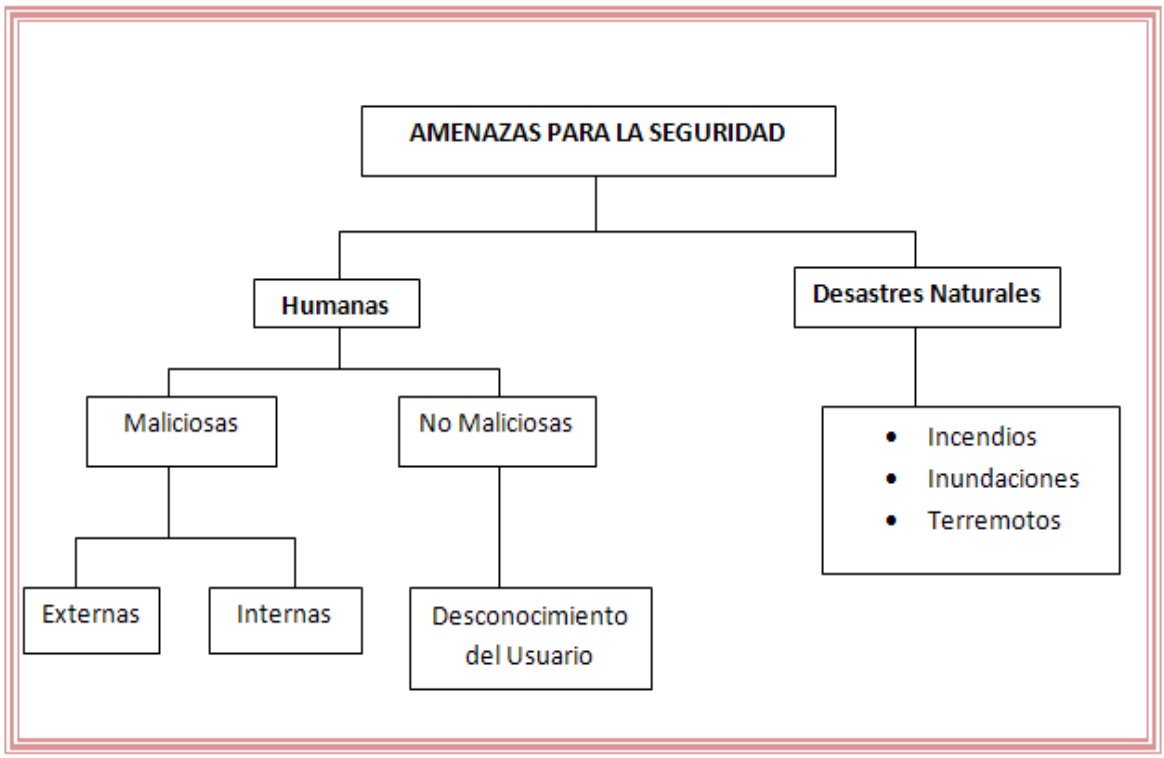

Fig. 2.1: "Amenazas para la Seguridad Informática"

Fuente: Seguridad Informática, Universidad Tecnológica Nacional

2.1.3. TIPOS DE INTRUSOS

Se conoce como intruso o atacante a quien accede (o intenta acceder) sin autorización a un sistema ajeno, ya sea en forma intencional o no. Se conocen algunos tipos de intrusos, según el nivel de ataque que propicien a la información, y según el nivel de conocimiento para ejecutar acciones que desencadenen intromisiones:

1.- Clase A: el 80% son novatos que prueban pequeños programas para accesos no autorizados a información sin demasiada protección.

2.- Clase B: el 12% son más peligrosos, saben compilar programas, conocen como detectar que sistema operativo está en uso, testean las vulnerabilidades del mismo e ingresan por ellas.

3.- Clase C: el 5%, son usuarios que tienen un objetivo planteado y acceden a una red de información a través de accesos remotos.

4.- Clase D: el 3% restante, cuando acceden a determinados sistemas buscan la información que necesitan.[4]

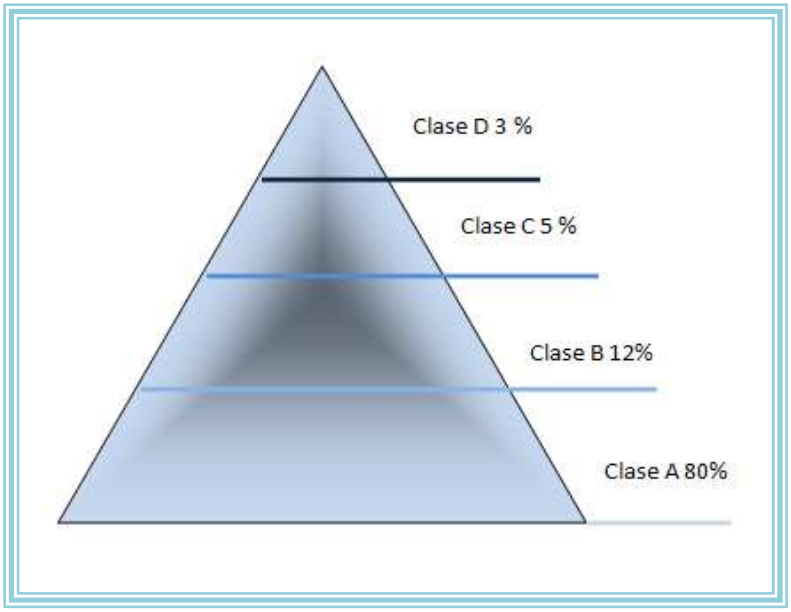

Fig. 2.2: "Tipos de Intrusos"

Fuente: CybSec S.A. www.cybsec.com

2.1.4. PROTECCIÓN

En cualquier sistema informático existen tres elementos básicos a proteger:

- Hardware: se entiende como el conjunto de todos los elementos físicos del sistema informático: CPU, cableado, impresoras, CD–ROM, cintas, componentes de comunicación.

- Software: son todos los elementos lógicos que hacen funcional al hardware: sistema operativo, aplicaciones, utilidades.

[4]ARDITA Julio César. Director de Cybsec S.A. Security System y ex–Hacker, http://www.cybsec.com

- Datos: conjunto de información lógica que maneja el software y el hardware: bases de datos, documentos, archivos.

De los elementos mencionados, son los datos, el componente más frágil y que si no existiera una copia de seguridad o respaldo, su recuperación sería difícil.

Para cada elemento descrito, existen multitud de amenazas y ataques que se los puede clasificar en:

1.- Ataques Pasivos:

El atacante no altera la comunicación, sino que únicamente la “escucha” o monitoriza, para obtener información que está siendo transmitida.

Sus objetivos son la intercepción de datos y el análisis de tráfico.

Generalmente se emplean para:

- Obtención del origen y destinatario de la comunicación, a través de la lectura de las cabeceras de los paquetes monitorizados.

- Control del volumen de tráfico intercambiado entre las entidades monitorizadas, obteniendo así, información acerca de actividad o inactividad inusual.

- Control de las horas habituales de intercambio de datos entre las entidades de la comunicación, para extraer información acerca de los períodos de actividad.

Es posible evitar el éxito, si bien no el ataque, mediante el cifrado de la información.

2.- Ataques Activos:

Estos ataques implican algún tipo de modificación del flujo de datos transmitido o la creación de un falso flujo de datos.

Generalmente son realizados por hackers, piratas informáticos o intrusos remunerados y se los puede subdividir en cuatro categorías:

- **Interrupción:** si hace que un objeto del sistema se pierda, quede inutilizable o no disponible.

- **Intercepción:** si un elemento no autorizado consigue el acceso a un determinado objeto del sistema.

- **Modificación:** si además de conseguir el acceso consigue modificar el objeto.

- **Fabricación:** se consigue un objeto similar al original atacado de forma que es difícil distinguirlos entre sí.

- **Destrucción:** es una modificación que inutiliza el objeto.

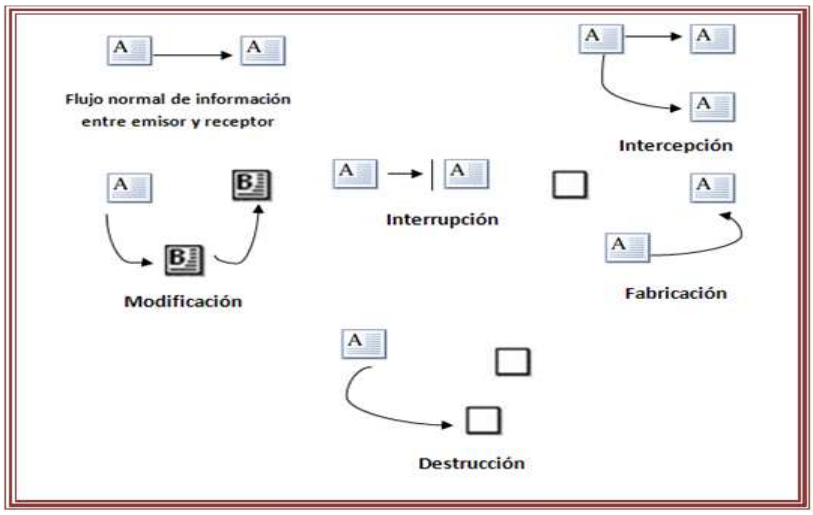

Fig. 2.3: "Tipos de Ataques a Sistemas Informáticos"

Fuente: HOWARD John D. Thesis: An Analysis of security on the Internet, www.cert.org.

2.2. SEGURIDAD CON RESPECTO A LA NATURALEZA DE LA AMENAZA

Existen dos tipos de seguridad con respecto a la naturaleza informática, estas son:

2.2.1. SEGURIDAD LÓGICA:[5]

Dentro de la seguridad informática, la seguridad lógica hace referencia a la aplicación de mecanismos y barreras para mantener el resguardo y la integridad de la información dentro de un sistema informático. La seguridad lógica contrasta con la seguridad física.

La seguridad lógica de un sistema informático incluye:

- Restringir al acceso a programas y archivos mediante claves y/o encriptación.

- Asignar las limitaciones correspondientes a cada usuario del sistema informático. Esto significa, no darle más privilegios extras a un usuario, sino sólo los que necesita para realizar su trabajo.

- Asegurarse que los archivos y programas que se emplean son los correctos y se usan correctamente. Por ejemplo, el mal uso de una aplicación puede ocasionar agujeros en la seguridad de un sistema informático.

- Control de los flujos de entrada/salida de la información. Esto incluye que una determinada información llegue solamente al destino que se espera que llegue, y que la información llegue tal cual se envió.

[5] s/a, Definición de seguridad Lógica, www.alegsa.com.ar

Los controles anteriormente mencionados se pueden hacer a nivel del sistema operativo, a nivel aplicación, a nivel base de datos o archivo, o a nivel firmware[6].

2.2.2. SEGURIDAD FÍSICA:[7]

Dentro de la seguridad informática, la seguridad física hace referencia a las barreras físicas y mecanismos de control en el entorno de un sistema informático, para proteger el hardware de amenazas físicas. La seguridad física contrasta con la seguridad lógica

Los mecanismos de seguridad física deben resguardar de amenazas producidas tanto por el hombre como por la naturaleza. Básicamente, las amenazas físicas que pueden poner en riesgo un sistema informático son:

- Desastres naturales, incendios accidentales, humedad e inundaciones.
- Amenazas ocasionadas involuntariamente por personas.
- Acciones hostiles deliberadas como robo, fraude o sabotaje.

2.3. IMPLEMENTACIÓN DE BARRERAS DE SEGURIDAD

En informática, se considera como barrera de seguridad, a todo objeto o sistema que se aplique para mantener la seguridad de un sistema informático.

2.3.1. BARRERAS DE SEGURIDAD A NIVEL SOFTWARE (SEGURIDAD LÓGICA):

- Cortafuegos: Aplicación o herramienta que funciona como medio de defensa, que evita cualquier tipo de acceso a un determinado sistema.

---

[6] Es un programa que es grabado en memoria ROM y establece la lógica de más bajo nivel que controla los circuitos electrónicos de un dispositivo

[7] s/a, Definición de Seguridad Física, www.alegsa.com.ar

- Antivirus: Aplicación o grupo de aplicaciones dedicadas a la prevención, búsqueda, detección y eliminación de programas malignos en sistemas informáticos.

- Antispam: Aplicación o herramienta informática que se encarga de detectar y eliminar el spam y los correos no deseados.

- Antispyware: Tipo de aplicación que se encarga de buscar, detectar y eliminar spywares o espías en el sistema.

- Números de serie: En software, conjunto de caracteres (letras y/o números) que permiten activar una aplicación o servicio.

2.3.2. BARRERAS DE SEGURIDAD A NIVEL HARDWARE (SEGURIDAD FÍSICA):

- UPS o SAI (Sistema de alimentación ininterrumpida): es una fuente de suministro eléctrico que posee una batería con el fin de seguir dando energía a un dispositivo en el caso de interrupción eléctrica.

- Extintores, guardia humana, cámaras de seguridad, etc.

2.4. DESCRIPCIÓN DE TECNOLOGÍA JAVA[8]

2.4.1. LA TECNOLOGÍA JAVA[9]

La tecnología Java es una revolucionaria plataforma informática presentada por Sun Microsystems en 1995, originalmente fue denominada OAK, el lenguaje de programación fue llamado Java en 1995.

[8] Almazán Zapata Hilda Laura, Administración de sistemas
[9] JAVA, www.java.com

La tecnología Java despliega una multitud de posibilidades para los usuarios, pues permite que prácticamente cualquier aplicación se ejecute en casi cualquier equipo o dispositivo.

2.4.2. LENGUAJE DE PROGRAMACIÓN JAVA

Java, es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems, el lenguaje en sí, toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode[10] , en el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución.

Desde 1995, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process.

2.4.3. FILOSOFÍA

El lenguaje Java se creó con cinco objetivos principales:

- Usar la metodología de la programación orientada a objetos.
- Permitir la ejecución de un mismo programa en múltiples sistemas operativos.
- Incluir por defecto soporte para trabajo en red.

[10] Es un código binario intermedio, tratado como un programa ejecutable similar a un módulo objeto, que es un fichero binario producido por el compilador cuyo contenido es el código objeto o código máquina.

- Ejecutar código en sistemas remotos de forma segura.

- Su uso deberá ser fácil y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

2.4.4. MÁQUINA VIRTUAL JAVA

Una Máquina virtual Java, es un programa nativo, es decir, ejecutable en una plataforma específica, capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial, el cual es generado por el compilador del lenguaje Java.

El código binario de Java no es un lenguaje de alto nivel, sino un verdadero código máquina de bajo nivel, viable, incluso como lenguaje de entrada para un microprocesador físico.

La JVM[11] es una de las piezas fundamentales de la plataforma Java, se sitúa en un nivel superior al Hardware del sistema sobre el que se pretende ejecutar la aplicación, y este actúa como un puente que entiende tanto el bytecode, como el sistema sobre el que se pretende ejecutar.

Así, cuando se escribe una aplicación Java, se hace pensando que será ejecutada en una máquina virtual Java, siendo ésta la que en última instancia convierte de código bytecode a código nativo del dispositivo final.

La gran ventaja de la máquina virtual java es aportar portabilidad al lenguaje de manera que pueda ser ejecutado en diferentes arquitecturas.

[11] Máquina Virtual Java

2.4.5. ENTORNO DE EJECUCIÓN

Para ejecutar una aplicación en una Máquina Virtual de Java, el programa código debe compilarse de acuerdo a un formato binario portable, estandarizado, normalmente en forma de ficheros con extensión .class.

Un programa puede componerse de múltiples clases, en cuyo caso cada clase tendrá asociada su propio archivo .class.

Para facilitar la distribución de aplicaciones, los archivos de clase pueden empaquetarse juntos en un archivo con formato .jar.

El código resultante de la compilación es ejecutado por la JVM que lleva a cabo la emulación del conjunto de instrucciones, bien por un proceso de interpretación o más habitualmente mediante un compilador.

2.4.6. VENTAJAS

Un applet de Java[12] puede tener las siguientes ventajas:

- Funcionan en Linux, Windows y Mac OS, son multiplataforma.
- El mismo applet pueden trabajar en cualquier versión de java. Sin embargo, si un applet requiere una versión posterior de la JRE[13], el cliente deberá actualizar la versión.
- Es soportado por la mayoría de los navegadores Web.

[12] Es un pequeño programa basado en internet y escrito en Java. Los applets generalmente están embebidos en páginas web y pueden ser ejecutados directamente desde un navegador con soporte para Java.

[13] Java Runtime Environment (Entorno en Tiempo de Ejecución de Java), es el software necesario para ejecutar cualquier aplicación desarrollada para la plataforma Java.

- Puede ser almacenado en la memoria cache de la mayoría de los navegadores Web, de modo que se cargará rápidamente cuando se vuelva a cargar la página Web.

- Puede tener acceso completo a la máquina en la que se está ejecutando, si el usuario lo permite.

- Puede ejecutarse con velocidades comparables a la de otros lenguajes compilados, como C + +.

- Puede trasladar el trabajo del servidor al cliente, haciendo una solución Web más escalable tomando en cuenta el número de usuarios / clientes.

2.4.7. DESVENTAJAS

Un applet de Java puede presentar desventajas como:

- Requiere el plug-in de Java, que no está disponible por defecto en todos los navegadores web.

- Sun no ha creado una implementación del plug-in para los procesadores de 64 bits.

- No puede iniciar la ejecución hasta que la Máquina virtual de Java está en funcionamiento, y esto puede tomar tiempo la primera vez que se ejecuta un applet.

- Si no está firmado como confiable, tiene un acceso limitado al sistema del usuario.

- Algunas organizaciones sólo permiten la instalación de software a los administradores. Como resultado, muchos usuarios no pueden ver los applets.

- Un Applet podría exigir una versión específica del JRE.

2.4.8. TECNOLOGÍAS JAVA

La Plataforma Java se compone de una amplia gama de tecnologías, cada una de estas, ofrece una parte del entorno de ejecución en tiempo real.

Las aplicaciones Java pueden usarse de forma variada, como por ejemplo en una página Web.

Para el desarrollo de aplicaciones, se utiliza un conjunto de herramientas conocidas como JDK.[14]

2.4.8.1. Lenguajes

La palabra Java, por sí misma, se refiere habitualmente al lenguaje de programación Java, que fue diseñado para usar con la Plataforma que lleva el mismo nombre.

El lenguaje de programación Java es uno de los componentes fundamentales de la plataforma Java, es un lenguaje orientado a objetos, que se puede utilizar para crear subprogramas o programas que se pueden distribuir como adjuntos documentos web.

[14] Java Development Kit, son un conjunto de herramientas de desarrollo para Java.

2.5. DESCRIPCIÓN DE PLATAFORMAS OPERATIVAS

El sistema operativo es el software más importante de un computador. Para que funcionen los otros programas, cada ordenador de uso general debe tener un sistema operativo.

Los sistemas operativos, en su condición de capa software, posibilitan y simplifican el manejo de la computadora, desempeñan una serie de funciones básicas y esenciales para la gestión del equipo.

Un sistema operativo desempeña cinco funciones básicas en la operación de un sistema informático:

- suministro de interfaz al usuario,
- administración de recursos,
- administración de archivos,
- administración de tareas y
- servicio de soporte y utilidades.

Interfaces del usuario

Es la parte del sistema operativo que permite comunicarse con él, de tal manera que se puedan cargar programas, acceder archivos y realizar otras tareas.

Administración de recursos

Sirven para administrar los recursos de hardware y de redes de un sistema informático, como la CPU, memoria, dispositivos de almacenamiento secundario y periféricos de entrada y de salida.

Administración de archivos

Un sistema de información contiene programas de administración de archivos que controlan la creación, borrado y acceso de archivos de datos y de programas.

Administración de tareas

Los programas de administración de tareas de un sistema operativo administran la realización de las tareas informáticas de los usuarios finales. Los programas controlan qué áreas tienen acceso al CPU y por cuánto tiempo. Las funciones de administración de tareas pueden distribuir una parte específica del tiempo del CPU para una tarea en particular, e interrumpir al CPU en cualquier momento para sustituirla con una tarea de mayor prioridad.

Servicio de soporte

Los servicios de soporte de cada sistema operativo, dependerán de la implementación particular de éste con la que se este trabajando. Estos servicios de soporte suelen consistir en:

- Actualización de versiones.
- Mejoras de seguridad.
- Inclusión de alguna nueva utilidad (un nuevo entorno gráfico, un asistente para administrar alguna determinada función).
- Controladores para manejar nuevos periféricos (este servicio debe coordinarse a veces con el fabricante del hardware).
- Corrección de errores de software.

El sistema operativo es responsable de la seguridad, denegando el acceso a los usuarios no autorizados al sistema. Un sistema operativo además debe ser:

- **Multiusuario**: Permite que dos o más usuarios utilicen sus programas al mismo tiempo.
- **Multiprocesador**: soporta el abrir un mismo programa en más de un CPU.
- **Multitarea**: Permite que varios programas se ejecuten al mismo tiempo.
- **Multitramo**: Permite que diversas partes de un solo programa funcionen al mismo tiempo.
- **Tiempo Real**: Responde a las entradas inmediatamente.

De esta manera, el proyecto que se realizará, podrá ser ejecutado en dos plataformas operativas, las cuales se describen a continuación:

2.5.1. EL SISTEMA OPERATIVO WINDOWS

Microsoft Windows es una familia de sistemas operativos desarrollados y comercializados por Microsoft.

Existen versiones para hogares, empresas, servidores y dispositivos móviles, como computadores de bolsillo y teléfonos inteligentes, esta plataforma operativa, incorpora diversas aplicaciones.

Es el sistema operativo más difundido y usado, por tal motivo, una gran cantidad de programas son desarrollados para este sistema.

Windows, es un sistema operativo, cuyo diseño y creación se caracteriza por ser:

- Extensible
- Portable
- Fiable
- Adaptable
- Robusto
- Seguro y
- Compatible con sus versiones anteriores.

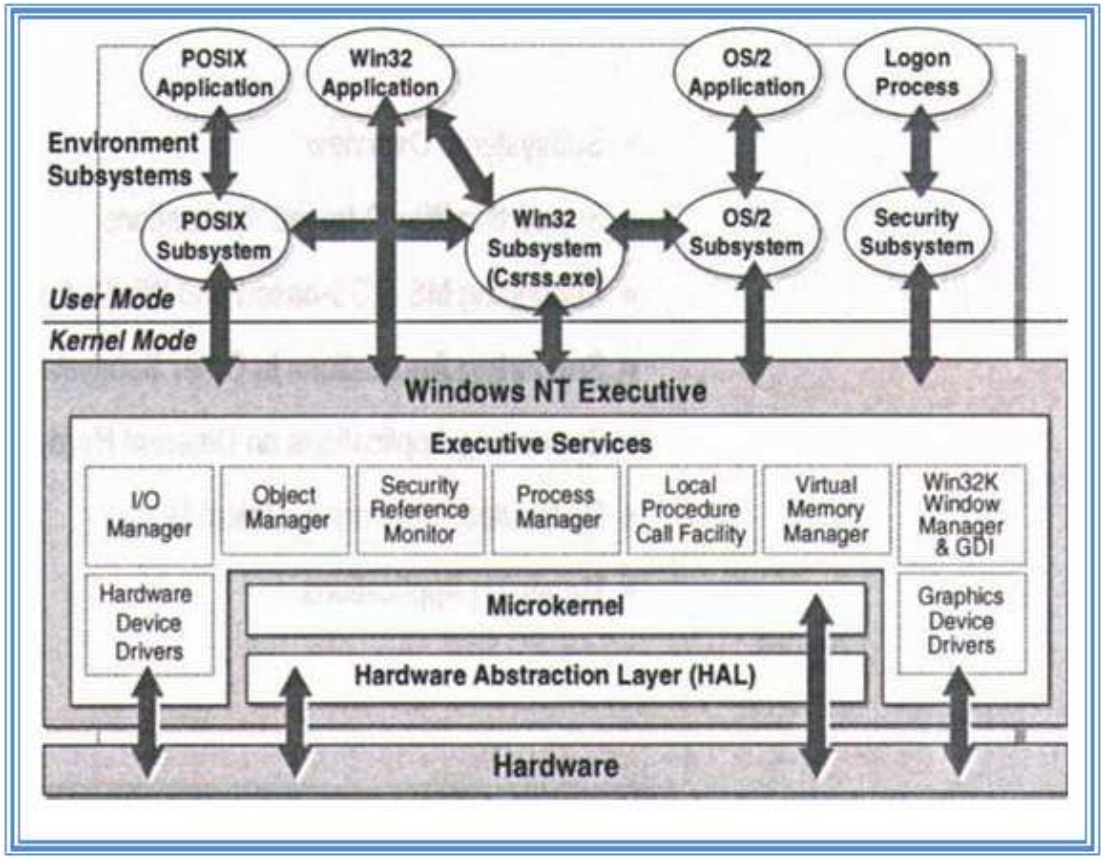

Fig. 2.4: "Arquitectura modular de Windows"

Fuente: Sistema Operativo Windows, www.monografias.com

En la Fig. 2.4, se muestra la arquitectura modular de Windows, la cual está compuesta por una serie de componentes separados, donde cada cual es responsable de sus funciones y brindan servicios a otros componentes.

Esta arquitectura es del tipo cliente – servidor, porque los programas de aplicación son contemplados por el sistema operativo como si fueran clientes a los que hay que servir, por lo que está equipado con distintas entidades servidoras.

Es preciso recalcar que una de las características que Windows tiene en común con el resto de Sistemas Operativos, es la división de tareas del Sistema Operativo en múltiples categorías, las cuales son soportadas por microprocesadores.

Estos modos proporcionan a los programas cierta jerarquía y privilegios para acceder al hardware o a otros programas, siendo el principal, el que corre en el kernel y de un modo secundario los que están en opción de usuario.

- **El Modo Kernel:**

Es un modo muy privilegiado de funcionamiento, donde el código tiene el acceso directo a todo el hardware y toda la memoria, incluso a los espacios de dirección de todos los procesos del modo usuario.

La parte de Windows que corre en el modo Kernel se llama Ejecutor de Windows, este no es más que un conjunto de servicios disponibles a todos los componentes del Sistema Operativo, donde cada grupo de servicios es manipulado por componentes que son totalmente independientes entre sí y se comunican a través de interfaces bien definidas.

- **El Modo Usuario:**

Es un modo menos privilegiado de funcionamiento, sin el acceso directo al hardware. El código que corre en este modo sólo actúa en su propio espacio de

dirección. Este usa API[15] para pedir los servicios del sistema. Todos los programas que no corren en Modo Kernel corren en Modo Usuario.

El software prototipo desarrollado, se ejecutará específicamente sobre una de las versiones de Windows, precisamente en Windows XP.

2.5.1.1. Windows XP (eXPerience)

Windows XP, es el resultado de la conjunción de Windows NT/2000 y la familia de Windows 9.x, es una plataforma operativa comercializada desde el año 2001 en su versión Home y Professional.

Esta versión de Windows, presenta una serie de características como:

- Multitarea mejorada,
- Soporte para redes inalámbricas
- Asistencia remota.

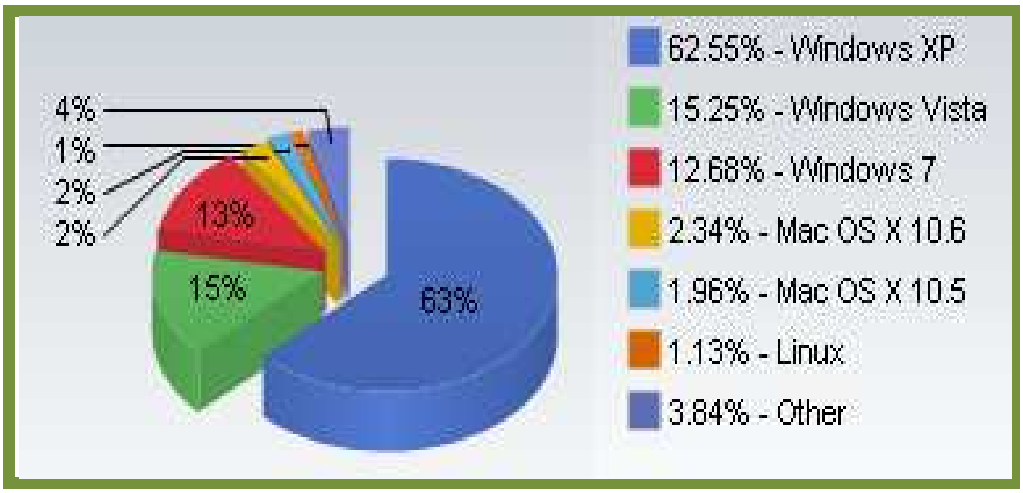

Fig. 2.5: "Uso de Sistemas Operativos"

Fuente: Operating System Market Share, www.marketshare.hitslink.com

[15] Application Programming Interface (Interfaz de Programación de Aplicaciones), representa un interfaz de comunicación entre componentes.

En la Fig. 2.5, se muestra el porcentaje estimado del uso actual de sistemas operativos según una muestra de computadoras con acceso a Internet, en donde se observa claramente que el sistema operativo Windows, en su versión XP tiene un porcentaje de uso del 63%, lo cual permite conocer el buen desempeño y por ende la aceptación de los usuarios ante dicha plataforma operativa.

Seguridad

Una de las principales falencias con las que cuentan las versiones de la plataforma operativa Windows, es la debilidad del sistema en lo que a seguridad se refiere y el alto índice de vulnerabilidades críticas.

Cabe recalcar que no existe un sistema completamente libre de errores, estos se pueden presentar y a su vez corregir según el constante uso del mismo.

2.5.1.2. Windows 7

Es la versión más reciente que ofrecida al mercado por Microsoft Windows, es un sistema operativo producido para uso en computadores, sean estos equipos de escritorio o portátiles[16].

El desarrollo de dicho sistema operativo culminó el 22 de julio del año 2009, siendo comercializado oficialmente el 22 de octubre de 2009 junto a su equivalente para servidores Windows Server 2008 R2[17].

A diferencia del gran salto arquitectónico y de características que sufrió su antecesor Windows Vista con respecto a Windows XP, Windows 7 fue concebido como una actualización incremental y focalizada de Vista y su núcleo NT 6.0, lo

[16] Windows 7, www.news.cnet.com

[17] Comercialización de Windows 7, www.publico.es

que permitió mantener cierto grado de compatibilidad con aplicaciones y hardware en los que éste ya era compatible.

Sin embargo, entre las metas de desarrollo para Windows 7 se dio importancia en mejorar su interfaz para volverla más accesible al usuario e incluir nuevas características que permitieran hacer tareas de una manera más fácil y rápida, al mismo tiempo en que se realizarían esfuerzos para lograr un sistema más ligero, estable y rápido.

Características

Entre las principales características que presenta Windows 7, están:

- Reconocimiento de voz, táctil y escritura
- Soporte para discos virtuales
- Mejor desempeño en procesadores multinúcleo
- Mejor arranque
- Mejoras en el núcleo.
- Inclusion de programas como: Windows Mail, Windows Movie Maker y Windows Photo Gallery.

Requisitos de hardware para la instalación de Windows 7[18]:

A continuación, se presentan los principales requisitos que deben tener los equipos para instalar el sistema operativo Windows 7:

[18] Requerimientos mínimos para la instalación de Windows 7, www.blogs.zdnet.com

Arquitectura	32 bits	64 bits
Procesador	1 GHz	
Memoria RAM	1 GB de RAM	2 GB de RAM
Tarjeta gráfica	Dispositivo de gráficos DirectX 9 con soporte de controladores WDDM 1.0	
Disco duro	16 GB de espacio libre	20 GB de espacio libre
Unidad óptica	DVD-R/RW	

Cuadro 2.1: "Requisitos mínimos para instalar Windows 7"

Fuente: www.blogs.zdnet.com

Opcionalmente, se requiere un monitor táctil para poder acceder a las características "multitáctiles" nuevas encontradas en este sistema.

2.5.2. EL SISTEMA OPERATIVO GNU/LINUX

GNU/Linux[19] es un sistema operativo diseñado bajo el objetivo de impulsar el software de libre distribución junto con su código fuente para que pueda ser modificado por cualquier persona, de esta manera se fortalecería su código la aplicación.

Las principales funciones de esta plataforma operativa son:

- Sistema multitarea: es posible ejecutar varios programas a la vez sin necesidad de detener la ejecución de cada aplicación.

[19] Sistema Operativo Linux, www.monografias.com

- Sistema multiusuario: varios usuarios pueden acceder a las aplicaciones y recursos del sistema Linux al mismo tiempo.

- Shells programables: un shell conecta las órdenes de un usuario con el Kernel y al ser programables se puede modificar para adaptarlo a las necesidades o requerimientos de cada usuario.

- Independencia de dispositivos: linux admite cualquier tipo de dispositivo de manera que posee una gran adaptabilidad y no se encuentra limitado.

La plataforma operativa Linux cuenta con características como:

- Distribución de código fuente

- Es desarrollado en forma abierta por cientos de usuarios

- Cuenta con un amplio y robusto soporte para comunicaciones y redes

- Da soporte a una amplia variedad de hardware y se puede correr en una multitud de arquitecturas de hardware.

Seguridad

Considerando a Linux como un sistema gratuito, flexible, potente y robusto, incorpora las características de seguridad comunes a todos los sistemas tipo Unix, por lo que se debe tener en cuenta el incremento de las medidas de protección para evitar la pérdida de información y el acceso de agentes no autorizados al sistema.

Como se mencionó antes, el software que se desarrollará, en su carácter de multiplataforma, podrá ser ejecutado en la plataforma operativa Linux, específicamente en su distribución Ubuntu.

2.5.2.1. Ubuntu[20]

Linux, al igual que Windows, incorpora diferentes versiones de sistema operativo pero para esta plataforma se las conoce como distribuciones Linux[21].

Es una distribución GNU/Linux que ofrece un sistema operativo predominantemente enfocado a computadores de escritorio aunque también proporciona soporte para servidores. Esta distribución está basada en Debian GNU/Linux, y concentra su objetivo en la facilidad de uso, la libertad de uso y la facilidad en la instalación.

La distribución Ubuntu, proporciona un sistema operativo actualizado y estable, enfocado en un usuario promedio, con facilidad de uso y de instalación del sistema, se compone de múltiples paquetes de software que están distribuidos bajo una licencia libre o de código abierto.
Ubuntu es una de las distribuciones más populares, llegando a alcanzar aproximadamente el 30% de las instalaciones de Linux en computadoras de escritorio.

Principios de Ubuntu

La filosofía de Ubuntu se basa en los siguientes principios:

- Ubuntu siempre será gratuito, y no habrá un coste adicional por tal motivo, estará libremente disponible para todos.

- Ubuntu emplea las mejores herramientas de traducción y accesibilidad que la comunidad del Software Libre es capaz de ofrecer.

[20] Ubuntu, www.doc.ubuntu-es.org

[21] Una distribución, es un software basada en el núcleo Linux que incluye determinados paquetes de software para satisfacer las necesidades de un grupo específico de usuarios, dando así origen a ediciones domésticas, empresariales y para servidores.

- Ubuntu publica de manera regular y predecible, una nueva versión cada seis meses. Puede usar la versión estable o de prueba.

- Ubuntu está totalmente comprometido con los principios de desarrollo del software de código abierto, para mejorarlo y compartirlo.

CAPÍTULO III

3. ANÁLISIS DEL PROTOTIPO

En el capítulo III, se procederá a realizar un estudio sobre las factibilidades tanto operativas, técnicas y económicas con respecto al software prototipo que se desarrollará en el presente trabajo investigativo.

3.1. FIREWALL Y SERVIDOR PROXY

3.1.1. ANÁLISIS \ FACTIBILIDAD

El análisis de factibilidad, es el proceso mediante el cual, se miden distintos aspectos del posible éxito que tendrá el proyecto a realizarse, mediante el producto que se genera, en este caso será los resultados obtenidos, al implementar el software prototipo.

Para dicho proyecto, se tomará en cuenta el análisis de factibilidad operativa, factibilidad técnica y factibilidad económica, específicamente el análisis costo/ beneficio.

3.1.1.1. Factibilidad Operativa

Para el análisis de la factibilidad operativa, se tomará en cuenta cuatro aspectos:

1.- Complejidad en el manejo del sistema:

El sistema que se desarrollará, será diseñado básicamente con el propósito de brindar seguridad a la red, a través de la creación de un software prototipo de firewall y servidor proxy multiplataforma, a través del cual, será posible conocer cómo trabajan dichos programas y cuál es la función que cumplen dentro de una red, en función de la protección que brindan a la misma.

2.- Funcionalidad del sistema:

El sistema planteado, no busca desplazar a ningún programa ya diseñado, sino que permitirá conocer como es el funcionamiento del mismo y además constituirá un soporte para el software ya existente, con el propósito de brindar mayor seguridad a la red de comunicaciones.

3.- Capacitación en el manejo del sistema:

Si el software, fuera distribuido en el mercado, este debe contar con un manual de usuario, en donde se detallará paso a paso la instalación, configuración y manejo de dicho sistema para que los posibles usuarios no tengan inconvenientes de uso.

4.- Vigencia del Sistema

El software, será desarrollado en un lenguaje de programación que está a la vanguardia, y que hoy por hoy es manejado por múltiples personas conocedoras de sistemas, lo que permitirá en un futuro adaptarlo a las necesidades que se presenten en la red que haga uso del mismo.

Lenguajes de programación para la web[22]

En la actualidad, existen diferentes lenguajes de programación para desarrollar aplicaciones en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas operativas que se encuentran en el medio.

Entre los lenguajes de programación más usados por su dinamismo y empleo de bases de datos para manejo de la información, están:

Lenguaje PHP

Es un lenguaje de programación utilizado para la creación de sitios web. PHP, significa "PHP Hypertext Pre-processor", surgió en 1995, desarrollado por PHP Group.

PHP es un lenguaje de script interpretado en el lado del servidor utilizado para la generación de páginas web dinámicas, embebidas en páginas HTML y ejecutadas en el servidor, dicho lenguaje no necesita ser compilado para ejecutarse.

Para su funcionamiento necesita tener instalado Apache o IIS[23] con las librerías de PHP.

La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas. Los archivos cuentan con la extensión (php).

Ventajas:

- Muy fácil de aprender.
- Se caracteriza por ser un lenguaje muy rápido.

[22] Lenguajes de programación para aplicaciones web, www.maestrosdelweb.com

[23] Internet Information Server

- Soporta en cierta medida la orientación a objetos, clases y herencia.
- Es un lenguaje multiplataforma
- Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.
- Capacidad de expandir su potencial utilizando módulos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Incluye gran cantidad de funciones.

Desventajas:

- Se necesita instalar un servidor web.
- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.
- La legibilidad del código puede verse afectada al mezclar sentencias HTML y PHP.
- La programación orientada a objetos es aún muy deficiente para aplicaciones grandes.
- Dificulta la modularización.
- Dificulta la organización por capas de la aplicación.

Lenguaje ASP.NET

Es un lenguaje comercializado por Microsoft, y usado para desarrollar entre otras funciones, sitios web ASP[24].NET es el sucesor de la tecnología ASP, fue lanzada al mercado mediante una estrategia denominada .NET.

El ASP.NET fue desarrollado para resolver las limitantes que brindaba tu antecesor ASP.

Para el desarrollo de ASP.NET se puede utilizar C#, VB.NET o J#. Los archivos cuentan con la extensión (aspx).

Para el funcionamiento de las páginas se necesita tener instalado IIS con el Framework .Net.

Ventajas:

- Completamente orientado a objetos.
- Controles de usuario personalizados.
- División entre la capa de aplicación o diseño y el código.
- Facilita el mantenimiento de grandes aplicaciones.
- Incremento de velocidad de respuesta del servidor.
- Mayor velocidad.
- Mayor seguridad.

[24] Active Server page

Desventajas:

- Mayor consumo de recursos.

Lenguaje JSP

Es un lenguaje para la creación de sitios web dinámicos, acrónimo de Java Server Pages. Está orientado a desarrollar páginas web en Java.

JSP es un lenguaje multiplataforma, creado para ejecutarse del lado del servidor.

JSP fue desarrollado por Sun Microsystems, posee un motor de páginas basado en los servlets[25] de Java.

Para su funcionamiento se necesita tener instalado un servidor Tomcat.

Características:

- Código separado de la lógica del programa.
- Las páginas son compiladas en la primera petición.
- Permite separar la parte dinámica de la estática en las páginas web.
- Los archivos se encuentran con la extensión (jsp).
- El código JSP puede ser incrustado en código HTML.

[25] Pequeño programa que corre en un servidor, no presentan ningún tipo de interfaz gráfica puesto que se encargan de hacer el trabajo oculto.

Elementos de JSP

Los elementos que pueden ser insertados en las páginas JSP son los siguientes:

- Código: se puede incrustar código “Java”.
- Directivas: permite controlar parámetros del servlet.
- Acciones: permite alterar el flujo normal de ejecución de una página.

Ventajas:

- Crear páginas del lado del servidor.
- Multiplataforma.
- Código bien estructurado.
- Integridad con los módulos de Java.
- La parte dinámica está escrita en Java.
- Permite la utilización de servlets.

Desventajas:

- Complejidad e inversión de tiempo para su aprendizaje.

LENGUAJE JSF

Java Server Face (JSF), es un marco de trabajo de interfaces de usuario del lado del servidor para aplicaciones web.[26]

Componentes

API e implementación de referencia para representar componentes de interfaz de usuario y manejar su estado, manejo de eventos, validación del lado del servidor y conversión de datos; también permite definir las reglas de navegación entre páginas.

Librería de etiquetas Java Server Pages, personalizadas para dibujar componentes de interfaz dentro de una página JSP.

Una aplicación JSF contiene:

- JavaBeans, para contener datos y funcionalidades específicas de la aplicación.
- Páginas (JSP).
- Beans, clases de utilidad del lado del servidor.
- Librería de etiquetas personalizadas para dibujar componentes UI en una página.
- Librería de etiquetas personalizadas para representar manejadores de eventos, validadores y otras acciones.
- Validadores, manejadores de eventos y manejadores de navegación.

[26] Álvarez Miguel Ángel, "Java Server Faces", www. proyectoremar.tripod.com

CONCLUSIÓN:

Existen diferentes tipos de lenguajes de programación, con sus características propias, así como sus ventajas y desventajas, pero cada uno permite llegar a un fin común, que es el desarrollar aplicaciones orientadas a la web, a través del uso de las herramientas que cada uno de estos proporcionan a los programadores.

Es así que tomando en consideración el presente trabajo, se ha decidido utilizar como herramienta de desarrollo del software prototipo de firewall y servidor proxy multiplataforma, el lenguaje de programación JSP, porque el proyecto, basará su diseño en la tecnología Java, es así que considerando sus características, ventajas y desventajas, JSP es el lenguaje más apto en base al cual se construirá el prototipo mencionado.

Así como también se utilizará JSF, como el marco de trabajo base para el diseño de las interfaces de usuario, por su versatilidad y adaptación a JSP, porque los dos lenguajes son parte de la tecnología Java.

BASES DE DATOS PARA APLICACIONES WEB

ORACLE DATABASE.

"Oracle es un sistema de gestión de base de datos relacional (RDBMS)[27] desarrollado por Oracle Corporation" [28] El paquete informático Oracle es uno los sistemas de almacenamientos de datos más completos en la actualidad.

Las características más destacadas de Oracle son:

- Soporte de transacciones

[27] RDBMS: Relational Data Base Management System

[28] Oracle Database, www.wikipedia.org

- Estabilidad
- Escalabilidad
- Soporte multiplataforma
- Rapidez
- Interactividad
- Facilidad de Administración

Ventajas y Desventajas

- La principal ventaja que posee Oracle es la capacidad ilimitada de almacenamiento de datos.

- La principal desventaja es la cantidad de tiempo en programar y sobre todo la comprensión del funcionamiento de Oracle.

SQL SERVER.

"Es un sistema de gestión de base de datos relacionales (SGBD), basado en el lenguaje Trasnsact-SQL, y específicamente en Sybase IQ, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea".[29]

El lenguaje de programación Microsoft SQL Server constituye una alternativa para la gestión de bases de datos.

Entre las principales características están:

- Soporte de transacciones.

[29] SQL Server, www.wikipedia.org

- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye un potente entorno grafico de administración
- Permite trabajar en modo cliente –servidor, donde la información y los datos se alojan en el servidor y las terminales o clientes de la red solo acceden a la información.
- Permite administrar información de otros servidores de datos.

Ventajas y Desventajas.

- La principal ventaja son las sentencias estándar, las mismas que sirven para cualquier lenguaje de base de datos.

- Entre las principales desventajas está la capacidad de almacenamiento de datos limitado.

MySQL

"Es un sistema de gestión de base de datos relacional, multihilo y multiusuario"[30].

Se puede decir que desde el punto de vista este programa se ofrece bajo GNU y GPL, que permite el uso de cualquier licencia compatible, pero existe un inconveniente en la empresa al implementar o mejorar sus características porque necesitan la adquisición de una licencia específica para que les permita el uso.

Entre sus principales características son:

- Uso de multihilos mediante hilos del kernel.

[30] MySQL, www.wikipedia.org

- Usa tablas para búsquedas rápidas con compresión de índice.

- Tablas en memoria temporales.

- Completo soporte para operadores y funciones en cláusulas select y where.

- Completo soporte para cláusulas group by y order by, soporte de funciones de agrupación

- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.

- Soporta gran cantidad de datos. MySQL Server tiene bases de datos de hasta 50 millones de registros.

- Los clientes se conectan al servidor MySQL usando sockets TCP/IP en cualquier plataforma.

- MySQL contiene su propio paquete de pruebas de rendimiento proporcionado con el código fuente de la distribución de MySQL.

Ventajas y Desventajas

- La principal ventaja que posee MySql es la capacidad ilimitada de almacenamiento de datos.

- La principal desventaja es la escaza cantidad de personas preparadas para trabajar bajo este gestor de manejo de datos.

CONCLUSIÓN:

Son diversas las bases de datos que existen para el almacenamiento de información, cada una de ellas reúne características propias que la hacen útiles para determinadas aplicaciones.

Para el almacenamiento de los datos que se manejarán en el presente proyecto, se hará uso de las características que tiene la base de datos MySQL, teniendo en cuenta que el proyecto será aplicado en dos plataformas operativas diferentes como son Windows y Linux, esta base de datos, puede ser utilizada en los dos sistemas operativos indicados, teniendo en consideración su principal característica multiplataforma y además posee una licencia GPL, que permite su utilización libremente, sin ningún tipo de restricciones.

3.1.1.2. Factibilidad Técnica

Para el análisis este tipo de factibilidad se tomará en cuenta:

Requisitos técnicos de los equipos:

Los requisitos mínimos con los que deberán contar los equipos en donde se instalará el software prototipo de firewall y servidor proxy multiplataforma con tecnología java, serán:

Instalación de Windows XP Professional: [31]

- Procesador Pentium a 233 megahercios (MHz) o mayor velocidad (se recomienda 300 MHz)

- Al menos 64 megabytes (MB) de RAM (se recomienda 128 MB)

[31] Requisitos del sistema para los sistemas operativos Windows XP, www.microsoft.com

- Un mínimo de 1,5 gigabytes (GB) de espacio disponible en el disco duro
- Unidad de CD-ROM o DVD-ROM
- Un teclado y un mouse de Microsoft, o algún otro dispositivo señalador compatible
- Adaptador de vídeo y monitor con una resolución Super VGA (800 x 600) o mayor
- Tarjeta de sonido
- Altavoces o auriculares

Los requisitos mínimos de hardware que debe tener un equipo para instalar Linux en du distribución Ubuntu son:[32]

- Procesador Intel™ o compatible a 200 MHz
- 256 Mb de RAM
- Tarjeta SVGA
- 3 GB de espacio libre en el disco duro

Los requisitos mínimos de hardware que debe tener un equipo para instalar la base de datos MySQL 5.0 son:

- Procesador: Pentium III, 600 MHz o superior
- Memoria: 256 MB o superior

[32] Requisitos mínimos de instalación de Ubuntu, www.blogspot.com

- Disco Duro: 300 Mb disponibles

- Video: 8 MB resolución 800 x 600

CONCLUSIÓN:

Es factible técnicamente el desarrollo del software prototipo, porque los equipos en los que se realizará la implementación y pruebas, reúnen los requerimientos mínimos de instalación del programa, tanto de las plataformas operativas como de la base de datos, para que la ejecución del mismo tenga los resultados esperados.

3.1.1.3. Factibilidad Económica

Análisis Costo / Beneficio

EL software prototipo de firewall y servidor proxy multiplataforma que se desarrollará, tiene como objetivo proporcionar seguridad a una red de comunicaciones, a través del control de puertos y accesos de agentes no autorizados a la misma.

Para verificar la vialidad que tiene el desarrollo y ejecución del programa mencionado, se realizará el análisis costo/beneficio, para lo cual se concentrará el estudio en los puntos claves en los que se basa la implementación del sistema propuesto, el mismo que permitirá registrar la factibilidad económica de la aplicación, si fuera puesta al alcance del mercado tecnológico.

El Análisis de Costo / Beneficio para el desarrollo durante el primer año será:

DESCRIPCIÓN	CANTIDAD	COSTO (Dólares)	BENEFICIO	COSTO/BENEFICIO (Dólares)
RECURSO HARDWARE				
Equipos	2	$ 800,00	Incremento del nivel académico en cuanto se refiere a manejo de software	$ 800,00
RECURSOS SOFTWARE				
Windows versión XP	2	$ 150,00	Conocer como es el funcionamiento interno de un firewall y las funciones que desempeña un servidor proxy al desarrollar un prototipo de funcionalidad.	$ 300,00
Linux distribución Ubuntu	2			
RECURSO HUMANO				
Director de Proyecto	1		Trabajo con dos sistemas operativos a través del desarrollo de una misma aplicación, en un lenguaje con característica multiplataforma.	$ 800,00
Programadores	2			
GASTO MENSUAL				
Internet	300 hrs	$ 225,00		
Transporte		$ 150,00	Mayor conocimiento de herramientas de programación orientada a la web	$ 500,00
Insumos de Papelería		$ 240,00		
Derecho Denuncia Tesis	2	$ 300,00		
Derecho de Grado	2	$ 270,00		
Derecho de especies	2	$ 130,00		
Delegado consejo carrera	2	$ 50,00		
Derecho de Mención	2	$ 10,00		
COSTO TOTAL		$ 2325,00		$ 2.400,00

Cuadro 3.1: "Costo/Beneficio Sistema Prototipo de Firewall y Servidor Proxy Multiplataforma con Tecnología Java"

Nota: Los valores considerados en la columna costo/beneficio, son un estimado del posible valor que se podría obtener si el software tuviera una aplicación comercial.

Análisis

La relación que da como resultado este análisis es de $1.03 de ingreso por cada dólar invertido ($2400/$2325), este ingreso es factible para el desarrollo de la aplicación deseada.

Punto de Equilibrio:

Para encontrar este valor se debe tomar en cuenta la inversión inicial y el beneficio esperado, el cual se muestra en la siguiente operación:

Datos:

- Costo Inicial: $2325
- Beneficio $2400 para un tiempo estimado de un año

Desarrollo:

Fórmula= (Costo /Beneficio)*12(meses)

El punto de equilibrio es: 11.63

Periodo de Devolución:

Es el tiempo requerido para recuperar el monto inicial de una inversión de capital, a través de este método, se calcula la cantidad de tiempo que se tomaría para lograr un flujo de caja positivo igual a la inversión total necesaria para el desarrollo del proyecto mencionado.

Datos:

- Costo Inicial: $2325

- Beneficio $2400 para un tiempo estimado de un año

Valor asegurado: $500

Fórmula:

Período de Devolución = [(Costo – Valor asegurado) /total ingresos incrementados y/o reducción de gastos] X 12 (Meses)

PD= [($2325 - $500) /$2400] * 12(meses)

El Período de Devolución es: 9.13

Conclusión:

Se ha generado ganancia como beneficio de inversión dentro de un tiempo determinado, por lo tanto, el desarrollo del proyecto mencionado, es viable y factible, porque los beneficios que trae consigo superan a los costos necesarios para la construcción del prototipo de firewall y servidor proxy multiplataforma con tecnología java.

3.1.2. ESTUDIO DE POLÍTICAS DE SEGURIDAD[33]

Una Política de Seguridad es un conjunto de reglas, por medio de las cuales se determinará y definirá las normas de uso que se deberá seguir en una red de comunicaciones para la protección de la información manejada.

El estudio de las políticas de seguridad, permite la correcta toma de decisiones en cuanto a medidas de seguridad se refiere, para lo cual es necesario analizar cuál

[33] Políticas de Seguridad de Firewalls, www.textoscientificos.com

es la seguridad con la que cuenta la red y que nivel de funcionalidad se pretende ofrecer al proponer nuevas normas de seguridad informática.

Es preciso determinar objetivos de seguridad, los cuales permitan resolver la selección de las herramientas que harán efectivos dichos objetivos, para lo cual se debe considerar las necesidades que se desee satisfacer y encontrar un punto de equilibrio en base al análisis de:

- Servicios ofrecidos vs la seguridad provista: cada servicio ofrecido a un usuario tiene su propio riesgo de seguridad.

- Facilidad de uso vs seguridad: un sistema muy fácil de usar permitirá el acceso a casi todos los usuarios y por lo tanto serán menos seguro.

- Costo de la seguridad vs riesgo de pérdida: existen muchos costos de seguridad: monetarios, de desempeño y facilidad de uso. Los riesgos de pérdida pueden ser de privacidad, de datos, y servicios.

Cada tipo de costo debe ser balanceado con respecto a cada tipo de perdida.

El objetivo principal del uso de una política de seguridad es:

- Informar a los usuarios de la red sus obligaciones para proteger a los recursos de la red.

- Especificar los mecanismos a través de los cuales estos requerimientos pueden ser logrados.

- Proveer una guía que permitirá implementar, configurar y controlar los sistemas de la red para determinar su conformidad con la política.

Una política de seguridad debe asegurar cuatro aspectos fundamentales en una solución de seguridad:

- autenticación,
- control de acceso,
- integridad y
- confidencialidad.

Los principales componentes de una política de seguridad son:

- Una política de privacidad: define expectativas de privacidad con respecto a funciones como monitoreo, registro de actividades y acceso a recursos de la red.

- Una política de acceso: que permite definir derechos de acceso y privilegios para proteger los objetivos clave de una pérdida o exposición mediante la especificación de guías de uso aceptables para los usuarios con respecto a conexiones externas, comunicación de datos, conexión de dispositivos a la red, incorporación de nuevo software a la red, etc.

- Una política de autenticación: que establece un servicio de confiabilidad mediante alguna política de contraseñas o mecanismos de firmas digitales, estableciendo guías para la autenticación remota y el uso de dispositivos de autenticación.

- Un sistema de IT (tecnología de la información) y una política de administración de la red: describe como pueden manipular las tecnologías los encargados de la administración interna y externa. De aquí surge la consideración de si la administración externa será soportada y, en tal caso, como será controlada.

Para un adecuado diseño de las políticas de seguridad que se pretende satisfacer con el desarrollo del proyecto, es preciso conocer:

- Qué recursos serán protegidos,
- De qué se pretende proteger a la red,
- Cuáles serán las amenazas que se presentarán y afectarán a los recursos manejados,
- Cuáles son las medidas que se implementarán para proteger dichos recursos.

El diseño de las políticas de seguridad serán determinadas en base a:

Análisis de riesgo

Mediante el desarrollo del sistema prototipo de firewall y servidor proxy, se desea conocer que se protegerá y cuáles serán los principales agentes externos considerados una amenaza para la red, así como las medidas adecuadas para salvaguardar la información manejada.

Identificar los objetivos clave

Los recursos que serán protegidos mediante el sistema son:

- Hardware: Unidades de procesamiento y puertos
- Software: Plataformas operativas
- Datos: Bases de datos, e información en general.

Identificar las amenazas

Las amenazas de las cuales se pretende proteger a la red, con la implementación del software mencionado es:

- Accesos no autorizados
- Instalación de Programas no autorizados
- Acceso de usuarios sin permiso

El diseño de buenas políticas de seguridad, junto con el adecuado manejo que el usuario proporcione al sistema, permitirá controlar y restar las amenazas que se presentan en una red para evitar la pérdida de información en la misma.

3.2. FIREWALLS

Un firewall (cortafuegos) es una barrera que controla el flujo del tráfico entre los host, los sistemas de redes, y los dominio, es un tipo de tecnología que ayuda a prevenir el acceso de intrusos a un equipo, sean estos agentes externos o pertenecientes a la misma red y que no tengan debida autorización, controlando la entrada o salida de datos del sistema al cual brinda protección.

Los cortafuegos pueden ser implementados a nivel de hardware o software, o a su vez una combinación de ambos.

Para que un firewall sea efectivo, todo tráfico de información a través del Internet deberá pasar a través del mismo, en donde podrá ser inspeccionada la información.

El firewall podrá únicamente autorizar el paso del tráfico, y el mismo será inmune a la penetración; desafortunadamente, este sistema no puede ofrecer protección alguna una vez que el agresor lo traspasa o permanece entorno a este.

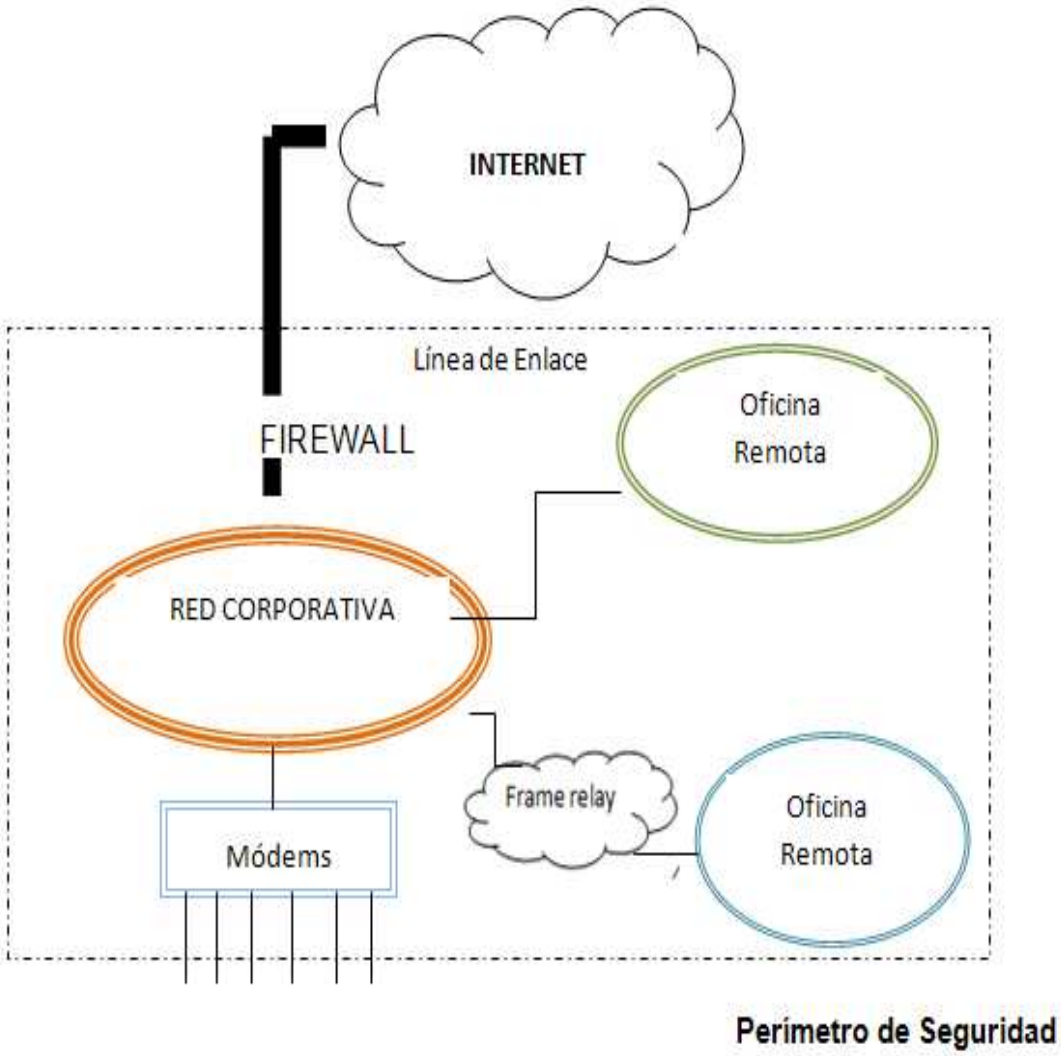

Fig. 3.1: "Política de Seguridad de un Firewall"

Fuente: Madrigal Daniel Ramón, "Firewall y Seguridad en la red", www.monografías.com

Como se muestra en la Fig. 3.1, el firewall es parte de una política de seguridad completa que crea un perímetro de defensa diseñada para proteger las fuentes de información, en donde, todos los puntos potenciales de ataque en la red podrán ser protegidos con el mismo nivel o perímetro de seguridad.

3.2.1. TIPOS DE FIREWALLS[34]

Los Firewalls, están clasificados de acuerdo a la funcionalidad que tienen, conociéndose los siguientes:

3.2.1.1. Firewall de Software

Dentro de este tipo de Firewall se conocen dos tipos:

1. Firewall de Nivel de Red

- **Cortafuegos de capa de red o de filtrado de paquetes**

Funciona a nivel de red (nivel 3) de la pila de protocolos (TCP/IP) como filtro de paquetes IP, este tipo de cortafuegos se permiten filtrados según campos de nivel de transporte (nivel 4) como el puerto origen y destino, o a nivel de enlace de datos (nivel 2) como la dirección MAC. Es uno de los principales cortafuegos, pues es eficaz y transparente pero difícil de configurar.

2. Firewall de Nivel Aplicación

- **Cortafuegos de capa de aplicación**

Trabaja en el nivel de aplicación (nivel 7), de manera que los filtrados se pueden adaptar a características propias de los protocolos de este nivel

Un cortafuego de nivel 7 de tráfico HTTP, suele denominarse proxy, y permite que los computadores de una organización entren a Internet de una forma controlada.

Un proxy oculta de manera eficaz las verdaderas direcciones de red.

[34] Firewalls, www.wikipedia.org, último acceso: 6 de octubre del 2009

3.2.1.2. Firewall de Hardware

Los firewall de hardware son dispositivos que se colocan entre el router y la conexión telefónica, no es necesario configurarlos cada vez que se instala el sistema operativo y no consumen recursos del sistema.

3.2.2. CARACTERÍSTICAS DE UN FIREWALL[35]

Un firewall, presenta las siguientes características y por ende ventajas en la seguridad para una red:

- Protección de la Red

Mantiene alejados a los piratas informáticos (crakers) de la red, al mismo tiempo que permite acceder a varios usuarios al sistema.

- Control de acceso a los recursos de la red

Al encargarse de filtrar los paquetes al resto de equipos de la red, el firewall es idóneo para implementar en el los controles de acceso.

- Control de uso de internet

Permite bloquear el material no adecuado, determinar que sitios se puede visitar en la red interna y llevar un registro de los mismos.

- Concentra la seguridad

Uno de las características de mayor relevancia que tiene un firewall, es la de enfrentar los ataques externos y vigilar la red a través de un continuo monitoreo.

---

[35] Karanjit Siyan y Chris Hare. (1997): "Firewalls y la seguridad en internet", Prentice-hall Hispanoamericana, S.A

- Control y estadísticas

Permite controlar el uso de internet en el ámbito interno, conocer los intentos de conexiones desde el exterior y detectar actividades sospechosas.

- Auditoría y registro de internet

Permite la detección de los cuellos de botella potenciales y el ancho de banda sobre el que trabaja la red.

3.2.3. LIMITACIONES DE UN FIREWALL

Como todo sistema, un firewall presenta limitaciones, entre las que se encuentran:

- No puede proteger contra aquellos ataques cuyo tráfico no pase a través de él.

- No puede proteger de las amenazas a las que está sometido por ataques internos o usuarios negligentes

- No puede proteger contra los ataques de ingeniería social.

- No puede proteger contra los ataques posibles a la red interna por virus informáticos a través de archivos y software.

- No protege de los fallos de seguridad de los servicios y protocolos cuyo tráfico esté permitido.

3.2.4. POLÍTICAS DE UN FIREWALL[36]

Hay dos políticas básicas en la configuración de un firewall:

- **Política restrictiva:**

Esta política, deniega todo el tráfico excepto el que está explícitamente permitido. El firewall obstruye todo el tráfico y hay que habilitar expresamente el tráfico de los servicios que se necesiten.

- **Política permisiva:**

Mediante esta política, se permite todo el tráfico excepto el que esté explícitamente denegado. Cada servicio potencialmente peligroso necesitará ser aislado básicamente caso por caso, mientras que el resto del tráfico no será filtrado.

De las políticas mencionadas, la más segura es la restrictiva, porque es más difícil permitir por error, un tráfico potencialmente peligroso; mientras que en la política permisiva es posible que no se haya contemplado algún caso de tráfico peligroso y sea permitido por omisión.

El firewall que se pretende desarrollar, tiene la característica multiplataforma, es decir, el software podrá ser utilizado en diferentes plataformas operativas, en este caso se lo implementará en el sistema operativo Windows, versión XP y Linz, distribución Ubuntu.

Será un firewall de software, para la capa aplicación y la capa red, mediante el desarrollo del mismo, se podrá conocer cómo trabaja en una red de

[36] Firewalls, www.wikipedia.org, último acceso: 6 de octubre del 2009

comunicaciones y de qué manera se proporciona la seguridad y protección de la información que se maneja.

3.3. SERVIDOR PROXY

Un servidor proxy es aquel que sirve de intermediario entre el internet y las computadoras de que conforman una red, constituye una herramienta de seguridad que permite controlar vulnerabilidades que se presenten en una red de comunicaciones.

3.3.1. FUNCIONAMIENTO DE UN SERVIDOR PROXY[37]

Un proxy permite a otros equipos conectarse a una red de forma indirecta a través de él. Cuando un equipo de la red desea acceder a una información o recurso, es realmente el proxy quien realiza la comunicación y a continuación traslada el resultado al equipo inicial.

En unos casos esto se hace así porque no es posible la comunicación directa y en otros casos porque el proxy añade una funcionalidad adicional, como puede ser la de mantener los resultados obtenidos en una caché que permita acelerar sucesivas consultas coincidentes.

En la Fig. 3.2, se especifica en detalle como es el funcionamiento del proxy en una red de datos con conexión a internet.

[37] Proxy, www.wikipedia.org, último acceso 12 de octubre 2009

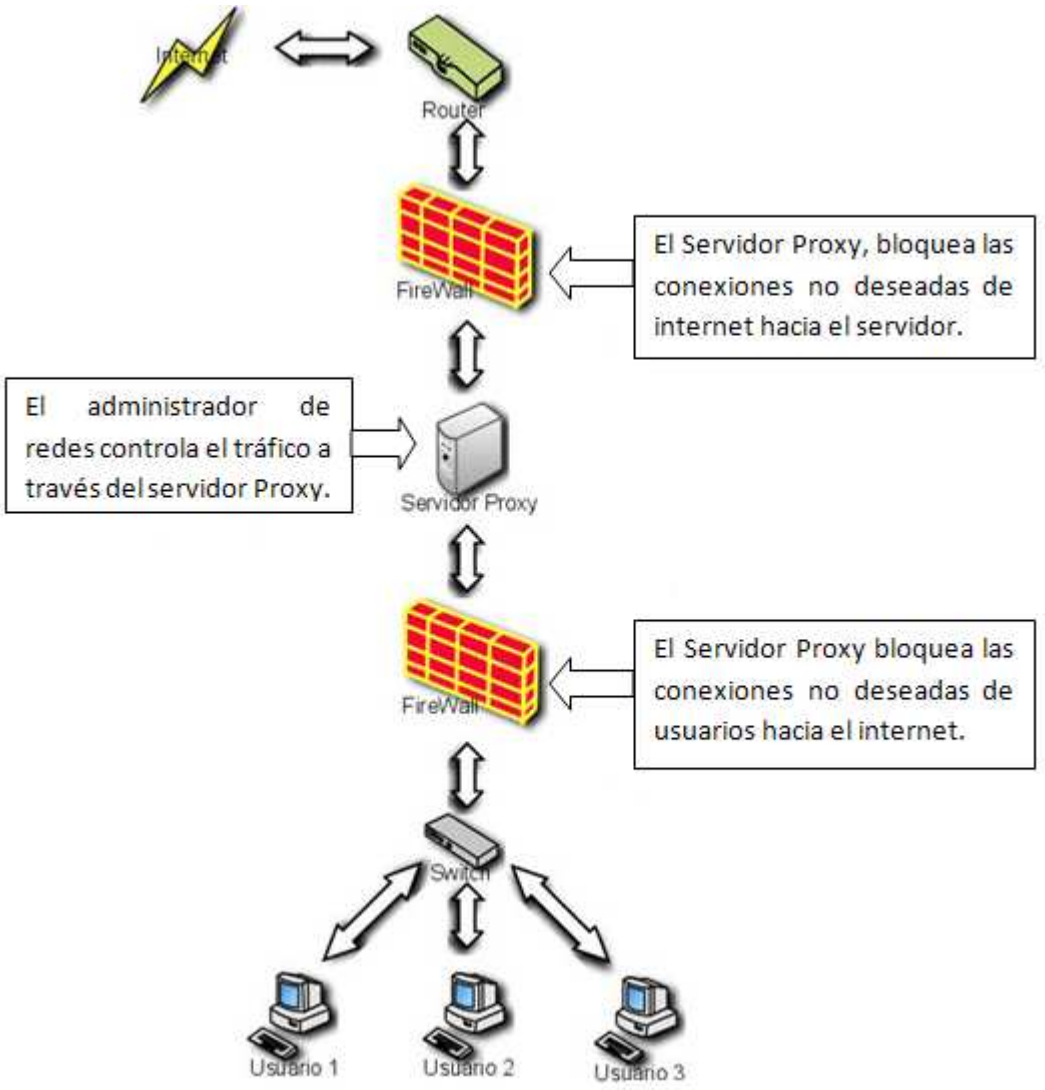

Fig. 3.2: "Funcionamiento de un Servidor Proxy"

Fuente: Servidores Proxy, www.tallerdigitalvw.com

3.3.2. VENTAJAS

El uso de un servidor proxy dentro de una red de comunicaciones, permitirá:

- **Control:** para limitar y restringir los derechos de los usuarios, y dar permisos sólo al proxy.

- **Ahorro:** Porque solo el proxy realizará el control de la red.

- **Velocidad:** Si varios clientes van a pedir el mismo recurso, el proxy, por medio del caché proporcionará una respuesta inmediata, porque guarda en memoria caché las peticiones que fueron solicitadas en un principio.

- **Filtrado:** El proxy puede negarse a responder algunas peticiones si detecta que están prohibidas.

- **Anonimato:** Si todos los usuarios se identifican como uno sólo, es difícil que el recurso accedido pueda diferenciarlos.

3.3.3. DESVENTAJAS

En general (no sólo en informática), el uso de un intermediario puede provocar:

- **Abuso:** Al estar dispuesto a recibir peticiones de muchos usuarios y responderlas, es posible que haga algún trabajo que no toque. Por tanto, ha de controlar quién tiene acceso y quién no a sus servicios, cosa que normalmente es muy difícil.

- **Carga:** Un proxy ha de hacer el trabajo de muchos usuarios.

- **Intromisión:** Es un paso más entre origen y destino, y algunos usuarios pueden no querer pasar por el proxy. Y menos si hace de caché y guarda copias de los datos.

- **Incoherencia:** Si hace de caché, es posible que se equivoque y dé una respuesta antigua cuando hay una más reciente en el recurso de destino. En realidad este problema no existe con los servidores proxy actuales, ya que se conectan con el servidor remoto para comprobar que la versión que tiene en cache sigue siendo la misma que la existente en el servidor remoto.

- **Irregularidad:** El hecho de que el proxy represente a más de un usuario da problemas en muchos escenarios, en concreto los que presuponen una comunicación directa entre 1 emisor y 1 receptor (como TCP/IP).

3.4. CONTROL DE ACCESOS

Se entiende como Control de acceso, al enfoque dado a la seguridad en redes de computadoras, el mismo que intenta unificar la tecnología de seguridad en los equipos finales siendo estos, usuarios o sistemas de autenticación para de esta manera reforzar la red de comunicaciones.[38]

El control de acceso, está dirigido en conjunto, tanto al computador o equipo perteneciente a la red, como al conjunto de protocolos empleados para definir como asegurar los nodos de la misma, antes de que algún agente externo pueda acceder a esta.

El principal objetivo que trae consigo el controlar el acceso a la red, es el diseño de políticas de seguridad y continuos controles de los recursos a los que pueden acceder los usuarios o dispositivos tanto internos como externos.

Los principales objetivos del control de accesos son:

- Prevenir en los equipos finales la falta de antivirus, parches, o software de prevención de intrusión de hosts y acceder así a la red poniendo en riesgo a otros equipos, expandiendo virus informáticos.

- Definir políticas de seguridad pertinentes al manejo de equipos dentro de una red, y a la aceptación de recursos ajenos a la misma.

[38] Control de accesos, www.wikipedia.org, último acceso: 26 de junio de 2009

- Reforzar las políticas de acceso en base a identidades de usuarios autenticados.

3.4.1. CONTROL Y MANEJO DE PUERTOS DE COMUNICACIÓN

Un puerto de comunicación, es una herramienta que permite manejar e intercambiar datos entre un computador y sus diferentes periféricos.
Por el contrario, un puerto TCP/IP, es una numeración lógica que se asigna a las conexiones, tanto en el origen como en el destino.

El permitir o denegar acceso a los puertos es importante porque las aplicaciones servidoras[39] deben 'escuchar' en un puerto conocido de antemano para que un cliente pueda conectarse; esto quiere decir que cuando el sistema operativo recibe una petición a ese puerto, la pasa a la aplicación que escucha en él, si existe alguna, y a ninguna otra.

Los servicios más habituales tienen asignados puertos de dominio más conocido, de esta manera, cuando se solicita un determinado servicio, el navegador realiza una conexión al puerto asignado para la aplicación pertinente, y si el número de puerto no se conociera, no se podría recibir el servicio solicitado.

Un puerto puede estar en tres estados[40]:

Abierto: Acepta conexiones. Hay una aplicación escuchando en este puerto. Esto no quiere decir que se tenga acceso a la aplicación, sólo que hay posibilidad de conectarse.

[39] Aplicaciones que aceptan conexiones originadas en otro equipo.

[40] Estados de los puertos, www.seguridades.com

Cerrado: Se rechaza la conexión. Probablemente no hay aplicación escuchando en este puerto, o no se permite el acceso por alguna razón. Este es el comportamiento normal del sistema operativo.

Bloqueado o Sigiloso: No hay respuesta. Este es el estado ideal para un cliente en Internet, de esta forma ni siquiera se conoce si el ordenador está conectado. Normalmente este comportamiento se debe a un cortafuego que impide el acceso de las aplicaciones, o a que el equipo está apagado.

Es preciso destacar que los puertos son puntos de acceso a aplicaciones corriendo en un ordenador, dichas aplicaciones pueden tener vulnerabilidades que pueden ser aprovechadas por otros usuarios.

Desde el punto de vista de seguridad, es recomendable permitir el acceso sólo a los servicios que sean imprescindibles, dado que cualquier servicio expuesto a Internet es un punto de acceso potencial para intrusos.

De la misma manera, es recomendable el funcionamiento sigiloso para no dar facilidades a las intromisiones de agentes externos a la red, porque cuando se descubre una vulnerabilidad, los intrusos están en disposición de atacar rápidamente a las máquinas que se consideran de tipo vulnerable.

3.5. SOCKETS

Un socket es un mecanismo de comunicación entre procesos, que les permiten emitir o recibir información de dos vías entre dos programas que se ejecutan a través de la red.

El cliente y el servidor deben ponerse de acuerdo sobre el protocolo que utilizarán.

A través de los tipos de socket, se puede definir las propiedades de las comunicaciones que se realizarán por medio de estos, es decir el tipo de comunicación que se puede dar entre cliente y servidor, las cuales pueden ser:

- Fiabilidad de transmisión.

- Mantenimiento del orden de los datos.

- No duplicación de los datos.

- El "Modo Conectado" en la comunicación.

- Envío de mensajes urgentes.

Por lo tanto se conocen dos tipos de sockets:

1.- Orientado a conexión:

- Establece un camino virtual entre servidor y cliente, fiable, sin pérdidas de información ni duplicados, la información llega en el mismo orden que se envía.

- El cliente abre una sesión en el servidor y este guarda un estado del cliente.

- El cliente utiliza la clase Socket

- El servidor utiliza la clase ServerSocket

2.- No orientado a conexión:

- Envío de datagramas de tamaño fijo. No es fiable, puede haber pérdidas de información y duplicados, y la información puede llegar en distinto orden del que se envía.

- No se guarda ningún estado del cliente en el servidor, por ello, es más tolerante a fallos del sistema.
- Tanto el cliente como el servidor utilizan la clase DatagramSocket.[41]

El mecanismo de comunicación vía sockets tiene los siguientes pasos:

1º) El proceso servidor crea un socket con nombre y espera la conexión.

2º) El proceso cliente crea un socket sin nombre.

3º) El proceso cliente realiza una petición de conexión al socket servidor.

4º) El cliente realiza la conexión a través de su socket mientras el proceso servidor mantiene el socket servidor original con nombre.

[41] Tipos de sockets, www.hispalinux.es

CAPÍTULO IV

4. DISEÑO Y CONSTRUCCIÓN DEL SOFTWARE PROTOTIPO

En este capítulo, se mostrará el diseño en base al cual se construirá el software prototipo de firewall y servidor proxy multiplataforma con tecnología Java, en donde a través del empleo del Lenguaje de Modelamiento Unificado UML, se presentarán los distintos diagramas que permitirán comprender la funcionalidad del programa mencionado, que de ahora en adelante se denominará “FIREPROX”.

Al mismo tiempo que se presentará el modelo de la interfaz, con la que el usuario tendrá interactividad, y la cual se mostrará en el software desarrollado.

De la misma manera, se conocerá cada uno de los módulos que se construyeron para manejar el software prototipo.

4.1.DISEÑO CONCEPTUAL

A través del diseño conceptual de la base de datos, se podrá conocer la información que se manejará en el sistema prototipo de firewall y servidor proxy multiplataforma, teniendo en cuenta que la base de datos está diseñada en el gestor de base de datos relacional MySQL 5.0 (Obsérvese en la Fig. 4.1).

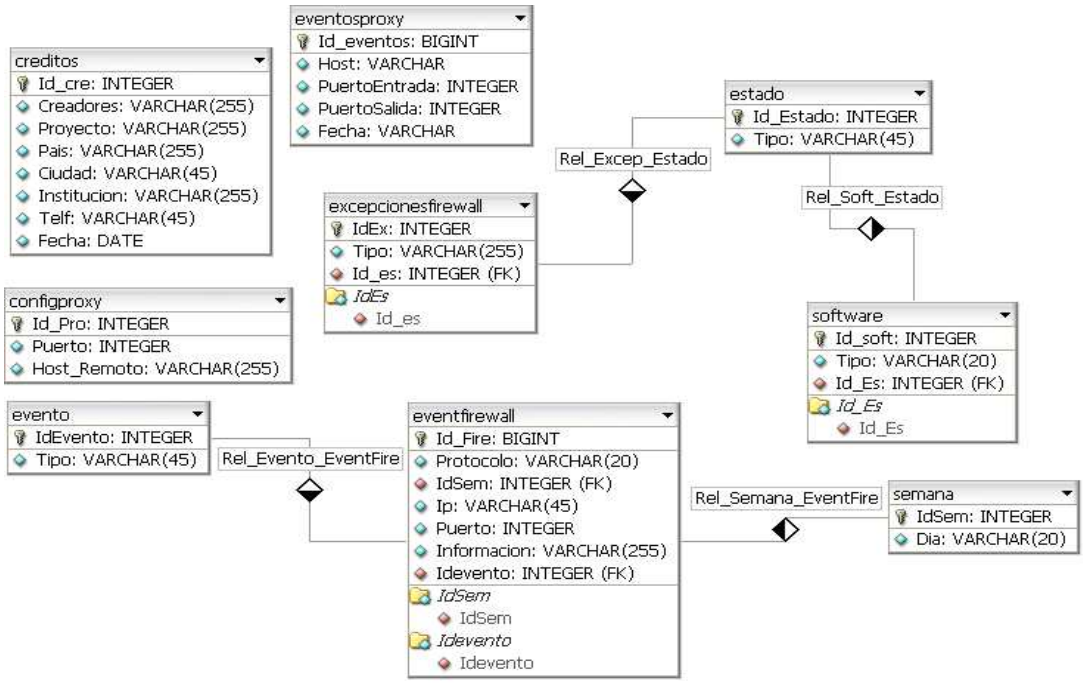

Fig. 4.1: "Modelo Conceptual FireProx desarrollado en DBDesigner"

4.2.DIAGRAMAS UML

A través del lenguaje de Modelamiento Unificado, se podrá visualizar, especificar, construir y documentar el software orientado a objetos que se desarrollará, a través de la simplificación de la realidad, para comunicar la estructura que tendrá el sistema, el comportamiento del mismo y las funciones relevantes que este cumple.

4.2.1. DIAGRAMA DE CLASES

En el siguiente diagrama, se mostrará las clases utilizadas para la construcción del prototipo de Firewall y servidor proxy multiplataforma, en donde se pueden observar las relaciones existentes entre cada una de estas. (Ver Fig. 4.2)

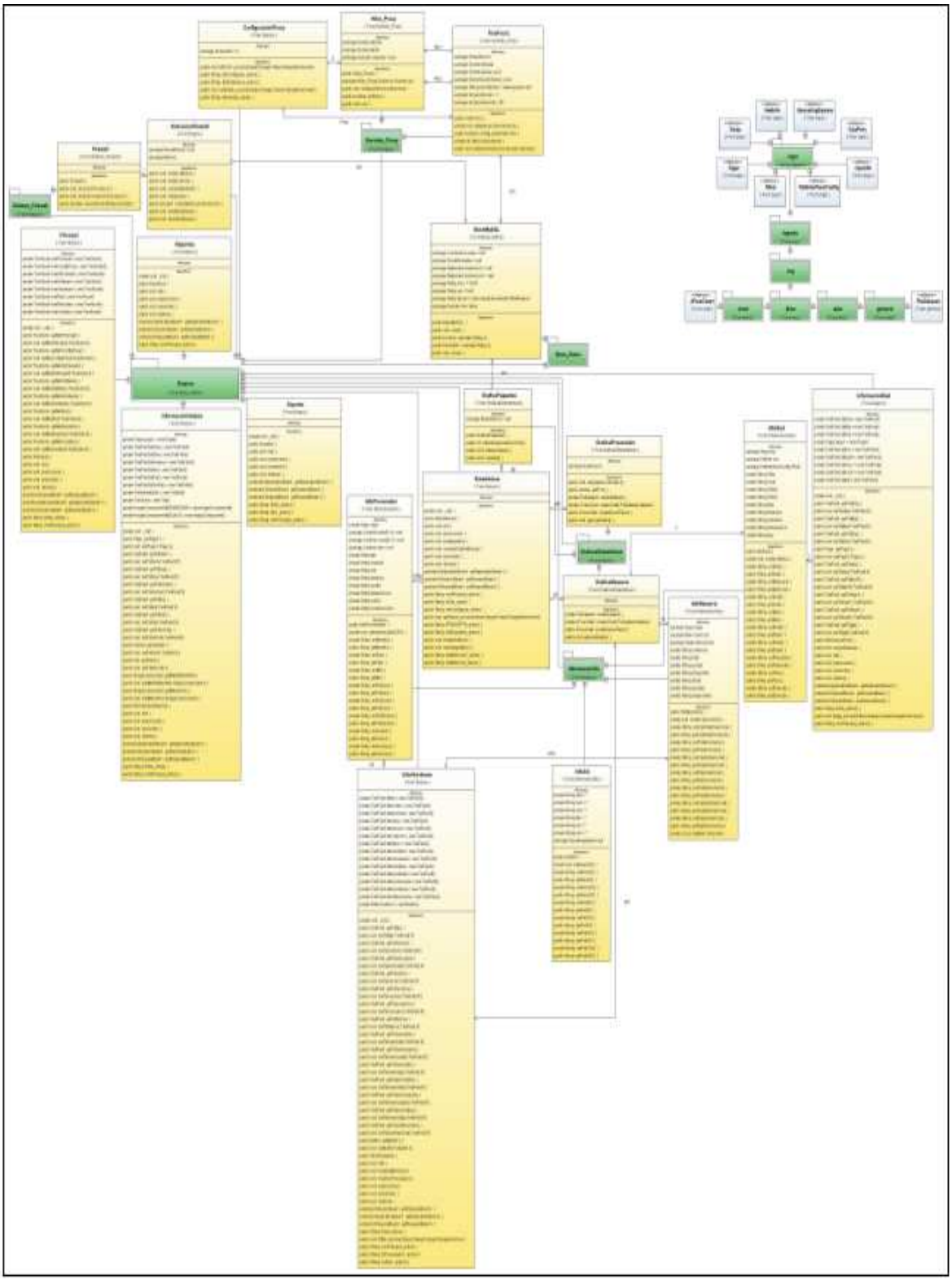

Fig. 4.2: “Diagrama UML de Clases Sistema FireProx”

Nota: Observar el diagrama completo en el Anexo Nº 2

4.2.2. DIAGRAMA DE CASOS DE USO

El siguiente diagrama permitirá conocer las distintas acciones que realizará el usuario, al interactuar con el software, entre estas están:

Presentación del Sistema: en esta etapa, el usuario, podrá conocer de que se trata el sistema, los requisitos necesarios tanto a nivel de software como de hardware, para poder ejecutar el sistema FireProx.

Configuración del Proxy: en esta etapa, el usuario, puede configurar el servidor proxy, a través de un puerto que previamente elija, para proceder con la activación de dicho servicio.

Configuración del Firewall: en esta etapa, el usuario, puede elegir activar o desactivar el sistema firewall, y al mismo tiempo configurar las principales excepciones del sistema, y a su vez capturar los paquetes que desean ingresar a la red a través de un sniffer.

Información del host: en esta etapa, el usuario, puede observar y verificar las propiedades del sistema, a nivel del software, en donde se indicará la plataforma operativa en la que se está ejecutando el sistema; a nivel de hardware, con información de los principales componentes y a nivel de red.

Estadísticas: En esta etapa, el usuario podrá visualizar un reporte gráfico del uso de la memoria, los paquetes interceptados y del uso del procesador.

Identificación de Usuarios: Se establece como único usuario al administrador quien va a ser el encargado de gestionar los servicios que proporciona FireProx

La Fig. 4.3 Muestra un diagrama UML de los principales casos de uso del sistema diseñado.

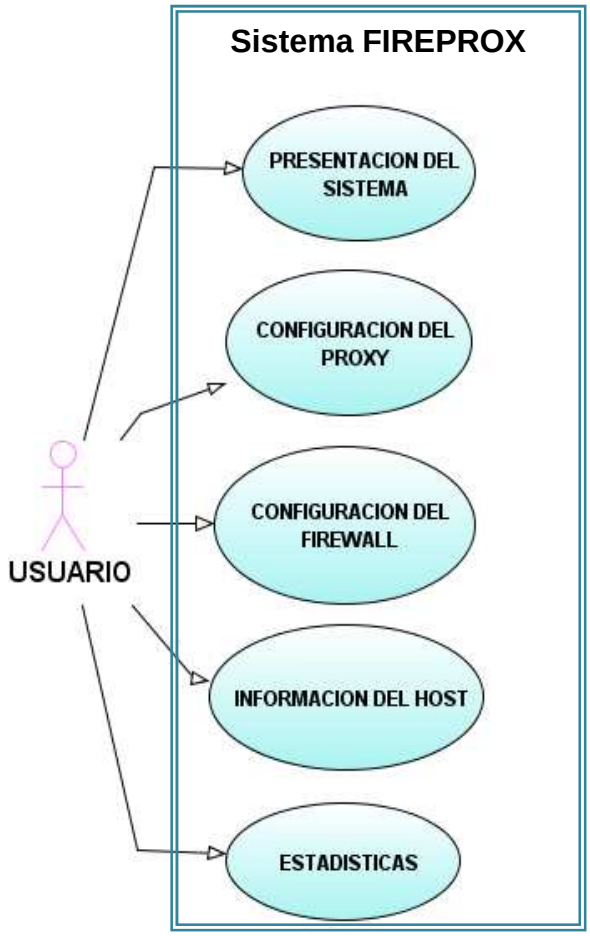

Fig. 4.3: "Diagrama UML de Casos de Uso Sistema FireProx"

Donde no hay restricción ni acciones específicas para un usuario determinado, porque el mismo usuario puede navegar y hacer uso de todo el sistema.

4.2.3. DIAGRAMA DE ACTIVIDADES

Mediante el siguiente diagrama de actividades, se conocerá el orden en el que se realizará determinada tarea dentro del sistema. (Ver Fig. 4.4)

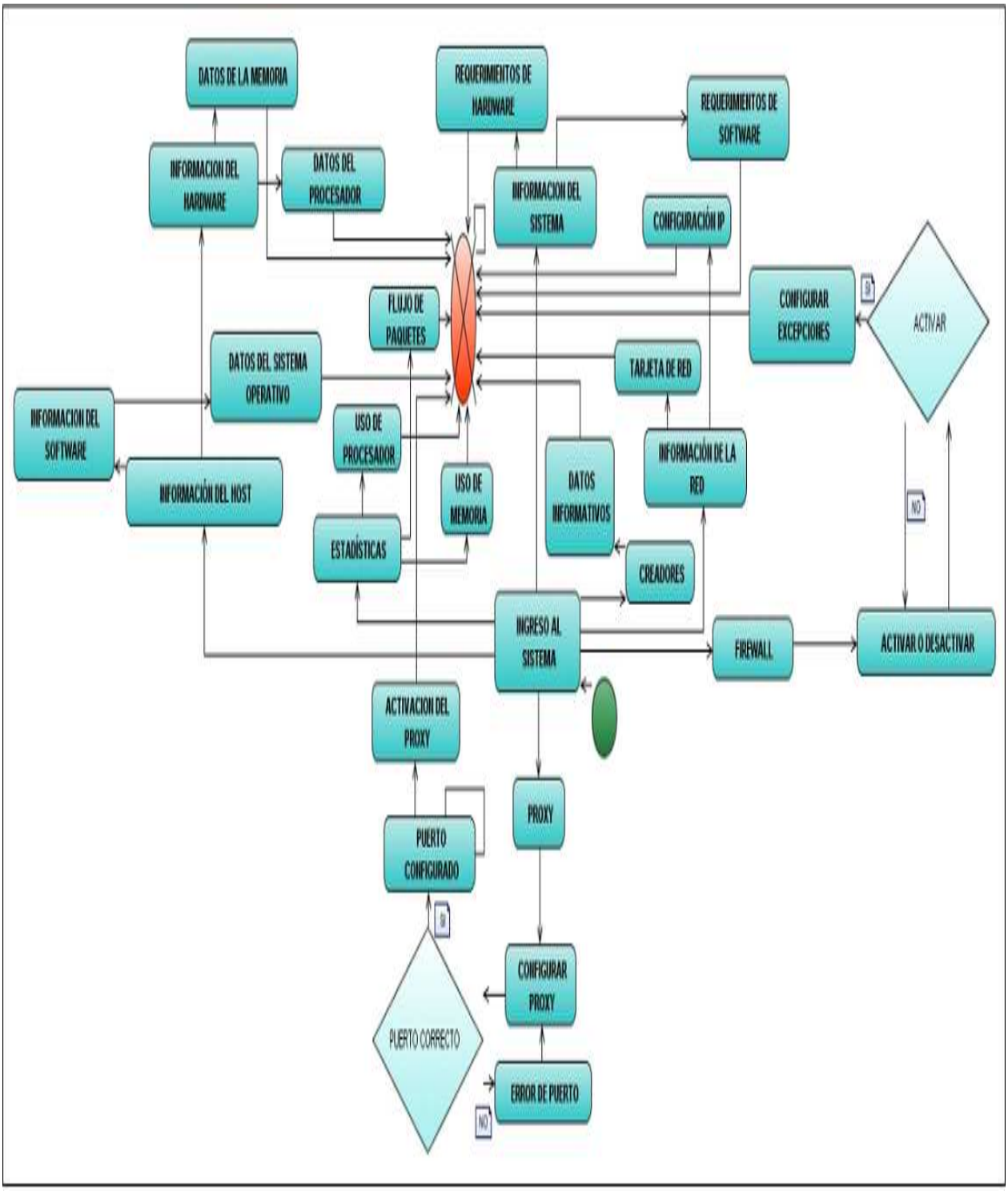

Fig. 4.4: “Diagrama UML de Actividades Sistema FireProx”

Nota: Observar diagrama completo en el Anexo Nº 3

4.2.4. DIAGRAMA DE SECUENCIA

El siguiente diagrama muestra la secuencia en la que se desencadenan las diferentes actividades a las cuales tiene acceso el usuario. La fig. 4.5, muestra el diagrama UML de Secuencia de sistema diseñado

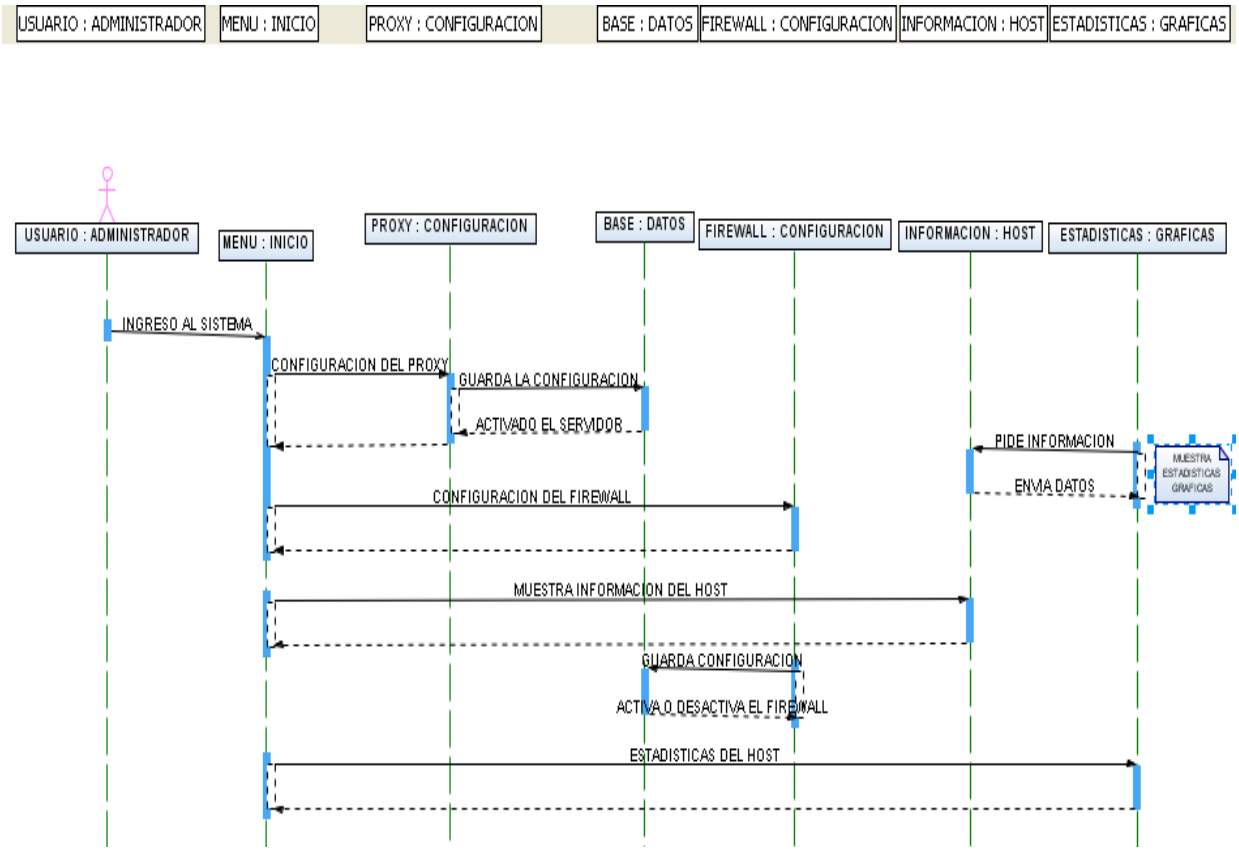

Fig. 4.5: "Diagrama UML de Secuencia Sistema FireProx"

4.2.5. DIAGRAMA DE COMPONENTES

Mediante el diagrama mostrado en la fig.4.6, se puede conocer los componentes que conforman el sistema desarrollado, los cuales son:

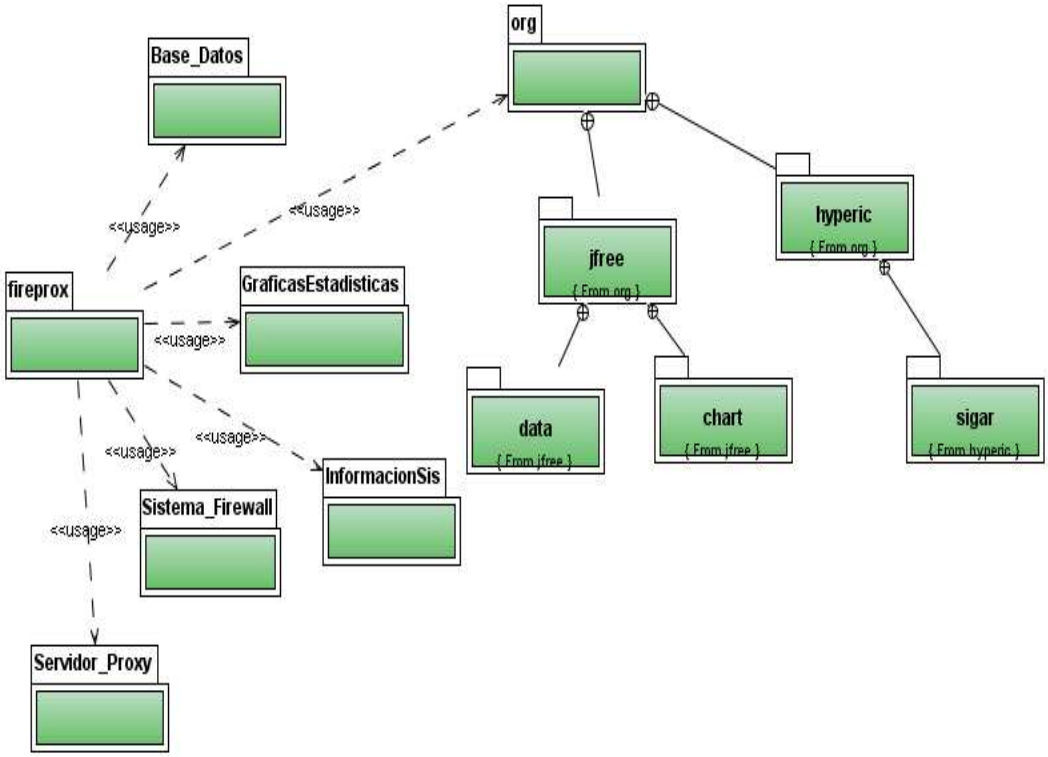

Fig. 4.6: "Diagrama UML de Componentes Sistema FireProx"

4.3. DISEÑO DE INTERFACES

Cada una de las interfaces diseñadas, tiene como base fundamental, la mostrada a continuación:

- **Logo:**

La imagen representativa del sistema

- **Menú Principal:**

En donde su ubican cada una de las opciones que permite conocer el sistema.

- **Opciones:**

Son pestañas adicionales de cada módulo.

- **Información y Formularios:**

Permite mostrar la información y realizar configuraciones en el sistema.

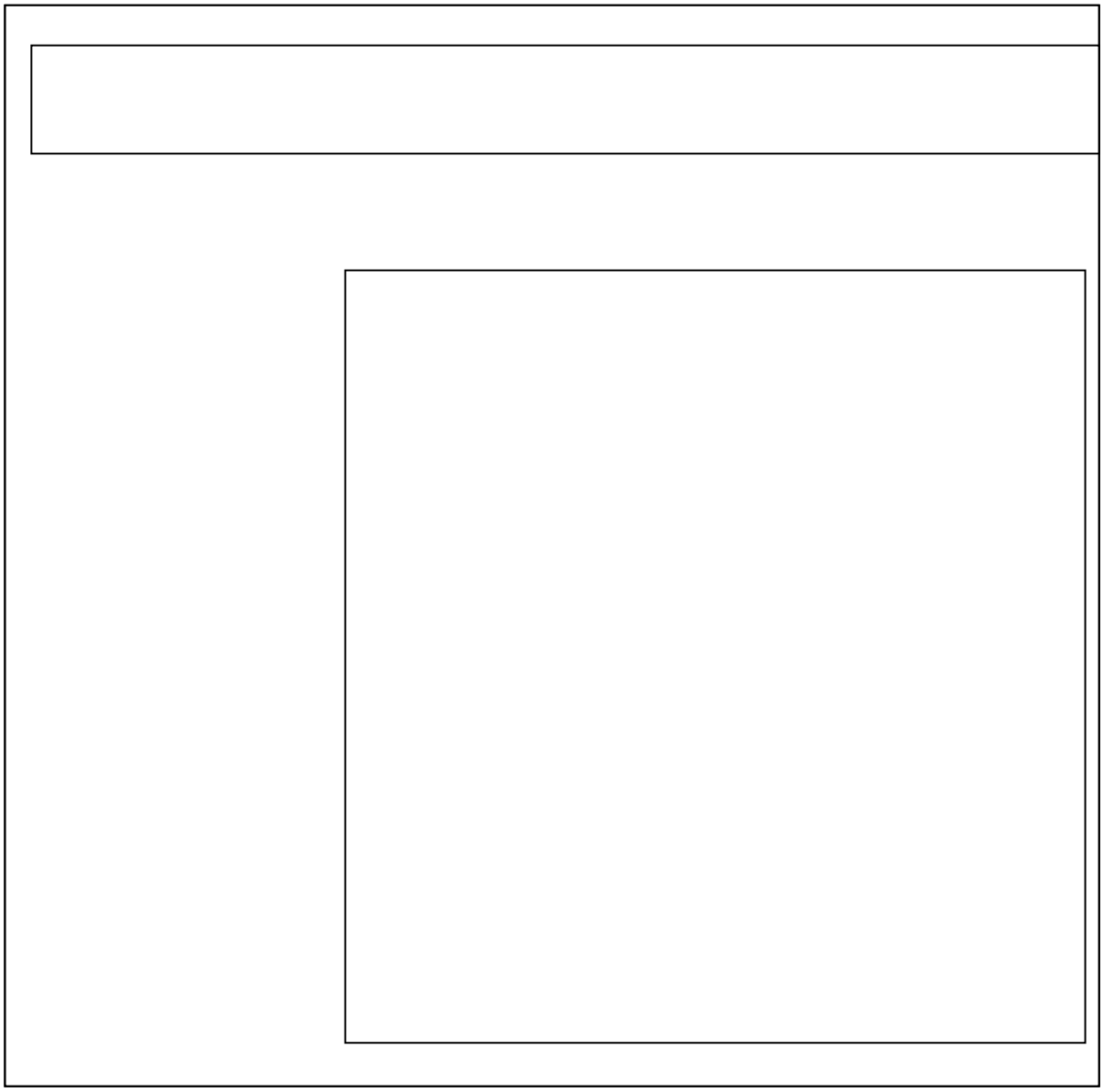

Fig. 4.7: “Maqueta base de Interface de usuario Sistema FireProx”

4.3.1. MAPA DE NAVEGACIÓN

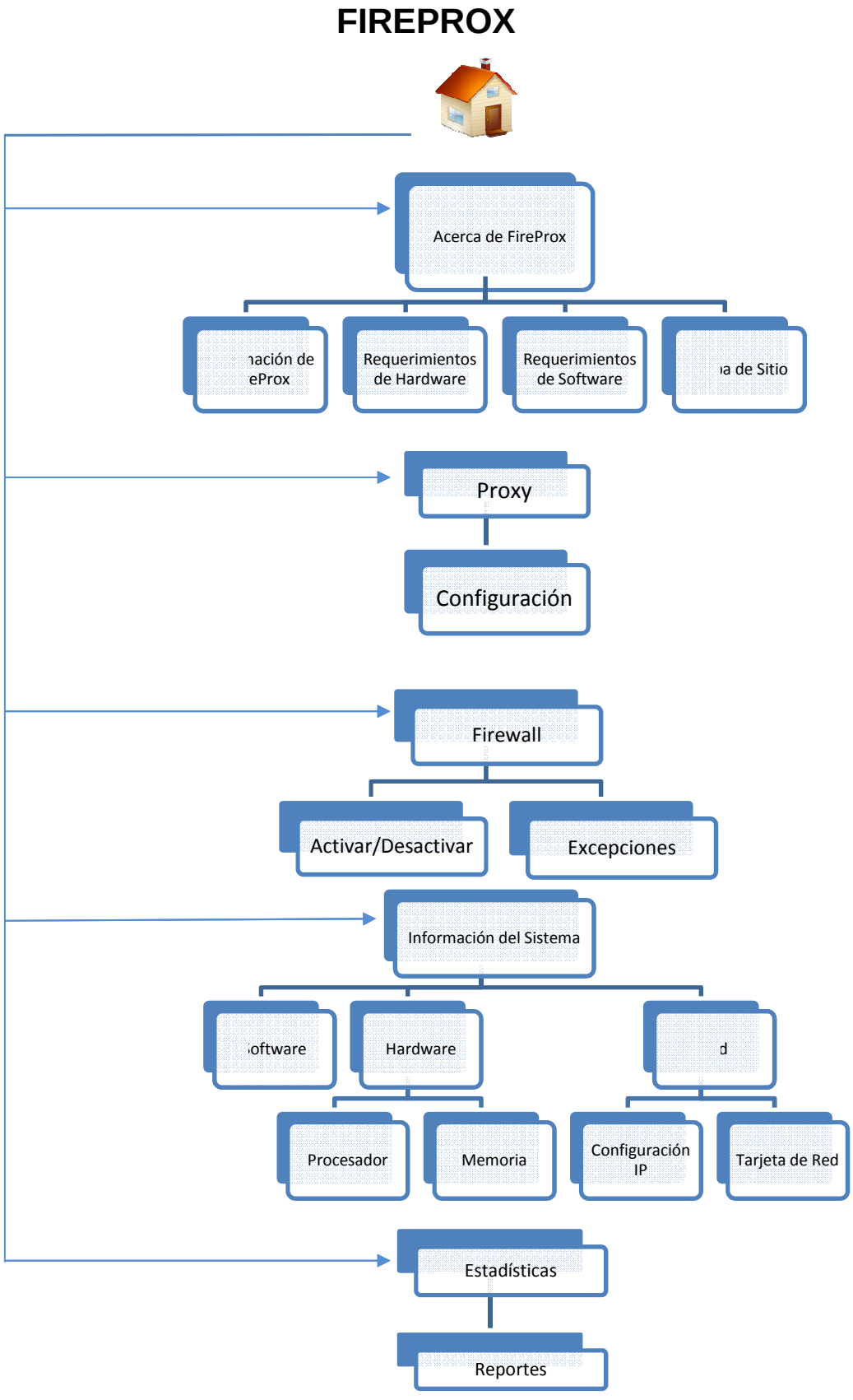

Fig. 4.8: "Mapa de Navegación Sistema FireProx"

4.4. CREACIÓN DE MÓDULOS

EL software se encuentra desarrollado en paquetes, cada paquete contiene clases que son utilizadas por cada una de las interfaces.
Los paquetes creados son los siguientes (Obsérvese en la Fig. 4.9):

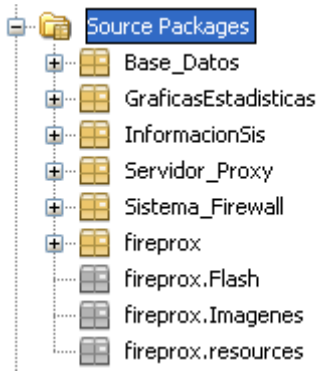

Fig. 4.9: "Paquetes del Sistema FireProx"

Paquete Base de Datos:

Permite ejecutar sentencias SQL y conexión de base de datos con MySQL, contiene la siguiente clase (Obsérvese en la Fig. 4.10):

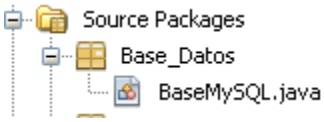

Fig. 4.10: "Paquete Base de datos del Sistema FireProx"

Paquete Gráficas Estadísticas:

Permite conocer las estadísticas de funcionamiento del sistema y de peticiones web por cliente, contiene las siguientes clases (Obsérvese en la Fig. 4.11):

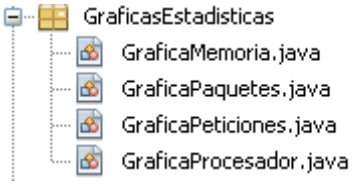

Fig. 4.11: "Paquete GraficasEstadisticas del Sistema FireProx"

Paquete Información Sistema:

Permite conocer información de hardware del host, contiene las siguientes clases (Obsérvese en la Fig. 4.12):

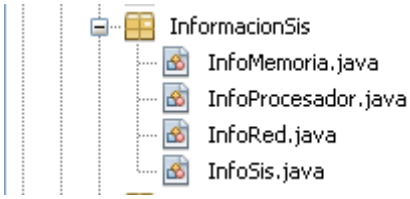

Fig. 4.12: "Paquete InformacionSis del Sistema FireProx"

Paquete Servidor_Proxy:

Permite la configuración del Servidor Proxy, a través de las siguientes clases (Obsérvese en la Fig. 4.13):

Fig. 4.13: "Paquete Servidor_Proxy del Sistema FireProx"

Paquete Sistema Firewall:

Permite establecer la configuración del sistema firewall, a través de la siguiente clase (Obsérvese en la Fig. 4.14):

Fig. 4.14: "Paquete Servidor_Firewall del Sistema FireProx"

Paquete Fireprox:

Contiene todas las clases que permiten operar a las interfaces que conforman el sistema Proxy y firewall respectivamente, estas clases son las siguientes (Obsérvese en la Fig. 4.15):

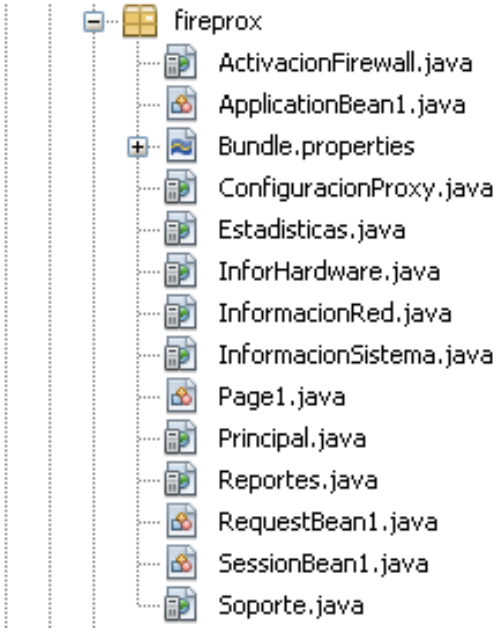

Fig. 4.15: "Paquete fireprox del Sistema FireProx"

4.4.1. CREACIÓN DEL MÓDULO DE FIREWALL

El modulo de programación del Firewall está compuesto por la clase firewall.java que se describe a continuación.

Descripción del Código de la Clase Firewall

El firewall de fireprox utiliza para su funcionamiento la clase Runtime del paquete io de java que permite la ejecución de comandos vía consola.

Se establece un método que establece la ejecución de los comandos de la siguiente manera (Observar el código 4.1):

```
    //Método que permite ejecutar un comando vía consola
public boolean ejecutacmd(String comando){
    boolean ban=true;

  String s = null;

  try{
                        // Ejecutamos el comando
      //Se utiliza la clase process del paquete io de java y la clase runtime
                        Process p = Runtime.getRuntime().exec(comando);
                        //Llamada al método que muestra la ejecución del comando
                        lecturaComando(p);
                        //Método que permite controlar los errores que pueden suceder al ejecutar un comando
                        errores(p);


                } catch (IOException e){
                        System.out.println("Excepción: ");
                        e.printStackTrace();
                        System.exit(-1);
              ban=false;
                }
  return ban;
}

}
```

Código 4.1: "Método que ejecuta un comando de la clase firewall del Sistema Fireprox"

Se establece un método que controla la salida estándar o respuesta del comando al ser ejecutado correctamente (Observar el código 4.2).

```
//leer la ejecucion del comando si se efectua
    public void lecturaComando(Process p1){
       String cad="";
       try{
         BufferedReader stdInput = new BufferedReader(new InputStreamReader(
                                   p1.getInputStream()));



                    // Leemos la salida del comando
                    System.out.println("Ésta es la salida standard del comando:\n");
                    while ((cad = stdInput.readLine()) != null) {
                            System.out.println(cad);
                    }
       }catch(IOException e){
            System.out.println("Excepción: ");
                    e.printStackTrace();
                    System.exit(-1);
       }

}
```

Código 4.2: "Método que muestra la respuesta del comando."

Se establece un método que controla la salida estándar o respuesta del comando al ser ejecutado y devolver un error (Observar el código 4.3).

```
//permite controlar errores y excepciones al ejecutar un proceso via consola
public void errores(Process p1){
    String cad="";
    try{
      BufferedReader stdError = new BufferedReader(new InputStreamReader(
                                p1.getErrorStream()));
    // Leemos los errores si los hubiera
                System.out.println("Ésta es la salida standard de error del comando (si la hay):\n");
                while ((cad = stdError.readLine()) != null) {
                        System.out.println(cad);
                }
    }catch(IOException e){
        System.out.println("Excepción: ");
                e.printStackTrace();
                System.exit(-1);
    }

}
```

Código 4.3: "Método que muestra la respuesta del comando al generar un error."

4.4.1.1 Descripción del Código de llamada a la clase Firewall desde la interfaz JSF (Activación Firewall.jsp)

Se establece los paquetes en la cabecera de la clase de activación del firewall con el objetivo de utilizar cada uno de los métodos que contienen; estos paquetes son los siguientes (Obsérvese el código 4.4):

```
import java.sql.*;
import Base_Datos.BaseMySQL;
import Sistema_Firewall.Firewall;
import capturadorsniffer.*;
```

Código 4.4: "Paquetes de la cabecera de la Interfaz ActivacionFirewall.jsp"

El espacio de código 4.4, muestra los paquetes que permiten utilizar la clase firewall.java para establecer los comandos de activación del firewall, la clase BaseMySQL.java para acceder a la información contenida en la base de datos y capturadorsniffer.java para husmear dentro de la red.

Descripción del Código de la Clase ActivacionFirewall.java

Se instancian los objetos de las clases a utilizar (Obsérvese el código 4.5):

```
  //Método para instanciar la base de datos
public void instanciaBase(){
      bm=new BaseMySQL();
    bm.canal();
}
//Método para instanciar el objeto de la clase Firewall
public void instanciaFire(){
    f=new Firewall();

}
```

Código 4.5: "Instancia de los Objetos de las clases firewall.java y BaseMySQL.java"

En el botón de activación se genera la secuencia de comandos para el estado del firewall (Activado/desactivado) se ingresa esta información a la base de datos y mediante comando Netsh (también llamado NetShell o Network Shell, es una herramienta basada en línea de comando que configura, interfaces de conexión de red, puertos para habilitarlos y deshabilitarlos; así como también opciones de firewall utilizado únicamente para Windows) para usuarios Windows y ufw (Herramienta en Linux que permite la configuración de un firewall a través de líneas de comando; para estos propósitos utiliza el kernel de Linux que incluye el subsistema *Netfilter*, que es usado para manipular o decidir el destino del tráfico de red entre o a través de su red) para usuarios Linux.

El Código para Windows es el siguiente (Obsérvese el código 4.6):

```
//Botón que activa/desactiva firewall
public String btnEjecutar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
     instanciaBase();
    //Intanciación de las clases basemysql y de la clase firewall
     instanciaFire();
     boolean r;
    if( this.optActiva1.isChecked()){
        //Llamada al método de la clase Firewall
        //Se envía un comando netsh para habilitar al firewall de windows
            r=f.ejecutacmd("cmd /c netsh firewall set opmode enable");
   if(r){
    //Se actualiza el estado del firewall, actualizando la base de datos
             boolean ver=bm.ejecutar("Update software set Id_es=1 where id_soft=1");
    if (ver){
        //Se consulta el estado actual del firewall
             consultarEstado();
    }
   }else{
            this.LBLESTADO.setText("NO SE PUDO ACTIVAR : ");
   }
    }else  if( this.optDesactiva1.isChecked()){
        r=f.ejecutacmd("cmd /c netsh firewall set opmode disable");
        if(r){
             boolean ver=bm.ejecutar("Update software set Id_es=2 where id_soft=1");
         if (ver){
             consultarEstado();
```

Código 4.6: "Botón que activa o desactiva al firewall en Windows"

Se establece la llamada a la clase firewall.java y se envía como parámetros los comandos a ejecutarse para habilitar o deshabilitar nuestro firewall estos comandos se describen a continuación.

Windows:

Comandos Para Activar o desactivar el Firewall

Comando	Descripción
Netsh firewall	Establece la configuración del subsistema de firewall de la Shell de Windows
Set Opmode	Con el podemos configurar el modo de operación del firewall. Existen dos Modos Enable (Activa) y disable (desactiva)

Cuadro 4.1: "Comandos Para Activar o desactivar el Firewall"

Comandos para excepciones predefinidas:

Comando	Descripción
Set service	Mediante el comando set service podemos permitir o bloquear las cuatro excepciones predefinidas del firewall de XP: compartir archivos e impresoras, administración remota, escritorio remoto y UPnP. Estos son los cuatro tipos (type) que XP los llama así: fileandprint, remoteadmin, remotedesktop, upnp
Add PortOpening	Con el podemos configurar excepciones mediante puertos ya sean TCP o UDP
add allowedprogram	Configura excepciones basadas en programas

Cuadro 4.2: "Comandos para excepciones predefinidas"

GNU/Linux:

Comandos Para Activar o desactivar el Firewall

Comando	Descripción
Ufw enable/disable	Establece la configuración del subsistema de firewall de Linux Activándolo o desactivándolo.
Ufw default allow	Abre todos los puertos por defecto
Ufw default deny	Cierra todos los puertos del host

Cuadro 4.3: "Comandos Para Activar o desactivar el Firewall"

Comandos para excepciones

Comando	Descripción
Ufw allow (numero del puerto)/protocolo	Añade regla para admitir el trafico por ese puerto
Ufw deny (número del puerto)/protocolo	Añade regla para no admitir el trafico por ese puerto
Ufw allow (aplicación) samba	Cierra la conexión a la compartición de recursos con otros sistemas

Cuadro 4.4: "Comandos para excepciones"

```
public String btnEjecutar_action() {
    // TODO: Process the action. Return value is a navigation
    // case name where null will return to the same page.
     instanciaBase();
    instanciaFire();
     boolean r;
    if( this.optActiva1.isChecked()){

   r=f.ejecutacmd("ufw enable");
    r=f.ejecutacmd("ufw default deny");
   if(r){
    boolean ver=bm.ejecutar("Update software set Id_es=1 where id_soft=1");
    if (ver){

        consultarEstado();
    }
```

Código 4.7: "Botón que activa o desactiva al firewall en GNU/Linux"

Para controlar las excepciones que se desean activar o desactivar se las ejecutara a través de casillas checkbox y botones para la ejecución de las peticiones hechas por el usuario.

El código para activar las excepciones para versiones Windows es el siguiente (Obsérvese el código 4.8):

```
    //Método que permite la actualización de excepciones del firewall
public void actualizarExcep(){
      boolean com;
      instanciaBase();
         instanciaFire();
        if(chkCompartir.isChecked()){
            //Habilita el servicio de compartición
              com=f.ejecutacmd("cmd /c netsh firewall set service fileandprint enable");
            if(com){
                boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=1 where IdEx=3");
                msgexcep();
            }
        }
         //Habilita la administración remota
          if(chkAdmin.isChecked()){
              com=f.ejecutacmd("cmd /c netsh firewall set service remoteadmin enable");
            if(com){
                 boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=1 where IdEx=2");
                msgexcep();
            }
        }
         //Habilita el acceso remoto
          if(chkRemoto.isChecked()){
              com=f.ejecutacmd("cmd /c netsh firewall set service remotedesktop enable");
            if(com){
                 boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=1 where IdEx=1");
                msgexcep();
            }
```

Código 4.8: “Botón que activa las excepciones del firewall en Windows”

El código para activar las excepciones para versiones GNU/Linux es el siguiente (Obsérvese el código 4.9):

```
public void actualizarExcep(){
       boolean com;
       instanciaBase();
          instanciaFire();
         if(chkCompartir.isChecked()){
               com=f.ejecutacmd("ufw allow samba");
              if(com){

                  boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=1 where IdEx=3");
                  msgexcep();
              }
          }
            if(chkAdmin.isChecked()){
                com=f.ejecutacmd("ufw allow 22");
              if(com){
                   boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=1 where IdEx=2");
                  msgexcep();
              }
          }
            if(chkRemoto.isChecked()){
                com=f.ejecutacmd("ufw allow 23");
              if(com){
                   boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=1 where IdEx=1");
                  msgexcep();
              }
```

Código 4.9: “Botón que activa las excepciones del firewall en GNU/Linux”

Desactivar excepciones en Windows (Obsérvese el código 4.10):

```
//Método que desactiva las excepciones del firewall
public void desactivaExcep(){
        boolean com;
        instanciaBase();
           instanciaFire();
           //Deshabilita la compartición de archivos
          if(chkCompartir.isChecked()){
                com=f.ejecutacmd("cmd /c netsh firewall set service fileandprint disable");
              if(com){


                  boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=2 where IdEx=3");
                  msgexcep();
              }
          }
```

Código 4.10: "Botón que desactiva las excepciones del firewall en Windows"

Desactivar excepciones en GNU/Linux (Obsérvese el código 4.11):

```
public void desactivaExcep(){
      boolean com;
      instanciaBase();
         instanciaFire();
        if(chkCompartir.isChecked()){
              com=f.ejecutacmd("ufw deny samba");
            if(com){


                boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=2 where IdEx=3");
                msgexcep();
            }
        }
          if(chkAdmin.isChecked()){
              com=f.ejecutacmd("ufw deny 22");
            if(com){
                 boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=2 where IdEx=2");
                msgexcep();
            }
        }
          if(chkRemoto.isChecked()){
              com=f.ejecutacmd("ufw deny 23");
            if(com){
                 boolean r=bm.ejecutar("Update excepcionesfirewall set Id_es=2 where IdEx=1");
                msgexcep();
            }
```

Código 4.11: "Botón que desactiva las excepciones del firewall en GNU/Linux"

Es necesario saber que excepciones están activadas y cuál es el estado del actual del firewall utilizando el siguiente código:

Consultar excepciones (Obsérvese el código 4.12):

```
//Metodo que permite consultar las excepciones desde la base de datos recibe un identificador
public boolean consultarExcepciones(int id){
  boolean r=false;
   String respon="";
  instanciaBase();
 //metodo de la clase BaseMySql para ejecutar una consulta
  rst=bm.consulta("Select ef.Tipo, e.Tipo From excepcionesfirewall ef, Estado e where e.Id_Estado=ef.Id_es and
  try{
      while(rst.next()){
         respon=rst.getString(2);

   }
      if(respon.equalsIgnoreCase("Activado"))
   r=true;
      else
          r=false;
  }catch(SQLException e){
      r=false;
  }
  return r;
}
```

Código 4.12: "Método de consulta de excepciones"

El método de consulta de excepciones recibe como parámetros un identificador que es la clave del servicio a comprobar si esta activo o no.

Para mostrar Información en pantalla de las excepciones mediante gráficos (Obsérvese el código 4.13).

```
//Control de excepciones
public void msgexcep(){
 boolean validarmsg;
  int num=1;
 // acceso remoto
  //comprueba si esta habilidato o no el acceso remoto
  // el metodo recibe un identificador para determinar el valor
  validarmsg=consultarExcepciones(num);
  if(validarmsg){
      this.imgok3.setVisible(true);
       this.imgst3.setVisible(false);
  }else{
       this.imgok3.setVisible(false);
        this.imgst3.setVisible(true);
  }
  // administracion remota
 //comprueba si esta habilidato o no la administracion remota
  // el metodo recibe un identificador para determinar el valor
```

Código 4.13: "Método que muestra de manera grafica las excepciones"

Es necesario crear un metodo de mensajes que envia al metodo de consulta de excepciones el identificador de cada servicio y establece de manera grafica las excepciones que estan activas y cuales no.

4.4.2. CREACIÓN DEL MÓDULO DE CONTROL DE ACCESOS

Para el control de accesos se creó un pequeño software Sniffer que es el encargado de capturar e interceptar los paquetes que están entrando o saliendo del host.

El software esta creado en una aplicación de java y colocado dentro de fireprox como una librería .jar que puede ser llamada desde un botón de la interfaz web.

A continuación se muestra las clases que contiene capturadorsniffer.jar (Obsérvese en la Fig. 4.16).

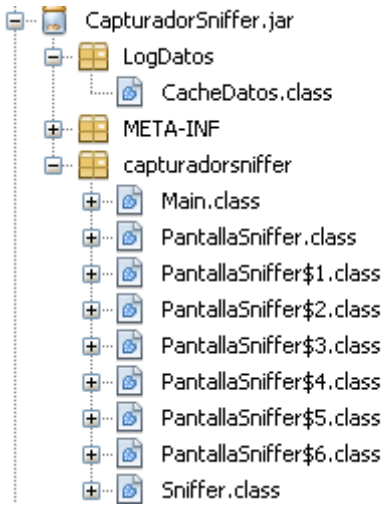

Fig. 4.16: “Contenido de la Librería capturadorsniffer.jar”

Paquetes y Clases de capturadorsniffer.jar

Paquete LogDatos

Permite conectarse a una base de mysql, contiene la clase CacheDatos.java (Obsérvese en la Fig. 4.17).

Fig. 4.17: "Clase del paquete LogDatos de la Aplicación CapturadorSniffer"

Paquete Capturador Sniffer

Contiene los métodos para capturar los paquetes que ingresan a la red; está conformado por las clases main.java, sniffer.java, pantallasniffer.java (Obsérvese en la Fig. 4.18).

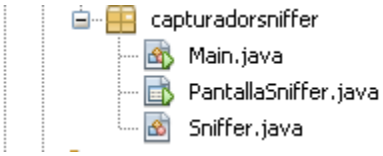

Fig. 4.18: "Clase del paquete LogDatos de la Aplicación CapturadorSniffer"

4.4.2.1. Descripción de la Clase Sniffer.java

Permite husmear entre los paquetes que ingresan y salen de una tarjeta de red instalada al equipo; utiliza las librerías jpcap y winpcap que contienen clases que ayudan a manejar código nativo del sistema operativo. Para su creación es necesario importar en la cabecera de la clase los siguientes paquetes (Obsérvese el código 4.14):

```
import java.io.*;
//importa paquete jpcap la clse jpcaptor para la captura de paquetes
import jpcap.JpcapCaptor;
//permite importar la clase networdinterface para selecionar las tarjetas de red
import jpcap.NetworkInterface;
//Permite importar la clase packet para obtener los paquetes
import jpcap.packet.*;
//Importa el paquete logDatos para el registro de la información en la base de datos
import LogDatos.CacheDatos;
import java.util.*;
```

Código 4.14: "Cabecera de paquetes de la clase Sniffer"

Los paquetes de jpcap permiten el acceso a la información que pasara a través de la red y podrán ser utilizados dentro de la clase sniffer.

Se necesita establecer objetos que permitan la utilización de los diferentes métodos que conforman al paquete jpcap para ello realizamos lo siguiente (Obsérvese el código 4.15):

```
    //Declaración de objetos de clase
PantallaSniffer ps;
NetworkInterface[] netInter, esNet;
CacheDatos cd;
JpcapCaptor captor;
String str,info;
int x, choice;
Calendar c;
```

Código 4.15: "Declaración de objetos dentro de la clase Sniffer"

Se crea un método que permite escoger los diferentes dispositivos de red que se encuentran instalados en el host para poder capturar la información utilizando el siguiente código (Obsérvese el código 4.16):

```
//método que extrae las interfaces de red
public NetworkInterface[] getTarRed(){

    netInter=JpcapCaptor.getDeviceList();

return netInter;
}
```

Código 4.16: "Método que extrae los dispositivos de red de la clase Sniffer"

Una vez establecidas las interfaces se procede a la creación del método que permitirá la captura de los paquetes dentro de la red cuyo código es el siguiente (Obsérvese el código 4.17):

```
//método que permite la captura de los paquetes
public void capturapaquetes(int i){
    ps=new PantallaSniffer();
   try {
       //establece la tarjeta de red a utilizarse
       esNet=getTarRed();
       //se habre la tarjeta de red y se escanean todos los puertos
                    captor=JpcapCaptor.openDevice(esNet[i], 65535, false, 20);
                    /* escuchará solo conexiones TCP/IP */
                    captor.setFilter("tcp and ip", true);
                }
                    catch(IOException ioe) { ioe.printStackTrace(); }
                /* empezar  a husmear en los paquetes */
                boolean ban=true;
                int cont=0;
                while(ban){
                        //Captura la información contenida en los paquetes
                      Packet info1 = captor.getPacket();
                        if(info1 != null){
                        cont=cont+1;
                            System.out.print(getPaqueteTexto(info1));
                        }
                      //Contador que permite detener la captura de los paquetes
                      //Se limita a capturar únicamente 1000 paquetes
                      if(cont==1000){
                          ps.ejecutaMsg();
                          ban=false;
                      }
                      }
```

Código 4.17: "Método que captura los paquetes de la clase Sniffer"

Clase Pantallasniffer.java

Llama a todos los métodos de la clase sniffer para poder capturar los paquetes de la red. Está compuesta por una interfaz grafica diseñada en java como se muestra a continuación (Obsérvese en la Fig. 4.19):

Fig. 4.19: "Interfaz del Sniffer"

Desde esta aplicación se puede escoger la tarjeta de red que se desea husmear y se empieza a la captura de paquetes en dicha interfaz. Cabe mencionar que el Sniffer solo está disponible para la Versión de Windows.

4.4.3. CREACIÓN DEL MÓDULO PROXY

El modulo proxy está constituido por el paquete Servidor_Proxy que contiene siguientes clases (Obsérvese en la Fig. 4.20):

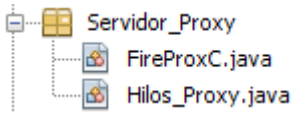

Fig. 4.20: "Paquete Servidor_Proxy y sus clases"

4.4.3.1 Descripción de la clase FireProxC.java

La cabecera de la clase incluye a la librería net de java que permite la utilización de sockets como se muestra a continuación (Obsérvese el código 4.18):

```
//Importa el paquete base de datos para realizar operaciones en MySql
import Base_Datos.BaseMySQL;
//Permite la utilización de sockets
import java.net.* ;
import java.io.* ;
import Base_Datos.BaseMySQL.*;
import java.sql.*;
import java.text.SimpleDateFormat;
import java.util.*;
```

Código 4.18: "Librerías de Cabecera de la clase FireProxC.java"

Esta clase establece el puerto en el que el servidor escuchara las peticiones hechas por los clientes que intenten ingresar a la web utilizando el siguiente método (Obsérvese el código 4.19):

```
 //método que permite cofigurar el puerto del proxy
public boolean config_puertoSer(int n){
    boolean r=false;
    //llamada al método que establece la conexión a la base de datos
this.establecerComunicacion();
    r=bm.ejecutar("Insert into configproxy(Puerto) values (" + n + ")");

    return r;

}
```

Código 4.19: "Método para la configuración del puerto del servidor proxy de la clase FireProxC.java"

El método mostrado en el espacio de código 4.19, necesita establecer comunicación con la base de datos para guardar la configuración realizada por el usuario.

Realizada la configuración del proxy se activa al servidor mediante el siguiente método (Obsérvese el código 4.20):

```
//Método que permite activar el servidor
   public void ActivarServidorprox(boolean bandera){
       //Llamada al método que seleciona el puerto del servicio proxy
     puertolocal=SeleccionpuertoS();
             // verifico para validar que el puerto es correcto el host remoto y el puerto del
    System.out.println("Verificando: Puerto :" + puertolocal + " a " + remotehost + " Puerto "
     System.out.println("LISTO PARA EMPEZAR...!");

    //Creo mi servidor de sockets en el puerto local para escuchar peticione
    try{
        Server = new ServerSocket(puertolocal);
    }
    catch(IOException e) {
        e.printStackTrace();
    }

    //Verifico las peticiones de entrada para poder solventarlas
int acum=0;
```

Código 4.20: "Método para la activación del servidor proxy de la clase FireProxC.java"

Es necesario establecer el puerto del servidor con el método SeleccionpuertoS (Obsérvese el código 4.21):

```
private int SeleccionpuertoS(){
    int port=0, max=0;
    this.establecerComunicacion();
    rst=bm.consulta("Select Max(Id_Pro) From configproxy");
  try{
      while(rst.next()){
        max=rst.getInt(1);

    }
        this.establecerComunicacion();
    rst=bm.consulta("Select Puerto From configproxy where Id_Pro=" + max + "");
     while(rst.next()){
        port=rst.getInt(1);
    }
  }catch(SQLException e){
  }

    return port;
```

Código 4.21: "Método para la selección del puerto del servidor proxy de la clase FireProxC.java"

El método del espacio de código 4.21, se establecerá una consulta a la base de datos para cargar la configuración establecida por el usuario.

Establecido el puerto se crea los sockets del servidor y del cliente escuchando cada una de las peticiones hechas a través del servicio web; para atender a todos los requerimientos hechos se crea hilos de la clase Hilos_Proxy que son los encargados de ejecutar las órdenes que son recibidas a través del puerto del servidor.

A continuación se muestra el código generado (Obsérvese el código 4.22):

```
        try{
            Server = new ServerSocket(puertolocal);
        }
        catch(IOException e) {
            e.printStackTrace();
        }

        //Verifico las peticiones de entrada para poder solventarla
    int acum=0;
        while(bandera)
            {
                try{
                    //acepta las peticiones hechas por el cliente
                    entrada = Server.accept();
                  this.establecerComunicacion();
                   boolean ingre=false;
                 String s=entrada.getInetAddress().toString();
```

Código 4.22: "Creación del Socket Servidor y del Socket de entrada"

Las peticiones realizadas por los clientes necesitan ser solventadas de manera constante para ello se crean hilos de la clase Hilos_Proxy como se detalla a continuación (Obsérvese el código 4.23):

```
//Envia a la calse Hilos Proxy las peticiones de entrada y salida
                hilo1 = new Hilos_Proxy(entrada, salida);
                hilo1.start();
                //Envia a la clase Hilos Proxy las peticiones de entrada y salida de manera invertida para qu
              hilo2 = new Hilos_Proxy(salida, entrada);
                hilo2.start();
```

Código 4.23: "Hilos para peticiones de entrada y de salida"

4.4.3.2 Descripción de la clase Hilos_Proxy.java

Solventa de manera constante las solicitudes de servicio que llegan de cada uno de los clientes de la red.

Para ello redirecciona las peticiones desde el servidor hacia el host remoto y viceversa para mantener el anonimato de las solicitudes como se detalla a continuación (Obsérvese el código 4.24):

```
//Escribo un metodo run para arrancar el hilo y para la transferencia de datos
public void run(){
    byte[] buffer = new byte[2048];
  int nLectura = 0;
  // lectura de datos de salida en un socket
  OutputStream aCliente;
  // colocar datos en un socket
  InputStream deCliente;

  try{
      // extraccion  de la informacion colocada en los sockets de entrada y salida
      aCliente = salida.getOutputStream();
      deCliente = entrada.getInputStream();
      empezar=getemp();
    while(true){

       //lectura de los datos del socket
      nLectura = deCliente.read(buffer, 0, 10);
     // System.out.println("Leer " + nLectura);
      //buffer[nLectura] = buffer[0] = (byte)'+';

       if(nLectura == -1){
         entrada.close();
         salida.close();
```

Código 4.24: "Redireccionamiento de entrada y de salida de datos"

4.4.3.3. Clase que permite la ejecución de la interfaz ConfiguracionProxy.java

Permite ejecutar y configurar a nuestro servidor proxy para el acceso a la web.
Se importa las clases del servidor (Obsérvese el código 4.25):

```
import Servidor_Proxy.FireProxC.*;
import Servidor_Proxy.Hilos_Proxy;
```

Código 4.25: "Importar librerías para Proxy"

La configuración del proxy se la realiza a través de un botón que contiene las siguientes líneas de código (Obsérvese el código 4.26):

```
//método de Configuración del Servidor Proxy a través de un botón
    public String btnConfigurar_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.
        boolean ejec=false;
        instanciaProxy();

        puertoS=Integer.parseInt(this.TXTPUERTO.getText().toString());

        if (puertoS >= 1000 && puertoS <= 5000){
            try{
        ejec=Prox.config_puertoSer(puertoS);
            }catch(NullPointerException e){

            }
        if (ejec){

            this.LBLMSG.setText("OPERACION EXITOSA");
            this.imgres.setVisible(true);
             this.imgres1.setVisible(false);
        }else{
            this.LBLMSG.setText("OPERACION SIN EXITO");
            this.imgres1.setVisible(true);
             this.imgres.setVisible(false);
        }
        }else{

             this.LBLMSG.setText("PUERTO INCORRECTO ");
```

Código 4.26: “Configuración del puerto del servidor del Proxy”

Ya configurado el puerto se procede a activar el servicio (Obsérvese el código 4.27):

```
//Botón que permite activar el servidor Proxy
    public String btnacepta_action() {
        // TODO: Process the action. Return value is a navigation
        // case name where null will return to the same page.


            instanciaProxy();
        if( this.optActivar.isChecked()){
            try{
           Prox.ActivarServidorprox(true);
           this.lblac.setVisible(true);
           this.imgac.setVisible(true);
            }catch(NullPointerException e ){

            }


        }
        return null;
```

Código 4.27: “Configuración del puerto del servidor del Proxy”

4.4.4. CREACIÓN DEL MÓDULO ESTADÍSTICO Y DE REPORTES

A través de las gráficas estadísticas, se puede conocer un reporte de uso tanto de la memoria, como del procesador, de los paquetes capturados en la red en un periodo de tiempo y clientes que mas solicitan el servicio web.
Es así como el paquete Gráficas estadísticas contiene las siguientes clases:

- GráficaMemoria.java
- GráficaPaquetes.java
- GráficaProcesador.java
- GraficaPeticiones.java

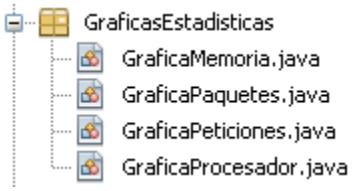

Fig. 4.21: "Contenido del Paquete GraficasEstadisticas"

Clase GráficaMemoria.java

Crea Un Grafico de pastel tomando los datos proporcionados por la clase de InfoMemoria.java contenida en el paquete InformacionSis.

Para su creación se necesita la librería org.jfreechart que proporciona un sin número de clases para la creación de Gráficos de pastel, barras, puntos, dispersión, etc. (Obsérvese el código 4.28):

```
//permite importar propiedades de interfaces gráficas de java
import java.awt.Font;
//permite importar la clase de construción de gráficas
import org.jfree.chart.ChartFactory;
//permite importar la clase principal del jfreechart
import org.jfree.chart.JFreeChart;
//Importa la clase que dibuja gráficos de pastel
import org.jfree.chart.plot.PiePlot;
//Importa la clase que permite colocar valores a nuestra gráfica
import org.jfree.data.general.DefaultPieDataset;
import org.jfree.data.general.PieDataset;
//Importa el paquete de información del sistema, especificamente de la red y memoria
import InformacionSis.InfoRed;
import InformacionSis.InfoMemoria;
//Importa la clase para mostrar la gráfica en un frame de java
import org.jfree.chart.ChartFrame;
```

Código 4.28: "Contenido del Paquete org.jfreechart"

Colocadas las cabeceras para importar librerías se procede a la construcción del grafico (Obsérvese el código 4.29):

```
//método que permite construir la estructura que tendrá la gráfica
private JFreeChart createChart(PieDataset dataset) {
   // ir=new InfoRed();
    JFreeChart chart = ChartFactory.createPieChart(
        "MEMORIA RAM DEL HOST " ,  // chart title
        dataset,             // data
        true,               // include legend
        true,
        false
    );

    PiePlot plot = (PiePlot) chart.getPlot();
   plot.setLabelFont(new Font("SansSerif", Font.PLAIN, 12));
   plot.setNoDataMessage("DATOS NO ENCONTRADOS");
   plot.setCircular(true);
   plot.setLabelGap(0.02);
    return chart;
```

Código 4.29: "Método que construye la gráfica de pastel"

Construida la grafica se coloca los datos que se desean mostrar (Obsérvese el código 4.30):

```
//Método para la creación de la gráfica de pastel
private PieDataset createDataset() {
    im=new InfoMemoria();
    DefaultPieDataset dataset = new DefaultPieDataset();
    dataset.setValue("Memoria Utilizada MB: " + im.getUsadamemoria(), Integer.parseInt(im.getUsadamemoria()))
    dataset.setValue("Memoria Libre MB: " + im.getDispmemoria(),Integer.parseInt(im.getDispmemoria()) );

    return dataset;
}

/**
```

Código 4.30: "Método que establece los valores de gráfica de pastel"

Para Mostrar el gráfico en un frame de java se establece el siguiente método (Obsérvese el código 4.31):

```
public  JFreeChart createDemoPanel() {
    JFreeChart chart = createChart(createDataset());
    return  chart;
}

/**
 * Starting point for the demonstration application.
 *
 * @param args  ignored.
 */

//método que permite mostrar la gráfica en pantalla
public void generaGrafica(){
 JFreeChart ch=createDemoPanel();
 ChartFrame fr=new ChartFrame("GRAFICA  MEMORIA DEL PC.- SISTEMA FIREPROX", ch );
 fr.pack();
 fr.setVisible(true);

}
```

Código 4.31: "Método que genera y muestra los resultados en un frame"

Clase GráficaProcesador.java

Crea un Grafico de pastel tomando los datos proporcionados por la clase de InfoProcesador.java contenida en el paquete InformacionSis.
El código es similar al de la gráfica de memoria lo único que varia son los valores mostrados extraídos del uso del procesador.

Clase GraficaPeticiones.java

Muestra una gráfica de barras en porcentajes de las peticiones realizadas por los clientes para que el servidor proxy les proporcione servicios web.

Para su implementación se importa las cabeceras necesarias para la generación de gráficos de barras y la extracción de la información de la base de datos (Obsérvese el código 4.32):

```
//importa el paquete para contruir la gráfica
import org.jfree.chart.ChartFactory;
//importa el paquete para mostrar la gráfica en pantalla
import org.jfree.chart.ChartFrame;
//permite la utilización de gráficos
import org.jfree.chart.JFreeChart;
//importa el paquete para la orientación de la gráfica
import org.jfree.chart.plot.PlotOrientation;
//importa el paquete para ubicar los datos que se mostraran en la gráfica
import org.jfree.data.category.DefaultCategoryDataset;
import Base_Datos.BaseMySQL;
import java.sql.*;
```

Código 4.32: “Cabecera de paquetes de la clase GraficaPeticiones.java”

Se establece un método que crea la grafica y extrae los datos de la base mediante una consulta de agrupación (Obsérvese el código 4.33 y código 4.34):

```
//método que permite la creación de la gráfica
public void creaGraf(){
    instanciabase();
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    try{
        rs=bm.consulta("Select Count(Host) As Numero_de_Peticiones, Host as Cliente From eventosproxy  gro

    double acump=0;
   while(rs.next()){
        acump=acump+Double.valueOf(rs.getInt(1));

    }
     rs=bm.consulta("Select Count(Host) As Numero_de_Peticiones, Host as Cliente From eventosproxy  group
    double valor1=0;
     while(rs.next()){
         double val=rs.getInt(1);
        valor1=this.obtenerporcentages(val, acump);
        dataset.addValue(valor1, "Ip " + rs.getString(2), "Ip " + rs.getString(2));

    }

    }catch(SQLException e){
```

Código 4.33: “Extracción de la información de la base”

```
//Crear el gráfico...

JFreeChart chart = ChartFactory.createBarChart("PETICIONES DE SERVICIO WEB", "DIR IP DE CLIENTES",
dataset, //Dataset
PlotOrientation.VERTICAL,
true,
true,
false);
//crear y visualizar una ventana...
ChartFrame frame = new ChartFrame("FIREPROX Sistema de Firewall y Proxy", chart);

frame.pack();

frame.setVisible(true);

}
    }
```

Código 4.34: “Creación del gráfico de barras”

Para poder mostrar la información obtenida en porcentajes se crea un método para la transformación de los datos (Obsérvese el código 4.35):

```
private double obtenerporcentages(double nc, double tp){
    double p=0.0;
    p=(nc*100.0)/tp;
    p=Math.rint(p);
    return p;
}
```

Código 4.35: “Método de Obtención de Porcentajes”

Clase GraficaPaquetes.java

Genera un grafico de barras del total de paquetes capturados por día.

El código es similar al de la clase GraficaPeticiones.java, únicamente se utiliza los datos del capturador sniffer contenidos en la base de datos.

CAPÍTULO V

5. IMPLEMENTACIÓN Y PRUEBAS

En este capítulo, se procederá a realizar la implementación del software desarrollado, en las plataformas operativas indicadas a lo largo del trabajo y las pruebas de funcionalidad se las realizará a través de máquinas virtuales.

FIREPROX, es un sistema que está programado enteramente en Java, de ahí su característica de ser un software multiplataforma.

5.1. IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA PROTOTIPO EN DOS PLATAFORMAS DIFERENTES LINUX Y WINDOWS CON CONEXIÓN A INTERNET A TRAVÉS DE MÁQUINAS VIRTUALES

La virtualización, es la técnica empleada sobre las características físicas de algunos recursos computacionales, para ocultarlas de otros sistemas, aplicaciones o usuarios que interactúen con ellos.

Esto implica hacer que un recurso físico, como un servidor, un sistema operativo o un dispositivo de almacenamiento, aparezca como si fuera varios recursos lógicos a la vez, o que varios recursos físicos, como servidores o dispositivos de almacenamiento, aparezcan como un único recurso lógico.

Esta tecnología permite la separación del hardware y el software, lo cual posibilita a su vez que múltiples sistemas operativos, aplicaciones o plataformas de

cómputo, se ejecuten simultáneamente en un solo servidor o PC según sea el caso de aplicación.

Para efectos de implementación y pruebas del sistema FireProx, se empleará el modo de Virtualización de plataforma, que se trata de simular una máquina real (servidor o PC) con todos sus componentes (los cuales no necesariamente son todos los de la máquina física) y prestarle todos los recursos necesarios para su funcionamiento.

En donde se virtualizará los sistemas operativos, como plataforma base de prueba del software prototipo diseñado.

5.1.1. IMPLEMENTACIÓN Y PRUEBAS DEL SERVIDOR PROXY

A continuación, se presentará las interfaces a través de las cuales se puede visualizar como es el funcionamiento del Servidor Proxy:

Se ejecuta el Sistema FireProx y al acceder a la opción Proxy del menú principal, actividad Configuración, se tiene la interfaz de usuario mostrada a continuación, en donde hay una opción de configuración de puerto, al que se debe ingresar valores comprendidos en el rango de 1000 a 5000,para elegir el puerto por donde se enviará la petición.

Una vez ingresado el puerto correcto la operación deseada es realizada, como se muestra en la Fig. 5.1 y fig. 5.2:

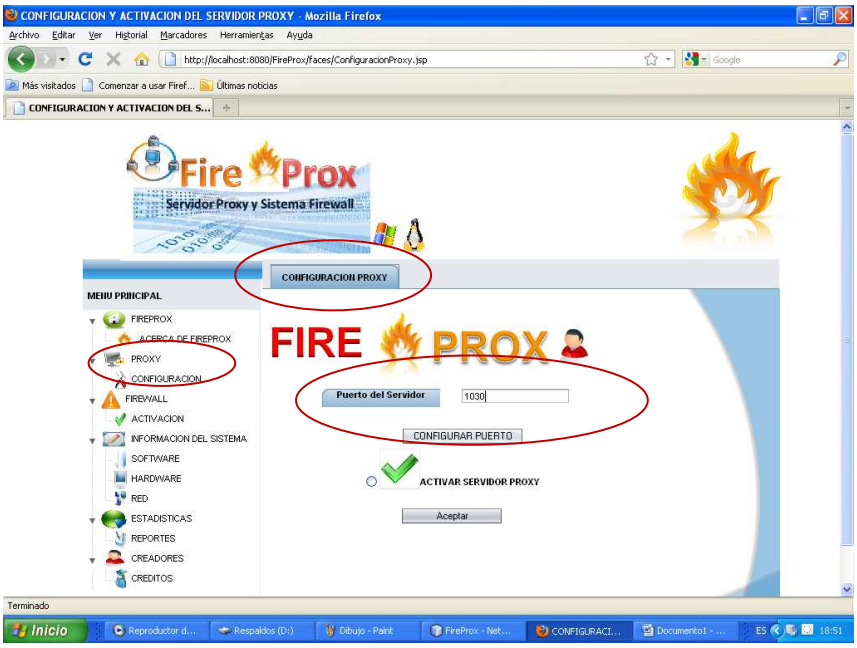

Fig. 5.1: “Configuración de puerto”

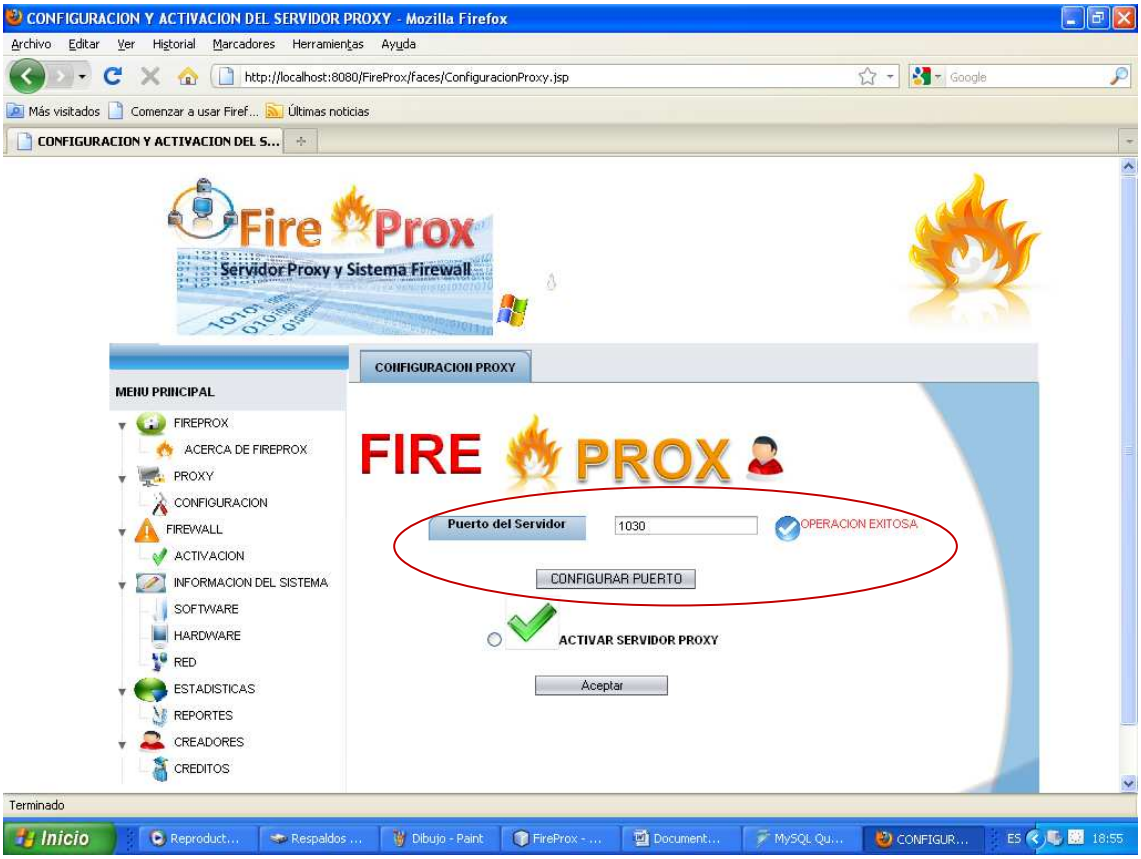

Fig. 5.2: “Puerto Correcto, operación Exitosa”

Se procede a realizar la activación del servicio:

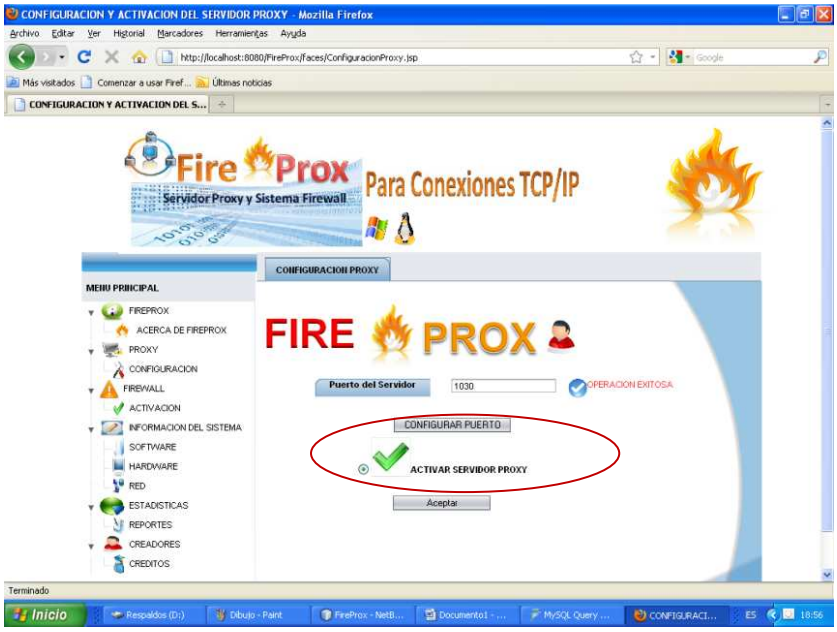

Fig. 5.3: "Activación del servicio"

Se accede a una página del navegador de internet, se escoge en la barra de menú: Herramientas→ Opciones→ Avanzado→ Red→ Configuración

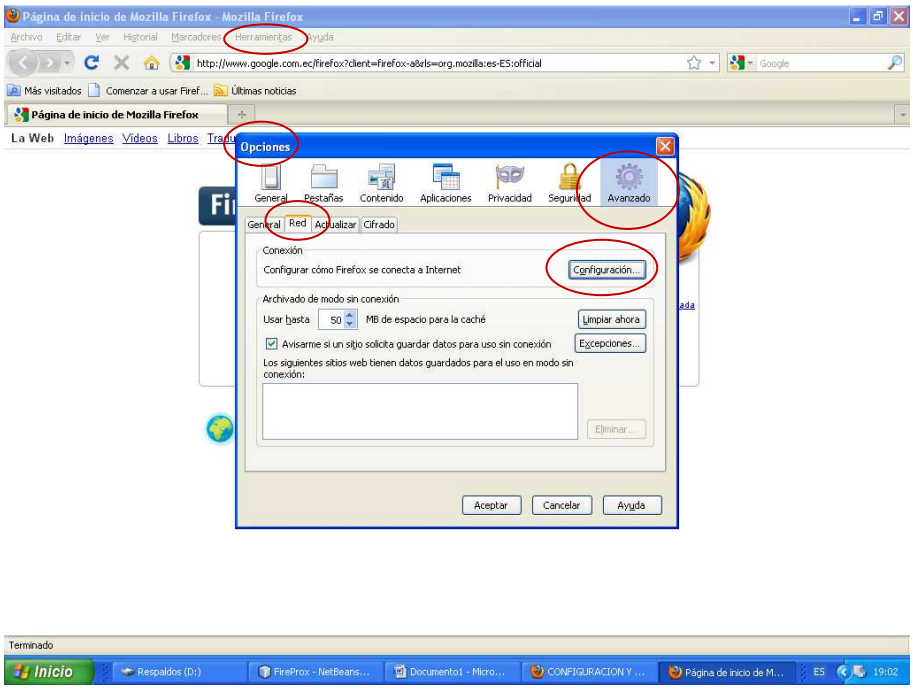

Fig. 5.4: "Navegador de Internet para configuración de servicio"

En la opción de configuración de conexión, se escoge Configuración general del Proxy.

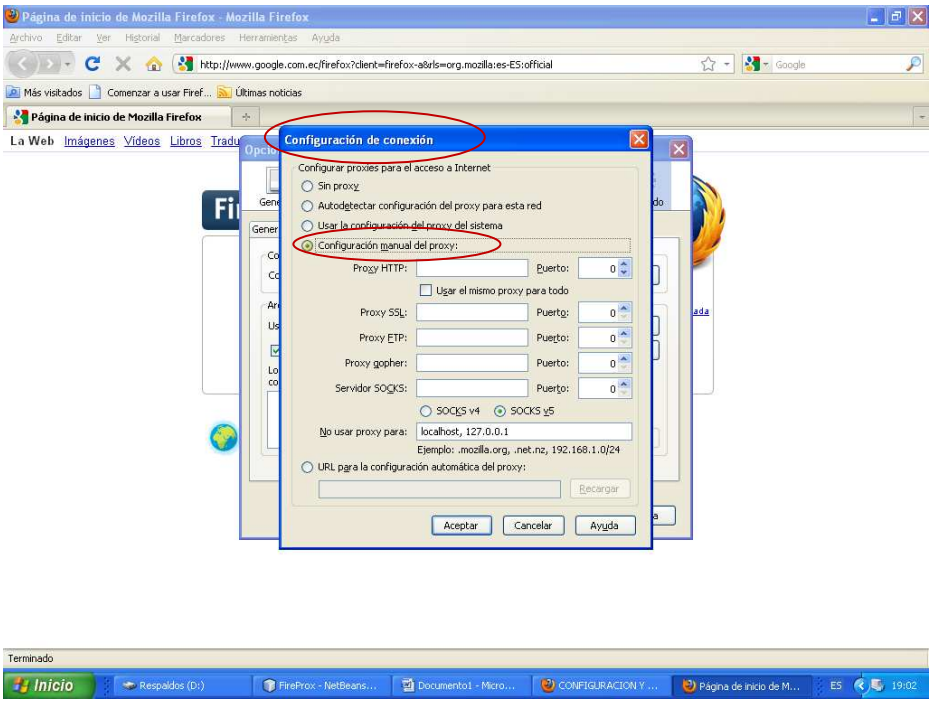

Fig. 5.5: "Configuración del servicio desde el Navegador de Internet"

Se coloca en la opción Proxy HTTP, la dirección IP del servidor, con el número del puerto que se configuró en el sistema FireProx.

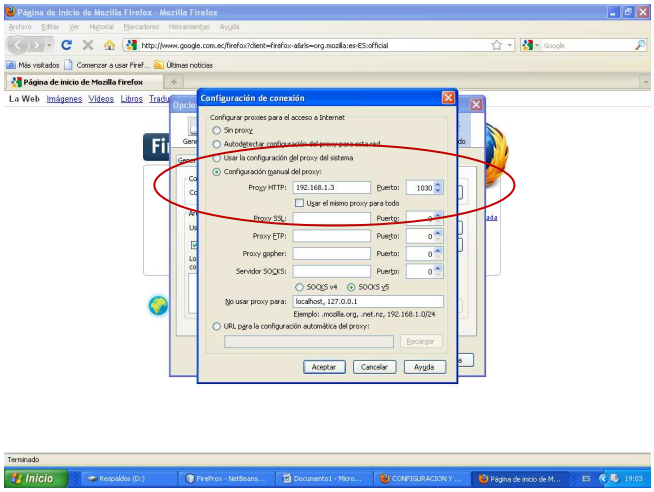

Fig. 5.6: "Configuración de puerto del servidor"

Entonces ingresará a la página direccionada por defecto, es decir www.google.com.

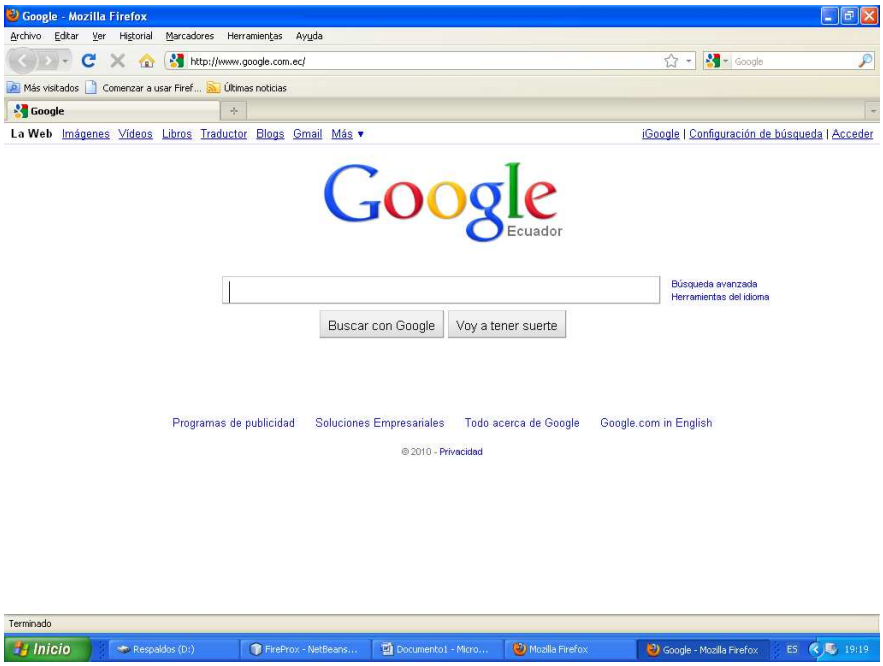

Fig. 5.7: "Página direccionada desde el proxy"

Se puede navegar por la página indicada por defecto.

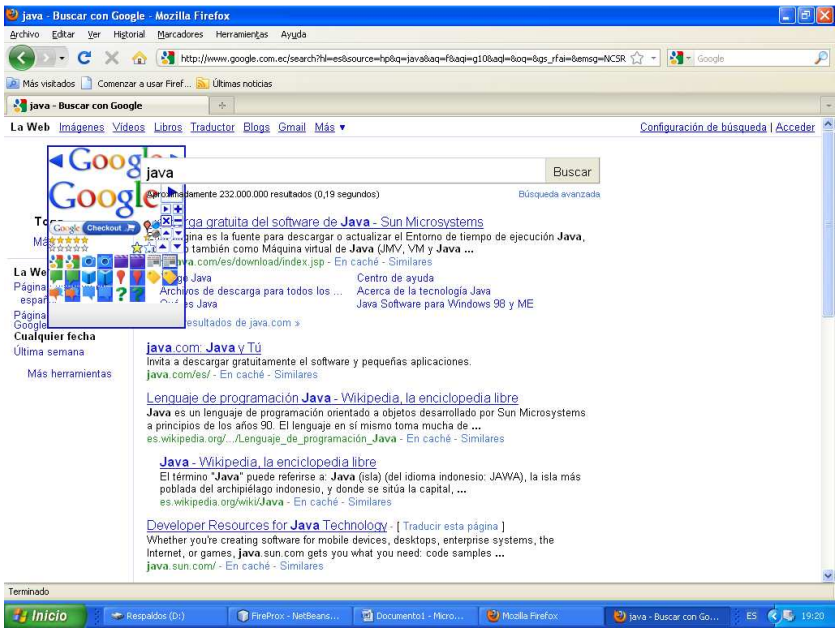

Fig. 5.8: "Página de Navegación"

Pero el momento que se desea acceder a otra página que no sea la predeterminada, se deniega el acceso.

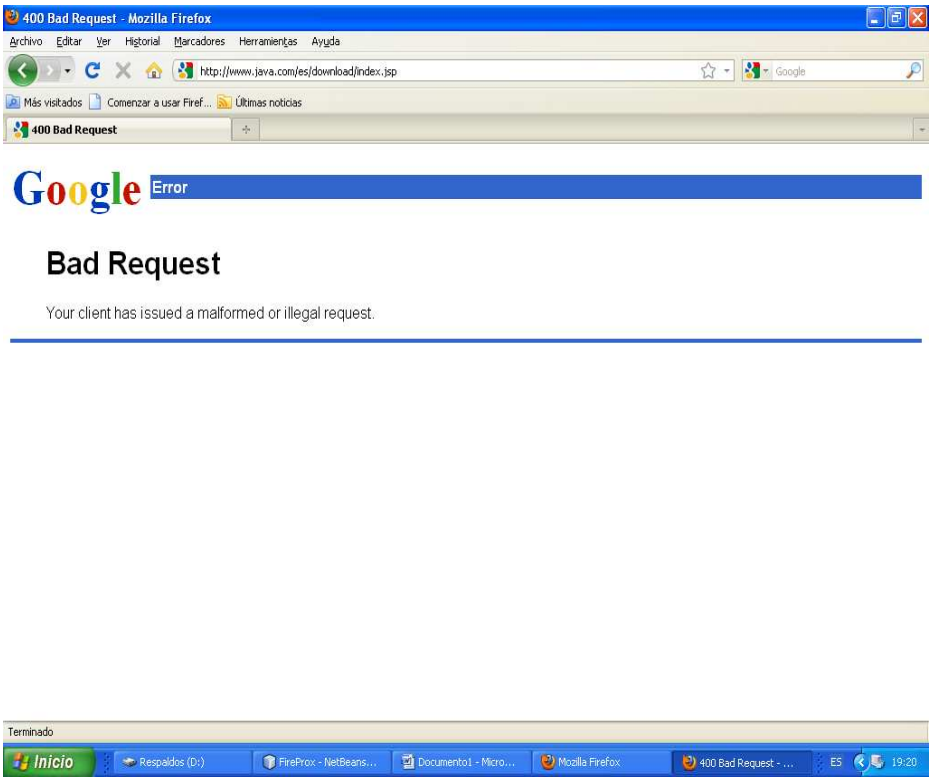

Fig. 5.9: "Acceso denegado para otras páginas"

5.1.2. IMPLEMENTACIÓN Y PRUEBAS FIREWALL

5.1.2.1.Pruebas en Windows

A través de las siguientes interfaces, se puede observar cómo se realiza la implementación del sistema Firewall, para controlar la seguridad de cada host dentro de una red de comunicaciones.

Mediante este software se conocerá como es el funcionamiento del firewall como sistema de seguridad.

Al ejecutar el sistema FireProx, se escoge del menú principal, opción Firewall, actividad Activación, se presenta la siguiente interfaz, en donde se puede seleccionar activar o desactivar firewall, según el estado actual en el que este se encuentre.

Fig. 5.10: “Configuración del Firewall”

Ser activa el firewall para realizar las pruebas.

Fig. 5.11: “Activación/Desactivación del sistema firewall”

A través de la máquina virtual se procederá a realizar las pruebas de funcionamiento, por medio del control de excepciones del firewall.

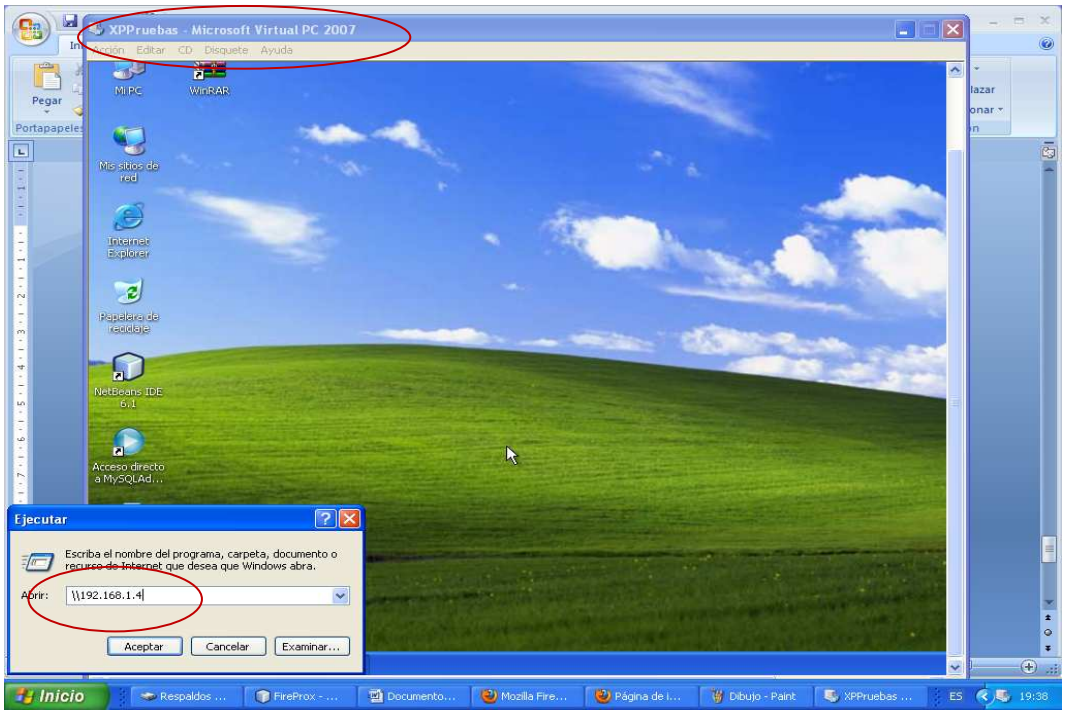

Fig. 5.12: "Acceso a máquinas virtuales"

La prueba se realizará a través de la excepcion denominada Compartir Impresoras y Archivos, en donde se habilitará la opción desde el sistema FireProx, y a través de máquinas virtuales se realizará la prueba de funcionalidad.

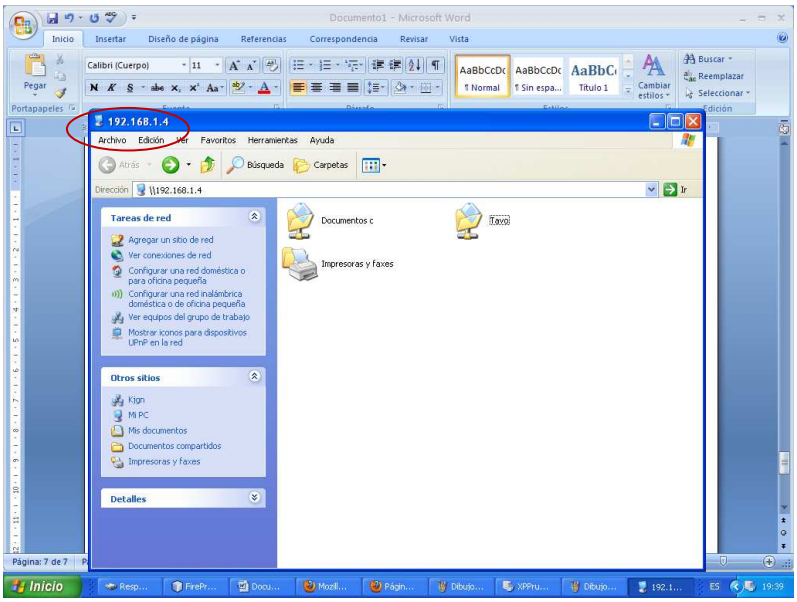

Fig. 5.13: "Compartir recursos"

La siguiente interfaz presenta las excepciones que se controlan a través del sistema, y el estado de cada una de ellas al momento de la ejecución del mismo.

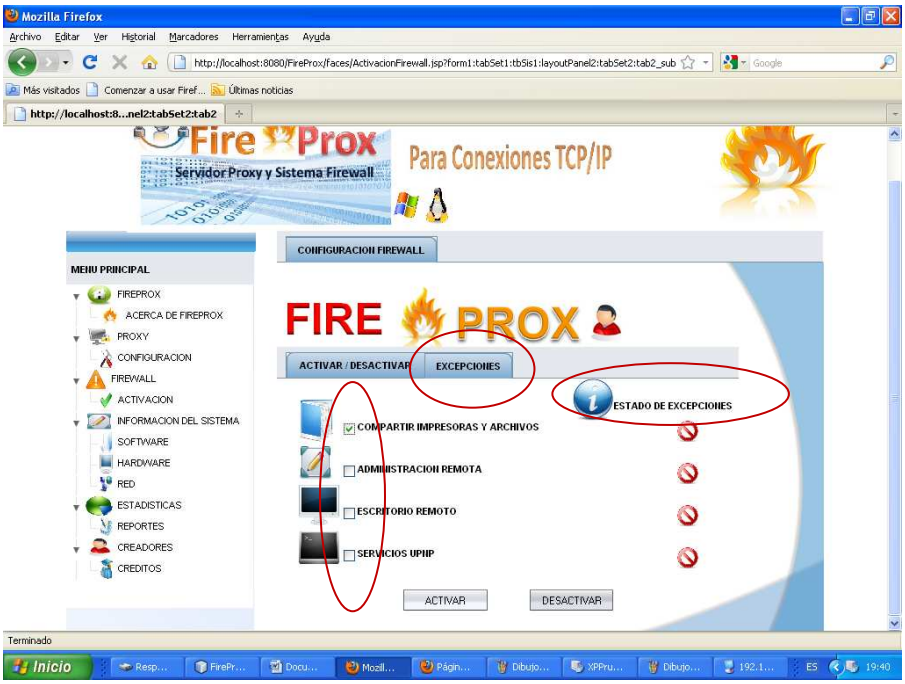

Fig. 5.14: “Manejo de Excepciones”

Se escoje la excepción que se desea habilitar y se presiona sobre el botón Activar

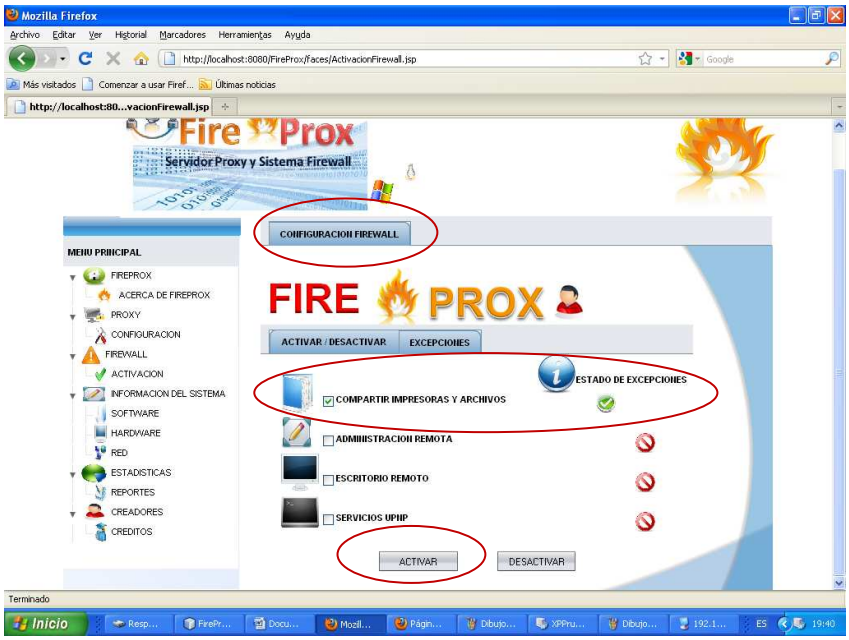

Fig. 5.15: “Activar/Desactivar Excepciones”

Se ingresa a la máquina virtual a través de la dirección IP

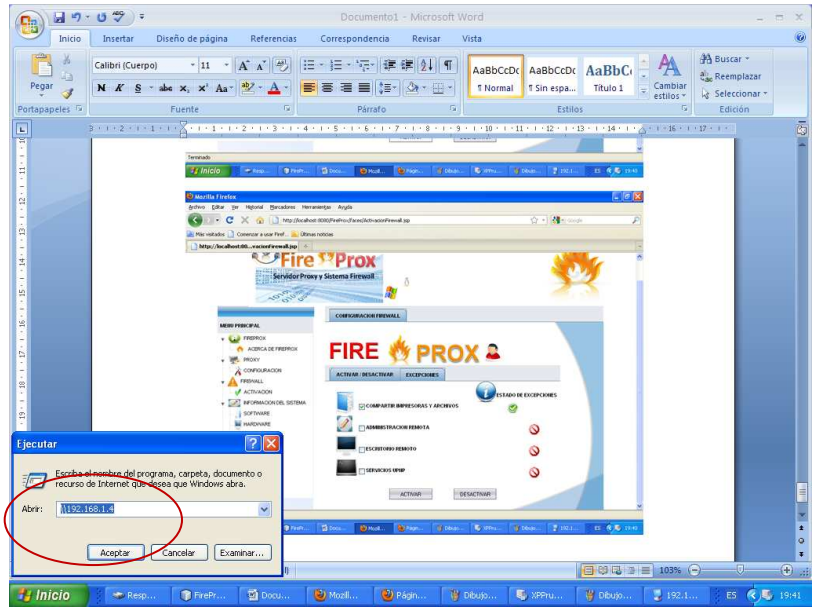

Fig. 5.16: "Acceso máquina Virtual"

Lo que se desea verificar, es que una vez desabilitada la excepción señalada,se dejen de compartir los recursos de un host con otro.

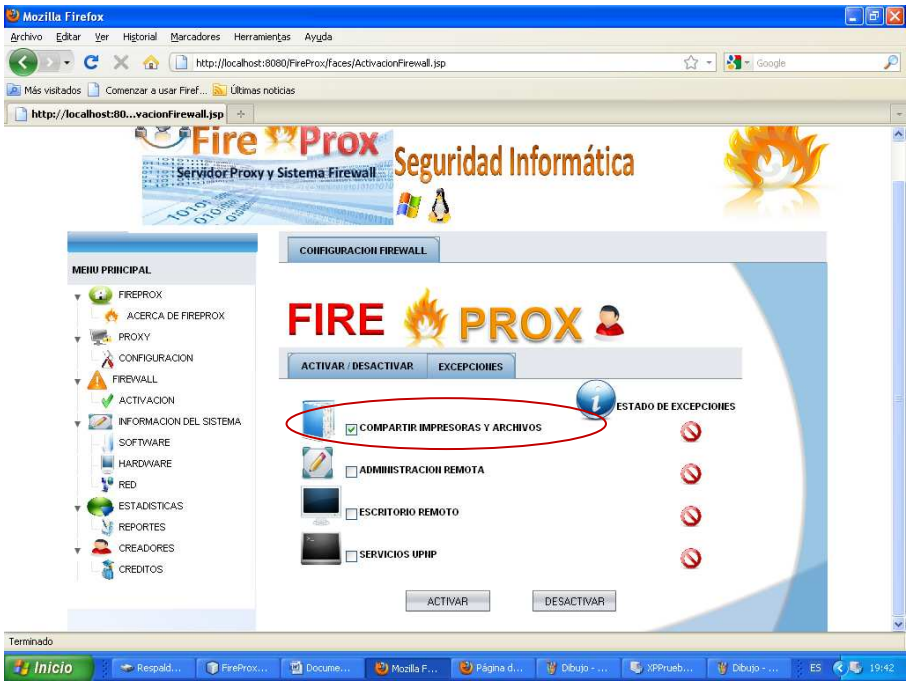

Fig. 5.17: "Desactivar Excepción"

Desde la Maquina Virtual, se accede al Servidor, ingresando a través de la dirección IP del mismo.

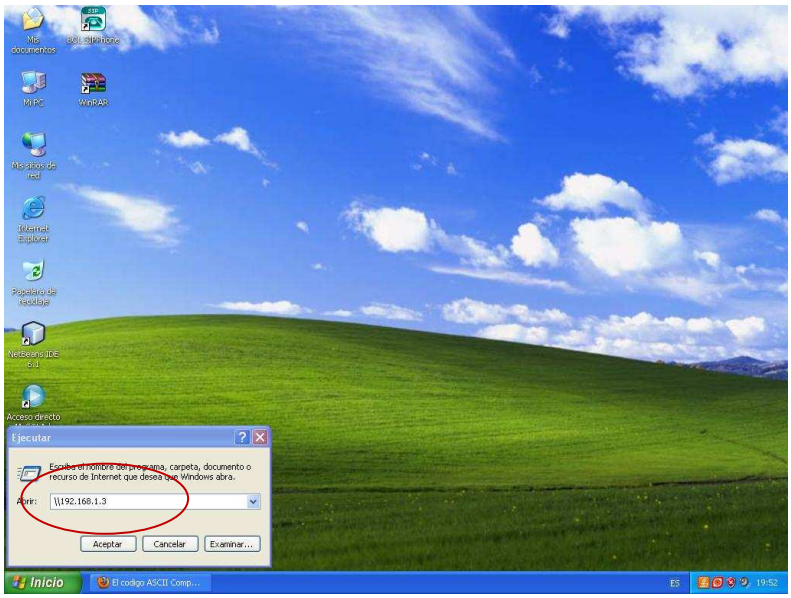

Fig. 5.18: “Servidor Virtual”

Se observa que ya no existen recursos compartidos entre el Servidor y la Máquina Virtual de prueba.

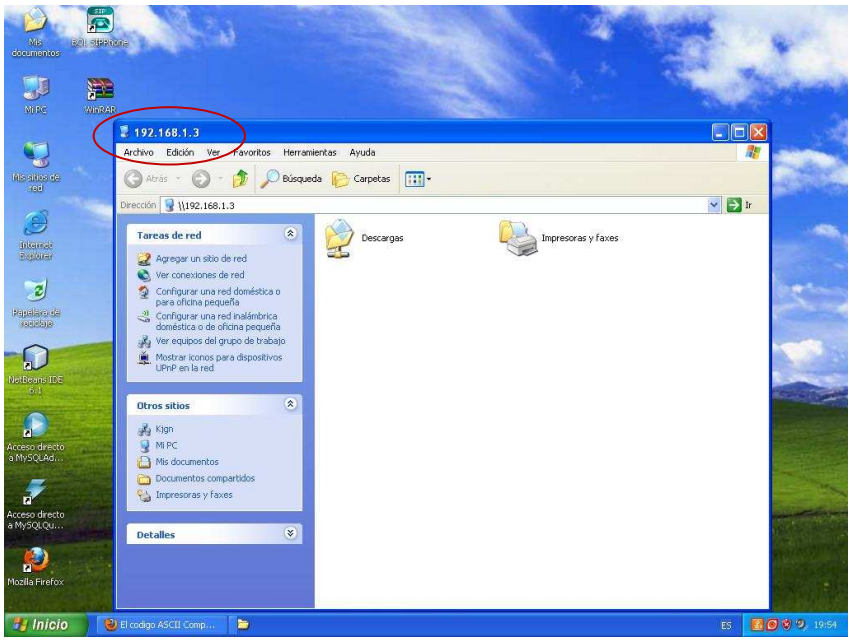

Fig. 5.19: “Recursos no Compartidos”

En la misma interfaz del Activación del Firewall, se encuentra el Capturador de paquetes o sniffer, que a través de una ventana permite conocer la interfaz de donde se va a capturar los paquetes, así como una descripción del host, el dispositivo y el tipo de red.

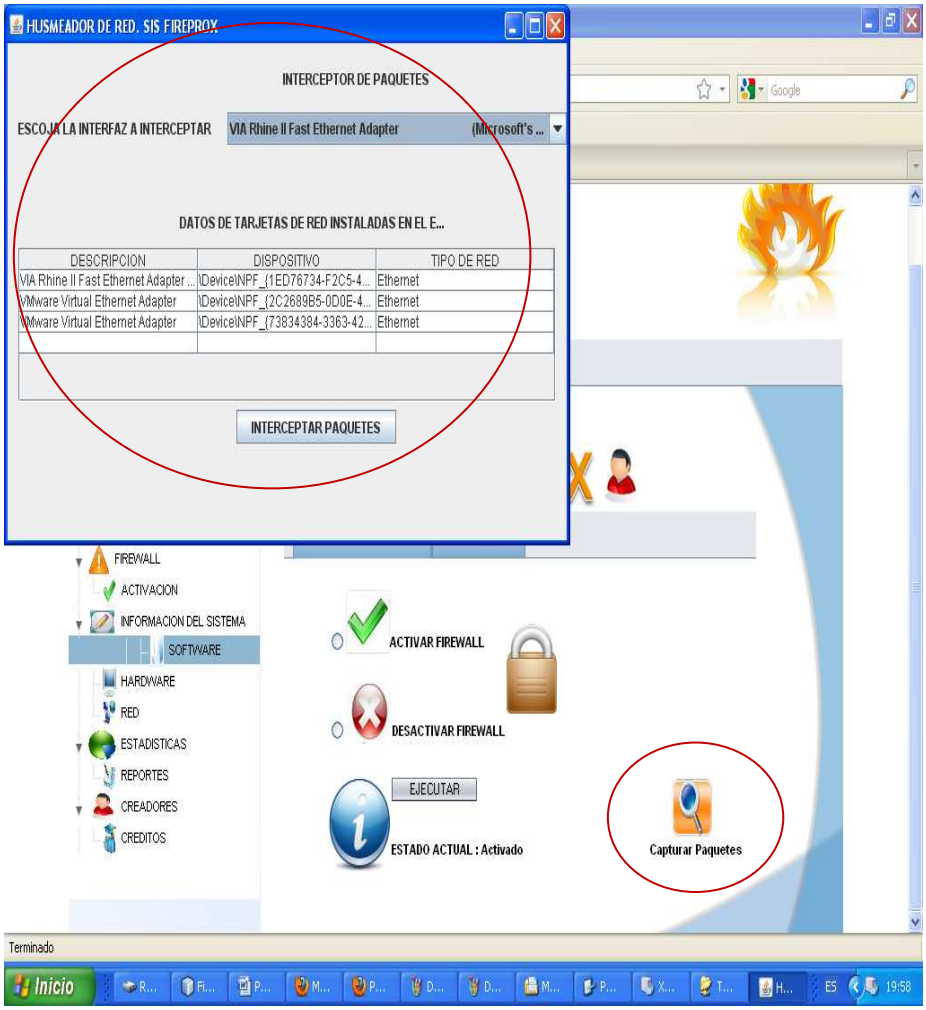

Fig. 5.20: “Capturador de paquetes”

En la programación del sistema, al ejecutar el sniffer, se observa como se procede a la captura de los paquetes que están ingresando a la red, esta información se visualiza a través de gráficas estadísticas en este caso son gráficas estadísticas de barras.

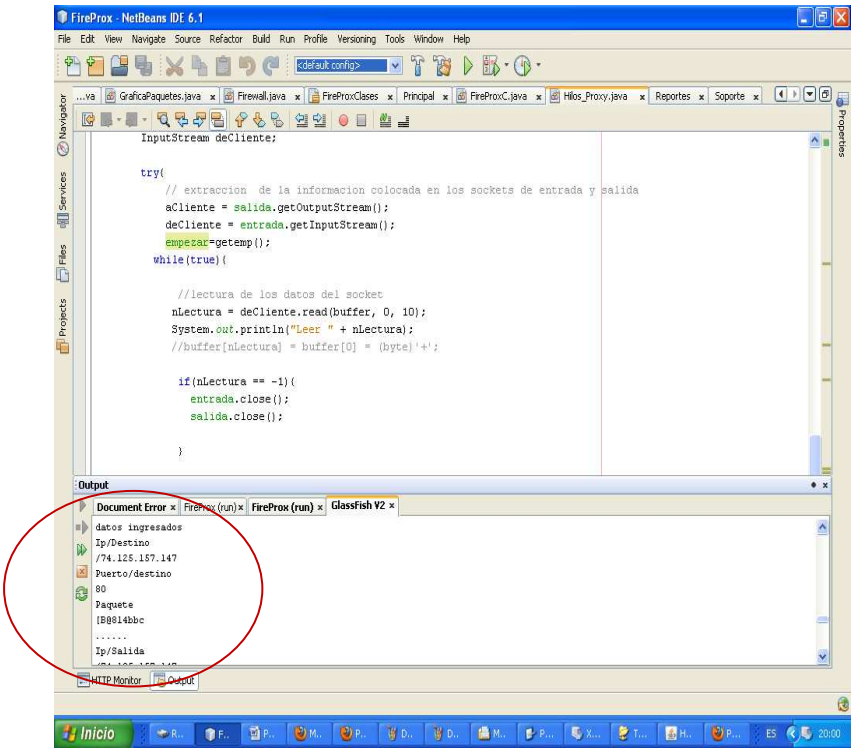

Fig. 5.21: "Paquetes ingresados a la red"

En este gráfico, se observa la cantidad de paquetes interceptados, según el día de la semana en que se desencadenó la acción.

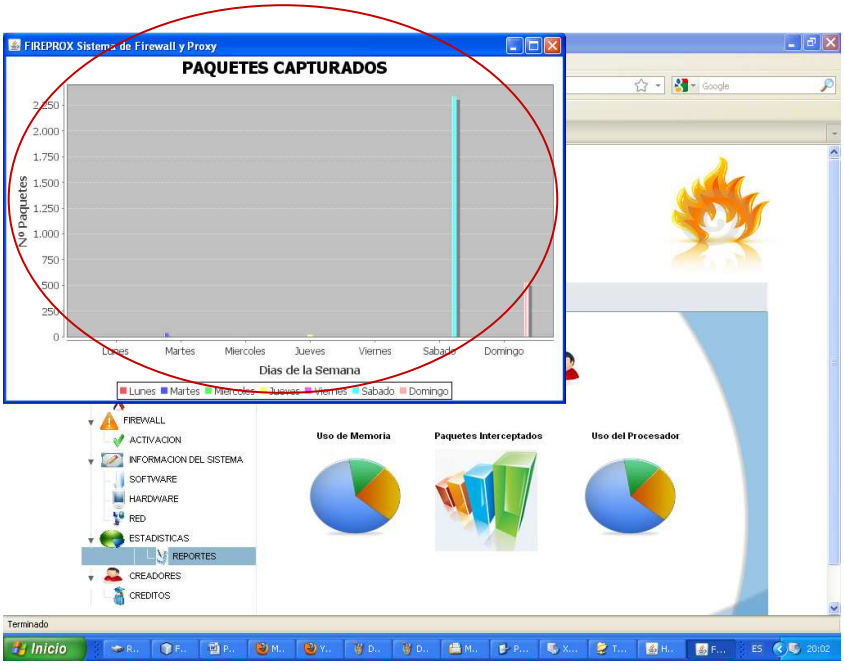

Fig. 5.22: "Reportes de Paquetes Capturados"

5.1.2.2.Pruebas en Linux

El sistema FireProx, como es de característica Multiplataforma se ejecuta en la plataforma operativa de GNU Linux, distribución Ubuntu, en donde se maneja la misma interfaz de usuario que en Windows, pero además desde un servidor ftp.

El cliente ftp Filezilla, instalado en Windows, puede acceder a las carpetas de descarga configurados en el servidor ftp en Ubuntu; esto se debe a que aún se encuentra habilitado el puerto 21, que se muestra en la Fig. 5.23:

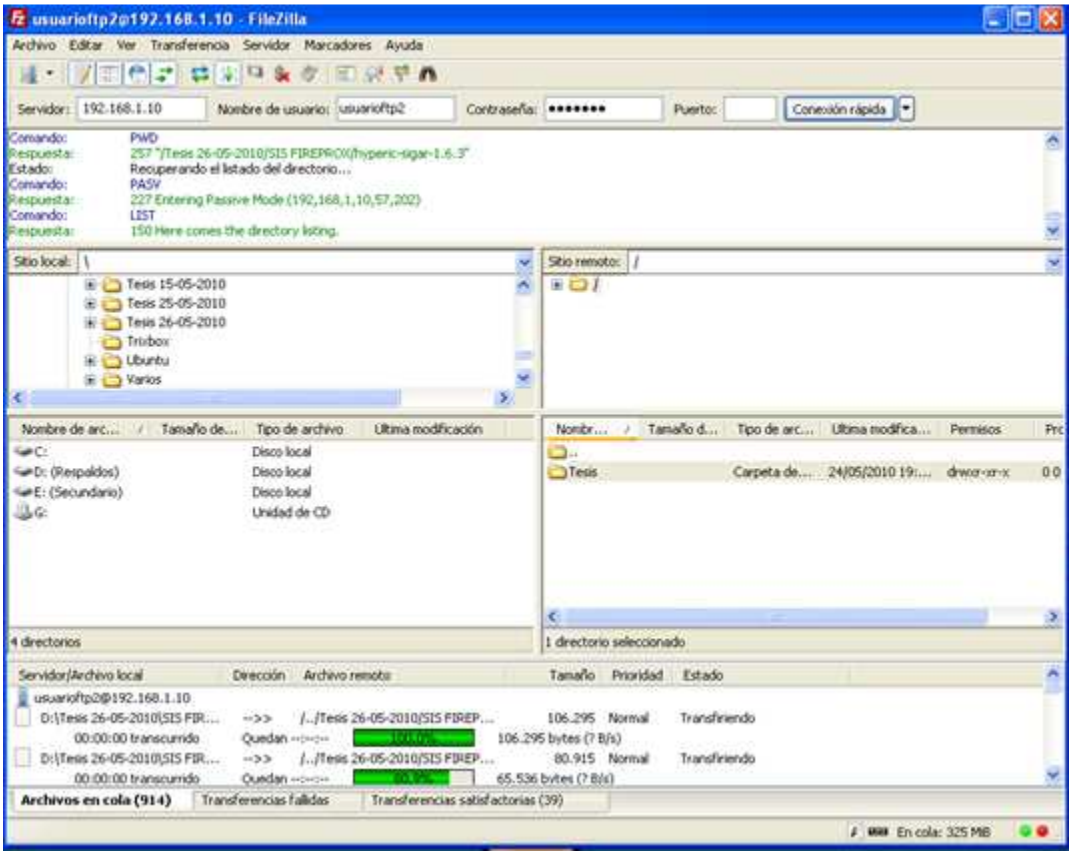

Fig. 5.23: “Cliente Filezilla accediendo al servidor ftp de Ubuntu”

La configuración del Firewall en Linux, permite controlar excepciones a través de puertos determinados, como se muestra en la Fig. 5.24, en donde se eligen las excepciones para ser activadas/desactivadas.

Fig. 5.24: “Excepciones del Firewall en Ubuntu”

Al activar el servidor, se puede compartir los recursos necesarios a través de las máquinas virtuales y de los usuarios creados para efectos de pruebas.

Fig. 5.25: “Activación de Excepciones – Compartir Recursos”

Al desactivar el Servicio ftp desde la interfaz de Fireprox en linux, se impide el acceso de los clientes, como se muestra en la Fig. 5.26:

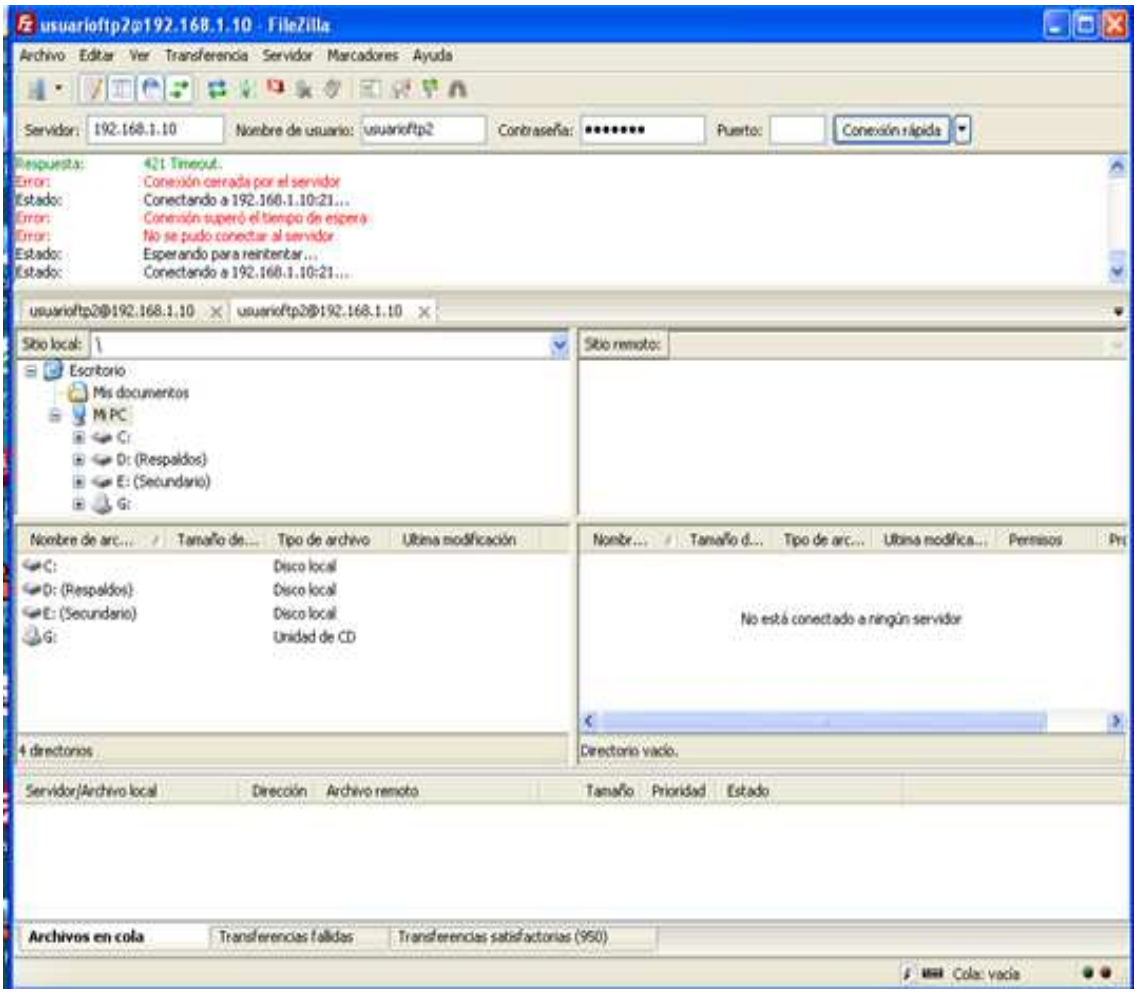

Fig. 5.26: “Desactivación de servicio ftp”

CONCLUSIONES Y RECOMENDACIONES

CONCLUSIONES

Al terminar el presente trabajo, se ha llegado a las siguientes conclusiones:

- La seguridad en las redes de comunicación, se ha convertido en un factor de relevancia al momento de diseñar nuevo software, porque cada vez se busca dar robustez y protección a la información manejada y proporcionar al usuario sistemas que permitan el control de intrusos a sus host, hoy en día es el anhelo de cualquier organización.

- En la actualidad, no existe solución técnica frente a un ataque de denegación de servicio, por lo tanto las infraestructuras de Internet representan el punto más débil de la seguridad global en un sistema. Por lo que se debe conocer como es el funcionamiento de un firewall, para identificar los puntos críticos y adherir políticas de seguridad que los fortalezcan.

- Los Servidores Proxy mejoran la seguridad y optimizan el rendimiento de una red informática dentro de una empresa, a través del anonimato, filtrado y reglas de uso del ancho de banda que pueden ser configuradas en el sistema.

- El uso de un firewall adecuado evita ataques y desastres que pueden ocasionar intrusos, males intencionados que intentan dañar o usufructuar la información contenida dentro de un Host.

- Las políticas de seguridad de red son de vital importancia a nivel empresarial, sin embargo por evitar costos que a largo plazo pueden

resultar beneficiosos, no se los realiza permitiendo que personas inescrupulosas manipulen los datos a su antojo.

- El manejo de interfaces amigables para el usuario mejoran la perspectiva y el funcionamiento de sistemas de seguridad; siendo mucho más atractivos para quienes los utilizan y menos confusos.

- El software libre y su proliferación ha permitido que la información ya sea parte de la globalización, fomentado el desarrollo humano, cognoscitivo e intelectual de desarrolladores de todo el mundo que desean fervientemente que la ciencia sea para todos y no solo para unos pocos.

- Java proporciona flexibilidad y adaptabilidad para ser ejecutado en cualquier tipo de sistema operativo; solo es necesario tener la máquina virtual adecuada y el software desarrollado se ejecutará sin ningún problema, porque java no trabaja directamente sobre el sistema operativo sino sobre su JVM(Java Virtual Machine), por lo tanto es necesario al tratar de obtener información del sistema, trabajar con clases o librerías elaboradas en código nativo y compiladas en código java para su utilización dentro de un software como es el caso de la librería sigar y jpcap.

- El uso de máquinas virtuales aunque ahorra recursos e inversión, produce un esfuerzo extra en la utilización de los recursos del host que la contiene.

- FireProx proporciona una Interfaz mucho mas intuitiva del manejo de un firewall, aunque posee limitaciones por ser un software prototipo, ha sido diseñado para que el usuario sepa lo que está haciendo y observando lo que ocurre al momento de establecer normas de seguridad; e inclusive su aplicación en Linux evite la configuración mediante el terminal de iptables para controlar los puertos de comunicación.

- Al ser probado en varios sistemas operativos se ha establecido que funciona sin problema alguno en Windows XP SP2, Vista y GNU/Linux Distribución Ubuntu.

- Los OverViews contienen ayuda técnica y de referencia de la construcción y funcionamiento de una determinada librería creada en java; capaz de brindar una idea clara al desarrollador de lo que se puede implementar dentro de un sistema con su inclusión.

- El Trabajar con Jsf, Jsp, y código Java mejora el desarrollo y la construcción de un sitio web, convirtiéndolo en dinámico, seguro y bastante confiable y sobre todo al ser software libre se reducen los costos en el desarrollo.

- Aunque se realice miles de estudios y se trate constantemente de mejor las seguridades informáticas siempre existirán usurpadores que violenten contra la integridad de la información de las instituciones, empresas u organizaciones, utilizando medios más sofisticados y causando daños; muchas veces irreversibles que pueden generar un caos en el buen funcionamiento de la estructura organizacional y del procesamiento adecuado de la información.

- Es relevante mencionar que diseñar sistemas bajo software libre que ayuden al estudio y desarrollo de nuevas herramientas que eviten la proliferación de ataques ocasionados por intrusos, beneficiará la comprensión lógica y conceptual del funcionamiento de un firewall y un servidor proxy; para divulgar su uso o realizar recomendaciones que mejoren su estructura funcional.

- El crear herramientas del tipo firewall y proxy; aunque ya existentes pero más fáciles de utilizar, ayuda a que los desarrolladores adapten sus sistemas convencionales; a las nuevas tendencias tecnológicas y de diseño del nuevo milenio que implican la creación de software mucho más

seguros, flexibles, adaptables a las necesidades de cada usuario y sobre todo fáciles de utilizar.

- Cada Sistema Operativo consta de su propio bloque de seguridad integrada a su Shell que permite el manejo de los diferentes puertos de comunicación como en el caso de Windows, en el que se pueden utilizar comandos netsh para el control de accesos y en GNU/Linux iptables específicamente en Ubuntu el UFW que permite o niega la utilización de un servicio. El correcto manejo de estos comandos ayudara a que cada vez encontremos nuevas y mejores maneras de hacer menos vulnerable la seguridad de la información.

RECOMENDACIONES

- Al trabajar con jsf se recomienda activar la opción Add Bindding Atribute colocando el cursor sobre el componente de la página y dando clic derecho sobre él; ya que de otra manera no se podrá realizar ningún tipo de operación.

- Es recomendable al añadir librerías con pantallas gráficas a una página web, cambiar de las propiedades del JForm de la opción defaultCloseOperation de exit por Dispose, porque de no ser así al momento de cerrar la pantalla se cerrara toda aplicación inclusive el sitio web.

- Se puede obtener librerías y códigos de clases nuevas de java a través de foros o simplemente visitando la página de Sun y el OverView de cada paquete de java.

- Para descargar código y paquetes que brindan funcionalidad a los programas en java, visitar la página de sourceforge.net, que día a día esta innovando con aportaciones de diferentes programadores que han enriquecido el desarrollo del software libre en diferentes lenguajes, permitiendo de esta manera que se puedan realizar programas mucho más elaborados y con gran capacidad de respuesta.

- Al programar en java es importante estructurar el código en paquetes y clases que realicen la mayoría de operaciones que requiera el sitio web, esto evitará redundancia del código y organizará el programa de tal manera que sea fácil identificar los errores y resolverlos.

- Si se posee una conexión a Internet; utilizar para la descarga de paquetes en Linux, el gestor Synaptic, porque realizará el engorroso trabajo de instalar los programas o software requeridos de manera rápida y sencilla sin la necesidad de utilizar líneas de comandos.

- Al Instalar MySql Sever o cualquier otro programa que requiera permisos de administrador como el Servidor Web GlassFish, es necesario iniciar sesión como usuario root en GNU/Linux distribución Ubuntu utilizando el siguiente comando: sudo passwd ingresado en la terminal del usuario creado por defecto; permitirá establecer una contraseña para el súper usuario root dejando configurar el software requerido dentro de GNU/Linux Distribución Ubuntu para el desarrollo de cualquier proyecto.

- Si no se maneja adecuadamente o se tiene conocimientos limitados en GNU/Linux no es recomendable habilitar al usuario root ya que esto podría ocasionar daños dentro del sistema operativo y la pérdida de información importante.

- Si no se posee la infraestructura necesaria, los recursos, equipos y demás; es recomendable la utilización de máquinas virtuales esto ahorrará costes en el desarrollo y facilitará las pruebas del sistema.

- Para la utilización de GNU/Linux en desarrollo de software son mucho más fáciles y amigables con el usuario las distribuciones de Fedora y Ubuntu; puesto que sus interfaces parecidas a las de Windows ayudan al desarrollador a que realice procesos de manera intuitiva.

BIBLIOGRAFÍA

- ALDEGANI, Gustavo, *Seguridad Informática,* Ediciones MP, Argentina. 1997

- CHAPMAN, D. & Fox, *Firewalls pix de cisco secure: reduzca el riesgo de ataques a las redes con el libro oficia,* Madrid, 2002

- GILBERT, David, *The JFreeChart Class Library*, 20 Abril 2009

- GONCALVES Marcus, *Firewalls: a complete guide Standards and Protocols Series*, 2ª Edición, EditorMcGraw-Hill, 2000

- KARANJIT Siyan y Chris Hare, *Firewalls y la seguridad en internet*, Editor Prentice-hall Hispanoamericana, 1997

- MARC Royer Jean, *Seguridad en la informática de Empresa*, Ediciones ENI, 2004

- MCCLURE, S. *Hackers 3: secretos y soluciones para la seguridad de redes*, Editor McGraw Hill, Madrid, 2002

- NORTHCUTT, S. & Novak, J. *Guía avanzada detección de intrusos*-2ª Edición, Editorial Pearson Educación, Madrid, 2001

- SOCORRO Pilar*, Glosario de términos para manejarnos en la red*, 2001

- SCAMBRAY, J. y otros, *Hackers 2: secretos y soluciones para la seguridad de redes*, Editor McGraw Hill, Madrid 2001

- MCNAB Andy, *Firewall*, Editor Simon & Schuster, 2001

- ORDINAS Barceló José y otros, *Protocolos y Aplicaciones Internet*, Primera Edición, Editorial UOC, 2008

- YANN Arthur Nicolás, *Introducción a JSF con NetBeans*, 15 mayo 2008

- ZWICKY Elizabeth, *Building Internet Firewall*, Editions O'Reilly & Associates, 2000

Sitios Web

- ÁLVAREZ Miguel Ángel, *Java Server Faces*, ww.proyectoremar.tripod.com

- ARDITA Julio César, *Definición de Seguridad* Lógica,www.cybsec.com

- CANALES Mora Roberto, *"Gráficas en Java con JfreeChart"*, Sección Tutoriales, www.adictosaltrabajo.com, último acceso: febrero del 2010.

- FRICK Didier. "Java Transparent Proxy", Sección FreeSoftware, www.dfr.ch/en/proxy.html, consultada en noviembre del 2009.

- GALLARDO José. "El firewall de XP SP2 desde la línea de comandos", Sección Foros, www.fermu.com, último acceso: marzo del 2010.

- HABLUTZEL Jaime, WordPress.com. "JFreeChart: Fácil creación de gráficos estadísticos en Java", Blog El espacio de Jaime, http://elespaciodejaime.wordpress.com/tag/jfreechart/, último acceso: diciembre del 2009.

- HOMEPAGE Gautam. “A Simple Packet Sniffer using Java”, http://gforgeek.blogspot.com/2005/04/simple-packet-sniffer-using-java.html, último acceso: febrero del 2010.

- LAGUNA Julio, RedRibera.es. “Un programa en Java. Series de tiempo Parte III.”, Sección Tutoriales, www.redribera.es/formacion/tutorialesl, último acceso: febrero 2010.

- MENÉNDEZ Maykel J. “Conceptos fundamentales y Servicios del Sistema Operativo”, Sección Monografías > computación > Sistemas Operativos, www.monografias.com/trabajos81/conceptos-fundamentales-servicios-sistema-operativo/conceptos-fundamentales-servicios-sistema-operativo.shtml, último acceso: diciembre del 2009.

- TELLA Llop José Manuel, “Configuración del firewall de XP - sp2 comandos de línea”, http://multingles.net/docs/jmt/fwxpsp2.htm, último acceso: marzo del 2010.

- s/a, “Seguridad informática”, Sección Seguridad informática | Seguridad de la Información, http://es.wikipedia.org, último acceso: noviembre del 2009.

- s/a, “Sistema operativo”, Sección Sistemas operativos, http://es.wikipedia.org/wiki/Sistema_operativo, último acceso: noviembre del 2009.

- s/a, “¿Qué es un Sistema Operativo?”, sección Preguntas frecuentes de la web, www.masadelante.com, último acceso: diciembre del 2009.

- s/a, “A simple proxy server”, Sección Java » Network Protocol, www.java2s.com/Code/Java/Network-Protocol/Asimpleproxyserver.htm, último acceso: diciembre 2009.

- s/a, “Java SOCKS Proxy”, Sección Find Software, http://jsocks.sourceforge.net/, último acceso: diciembre del 2009

- s/a, “Guía documentada para Ubuntu”, Sección home Page Ubuntu, www.guia-ubuntu.org, último acceso: enero del 2010.

- s/a, “Obtener información del sistema [memoria disponible, %CPU, espacio en disco] en Java”, http://casidiablo.net/capturar-informacion-sistema-operativo-java/, último acceso: enero del 2010.

- s/a, Object Refinery Limited. “Página Oficial de JfreeChart”, http://www.jfree.org/jfreechart/, último acceso: enero del 2010

- s/a, “Ubuntu”, Sección Ubuntu, http://es.wikipedia.org/wiki/Ubuntu, último acceso: enero del 2010.

- s/a, “Windows XP”, Sección Microsoft Windows, http://es.wikipedia.org/wiki/Windows_XP, último acceso: enero del 2010.

- s/a, “Jsniff - Tcp/Ip Packet Sniffer In Java Example console packet sniffer”, Sección foros, www.rohitab.com, último acceso: febrero del 2010.

- s/a, “Netsh”, Categoría Aplicaciones Informáticas, http://es.wikipedia.org/wiki/Netsh, último acceso: abril 2010.

- s/a, UFW es (C) 2008, Canonical Ltd. “UFW Firewall no complicado de Ubuntu”, http://ubuntusur.org/?p=385, último acceso: mayo del 2010.

- s/a, Ubuntu Documentation Team wiki page. “Firewall”, Sección Ubuntu Documentation > Ubuntu 10.04 > Ubuntu Server Guide > Security > Firewall, http://doc.ubuntu.com, último acceso: mayo del 2010.

GLOSARIO DE TÉRMINOS

Amenaza

Causa potencial de un incidente no deseado, el cual puede causar el daño a un sistema o la organización.

Análisis de riesgo de seguridad informática

Uso sistemático de la información para identificar fuentes y estimar el riesgo.

Base de datos

Colección de datos organizada de tal modo que el computador pueda acceder rápidamente a ella. Una base de datos relacional es aquella en la que las conexiones entre los distintos elementos que forman la base de datos están almacenadas explícitamente con el fin de ayudar a la manipulación y el acceso a éstos.

Certificado

Es un documento electrónico en el cual la Autoridad de Certificación (AC) acredita mediante su firma digital que la clave pública pertenece a su propietario. También se denominan Certificados de usuario y de clave pública.

Cifrado

Proceso utilizado para transformar un texto a una forma ininteligible de manera que los datos originales no puedan ser recuperados (cifrado de una vía) o sólo puedan ser recuperados usando un proceso inverso de descifrado (cifrado de dos vías).

Cliente:

Programa que se usa para contactar y obtener datos de un programa de servidor localizado en otro computador, a menudo a gran distancia. Cada programa cliente está diseñado para trabajar con uno o más tipos de programas servidores específicos, y cada servidor requiere un tipo especial de cliente. Programas de software que proporcionan acceso a los recursos de la red al trabajar con la información de un servidor.

Código Malicioso:

Es un término genérico que comprende todos los tipos de programas específicamente desarrollados para ejecutar acciones maliciosas en un ordenador.

Conexión:

Circuito virtual de transporte que se establece entre dos programas de aplicación con fines comunicativos.

Contraseña:

Palabra o cadena de caracteres, normalmente secreta, para acceder a través de una barrera. Se usa como herramienta de seguridad para identificar usuarios de una aplicación, archivo, o red. Puede tener forma de una palabra o frase de carácter alfanumérico, y se usa para prevenir accesos no autorizados a información confidencial.

Encriptación:

Este término describe la acción de codificar los datos contenidos en un mensaje o documento a fin de impedir que nadie, excepto el destinatario de los mismos puede leerlos. Actualmente existen varios tipos de programas de encriptación de

datos para preservar la seguridad de la transmisión de información a través de la red.

Enlace:

Conexión a otro documento web, por medio de la dirección URL. Los enlaces aparecen en el texto de un documento web en forma de texto subrayado y de distinto color. Permiten al usuario presionar el botón del ratón sobre dicho texto y automáticamente saltar a otro documento, o a otro servidor, o enlazar a otra parte del mismo documento.

Estándar:

Especificación que se utiliza como punto de partida para el desarrollo de servicios, aplicaciones, protocolos, etc. y que regula la realización de ciertos procesos o la fabricación de componentes para garantizar la interoperabilidad.

Estrategias de Seguridad de Información:

Es la elaboración de los procesos de cada una de las políticas del modelo de seguridad informática, según las propias características de la empresa.

Firewall:

Sistema de seguridad, encargado de chequear y bloquear el tráfico en una red, se coloca entre una red local e Internet para asegurar que todas las comunicaciones entre se realicen conforme a las políticas de seguridad de la organización que lo instala.

Además, estos sistemas suelen incorporar elementos de privacidad, autentificación, etc.

Host:

Ordenador que permite a los usuarios comunicarse con otros sistemas centrales de una red. Los usuarios se comunican utilizando programas de aplicación, tales como el correo electrónico, Tenet, WWW y FTP.

Ingeniería Social:

Se le dice ingeniería social a la acción de engañar a un usuario para que sea él quien actúe, ejecutando un archivo o relevando datos secretos como una clave.
Se utiliza la astucia para convencer a una persona a dar por si mismo información acerca de su sistema

Interface:

En su sentido más general, un internet es una gran red de equipos compuesta por un gran número de redes más pequeñas. Cuando este término está escrito en mayúsculas, hace referencia a la red física que compone el web y que hace posible el correo electrónico en todo el mundo. Es la mayor red Internet del mundo. Tiene una jerarquía de tres niveles formados por redes de eje central (backbones como, por ejemplo, NSFNET y MILNET), redes de nivel intermedio y redes aisladas (stub networks). Internet es una red multiprotocolo.

Modelo de Seguridad de información:

Un modelo formal permite probar teoremas y en particular comprender las propiedades de los objetos del modelo

Protocolo:

Descripción formal de formatos de mensaje y reglas que dos máquinas deben seguir para intercambiar dichos mensajes.

Proxy:

Programa que multiplexa, generalmente las conexiones TCP/IP, de usuario. Generalmente cuando se habla de Proxy hace referencia a conexiones Internet.

Servidor:

Entre dos equipos conectados es el que ofrece un servicio al otro que actúa como cliente.