INDICE GENERAL

Agradecimiento

Le agradecemos a Dios por habernos dado la fuerza suficiente para luchar contra todos los obstáculos que se nos presentaron a lo largo de nuestra carrera y sobre todo también a nuestros padres que de alguna manera estuvieron allí apoyándonos incondicionalmente para salir adelante y alcanzar nuestra meta; gracias por ese apoyo brindado.

Dedicatoria

Este trabajo es inspirado en el esfuerzo realizado por cada uno de nuestros padres, en el ejemplo de superación que siempre nos inculcaron al ser profesionales.

Solo les decimos gracias de todo corazón por el apoyo incondicional.

Atte.

Hugo Mauricio Cobo Rojas

Jaime Enrique Falcones Bermúdez

José Fernando Rivera Neira

Tribunal de Graduación

Presidente	**1er. Vocal**
2do. Vocal	**Secretario**

DECLARACIÓN EXPRESA

“La autoría de la tesis de grado corresponde exclusivamente a los suscritos, perteneciendo a la Universidad de Guayaquil los derechos que generen la aplicación de la misma”

(Reglamento de Graduación de la Carrera de Ingeniería en sistemas Computacionales, Art. 26)

Mauricio Cobo

mauricio_cobo@hotmail.com

Jaime Falcones

jfalcones0999@hotmail.com

José Rivera

joseriveraneira@hotmail.com

Resumen

Nuestra herramienta trata en si de formar un escudo de protección contra ataques informáticos de tipo maliciosos, implementado una interfaz Web, la cual será diseñada de una manera tan amigable que los usuarios no tengan contratiempos al momento de configurarla o manejarla.

Se debe utilizar una arquitectura de red adecuada la cual pueda detectar los protocolos de red que se encuentran habilitados para transferencia o navegación de información. Además el IDS debe estar bien ubicado dentro de la red para que pueda analizar todo el tráfico de la red.

Debe tener la facultad de obtener datos de los distintos recursos del sistema para evitar ataques internos y de uso indebido o fuera de lo normal por parte de los usuarios.

Todo esto le permitirá al sistema establecer mediante métodos y reportes estadísticos, las pautas del comportamiento necesario para detectar posibles anomalías pudieran afectar el correcto desempeño de la red.

Toda esta información servirá para el proceso de toma de decisiones y análisis por parte del administrador de la red.

Summary

Our tool question on whether to from a shield of protection against malicious computer attacks type, implemented a Web interface, which Hill be designed so friendly that user have no setbacks when configure or mange.

Must use an appropriate network architecture which can detect network protocols that are eligible for transfer or navigation information. In addition the IDS must be well located within the network so that it can analyza all network traffic.

You nust be able to extract data from multiple system resources to prevent attacks and internal misuse or out of the ordinary by users.

This will allow the system by establishing methods and statistical reports, the patterns of behavior necessary to detect possible anomalies could affect the proper performance of the network.

Índice de Contenido

Índice de Figuras

Índice de Cuadros

INTRODUCCIÓN

En la actualidad la cantidad de intentos de accesos no autorizados a la información que existe en Internet ha crecido durante estos últimos años. Muchas compañías, normalmente por motivos de coste, han migrado información clave a Internet, exponiéndola hacia el exterior. Además, para comodidad de los trabajadores que solicitan tele trabajo, las compañías han tenido que "abrir sus puertas" para permitir la conexión a la intranet de la oficina desde casa.

Razón por la cual nace la necesidad de usar dispositivos como el IDS (Sistema de Defección de Intrusos). Esta herramienta detecta anomalías que pueden ser indicio de la presencia de ataques o falsas alarmas y analiza el tráfico de paquetes IP en tiempo real

Cuando los atacantes comprometen los sistemas de entrada, ellos también tienen acceso a datos de la organización. La incorporación de cortafuegos con políticas correctas puede minimizar el que muchas redes queden expuestas. Sin embargo, los atacantes están evolucionando y aparecen nuevas técnicas como los Troyanos, gusanos y escaneos silenciosos que atraviesan los cortafuegos.

Las vulnerabilidades no solo afectan a sistemas tradicionalmente seguros, sino que afectan incluso a sistemas de seguridad: cortafuegos y a los propios IDSs. Esto se debe en parte a un crecimiento del número de auditorias que las empresas de software aplican a sus productos y por el aumento de interés en el campo de la seguridad por parte de los profesionales de la informática.

Los intrusos también intentan atacar a los IDS, ya sea saturándolos de tráfico a analizar o bien mediante herramientas que les proporcionan información falsa de lo que pasa por la red.

CAPITULO 1

ENTORNO PARA EL DESARROLLO DEL PROYECTO

1. Problemática

1.1. Problema

El número de equipos que se encuentran conectados a una red es muy relevante e importante, a mayor número de equipos conectados mayor es el número de eventos del sistemas a realizar es por esto que esta tarea se vuelve cada vez imposible para que un recurso humano del departamento de sistemas pudiera analizarlo o detectar alguna anomalía dentro de nuestra red.

La red puede verse afectada por ataques internos o externos. Los ataques internos son realizados por usuarios inexpertos o maliciosos que pudieran entrar de forma desautorizada a las máquinas principales de la empresa.

El problema se da cuando dentro de una empresa, no cuentan con una herramienta que les permita detectar, capturar, analizar los datos que circulan por la red, es que la información que circule por la misma este

siendo alterada o capturada por parte de usuarios maliciosos para un uso indebido.

Un ataque puede darse desde distintas máquinas, por lo cual nace la necesidad de registrar eventos que se suscita a lo largo de nuestra red y muchas veces estos se ven alterados por la mala ubicación de nuestro IDS en la arquitectura de la red.

1.2. Solución a Problemática

Nuestra herramienta trata en si de formar un escudo de protección contra ataques informáticos de tipo maliciosos, implementado una interfaz Web, la cual será diseñada de una manera tan amigable que los usuarios no tengan contratiempos al momento de configurarla o manejarla.

Se debe utilizar una arquitectura de red adecuada la cual pueda detectar los protocolos de red que se encuentran habilitados para transferencia o navegación de información. Además el IDS debe estar bien ubicado dentro de la red para que pueda analizar todo el tráfico de la red.

Debe tener la facultad de obtener datos de los distintos recursos del sistema para evitar ataques internos y de uso indebido o fuera de lo normal por parte de los usuarios.

Todo esto le permitirá al sistema establecer mediante métodos y reportes estadísticos, las pautas del comportamiento necesario para detectar posibles anomalías pudieran afectar el correcto desempeño de la red.

Toda esta información servirá para el proceso de toma de decisiones y análisis por parte del administrador de la red.

1.3. Misión

“Darle al administrador de red una herramienta fácil de usar y de configurar, que recoja datos importantes de en el monitoreo de la red y que muestre un informe detallado del tráfico y eventos que se produjeron en el monitoreo del mismo”.

1.4. Visión

Proporcionar un sistema que controle la seguridad de la información de cada usuario por medio de la red.

1.5. Objetivos Generales

Al momento de realizar el monitoreo, se utilizara una herramienta que nos proporcione un cuadro con reportes gráficos estadísticos del tráfico de red, ataques por medio de pitaras informáticos (hackers), que ayude al encargado de la red a tomar decisiones en base a reportes estadísticos, que sean de fácil entendimiento para el administrador que lo maneje.

Así también disponer de una base de datos de “firmas” que contengan los ataques que se hayan suscitado anteriormente.

Interfaz Web amigable que sea entendible para un operador cualquiera y fácil de configurar para el administrador de la red.

1.6. Objetivos Específicos

El aplicativo tendrá un proceso de seguridad en base al ingreso de un usuario con su respectiva contraseña, lo que le permitirá el ingreso o negación a la interfaz Web.

Se podrán crear dos tipos de perfiles de Usuario que son: Administrador y Operador; los cuales tendrán privilegios diferentes. El Administrador podrá crear nuevos usuarios, así como también asignar perfiles.

Los Operadores no tienen atributos para crear usuarios o modificar la configuración del IDS, solo tiene permisos para poder monitorear los reportes elaborados.

Los reportes se producirán en función de la hora y/o día (ej. backups, mantenimientos, etc.)

Se desarrollará nuestro aplicativo usando programación distribuida y haciendo uso de estándares para su implementación tanto a nivel Web como también con el IDS.

Se manejará una bitácora de ataques suscitados, los cuales nos servirán para prevenciones posteriores y saber que usuarios han tratado de atacar.

1.7. Alcances

- ✓ En lo que se refiere a los alcances de nuestro proyecto, trataremos de dar a conocer los de mayor importancia para el manejo, configuración, implementación tecnológica y de la metodología implantada en su desarrollo en nuestro aplicativo.

- ✓ El monitoreo nos proporcionará un cuadro con reportes gráficos estadísticos del tráfico de red, ataques suscitados y ayudará a la toma de decisiones por parte del Administrador que lo utilice. Los reportes se producen en función de la hora y/o día.

- ✓ Se implementará seguridades a nivel de accesos al aplicativo, en base al ingreso de usuarios con su respectiva contraseña.

- ✓ Se puede mejorar el control de intrusos agregando características de prevención, es decir, si el administrador desea restringir el acceso a un archivo o directorio, podrá configurar el envío de una señal al proceso infractor para impedir el acceso.

- ✓ Cada vez que se detecte un ataque se activará las seguridades del caso y se generará una alerta que será informada inmediatamente al Administrador de la Red por medio de correo electrónico y a la vez le enviara un mensaje de texto a su teléfono alertándolo de dicho ataque.

- ✓ Cuando se produzca un ataque externo nuestra aplicación bloqueará los puertos de comunicación y almacenará dicha dirección IP en una bitácora de direcciones maliciosas.

- ✓ Cuando un intruso viole la seguridad, nuestro sistema podrá obtener todos los datos disponibles del sistema que se esta utilizando (la maquina que esta violando la seguridad) como: versión, particiones, hora y fecha del ataque y fecha en la que se desconectó de la red.

- ✓ En el momento que se detecte que un usuario esta ocupando mayor cantidad de ancho de banda del ya establecido; el sistema lo que hará es bloquear a dicho usuario y enviara un mensaje al administrador indicando que maquina esta realizando dicho proceso.

- ✓ El administrador a través de la interfaz amigable podrá ver, administrar y controlar los accesos, reportes gráficos estadísticos del tráfico de red, los posibles ataques suscitados; tendrá la facultad de modificar a los usuarios.

1.8. Ventajas y Beneficios de la Solución

- ✓ Seguridad y monitoreo de la información que circula por toda la red.

- ✓ Reportes que ayudarán para identificar posibles causas de ataques futuros.

- ✓ Generación de reportes más completos ya que podrá almacenar patrones en la base de datos.

- ✓ El administrador de la red entrega reportes más detallados a los responsables ejecutivos de la empresa.

- ✓ Se obtendrá un mejor control de los protocolos y tipo de información que esta circulando por la red por medio del análisis que emite el administrador de la red en base a nuestro sistema.

- ✓ Poner en alerta al administrador o a los responsables de la red cuando ocurra un suceso anormal, emitiendo un e-mail al administrador.

- ✓ Facilitar una interfaz amigable, detallada, concisa y fácil de manejar para cualquier usuario de todos los procesos administrativos que conforma la herramienta.

- ✓ El aplicativo es multiplataforma razón por la cual puede adaptarse a cualquier sistema operativo.

- ✓ Utiliza la tecnología 3 capas para aplicaciones Web.

- ✓ El IDS estará en un solo equipo potente para un mejor desenvolvimiento cuando este analizando los paquetes.

- ✓ La base de datos estará en un solo equipo y la definición de nuestra tecnología 3 capas

- ✓ La Interfaz Web se la realizará utilizando páginas JSP que se levantan con un servidor de aplicaciones en nuestro caso será Apache Tomcat 5.0.

1.9. Desventajas

Los sistemas de prevención de intrusiones presentan también una serie de desventajas y generan serios cuestionamientos sobre su efectividad, algunos de ellos se listan a continuación:

- ✓ No existen tecnologías nuevas; fuera de una mayor integración entre controles de diferentes tipos, los IPS utilizan las mismas tecnologías que los IDS para la detección de eventos de seguridad (patrones, funciones estadísticas, algoritmos de inteligencia artificial, etc.)

- ✓ La concentración de elementos de seguridad en un solo punto genera los llamados “puntos únicos de fallo”.

- ✓ El compartir recursos puede traer problemas; cuando se combina un IDS con un Firewall, un equipo con el doble de capacidad de procesamiento no es lo mismo que 2 equipos con la mitad de capacidad de procesamiento (ambos utilizan un mismo procesador, así como las mismas entradas y salidas de datos; esto genera cuellos de botella si la carga de trabajo es excesiva).

1.10. Metodología

1.10.1. Metodología del Análisis

Usaremos el modelo orientado a objetos ya que nuestro sistema interactúa mucho con objetos más porque la programación es orientada a ello.

El hecho de que nuestra programación haga uso de herencias en java e invocación a otras clases ya definidas habla a la clara que estamos haciendo uso del método antes descrito, esto nos ha ayudado a establecer las bases para la realización del software a desarrollar.

1.10.1.1. Diagrama de Clases

Especifica a una clase que es una categoría o grupo de cosas que tienen atributos y acciones similares. Su diseño es un rectángulo.

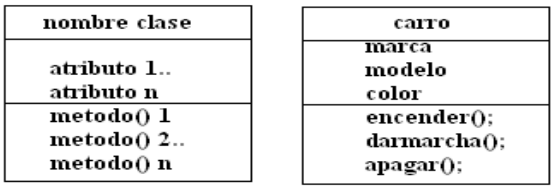


Figura 1

1.10.1.2. Diagrama de Objetos

Representa a un objeto que es una instancia de la clase (una entidad que tiene valores específicos de los atributos y acciones. Su diseño es un rectángulo, como una clase, pero el nombre esta subrayado. El nombre del objeto está a la izquierda separada por (**:**) del nombre de la clase al que pertenece.

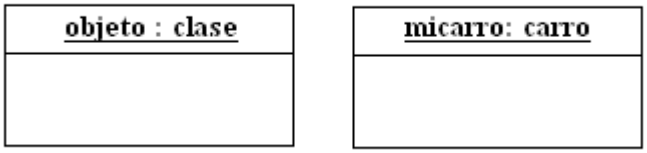


Figura 2

1.10.1.3. Diagrama Objeto-Relación

Este gráfico nos muestra como se realiza la relación entre las tablas de la base de datos.

1.10.1.4. Esquema de Datos

Es la representación de una arquitectura en general, de cómo está estructurada la arquitectura de datos.

1.10.1.5. Diagrama de Casos de Uso

Hay un actor que inicia un caso de uso y otro (posiblemente el que inició, pero no necesariamente) que recibirá algo de valor de él. La representación gráfica es directa. Una elipse representa a un caso de uso, una figura agregada representa un actor. El actor que inicia se encuentra a la izquierda del caso de uso y el que recibe esta a la derecha.

Una línea asociada conecta a un actor con el caso de uso. Se usa un rectángulo con el nombre del proceso para representar el confín del sistema, el rectángulo envuelve a los casos de uso del sistema.

Figura 3

1.10.1.6. Diagrama de Bloques

Los diagramas de bloque son útiles para entender como se relacionan los distintos departamentos, unidades operativas, etc., ante un determinado proceso; explica la forma en que interactúan los principales bloques que conforman la estructura arquitectónica.

1.10.2 Metodología del Diseño

Contiene los siguientes métodos:

1.10.2.1. Diseño de Subsistemas

Es una representación de cada uno de los subsistemas, para permitir al software conseguir sus requisitos definidos, por el cliente e implementar la infraestructura que soporte los requerimientos del cliente.

Se deriva considerando los requerimientos globales del cliente (representada por los casos de uso).

1.10.2.2. Diseño de Clases y Objetos

Contiene la jerarquía de clases, que permiten al sistema ser creado usando generalizaciones y cada vez especializaciones más acertadas. En cada capa también contiene representaciones.

Es trazado de la descripción de atributos, operaciones y colaboraciones contenidas en el modelo CRC.

1.10.2.3. Diseño de Mensajes

Contiene detalles de diseño, que permite a cada objeto comunicarse con sus colaboradores, esta capa establece interfaces externas e internas para el sistema.

Es manejado por el modelo objeto-relación

1.10.2.4. Diseño de Capa de Responsabilidades

Contiene estructuras de datos y diseños algorítmicos, para todos los atributos y operaciones de cada objeto.

Este diseño es derivado del uso de atributos, operaciones y colaboraciones descrito en el modelo CRC.

1.11. Arquitectura

La aplicación hace uso de una arquitectura muy flexible y adaptable a los conceptos actuales en la implementación de una herramienta IDS y su aplicativo bajo ambiente Web.

1.11.1. Snort

Este IDS capturará paquetes que circulan por la red para el proceso de almacenamiento de todas las instancias que ocurren en la red.

1.11.2. PostgreSQL

Es la base de datos Open Source que utilizaremos para integrarla con el Snort.

1.11.3. Tomcat 5.0

Es un servidor de aplicaciones, que nos ayudará a levantar nuestras páginas JSP.

1.12. Recursos a Utilizar

1.12.1. Humano

- ✓ 3 desarrolladores con sueldo de $ 1200 mensuales por 7 meses: $10500

1.13. Cronograma

Actividad	Tiempo en días	fecha _ inicio	fecha _ fin
Instalación IDS	9	14/05/2007	24/05/2007
Pruebas IDS	10	18/05/2007	31/05/2007
Instalación Base de Datos	23	22/05/2007	21/06/2007
Conectividad entre IDS y Base de Datos	13	22/06/2007	10/07/2007
Implementación	3	12/07/2007	14/07/2007
Desarrollo aplicativo parte 1	56	18/07/2007	30/09/2007
Pruebas	23	15/09/2007	15/10/2007
Desarrollo aplicativo parte 2	35	16/10/2007	30/11/2007
Pruebas	13	18/11/2007	04/12/2007
Documentación	4	01/09/2007	05/09/2007
Presentación	8	15/12/2007	14/12/2007

Cuadro 1

CAPITULO 2

Análisis de la Solución

2.1. Propósito del Análisis

El propósito de análisis de nuestro proyecto es que permitirá a las grandes, medianas y pequeñas empresas solucionar una de las dificultades que se presenta hoy en día que es el ataque de intrusos, robo de información y exceso en el ancho de banda de la red; por esta razón se creará un sistema de detección de intruso el mismo que esta diseño para contrarrestar estos ataques generando reportes estadísticos, una bitácora de cada intruso que haya tratado de violar la seguridad, además cada persona que utilice el sistema tendrá una nombre de usuario y contraseña.

2.2. Alcance

Los alcances que tendrá nuestro proyecto darán a conocer la configuración, implementación, tecnología y metodología que se implementara en nuestro sistema; así como también se podrá monitorear la red generando cuadros estadísticos del tráfico de red, de los ataques que se produjeron los cuales serán de mucha ayuda para que los administradores puedan tomar decisiones.

Además se crearan niveles de seguridad para contrarrestar los ataques, generando usuarios y contraseñas para cada una de las personas que utilizaran el sistema.

Estos niveles de seguridad estarán conformados por el Administrador y Operador; en el cual el administrador tiene la potestad de crear usuarios y realizar modificaciones en cambio el operador no puede realizar modificaciones ni cambios pero si podrá verificar los reportes.

Cuando se detecte un ataque de un intruso el sistema activara una de la seguridades que tendrá, activando una alerta que la recibirá el administrador por medio de un correo electrónico.

También podremos decir que si recibimos un ataque externo el aplicativo bloqueará los Puertos de comunicación y capturara la dirección IP de la maquina atacante y la guardar en una bitácora de direcciones.

Cuando un usuario este ocupando mayor cantidad de ancho de banda del ya establecido el sistema lo que hará es bloquear al usuario, y enviara un mensaje al administrador indicando que maquina esta violando dicha seguridad.

2.3. Levantamiento de Información

2.3.1. Herramienta IDS

El IDS es un programa que se lo utiliza para detectar accesos desautorizados a un computador o a una red; este suele tener sensores virtuales con los que el núcleo del IDS puede obtener datos externos. Su funcionamiento se basa en el análisis pormenorizado del tráfico de red, el cual al entrar al analizador es comparado con firmas de ataques conocidos, o comportamientos sospechosos, como puede ser el escaneo de puertos, paquetes malformados, etc.

2.3.1.1. NIDS

El NIDS (Sistema de detección de intrusos en una Red) busca detectar anomalías que inicien un riesgo potencial, tales como ataques de denegación de servicio, escaneadores de puertos o intentos de entrar en un ordenador, analizando el tráfico en la red en tiempo real y para ello, analiza todos los paquetes, buscando en ellos patrones sospechosos.

También podemos decir que los NIDS no sólo vigilan el tráfico entrante, sino también el saliente o el tráfico local, ya que algunos ataques podrían ser iniciados desde el propio sistema protegido.

2.3.1.2. SNORT

Es un sniffe*r* capaz de actuar como sistema de detección de intrusos en redes de tráfico moderado; con una facilidad de configuración, su adaptabilidad, sus requerimientos mínimos, y sobre todo su precio lo convierten en una óptima elección en multitud de entornos, frente a otros sistemas.

2.3.1.3. POSTGRESQL

Es un motor de base de datos relacional libre, soportan un modelo de datos que consisten en una colección de relaciones con nombre, que contienen atributos de un tipo específico. Los Postgres ofrecen una potencia adicional sustancial al incorporar los siguientes cuatro conceptos adicionales básicos: Clases, herencia, tipos y unciones.

2.3.1.4. TOMCAT 5.0

Es un servidor Web con soporte de servlets y JSPs (llamado también Jakarta Tomcat o Apache Tomcat), funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de Java Server Pages (JSP) de Sun Microsystems.

2.3.1.5. J2EE (JAVA 2 ENTERPRISE EDITION)

Es una plataforma de programación que sirve para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de n niveles, distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

2.3.1.6. JSP (JAVA SERVER PAGE)

Es una tecnología que se a utilizada para generar paginas HTML de forma dinámica a petición del usuario, el motor del JSP es un servlet que estará en un servidor que recibirá peticiones del usuario, estas peticiones llegarán del cliente, se analizarán y el servidor dará la respuesta adecuada.

2.3.1.7. UML

Lenguaje Unificado de Modelado, es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software, ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables.

2.3.1.8. RUP

Proceso Unificado de Racionales, es un proceso de desarrollo de software, que utiliza una metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos; en el Proceso Unificado los casos de uso se utilizan para capturar los requisitos funcionales y para definir los contenidos de las iteraciones.

2.4. Análisis de Requerimientos

El análisis de requerimientos es la tarea que plantea la asignación de software a nivel de sistema y el diseño de programas.

El sistema para administración y monitoreo de la herramienta IDS, que encapsula el dominio para la seguridad de la red, se han detectado 3 requerimientos que se detallan a continuación:

2.4.1. Proceso de Seguridad

A su vez contiene 2 subprocesos, cuando el usuario es un administrador y cuando es un operador.

2.4.2. Proceso Administrativo

Conlleva la configuración del IDS.

2.4.3 Proceso de Reportes del Monitoreo

Reportes de la información que se tiene del testeo de la red, dicha información es capturada por el IDS y se almacena en la Base de Datos SNORT.

2.5. Representación de la Arquitectura

Para la representación de la arquitectura de nuestro sistema utilizaremos algunos componentes del Lenguaje Unificado de Modelado UML, ya que ha sido de gran ayuda para el éxito de anteriores proyectos de desarrollo.

Por tal motivo utilizamos vistas, que estas nos muestran de mejor manera la arquitectura de un sistema.

2.5.1. Vista de Casos de Uso

La vista caso de uso es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización software.

Este artificio le muestra al usuario el funcionamiento del sistema, aquí se muestran los procesos operativos y los casos de uso más importantes que se generan de estos.

2.5.1.1. Procesos Operativos

Estos procesos van a ejecutarse para el debido control del tráfico de la red y son:

- ✓ ***Administración del IDS***.
  - Configuración del Sistema
  - Herramienta IDS
  - Ejecución del IDS

- ✓ ***Monitoreo de la información***
  - Control y Monitoreo de la información que viaja por la red
  - Generación de Reportes y Alarmas

2.5.1.2. Actores

Son aquellos usuarios que van a manipular el sistema, en este caso existen 2 tipos de usuarios previamente definidos:

- ✓ **Administrador**

  Se entiende como el súper usuario con privilegios y sin restricciones, que es el único que va a configurar las reglas del sistema.

- ✓ **Operador**

  Es el usuario común con privilegios limitados.

2.5.1.3. Casos de Uso

Es una estructura que nos ayuda a trabajar con los usuarios para determinar la forma en que se usará el sistema. Con una colección de casos de uso se puede hacer el bosquejo de un sistema en términos de lo que los usuarios intenten hacer con él

El caso de uso es mucho mejor cuando se lo visualiza por medio del UML, esta visualización le permitirá mostrar los casos de uso a los usuarios para que ellos le puedan dar mayor información.

2.5.1.3.1 Diagrama de Casos de Uso

Muestra las principales opciones que contendrán un sistema, las acciones y actores que influyen dentro de cada opción.

2.5.1.3.2. Diagrama de Casos de Uso de nuestro sistema

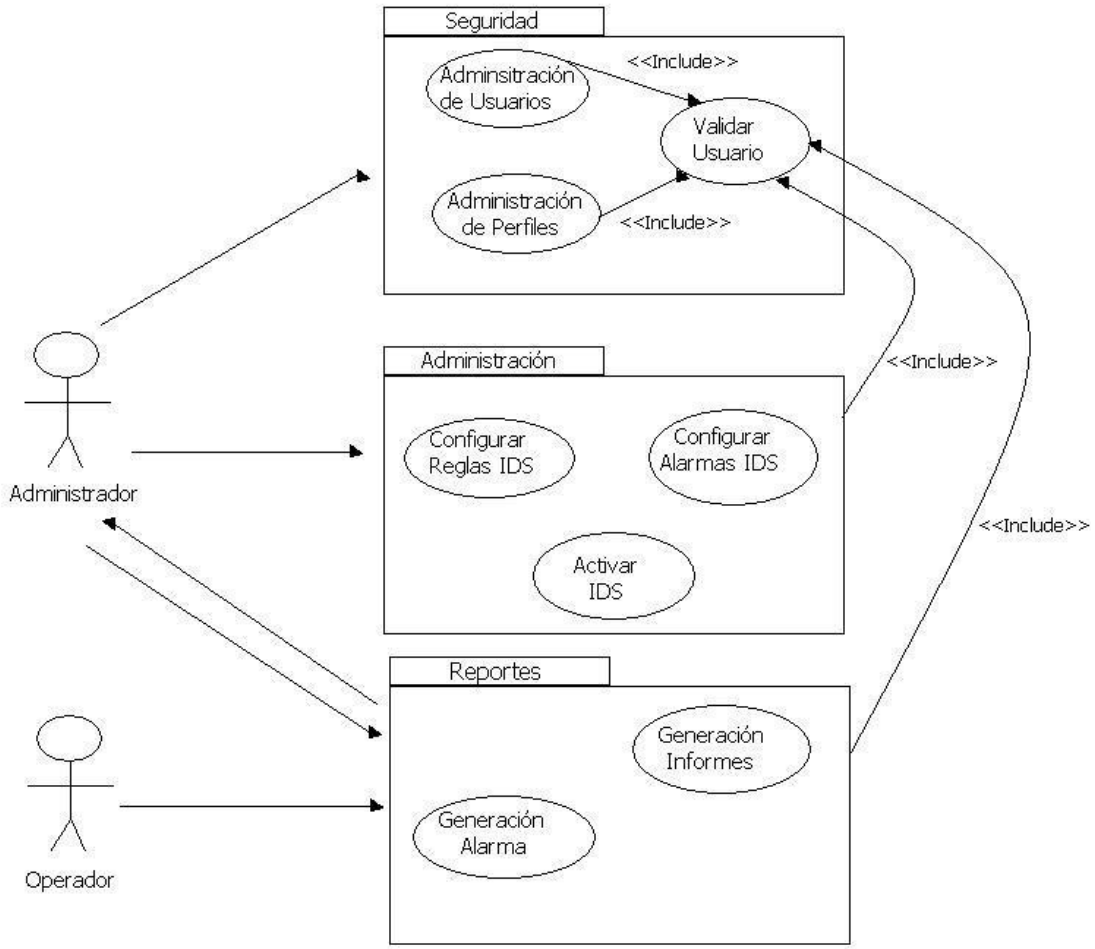


Figura 4

2.5.1.3.2.1. Descripción de los Casos de Uso del sistema

PAQUETE: USUARIO

Caso de Uso 1	
Nombre	Administración de usuarios
Actores	Administrador
Actividad	crear usuarios, cambiar clave, eliminación de usuarios y m de datos de registro
Descripción	este caso de uso comienza cuando le es solicitado al administrador la creación,cambio de clave, eliminación, modificación de los datos de registro necesarios para la operación o administración de la aplicación
Curso Típico de Eventos	administrador recibe petición para gestionar los usuarios
	administrador solicita ingreso al sistema
	sistema pide autentificación para ingreso al sistema
	el administrador selecciona la operación a realizar
	el administrador ejecuta la operación
	el sistema registra la operación realizada
Extensiones	***Crea nuevo usuario***
	el administrador ingresa datos del nuevo usuario
	Cambio de Clave
	el administrador puede cambiar su clave
	el operadorar puede cambiar clave
	Modifica datos de registro
	el administrador consulta el usuario a modificar
	el sistema muestra los datos consultados
	el administrador modifica los datos
	Eliminar usuario
	el administrador consulta usuarios a eliminar
	el sistema muestra datos consultados

Cuadro 2

Descripción.-Este caso de uso se da cuando se requiere la creación, cambio de clave, modificación, eliminación de los operadores de la aplicación. El único encargado de realizar estas operaciones es el administrador.

Caso de Uso 2

Paquete: PERFILES

Nombre: Administración de Perfiles	
Actores	Operadores
Actividad	ver todos los perfiles que se crearon con sus respectivos privilegios
Descripción	el caso de uso comienza cuando cuando el operador al administrador la creación, eliminación, modificación desea consultar de los perfiles de usuario necesarios para la operación o administración de la aplicación
Curso Típico de Eventos	administrador recibe orden a realizar
	administrador solicita ingreso al sistema
	sistema ingreso de datos al usuario
	administrador determina operación a realizar
	administrador ejecuta la operación
	sistema guarda la operación realizada
Extensiones	***Ver Perfiles***
	el operador consulta que perfiles hay creados
	el sistema muestra los posibles resultados
	el operador selecciona el resultado deseado

Cuadro 3

Descripción.- Este caso de uso se da cuando se crea, elimina, modifica o consulta perfiles por usuario, se crean perfiles con la finalidad de asociar a un grupo de usuarios determinados a que tengan ciertos privilegios de accesos a la aplicación. El único encargado de realizar estas operaciones es el administrador.

Caso de Uso 3

Nombre: Validar Usuario	
Actores	Administrador, operador
Actividad	Autentificación de usuarios que requieren el ingreso al sistema.
Descripción	Cuando un usuario quiere ingresar al sistema para realizar alguna transacción, el sistema debe verificar que el usuario exista.
Curso Típico de Eventos	sistema muestra pantalla de usuario para el ingreso de
	el usuario ingresa datos
	el sistema verifica los datos
	el sistema da paso al usuario
	el sistema levanta la interfaz de transacciones
	el usuario que ingreso.
Extensiones	***Datos ingresados son erróneos***
	el sistema pide el ingreso de datos válidos
	el sistema determina un número máximo de intentos
	el sistema bloquea la interfaz .

Cuadro 4

<u>Descripción.-</u> Este caso de uso se da cuando un usuario está solicitando acceso a la aplicación, en este caso el sistema implementará la autentificación del usuario para permitir el ingreso, está opción estará limitada por un intento de conexión máxima de 3 oportunidades, caso contrario la aplicación se cerrará.

PAQUETE: Administración y Monitoreo

Caso de Uso 4

Nombre	Configuración de Reglas
Actores	Administrador
Actividad	Sirve para que el administrador realice cambios en las reglas del IDS
Descripción	Este proceso es de mucha ayuda para el administrador ya que le permitirá reconfigurar nuestro detector de intrusos.
Curso Típico de Eventos	administrador necesita redefinir reglas del IDS
	administrador solicita ingreso al sistema
	sistema pide datos para autentificación
	sistema carga la pantalla para la configuración del Snort
	administrador ingresa y redefine reglas
	sistema ejecuta y analiza la sintaxis del registro
	sistema guarda registro
	sistema anuncia que debe reiniciarse el snort
Extensiones	Sistema no encuentra el archivo de configuración Sistema realiza la búsqueda del archivo Sistema encontró errores en la sintaxis del archivo redefinido Sistema anuncia errores en el archivo y no lo guarda

Cuadro 5

Descripción.- Este caso de uso se da cuando la aplicación requiere reconfigurar el detector de intrusos.

Caso de Uso 5

Nombre	Configuración de Alarmas
Actores	Administrador
Actividad	Realiza una definición de parámetros y un registro de patrones en la Base de Datos
Descripción	Este proceso es de mucha ayuda para el administrador ya que le permitirá reconfigurar la generación de alarmas nuestro detector de intrusos.
Curso Típico de Eventos	administrador necesita redefinir parámetros para envío de alarmas
	administrador solicita ingreso al sistema
	sistema pide datos para autentificación
	sistema carga la pantalla para redefinir reglas en IDS
	administrador define parámetros para generar alarmas
	sistema verifica la validez de parámetros ingresados
	sistema guarda registro
Extensiones	Sistema encontró errores en la sintaxis del archivo redefinido
	Sistema anuncia errores en el archivo y no lo guarda

Cuadro 6

Descripción.- Este caso de uso se da cuando la aplicación necesita reconfigurar la generación de alarmas del detector de intrusos.

Caso de Uso 6

Nombre	Activar IDS
Actores	Administrador
Actividad	levanta, para o reinicia el IDS
Descripción	El administrador la puede levantar, parar y reiniciar el IDS
Curso Típico de Eventos	administrador requiere gestionar la ejecución del IDS
	administrador solicita ingreso al sistema
	sistema pide datos de usuario para ingreso al sistema
	sistema carga pantalla para gestión del IDS
	administrador selecciona la operación a realizar
	administrador selecciona acción a ejecutar
	sistema muestra el estado actual del IDS
Extensiones	***habilitar el ids***
	Levanta el IDS
	parar el ids
	realiza la parada del IDS
	reiniciar el ids
	realiza el reset del IDS

Cuadro 7

Descripción.- Este caso de uso se da cuando la aplicación necesita que se levante, reinicie o para el IDS.

Paquete: Reportes

Caso de Uso 7

Nombre	Generación Alarmas
Actores	Administrador, operador
Actividad	envío de correo electrónico, recolección y procesamiento de datos
Descripción	Cuando existe algún ataque que se ha registrado se enviará un e-mail a los responsables de la red.
Curso Típico de Eventos	IDS recolecta información de paquetes circulantes de la red
	IDS procesa la información y la registra en la base de datos
	sistema encuentra infracción y la carga en la base de datos
	sistema procesa la información y genera los correos
	Administrador u operador recibe el e-mail enviado

Cuadro 8

***Descripción*.-** Este caso de uso se da cuando la aplicación ha detectado un ataque con la cual se enviará un e-mail a los responsables de la red.

Caso de Uso 8

Nombre	Generación de Informes
Actores	Administrador, operador
Actividad	Configuración de Reportes
Descripción	Realiza la configuración de Reportes para que se entreguen a los responsables de la red
Curso Típico de Eventos	Adm./operador desea generar un reporte
	Adm./operador requiere ingreso al sistema
	sistema solicita ingreso de datos de usuario
	sistema carga pantalla para configuración de informes
	Adm./operador ejecuta la generación del reporte
	sistema procesa configuración
	sistema recopila datos
	sistema informes
Extensiones	configuración de correos

Cuadro 9

Descripción.- Este caso de uso se da cuando se necesita entregar los informes a los administradores de la red.

2.5.2. Vista QoS

La calidad de servicio (QoS) basada en directivas de Microsoft, permite configurar perfiles de QoS que especifican el modo de marcar y limitar el tráfico red saliente.

Nuestra herramienta es transportable ya que es desarrollado en java que es un lenguaje multiplataforma y nos demuestra una gran usabilidad.

Es confiable ya que la arquitectura que posee nos da cuenta de un sistema robusto y adaptable a cualquier ambiente y posee un alto nivel de disponibilidad.

2.5.3. Vista Lógica

Muestra los componentes principales de diseño y sus relaciones de forma independiente de los detalles técnicos y de cómo la funcionalidad será implementada en la plataforma de ejecución, nos ayuda a representar en forma general de la estructura lógica del sistema y sus componentes.

2.5.3.1. Desarrollo mediante la división en Capas

Utilizaremos una estructura de 3 capas definidas:

- ✓ **Capa de Presentación**: GUI Interfaz gráfica, es la que se mostrará a los clientes de la red.

- ✓ **Capa Lógica**: es el motor de la aplicación, aquí se definen las reglas, conexiones, acciones y procesamiento de datos.

- ✓ **<u>Capa de Datos</u>**: Aquí se encuentran las bases de datos que contiene toda la información a usar.

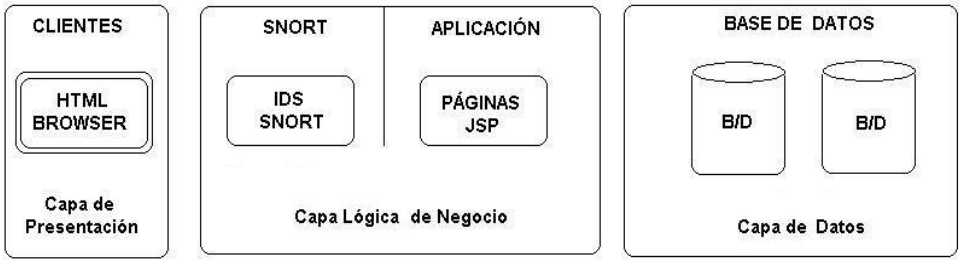


Figura 5

2.5.3.2. Arquitectura del Sistema

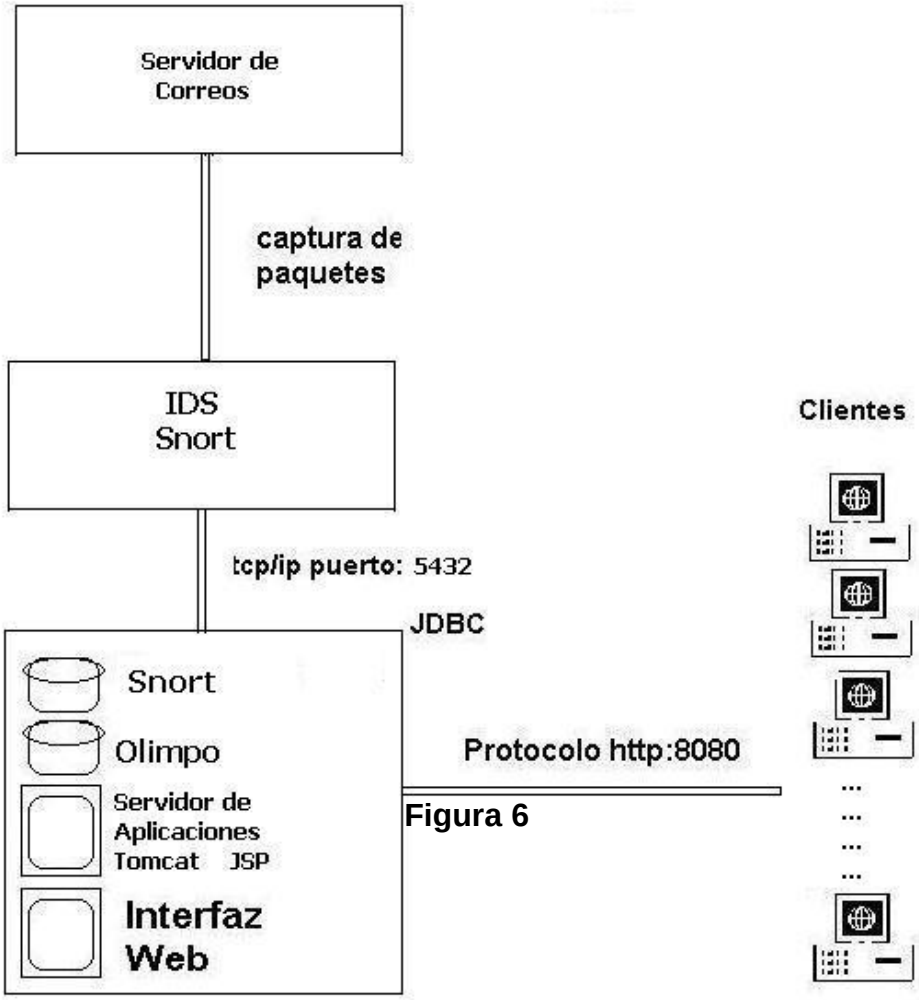


Figura 6

Los procesos de nuestra herramienta van a estar distribuidos en componentes, llamando componente a uno a más servicios específicos que trabajan en conjunto.

El primer componente contiene al servidor de correos el cual nos ayudará para avisarles a los responsables de la red (Administradores) sobre ataques suscitados.

El segundo componente contiene a la herramienta SNORT, el cual realizará el monitoreo de la red.

El tercer componente alberga a las bases de datos Snort y Olimpo (Accesos de usuarios), también al servidor de aplicaciones y a la aplicación como tal. Aquí se manejaran las solicitudes que el usuario realice mediante browser.

2.5.3.3. Diagramas de Bloques del Sistema

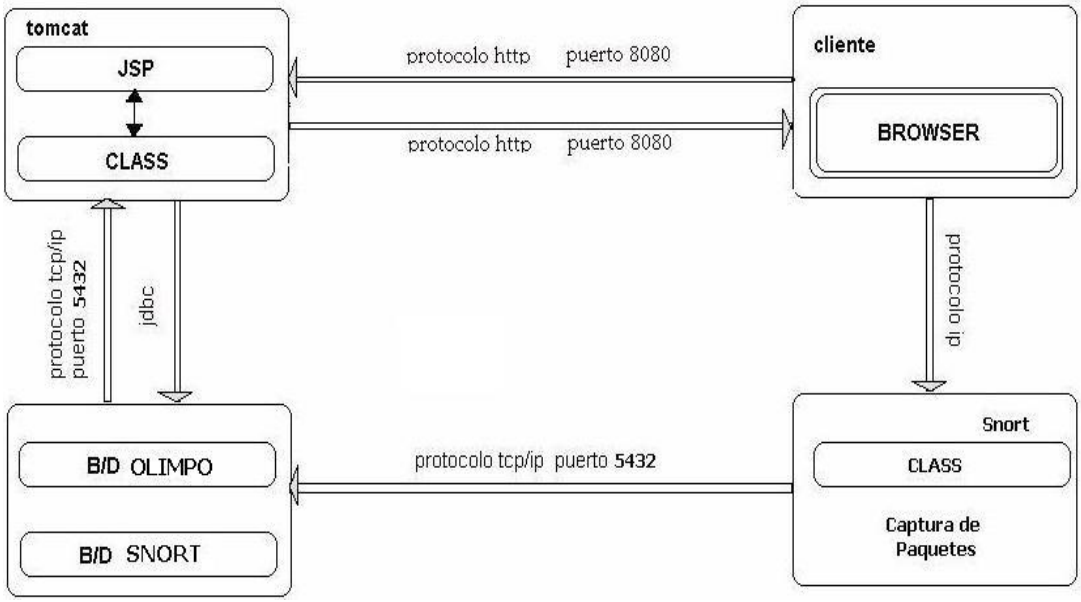


Figura 7

2.5.3.3.1. Breve descripción del diagrama de bloques.

- ✓ **Tomcat 5.0.-** Herramienta que utilizaremos para poder la pagina Web este se comunica por el puerto 8080.

- ✓ **JSP.-** son páginas dinámicas las cuales van a interactuar con nuestras clases java y demás componentes.

- ✓ **CLASS**.- Es la extensión compilada de archivos .java, la cual nos servirá de fuente para la manipulación de las clases elaboradas.

2.5.3.3.2. Base de Datos

- ✓ **Base de Datos Snort**.- En esta base se van a almacenar las incidencias generadas por el tráfico de la red

- ✓ **Base de Datos OLIMPO**.- Esta base contendrá la información de los usuarios (Administrador y Operadores) que tienen acceso a nuestra Interfaz Web, además se agregarán cuando haya nuevos usuarios.

2.5.3.3.3. Cliente

- ✓ **Clientes**.- Son las máquinas que van a estar dentro de nuestra red.

- ✓ **Browser**.- Es la forma como los clientes van a poder acceder a nuestro aplicativo Web utilizando el puerto 8080.

- ✓ **Snort.-** básicamente es un sniffer que nos permite estar testeando la red y será el encargado de monitoreo de la red y su protección.

- ✓ **Captura de Paquetes**.- Ira revisado cada paquete que circula por la red fin de encontrar paquetes maliciosos.

2.5.4. Vista de Datos

Es un diseñador de consultas para crear conjuntos de datos que recuperan meta datos de un origen de datos; éstos se utilizan para definir el diseño del informe. Se da a conocer los por menores del modelo de datos utilizado y su distribución.

2.5.4.1. Distribución de Datos

La Base de Datos Snort es la encargada de guardar toda la información correspondiente al testeo de la red de paquetes maliciosos.

La base de datos *OLIMPO* es la encargada de almacenar la información del usuario que puede tener acceso a la aplicación para administrar o monitorear la red.

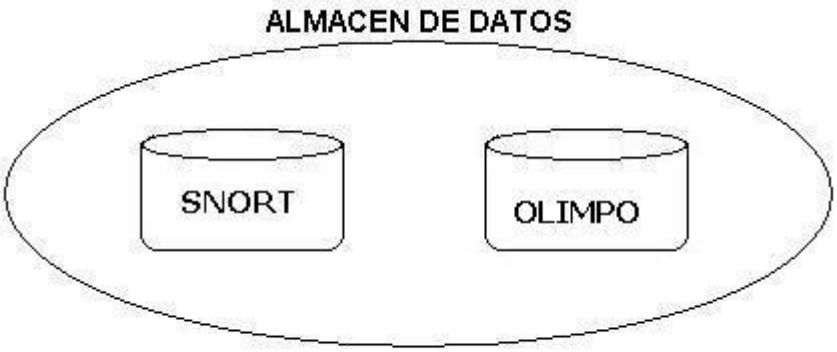


Figura 8

2.5.4.2. Diagrama de Objetos Relación Snort

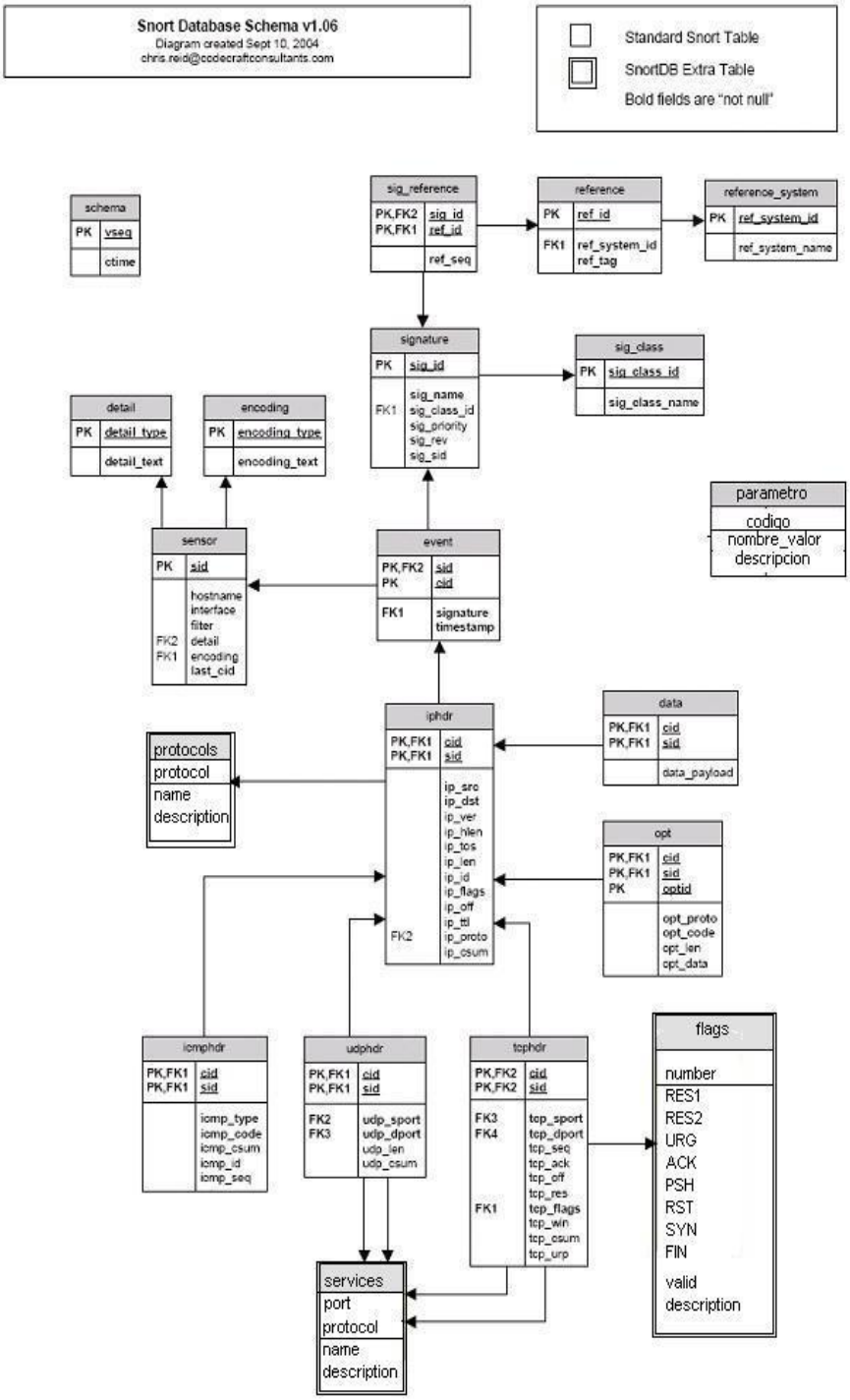


Figura 9

2.5.4.3. Diagrama de Objetos relación OLIMPO

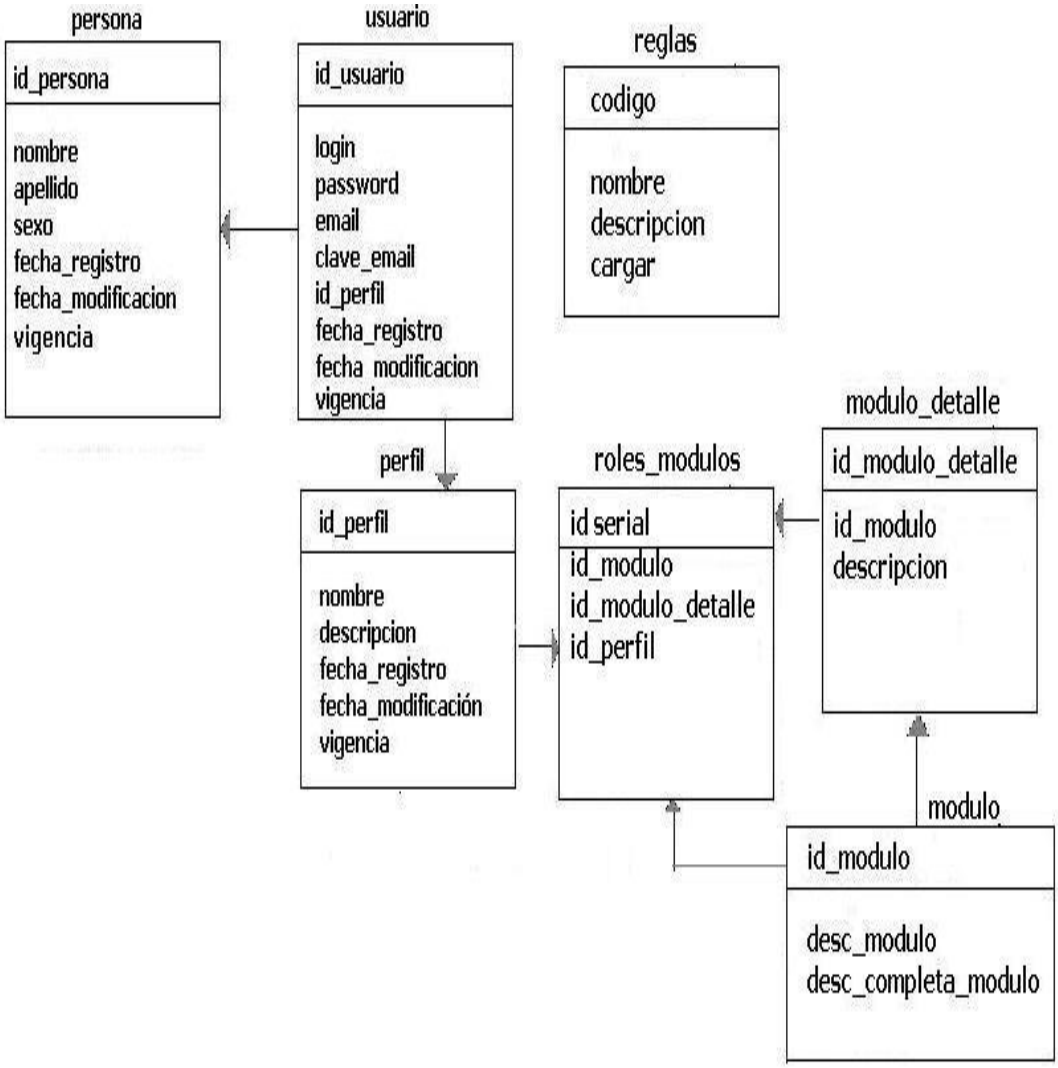


Figura 10.

2.5.4.4. Diccionario de Datos

Representa explícitamente las relaciones entre los objetos da datos y las restricciones entre los objetos de datos y las restricciones de los elementos de una estructura.

Nombre	Nombre de la Variable
Usado en la Base	Se describe la base en la que se la esta usando.
Usado en la Tabla	Se describe el nombre de la tabla
Tipo de Dato	Se detalla el tipo de la variable
Descripción	Descripción de lo comprende la variable

Cuadro 1.10

CAPITULO 3

DISEÑO DEL SISTEMA

3. Diseño

3.1. Diseño de la Solución.

El diseño tiene un alto nivel estratégico y decisión para resolver los problemas. Los grandes problemas se deben ver desde el punto del análisis y diseño, este sistema se divide en subsistemas, a su vez se divide en varios subsistemas de manera que puedan ser manejados y cada componente pueda ser comprensible

3.2. Diseño de Subsistemas

Se derivan en si de la explicación de los Casos de Uso antes mencionados y entender los diferentes estados de cómo funciona el sistema **GUARDIANSNORT**.

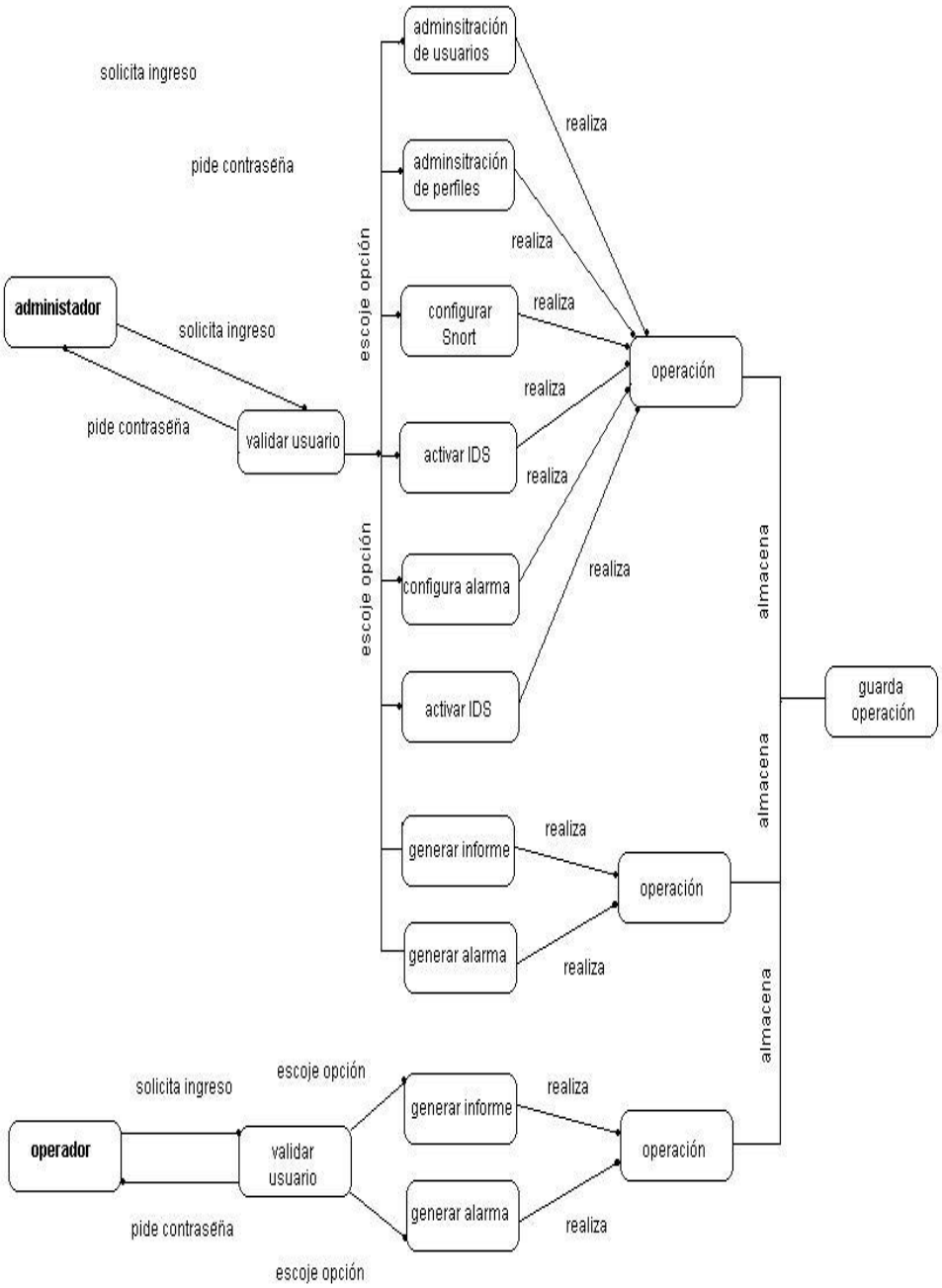

solicita ingreso
pide contraseña
administador
solicita ingreso
pide contraseña
validar usuario
escoje opción
escoje opción
adminsitración de usuarios
adminsitración de perfiles
configurar Snort
activar IDS
configura alarma
activar IDS
generar informe
generar alarma
realiza
realiza
realiza
realiza
realiza
realiza
realiza
realiza
operación
operación
almacena
almacena
almacena
guarda operación
solicita ingreso
escoje opción
operador
validar usuario
pide contraseña
escoje opción
generar informe
generar alarma
realiza
realiza
operación

Figura 11

3.2.1 Pagina de Inicio

Aquí se mostrara la página de inicio del snort la cual esta conformada por:

Home, Empresa, Acerca de, Sesión y Contacto.

Figura 12

3.2.1.1 Iniciar Sesión

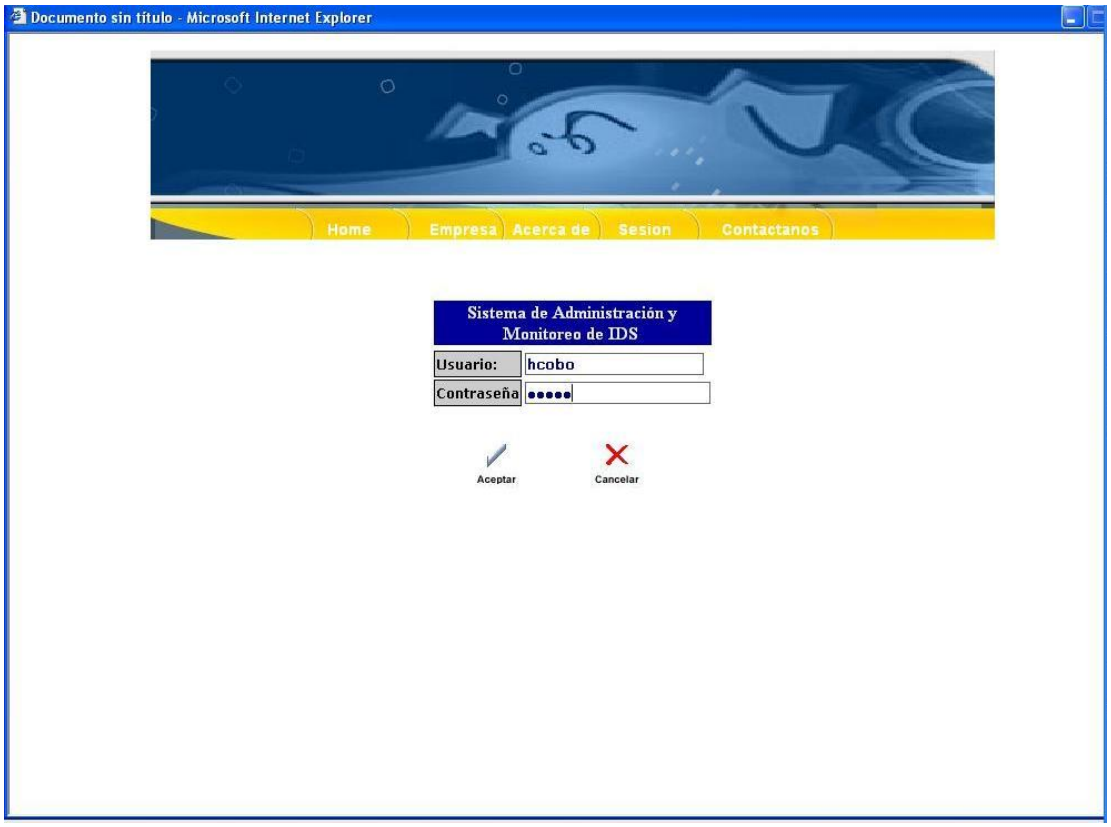


Figura 13

3.2.2. Menú Principal

Contiene todos los procesos de forma general que contiene el aplicativo va ha ejecutar, como: el Menú de Perfiles, Menú de Usuarios, Menú de Administración y Monitoreo y el Menú de Reportes.

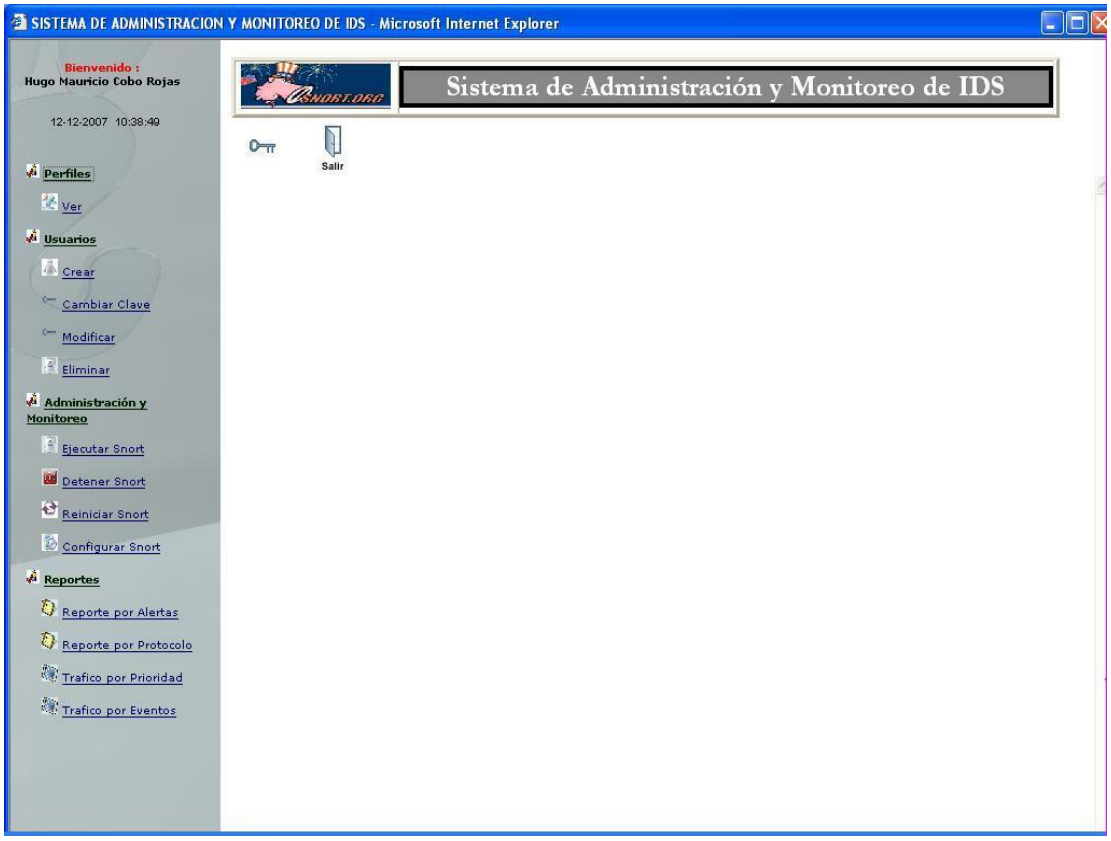


Figura 14

3.2.2.1. Menú Perfiles

Mostrará en pantalla una lista con los perfiles creados, estos perfiles tienen asignado tipo de transacciones que representan privilegios en el usuario para la manipulación del sistema.

3.2.2.1.1. Ver

Mostrara un listado de todos los perfiles que hayan utilizado el sistema así como también mostrara un código, nombres, apellidos, usuario, e-mail y teléfono.

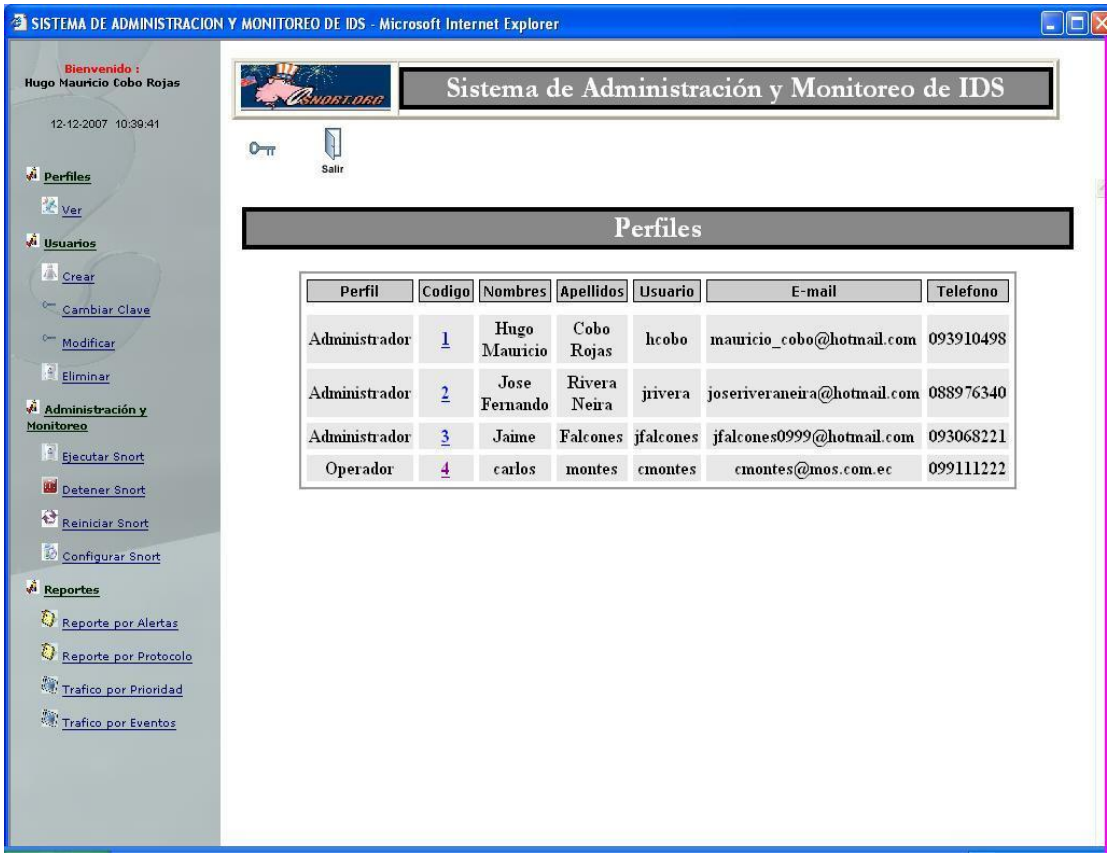


Figura 15

3.2.2.2. Menú de Usuarios

Este menú contiene ciertas opciones que permitirán a los Administradores y usuarios del sistema: crear, cambiar clave, modificar y eliminar usuarios.

3.2.2.2.1. Crear Nuevo Usuario

Creará un nuevo usuario y contraseña correspondiente para que pueda manipular el sistema, también se le ha de configurar un perfil.

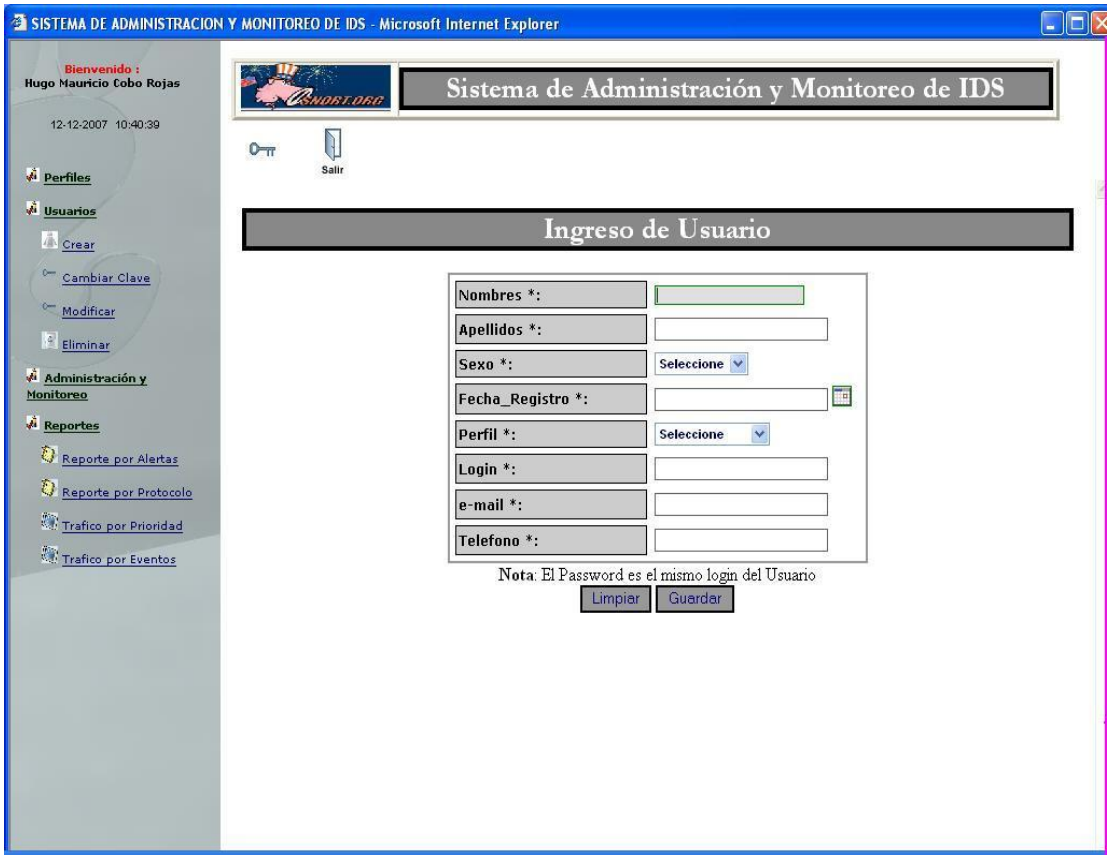


Figura 16

3.2.2.2.2. Cambiar Clave

Esta opción le sirve al administrador como al operador para que pueda cambiar su contraseña de acceso.

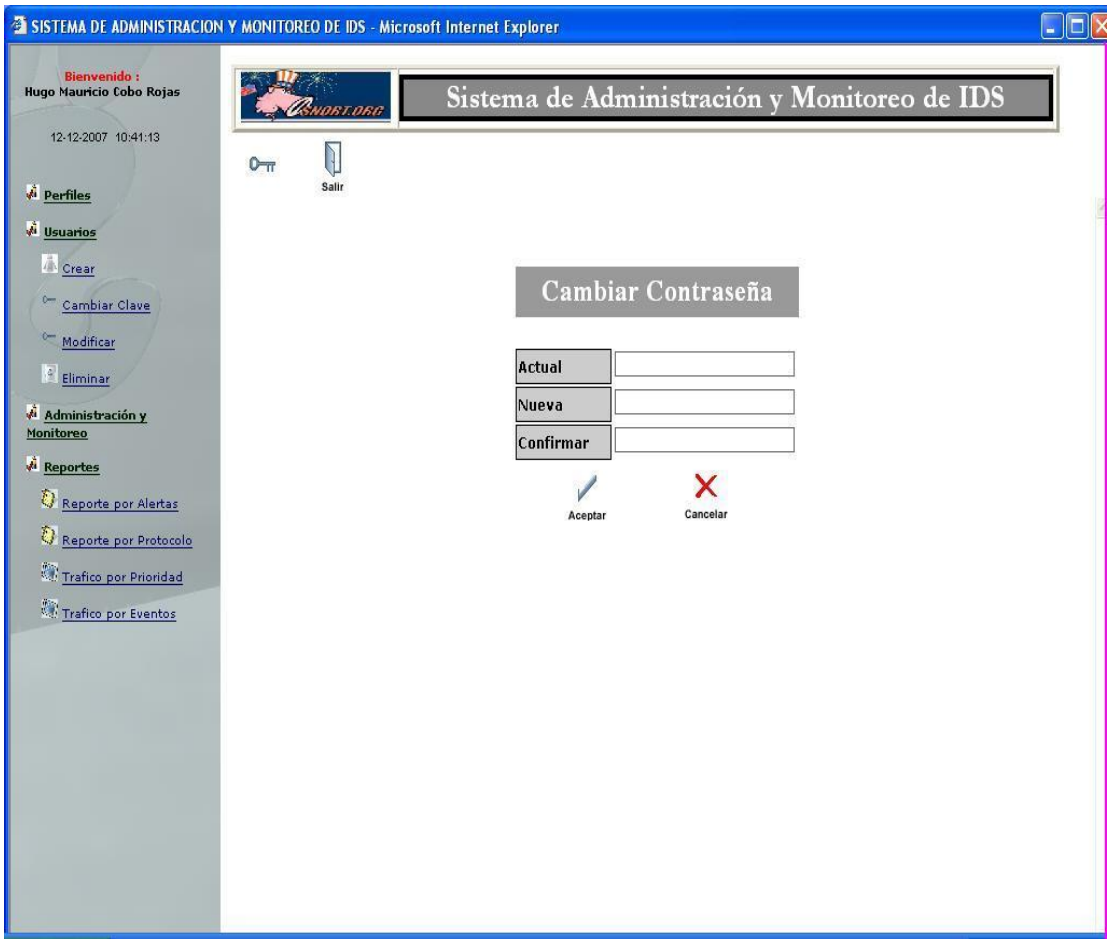


Figura 17

3.2.2.2.3. Modificar Usuario

Modificar los datos del usuario.

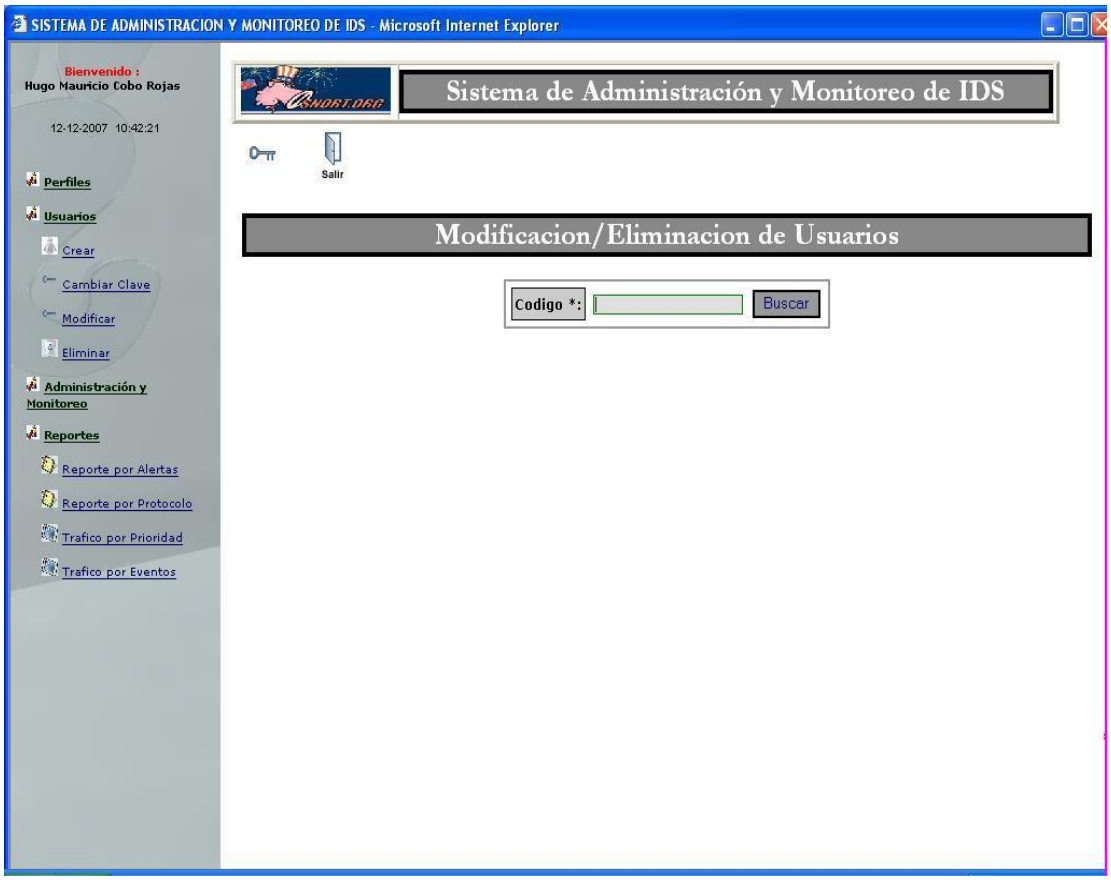


Figura 18

3.2.2.2.4. Eliminar Usuario

Eliminar usuarios que se escogieron.

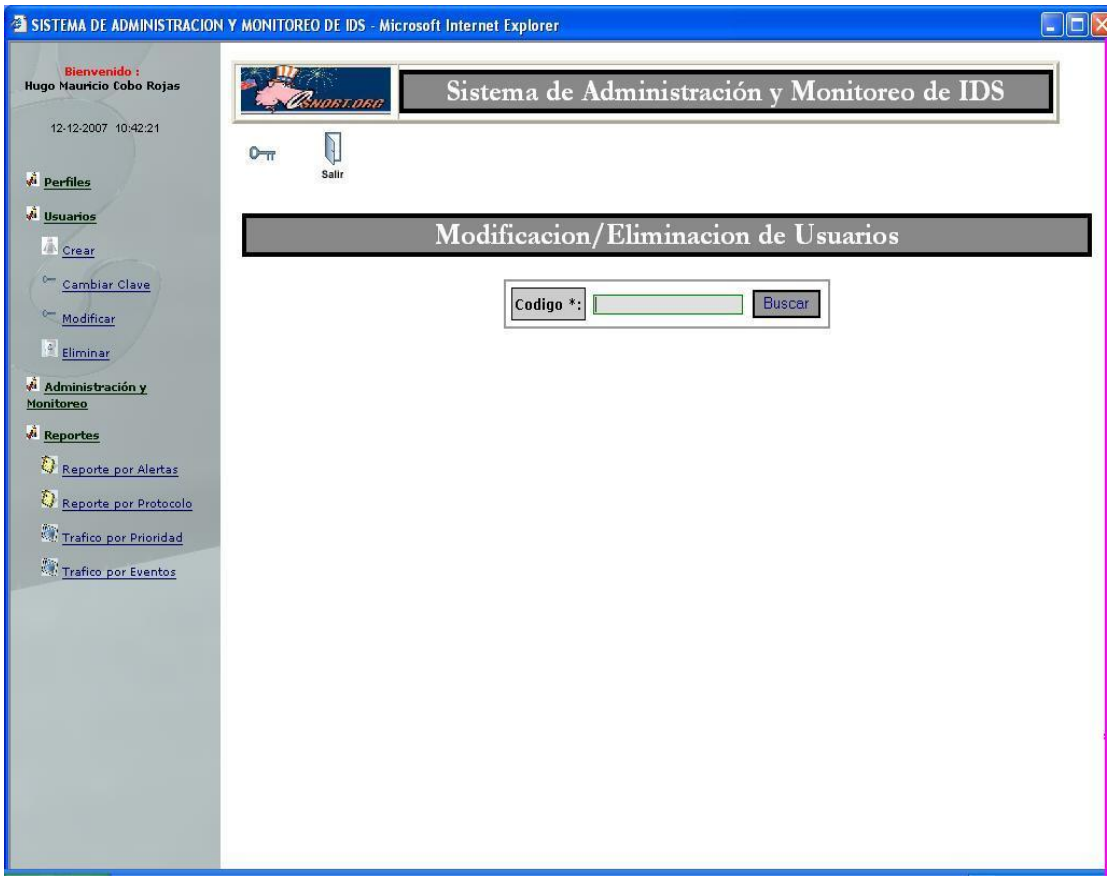


Figura 19

3.2.2.3. Menú de Administración y Monitoreo

Se podrá apreciar los estados del Snort como son: Ejecutar, Reiniciar, Detener y Configurar.

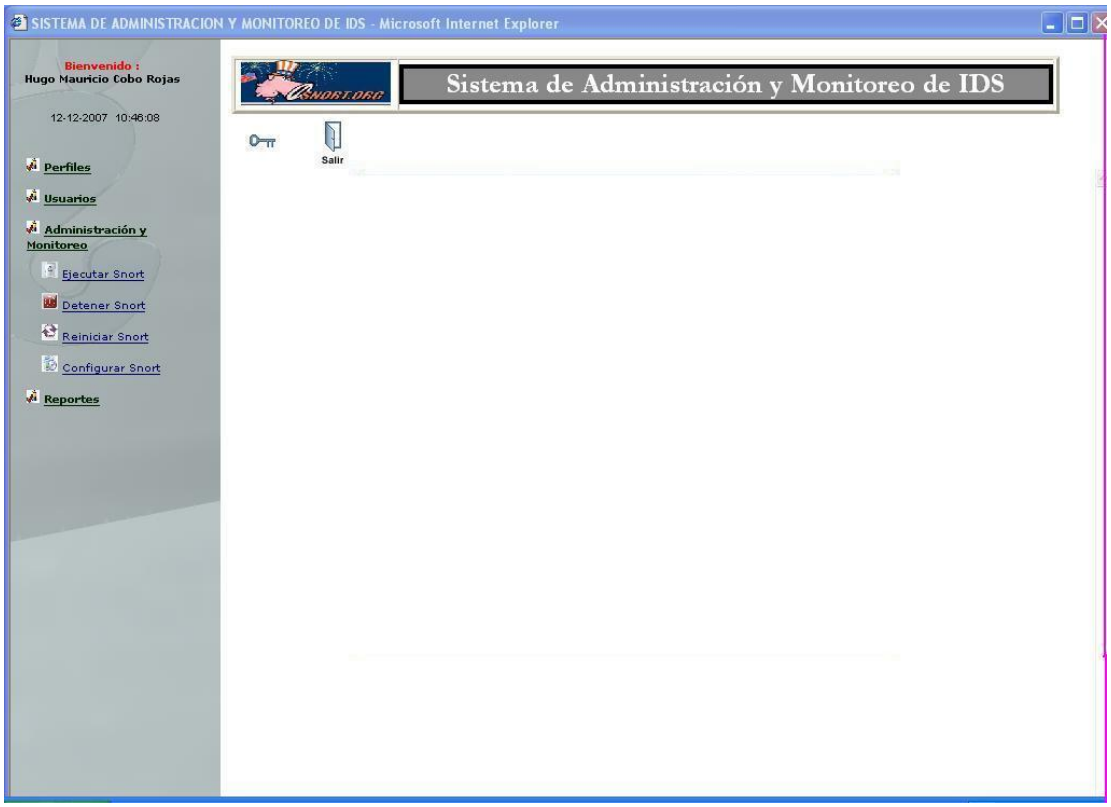


Figura 20

3.2.2.3.1. Ejecutar Snort

Iniciará el proceso de ejecución del IDS.

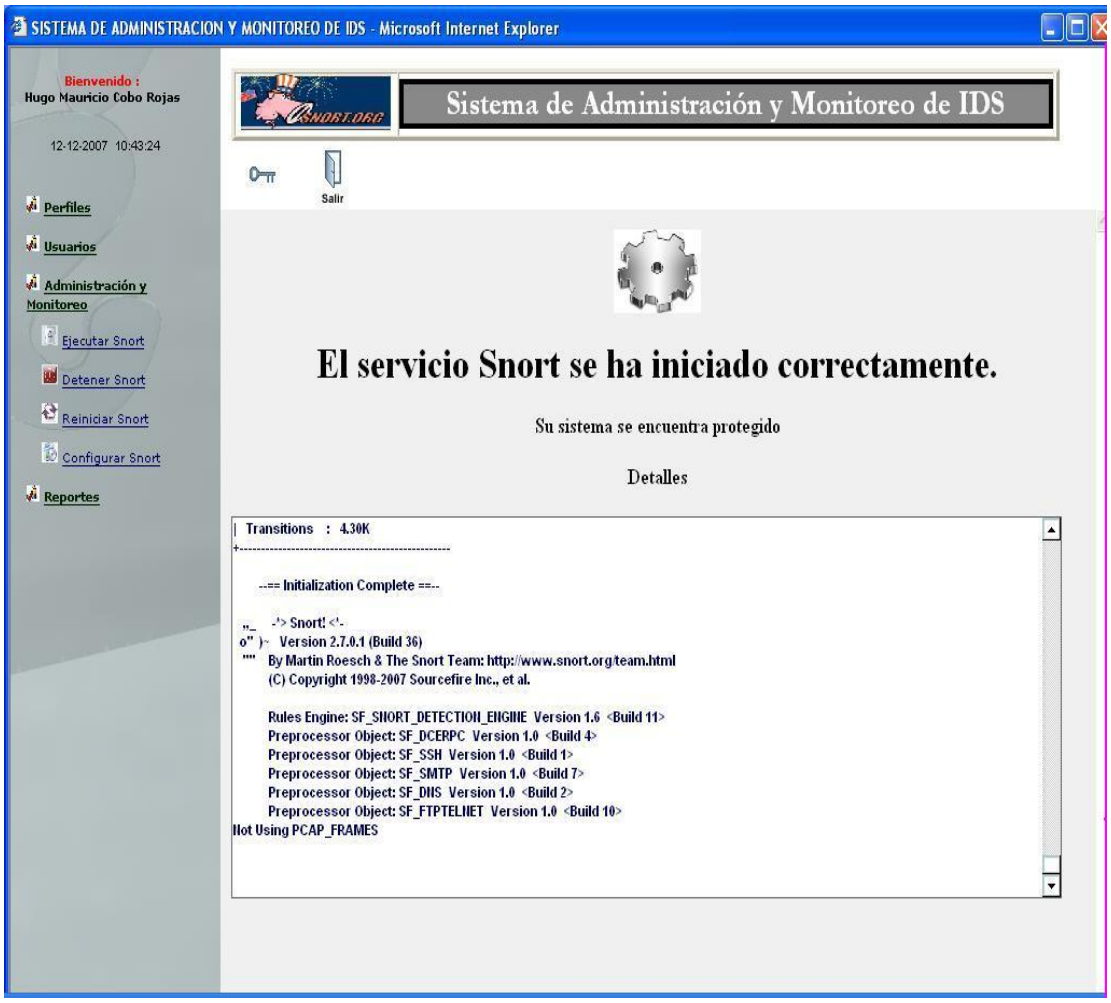


Figura 21

3.2.2.3.2. Detener Snort

Detendrá la ejecución del IDS.

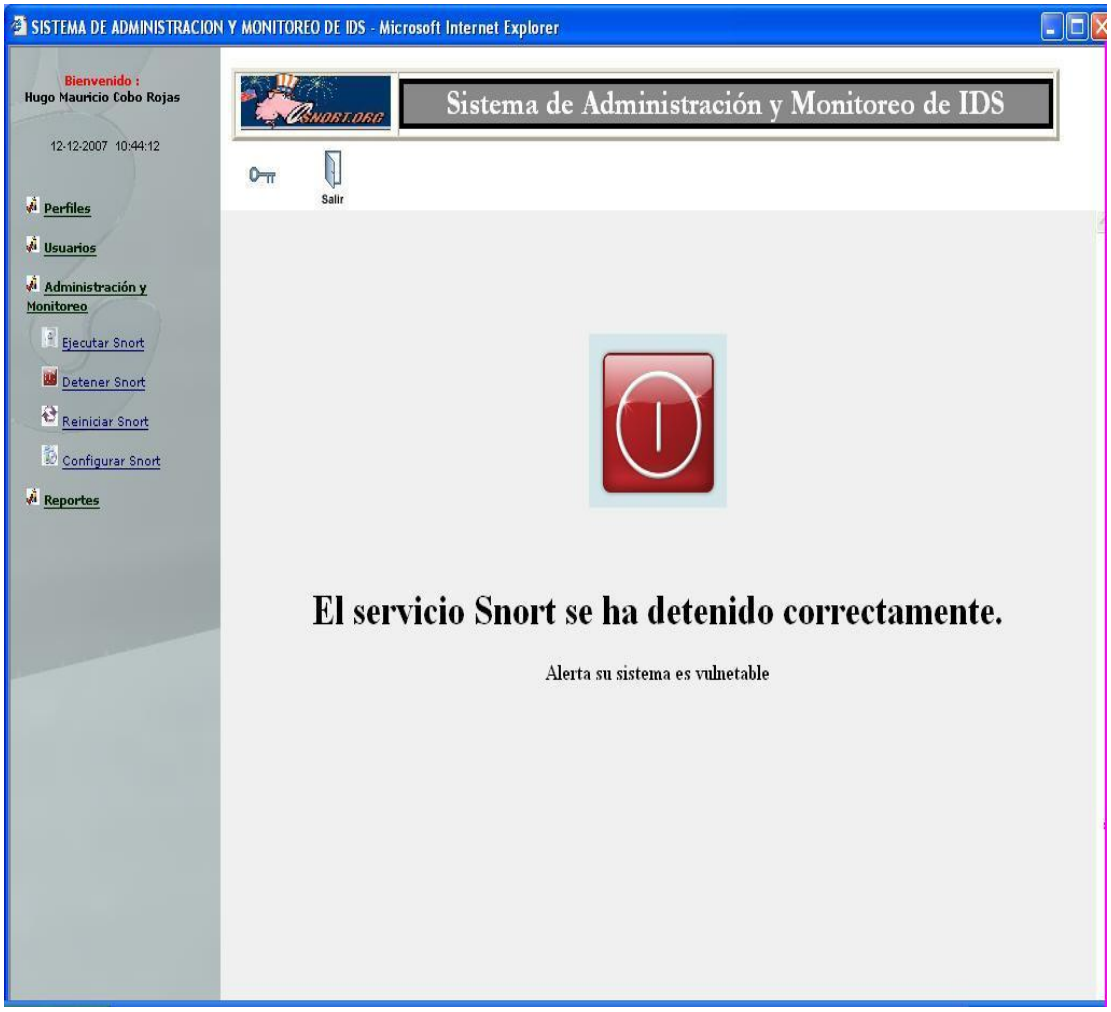


Figura 22

3.2.2.3.3. Reiniciar Snort

Detendrá y volverá a activar el IDS.

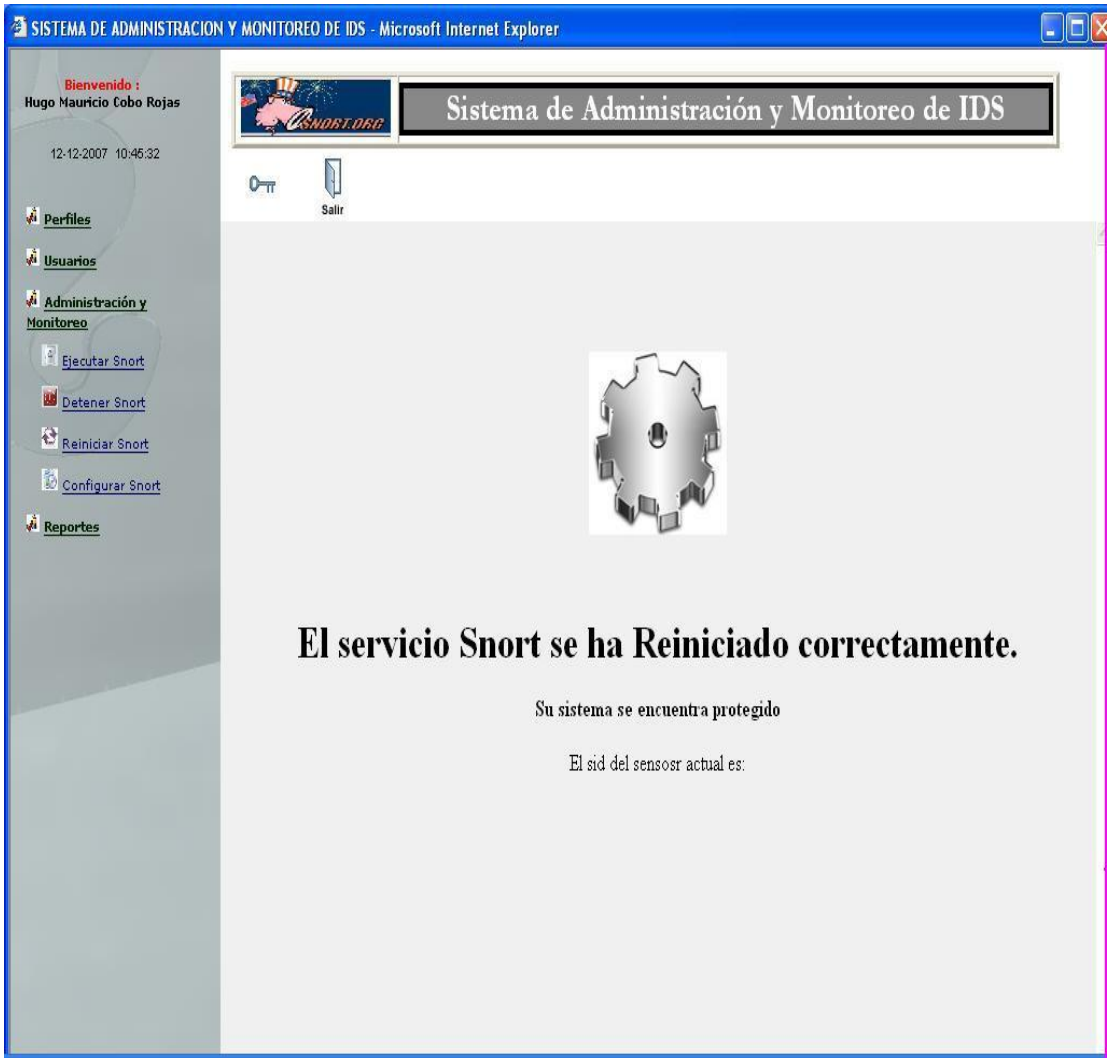


Figura 23

3.2.2.3.4. Configuración de Snort

Aquí trata de la configuración del archivo snort.conf.

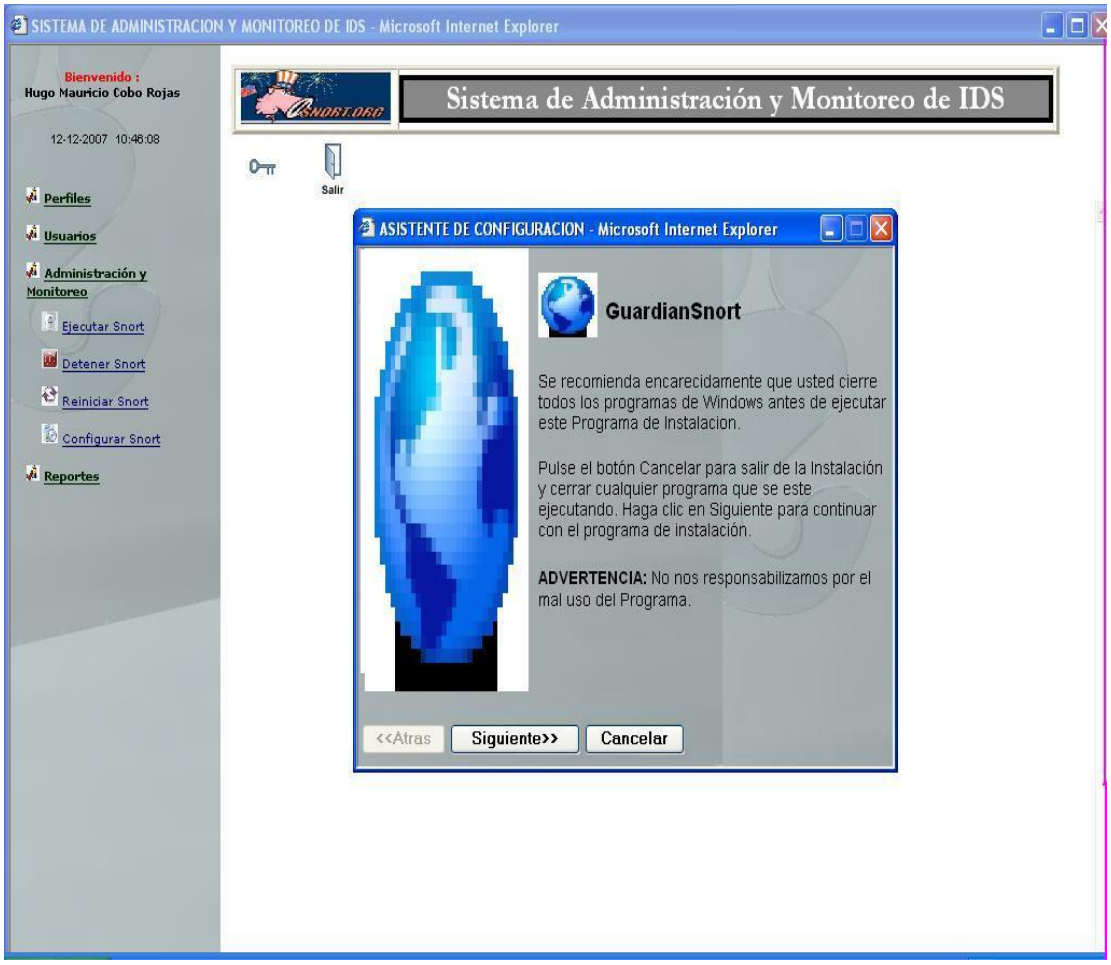


Figura 24

3.2.2.4. Menú de Reportes

Esta opción mostrara una lista con los diferentes tipos de reportes que se va a generar de forma detallada en nuestro sistema.

3.2.2.4.1. Reportes por Alerta

Mostrara en pantalla un grafico estadístico generado del tráfico por alertas, solo se mostraran las alertas generadas por el snort.

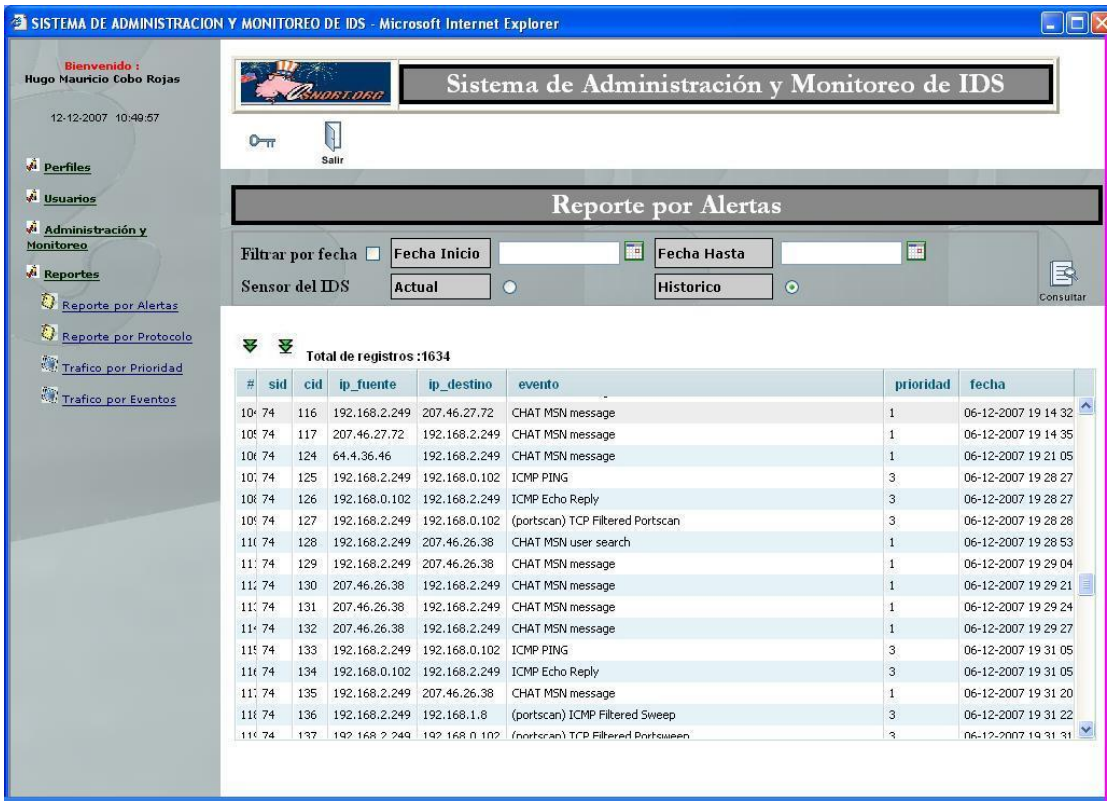
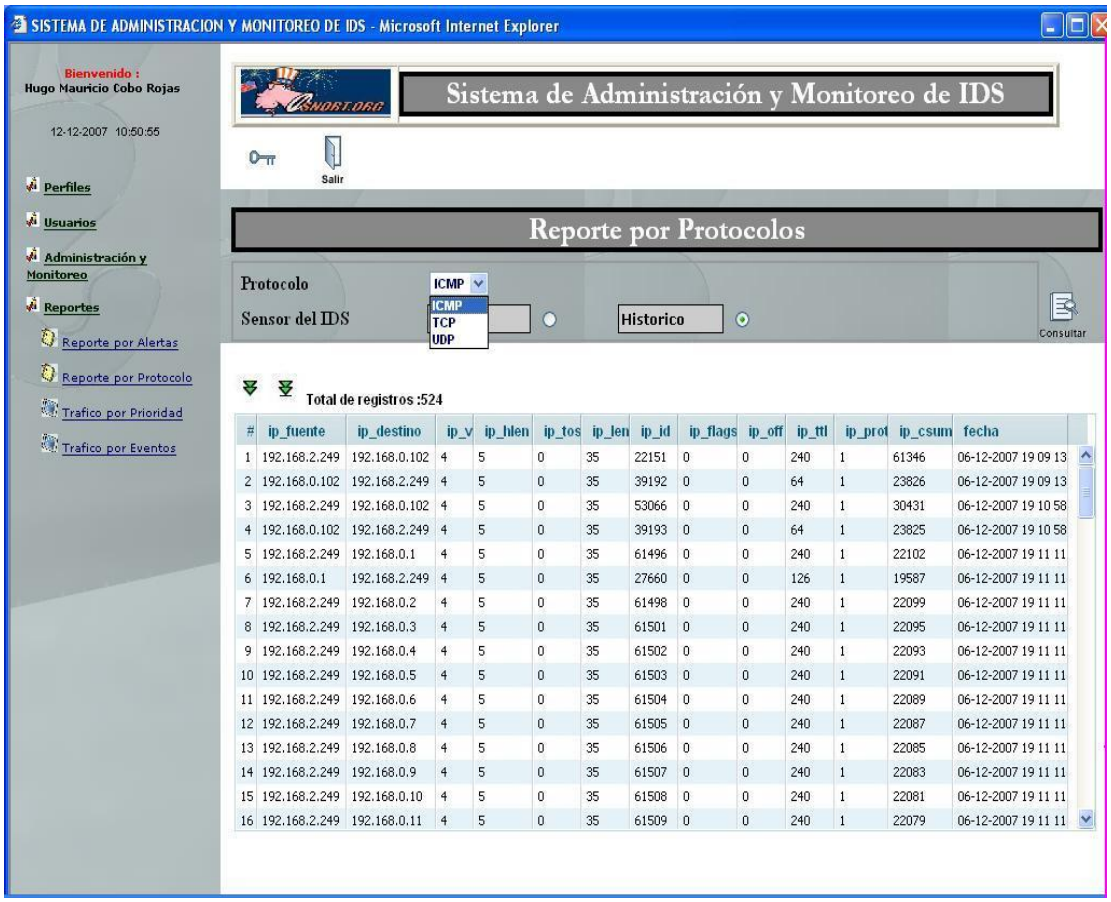


Figura 25

3.2.2.4.2. Reportes por Protocolo

Mostrará el tráfico por protocolo generado por el Snort mediante un grafico estadístico con su detalle.



#	ip_fuente	ip_destino	ip_v	ip_hlen	ip_tos	ip_len	ip_id	ip_flags	ip_off	ip_ttl	ip_prot	ip_csum	fecha
1	192.168.2.249	192.168.0.102	4	5	0	35	22151	0	0	240	1	61346	06-12-2007 19 09 13
2	192.168.0.102	192.168.2.249	4	5	0	35	39192	0	0	64	1	23826	06-12-2007 19 09 13
3	192.168.2.249	192.168.0.102	4	5	0	35	53066	0	0	240	1	30431	06-12-2007 19 10 58
4	192.168.0.102	192.168.2.249	4	5	0	35	39193	0	0	64	1	23825	06-12-2007 19 10 58
5	192.168.2.249	192.168.0.1	4	5	0	35	61496	0	0	240	1	22102	06-12-2007 19 11 11
6	192.168.0.1	192.168.2.249	4	5	0	35	27660	0	0	126	1	19587	06-12-2007 19 11 11
7	192.168.2.249	192.168.0.2	4	5	0	35	61498	0	0	240	1	22099	06-12-2007 19 11 11
8	192.168.2.249	192.168.0.3	4	5	0	35	61501	0	0	240	1	22095	06-12-2007 19 11 11
9	192.168.2.249	192.168.0.4	4	5	0	35	61502	0	0	240	1	22093	06-12-2007 19 11 11
10	192.168.2.249	192.168.0.5	4	5	0	35	61503	0	0	240	1	22091	06-12-2007 19 11 11
11	192.168.2.249	192.168.0.6	4	5	0	35	61504	0	0	240	1	22089	06-12-2007 19 11 11
12	192.168.2.249	192.168.0.7	4	5	0	35	61505	0	0	240	1	22087	06-12-2007 19 11 11
13	192.168.2.249	192.168.0.8	4	5	0	35	61506	0	0	240	1	22085	06-12-2007 19 11 11
14	192.168.2.249	192.168.0.9	4	5	0	35	61507	0	0	240	1	22083	06-12-2007 19 11 11
15	192.168.2.249	192.168.0.10	4	5	0	35	61508	0	0	240	1	22081	06-12-2007 19 11 11
16	192.168.2.249	192.168.0.11	4	5	0	35	61509	0	0	240	1	22079	06-12-2007 19 11 11

Figura 26

3.2.2.4.3. Tráfico por Prioridad

Detalle las prioridades capturados por el Snort.

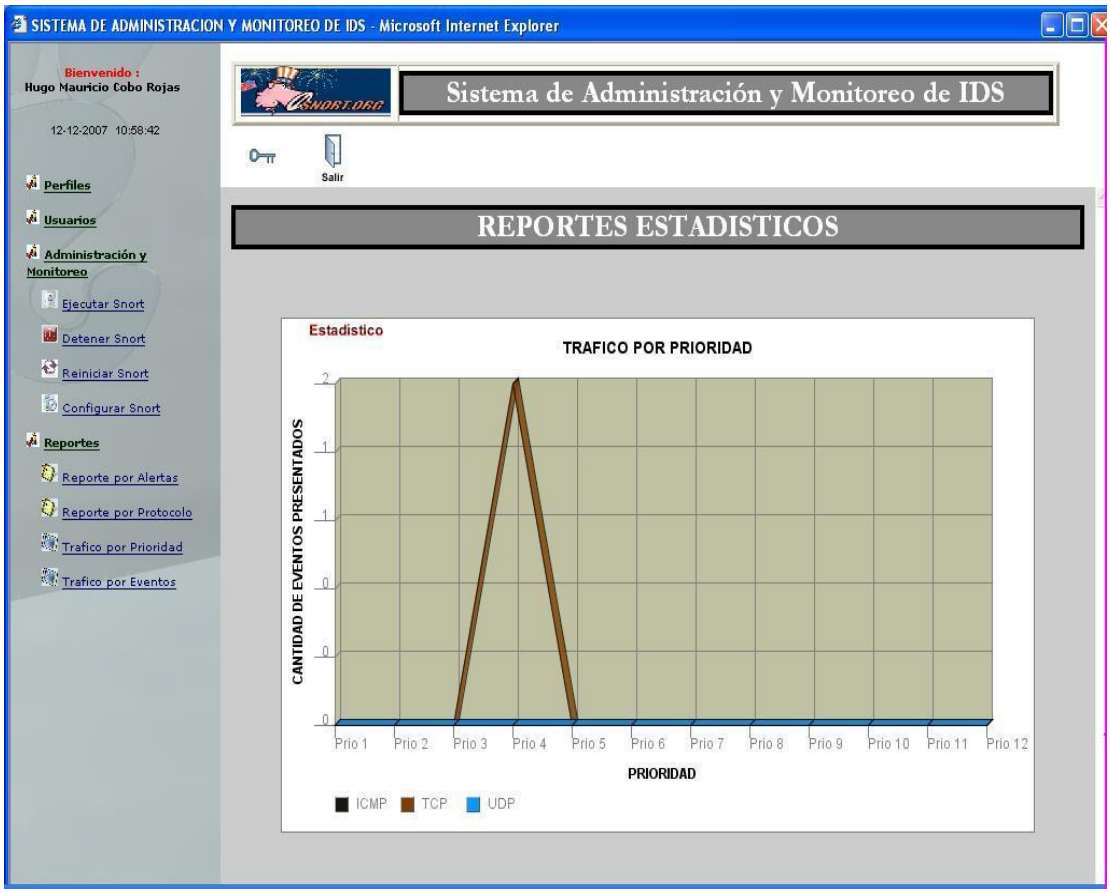


Figura 27

3.2.2.4.4. Tráfico por Evento

Reporte en detalle la cantidad de eventos capturadas por el Snort.

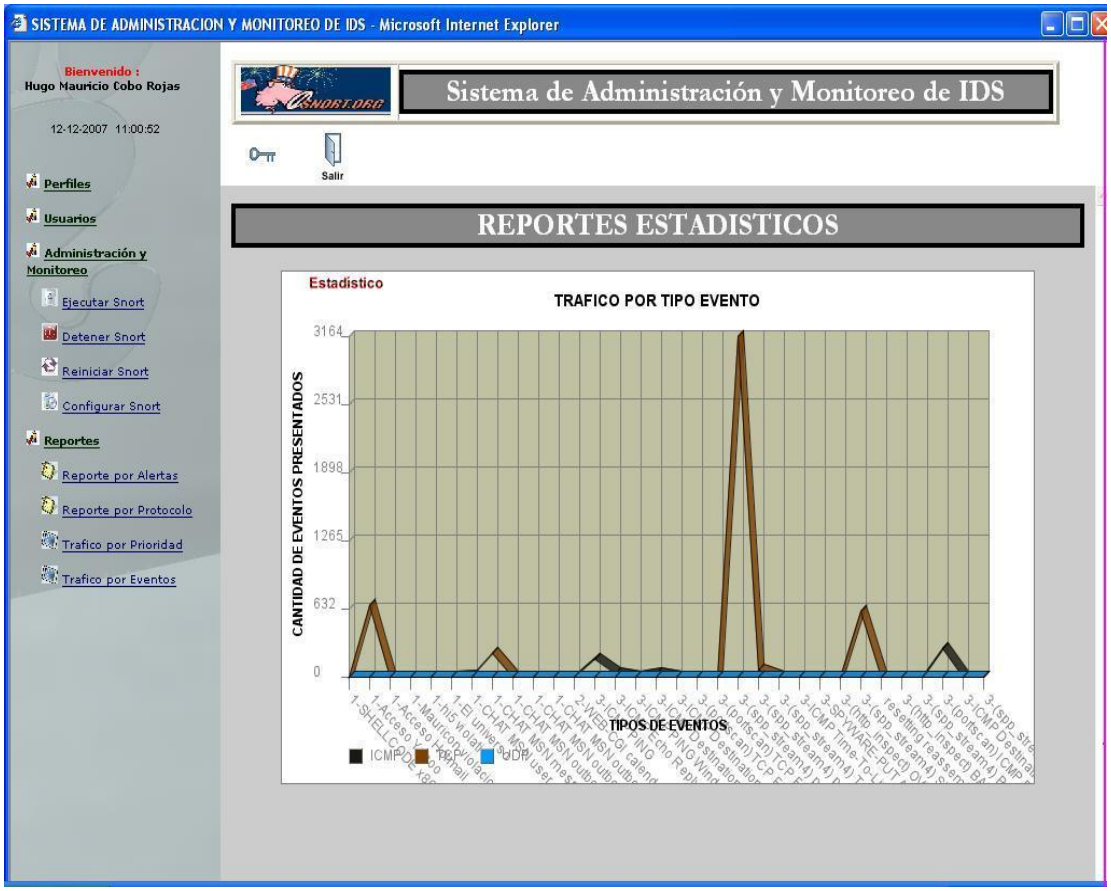


Figura 28

CAPITULO 4

Codificación

4. Principales Componentes

4.1. Servidor de Correo

Este servidor de correo se utilizara para poder enviar mensajes escritos a un teléfono celular cuando nuestro sistema detecte un intruso, alertando al administrador.

4.2. Servidor de Dominio

Se instalara el Sistema Operativo Linux en su distribución Fedora Core 7, en el mismo se habilitarán los servicios que conforman la estructura principal de un servidor de Dominio tales como DNS, DHCP entre los más importantes.

- ✓ **Servidor DNS**

  Aquí se configurarán los servicios de DNS el cual tendrá la responsabilidad de resolver nombres de maquina a direcciones IP y viceversa este es nuestro componente que hará las funciones de Servidor de DNS.

- ✓ **Servidor DHCP**

  Para asignarle una dirección IP a un equipo debemos configurar el servicio de DHCP el cual tendrá la función de asignar una dirección IP a los diferentes maquinas que forman parte de nuestro dominio,

4.3. Usuario Administrador

Es la persona encargada de llevar el control y monitoreo de nuestro sistema; es decir será la persona que va a manipular el sistema GuardianSnort.

4.4. Descripción del Diseño de Base de Datos Snort

Esta base contiene las tablas con las que trabaja la herramienta Ids Snort.

SCHEMA

Campo	Tipo de Dato	Detalle	Tipo de Campo
Vseq	int4	Identificador único de registro	primary key
Ctime	datetime	Identificador de tiempo de creación de la base	not null

Cuadro 11

SENSOR

Campo	Tipo de Dato	Detalle	Tipo de Campo
Sid	int4	identificador único de registro	primary key
hostname	varchar 15	identificador del host	not null
interface	varhcar 15	identificador de la interfaz de red	not null
Filter	varchar 15	filtro BPF	not null
detail	varchar 15	detalla el nivel del login	not null
encoding	varchar 15	detalla el formato de la carga	not null

Cuadro 12

EVENT

Campo	Tipo de Dato	Detalle	Tipo de Campo
Sid	int4	identificador del sensor	primary key
cid	int4	identificador del evento	primary key
Signatura	int4	identificador asignado	not null
Timestamp	datetime	tiempo que esta levantado el evento	not null

Cuadro 13

SIGNATURA

Campo	Tipo de Dato	Detalle	Tipo de Campo
sid_id	int4	indentificador de la asignatura	primary key
sig_name	varchar 255	nombre de la asignatura	not null
sig_class_id	int4	identificación de la clasificación	not null
sig_priority	int4	identificador de la prioridad	not null
sig_rev	int4	número de revisión	not null
sig_sid	int4	identificador interno de la signatura	not null

Cuadro 14

SIG_CLASS

Campo	Tipo de Dato	Detalle	Tipo de Campo
sid_class_id	int4	identificador de la referencia	primary key
sig_class_name	varchar 60	nombre de la clasificación	not null

Cuadro 15

SIG_REFERNCE

Campo	Tipo de Dato	Detalle	Tipo de Campo
sig_id	int4	indentificador de la asignatura	primary key
ref_seq	int4	número de secuencia de la referencia	primary key
ref_id	int4	identificador de la referencia	not null

Cuadro 16

REFERENTE

Campo	Tipo de Dato	Detalle	Tipo de Campo
ref_id	int4	identificador de la referencia	primary key
ref_system_id	int4	identificador de la referencia del sistema	
ref_tag	varchar 20	identificador de la referencia tag	not null

Cuadro 17

REFERENCE_SYSTEM

Campo	Tipo de Dato	Detalle	Tipo de Campo
ref_system_id	int4	identificador de la referencia del sistema	primary key
ref_system_id	int4	identificador de la referencia del sistema	not null

Cuadro 18

DATA

Campo	Tipo de Dato	Detalle	Tipo de Campo
Sid	int4	identificador de la referencia	primary key
cid	int4	identificador eventos	primary key
Data_payload	varchar	packed payload according	

Cuadro 19

IPHDR

Campo	Tipo de Dato	Detalle	Tipo de Campo
Sid	int4	identificador de sensor	primary key
cid	int4	identificador de enventos	primary key
ip_scr	int4	source de direcciones ip	
ip_dst	int4	direcciones ip destino	
ip_ver	int2	versión ip	
ip_hlen	int2	longuitud de la cabecera ip	
ip_tos	int2	tipo de servicio ip	
ip_len	int2	longuitud datagrama ip	
ip_id	int2	identificador ip	
ip_flags	int2	flags ip	
ip_off	int2	ip fragment offset	
ip_ttl	int2	tiempo de vida ip	
ip_proto	int2	protocolo ip	
ip_csum	int2	ip checksum	

TCPHDR

Campo	Tipo de Dato	Detalle	Tipo de Campo
Sid	int4	identificador de sensor	primary key
cid	int4	identificador de enventos	primary key
tcp_snort	int2	puerto tcp source (origen)	
tcp_dport	int2	puerto tcp destino	
tcp_seq	int4	número de la secuancia tcp	
tcp_ack	int4	número ASK TCP	
tcp_off	int2	tcp offset	
tcp_res	int2	tcp reservado	
tcp_flags	int2	flags tcp	
tcp_win	int2	ventana tcp	
tcp_csum	int2	tcp checksum	
tcp_urp	int2	tcp urgent pointer	

Cuadro 21

UDPHDR

Campo	Tipo de Dato	Detalle	Tipo de Campo
Sid	int4	identificador de sensor	primary key
cid	int4	identificador de enventos	primary key
Udp_snort	int2	puerto udp source (origen)	
Udp_dport	int2	puerto udp destino	
Udp_len	int2	longitud udp	
Udp_csum	int2	udp checksum	

Cuadro 22

ICMPHDR

Campo	Tipo de Dato	Detalle	Tipo de Campo
Sid	int4	identificador de sensor	primary key
cid	int4	identificador de enventos	primary key
Icmp_type	int2	tipo_icmp	
Icmp_code	int2	codigo_icmp	
Icmp_csum	int2	icmp_checksum	
Icmp_id	int2	identificador icmp	
Icmp_seq	int2	número de secuencia	

Cuadro 23

OPT

Campo	Tipo de Dato	Detalle	Tipo de Campo
Sid	int4	identificador de sensor	primary key
cid	int4	identificador de enventos	primary key
optad	int4	opciones de indentificador	primary key
opt_proto	int2	opciones de protocolo	
opt_code	int2	opción de código	
opt_len	int2	longitud de opción	
opt_data	varchar	opción de datos	

Cuadro 24

4.5 Descripción del Diseño de la Base de Datos OLIMPO

Esta base contiene los registros de los accesos a nuestra herramienta realizados por los tipos de usuarios que existen.

PERSONA

Campo	Tipo de Dato	Detalle	Tipo de Campo
id_persona	int4	identificador único de registro	primary key
nombre	varchar	nombre de persona	not null
apellido	varchar	apellido de persona	not null
Sexo	Char	M maculino, F femenino	not null
fecha_registro	Date	fecha en que se creo registro	not null
fecha_modificad	Date	fecha que se modifico el registro	
vigencia	Char	A activo, I inactivo	not null

Cuadro 25

USUARIO

Campo	Tipo de Dato	Detalle	Tipo de Campo
id_usuario	int4	identificador único de registro	primary key
login	varchar	nombre de usuario	not null
password	varchar	registra la clave del usuario	
email	varchar	registra email de usuario	not null
clave_email	varchar	registra clave del email del usuairo	
id_perfil	int4	FK fde la tabla perfil	not null
fecha_registro	date	fecha en que se creo registro	not null
fecha_modificación	date	fecha que se modifico el registro	
vigencia	char	A activo, I inactivo	not null

Cuadro 26

REGLAS

Campo	Tipo de Dato	Detalle	Tipo de Campo
codigo	int4	identificador único de registro	primary key
nombre	varchar	nombre de la regla	not null
descripcion	varchar	resgistra una descripción de laregla	not null
cargar	varchar	carga las reglas	not null

Cuadro 27

PERFIL

Campo	Tipo de Dato	Detalle	Tipo de Campo
id_perfil	int4	identificador único de registro	primary key
nombre	varchar	nombre de persona	not null
descripcion	varchar	descirpción	
fecha_registro	Date	fecha en que se creo registro	not null
fecha_modificac	Date	fecha que se modifico el registro	
vigencia	Char	A activo, I inactivo	not null

Cuadro 28

ID_MODULO_DETALLE

Campo	Tipo de Dato	Detalle	Tipo de Campo
id_modulo	int4	identificador único de registro	primary key
descripcion	varchar	nombre del modulo detalle	not null

ROLES_MODULOS

Campo	Tipo de Dato	Detalle	Tipo de Campo
idserial	int4	identificador único de registro	primary key
id_modulo	int4	FK de la tabla modulo	not null
id_modulo_detalle	int4	FK de la tabla modulo_detalle	not null
id_perfil	int4	FK de la tabla perfil	not null

Cuadro 29

MODULO

Campo	Tipo de Dato	Detalle	Tipo de Campo
id_modulo	int4	identificador único de registro	primary key
desc_modulo	int4	descripción del modulo	not null
desc_completa_modulo	int4	descripción completa del modulo	not null

Cuadro 30

CAPITULO 5

Desarrollo, Pruebas e Implementación del Sistema

5.- Desarrollo

5.1.- Creación de la Base de Datos

Para la creación de la base de datos nosotros:

- ✓ Seleccionamos el motor de PosgresQL 8.1.10
- ✓ Los objetos de conexión y manipulación de datos de la base están desarrollados en su totalidad en lenguaje java.

5.2. Pruebas del Sistema

5.2.1. Pruebas de Aplicación Ensambladas

Con esta prueba de aplicación lo que nosotros tratamos de buscar son los posibles errores que se puedan dar con respecto a la funcionalidad de nuestro sistema, ya que cada modulo de nuestra aplicación se deberá ensamblar de acuerdo a las etapas anterior.

5.2.2. Pruebas de la aplicación con varios usuarios

Nuestra aplicación es de fácil configuración tanto para los usuarios operadores como para el usuario administrador, además tiene un entorno de fácil entendimiento para que así ellos puedan manipular el sistema; para esta prueba se utilizan datos reales de los usuarios.

5.3.- Implementación del Sistema

5.3.1.- Componentes de Software

La implementación de nuestro dominio se realizara en la distribución de Fedorore Core; además se configuran otros servicios como:

- ✓ Sistema Operativo Linux distribución Fedora Core 6, 7
- ✓ Aplicativo IDS SNORT 2.7
- ✓ Base de Datos PostgreSql
- ✓ Administrador de PostgreSql PgAdmin
- ✓ Lenguaje de Programación Java
- ✓ Eclipse interfaz gráfica para Java
- ✓ Plugin versión español para Eclipse
- ✓ Servidor de Aplicación Apache Tomcat 5.0
- ✓ Servidor de Correo

5.3.2. Componentes del Hardware

- ✓ 1 PC 80 GB, 512 RAM, procesador de 2.8 Ghz que tenga instalado el Windows XP, además se instalara una maquina virtual como es el Vmware.

CAPITULO 6

Recomendaciones y Conclusiones

6.1.- Recomendaciones:

Este proyectos trata de la de un Detector de Intruso en el cual se debe tener muy en cuenta que cada componente que conforme el sistema como tal debe estar bien configurado porque de lo contrario cuando un componente falle esto conllevará a que todo el sistema falle ocasionando así una perdida irremediable de los datos que se encuentran almacenados en las bases de datos.

La seguridad en este tipo de sistemas es de suma importancia por lo cual desarrollamos un mecanismo de acceso por logoneo con lo cual estamos restringiendo el acceso a usuarios no permitidos para que nuestro sistema no pueda ser manipulado o modificado sin tener previo permiso o privilegios que se dan una vez que se crean usuarios, de esta forma cumplimos con estándares de seguridad a nivel se sistemas de computación.

6.2.- Conclusiones:

Para la implementación de nuestro sistema tuvimos que realizar varios pasos que empezaron con el análisis, el diseño, la codificación, y termino con las pruebas y puesta en marcha del sistema, con todo esto nos aseguramos que nuestro producto este estandarizado y sujeto a cambios para nuevos requerimientos del mismo.

Hoy en día poder tener el control de sobre cada maquina de una red de computadoras es algo complicado por ese motivo para poder llevar un control se creo un producto como es el GUARDIANSNORT el cual a través de una interfaz WEB trata de suplir las necesidades de los usuarios dándole así a ellos el control sobre cada recurso compartido en la red; de esta manera se podrá administrar la red de datos de una manera rápida.

Nuestro sistema podrá crear privilegios dependiendo el tipo de usuario que lo valla a ejecutar por ese motivo se podrán realizar consultas, crear nuevos usuarios, cambiar claves, modificar y eliminar usuarios además este sistema presenta una bitácora con todas las tareas que se han realizado en el día, en el mes, etc.

Bibliografía

http://www.snort.org	Página Oficial del snort
http://www.linuxparatodos.net	Información variada de Linux
http://linuxzeros.org	Foro de Snort en Linux
http://www.postgresql.org	Página Oficial de Postgresql

1. Manual de Usuario Operador

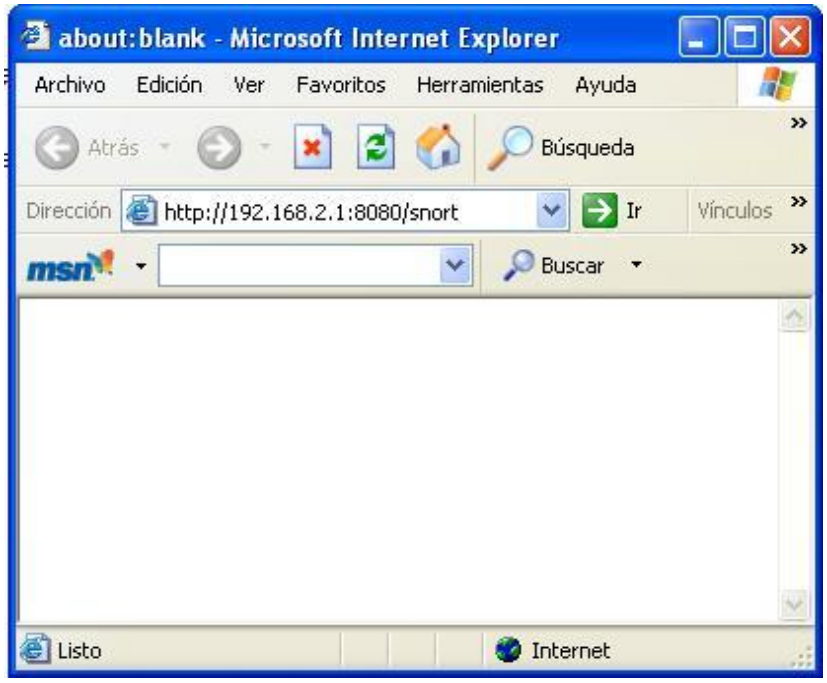
1.1. Introducción al Usuario.

El presente documento tiene como objetivo principal capacitar a los usuarios encargados del uso de la aplicación GuardianSnort, es necesario que el usuario en mención siga explícitamente los pasos y consejos escritos en este documento.

2. Accesos al Aplicativo

2.1. Formulario para Ingreso

Escribimos en el navegador la siguiente dirección: http://192.168.2.1:8080/snort para que así podamos ir a la página principal del Proyecto.



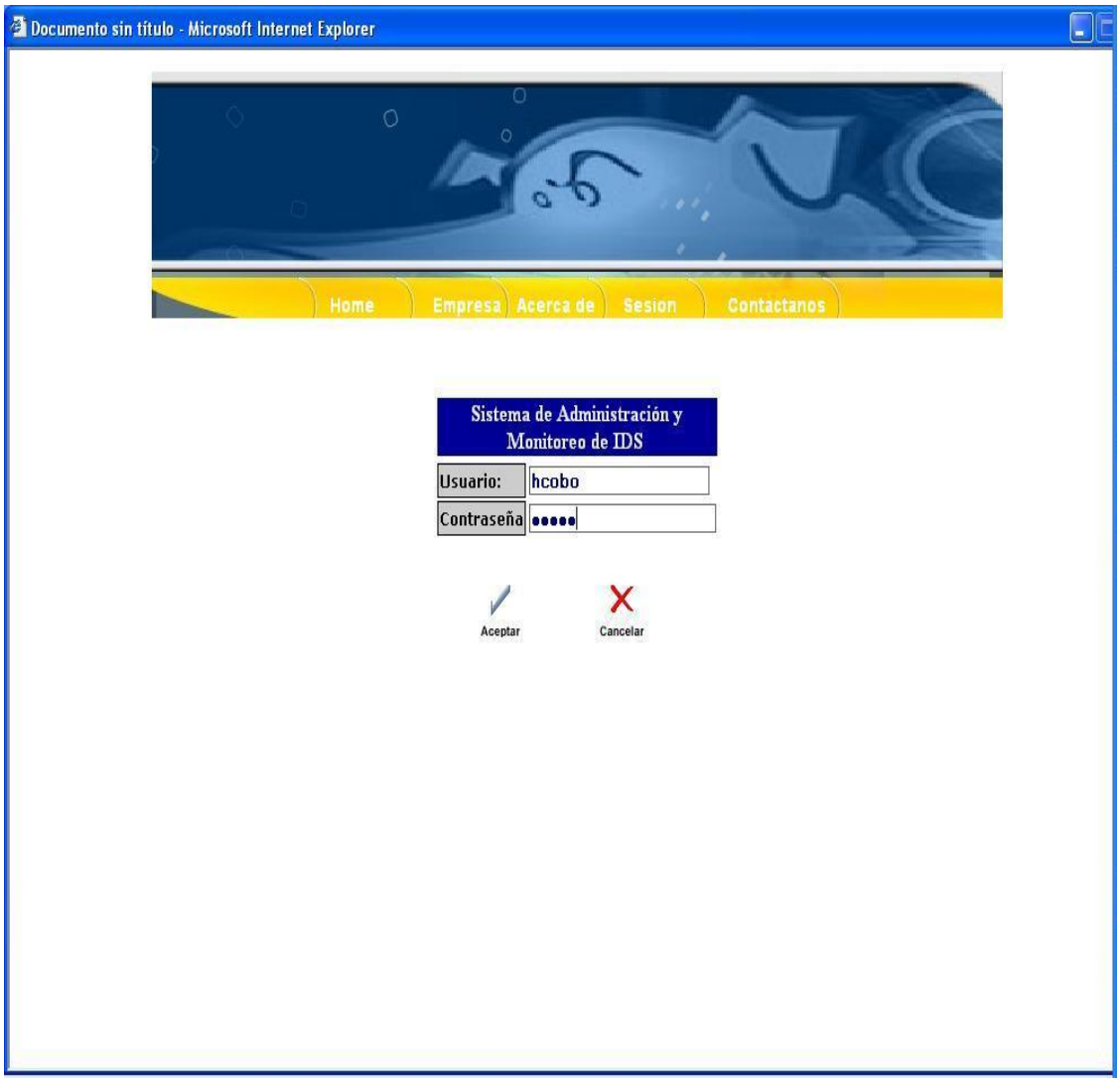
Una vez ingresado la dirección se levanta nuestra aplicación.

2.2. Iniciar Sesión

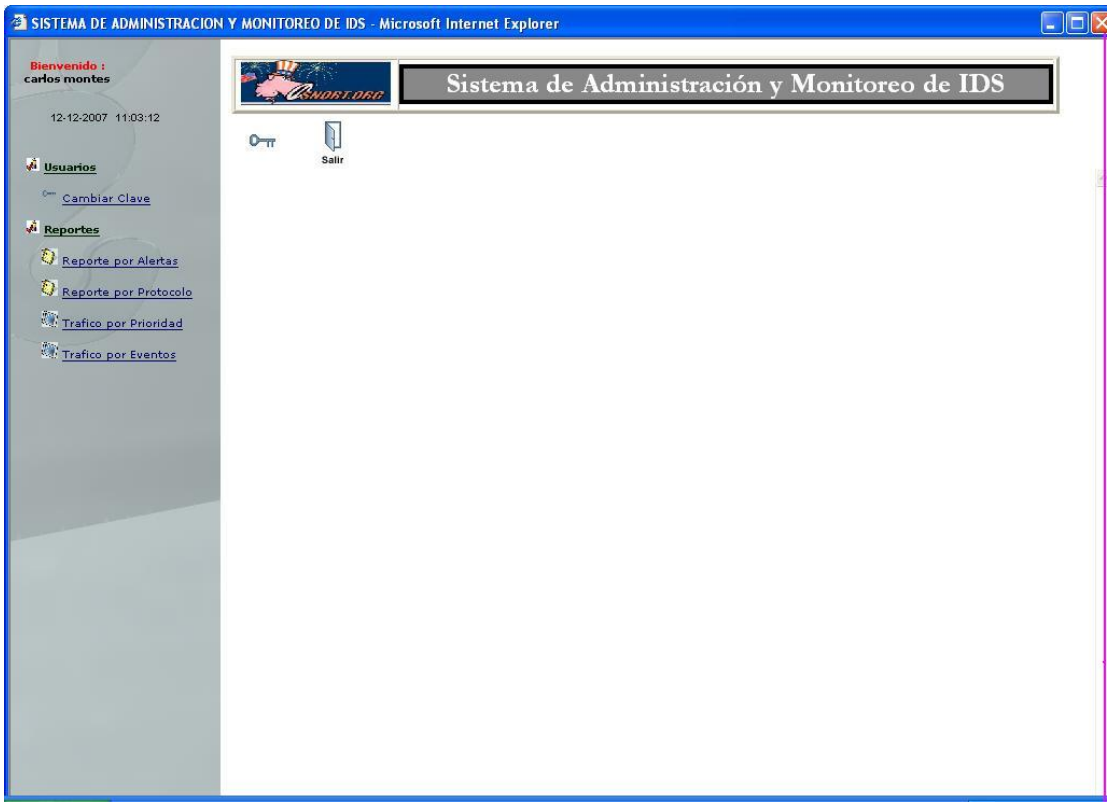
Haciendo Clic en **Sesión**:

Dependiendo que de que usuario lo va a usar, ya sea **Administrador** (con todos los privilegios de usuario) u **Operador** (con privilegios limitados) podrá ingresar al aplicativo.

Ingresamos el **usuario** con su respectiva **contraseña**, clic en **aceptar**



Se podrá visualizar el menú completo de nuestra aplicación con todas sus opciones:

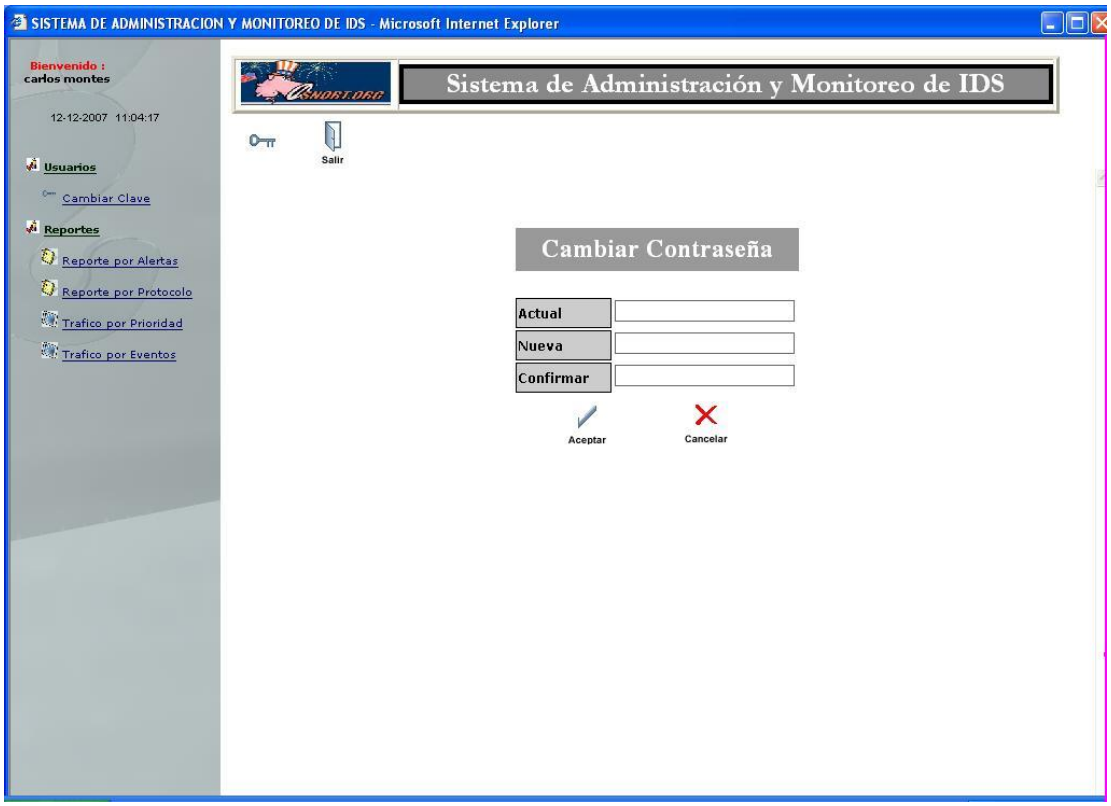


3. Menú Usuarios

Haciendo Clic en la opción Usuarios se tiene las opciones de configuración y administración de usuarios como:

3.1. Cambiar Clave

Cualquier usuario puede cambiar su contraseña de acceso, ingresará su código y procederá a cambiar su clave.

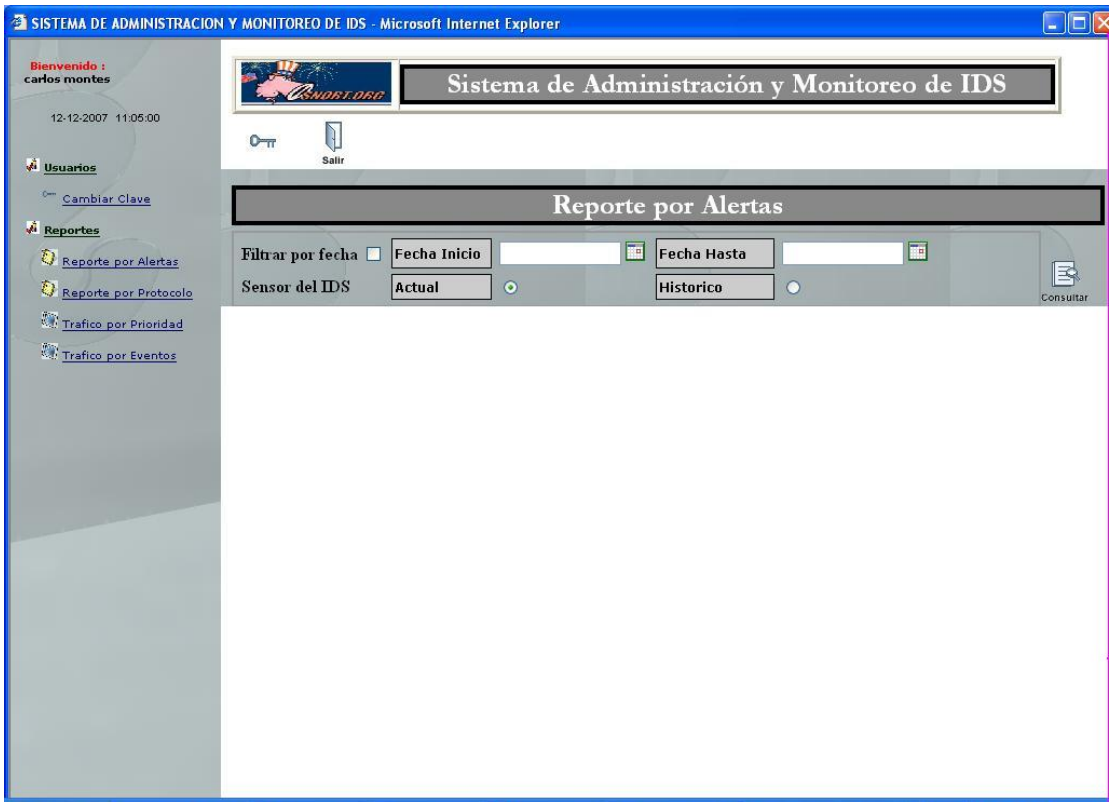


4. Reportes

Aquí el usuario podrá ver los reportes ya sea por alertas, protocolo trafico por IP y trafico por protocolo.

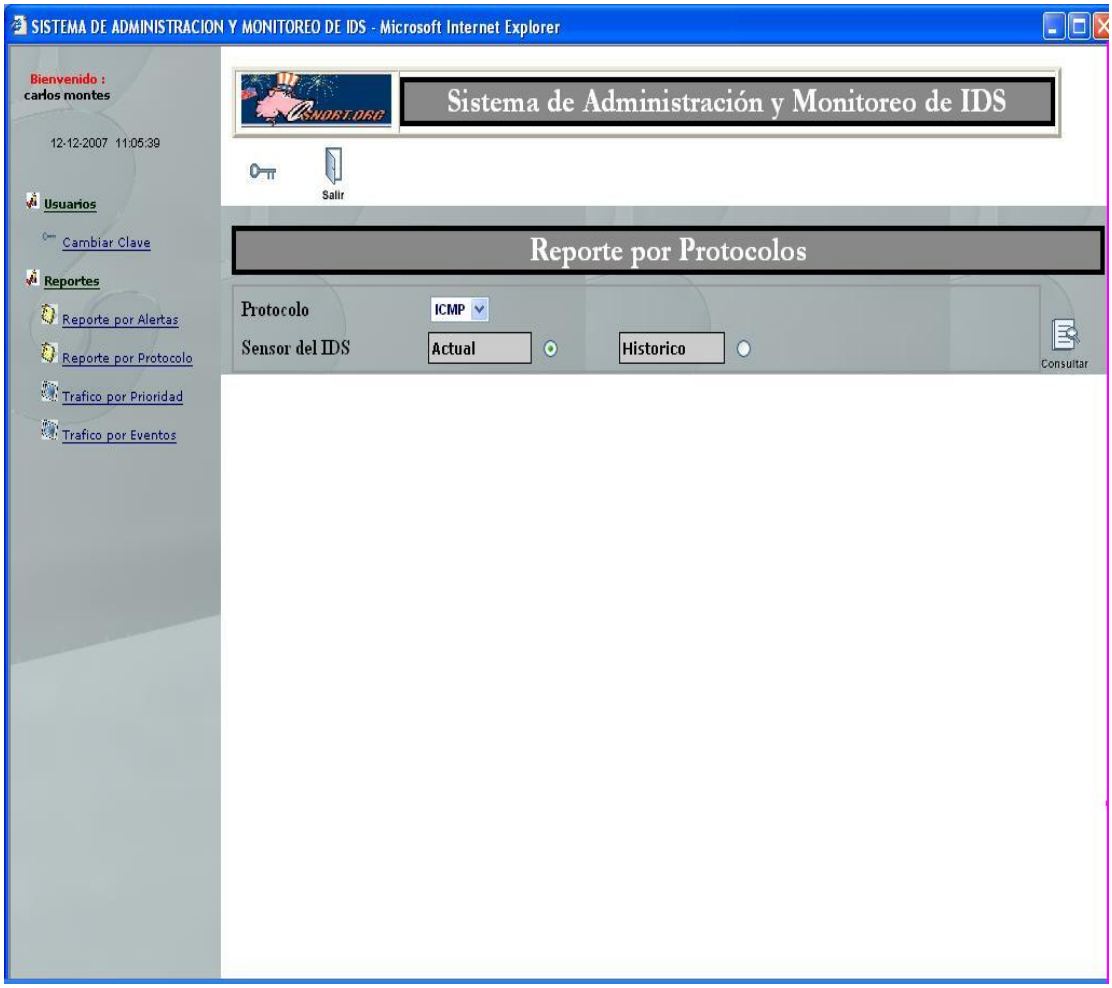
4.1. Reportes por Alertas

Clic en Reportes por Alertas, tiene que escoger un rango de fechas desde el icono del calendario tanto para **fecha inicio** como para **fecha hasta** y clic en la opción **consultar**.



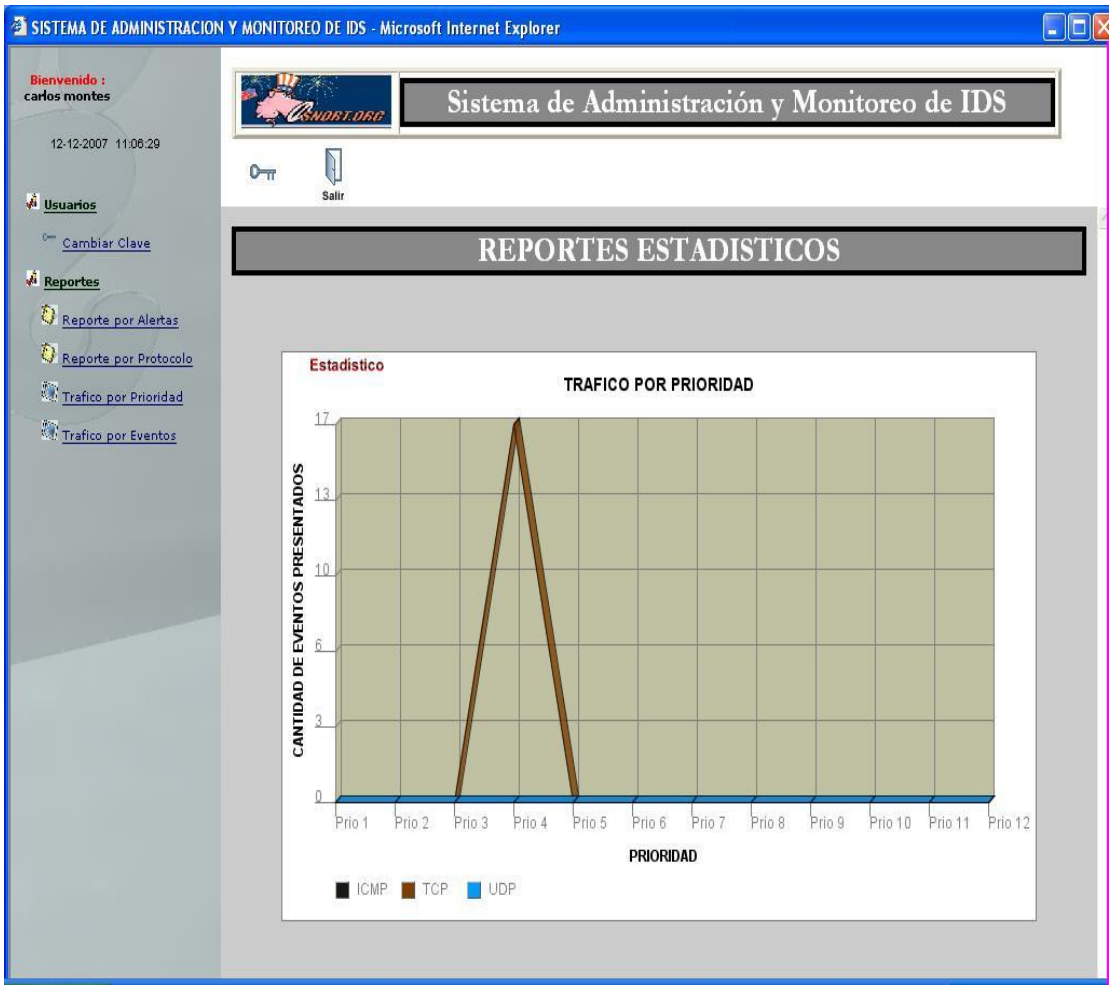
4.2. Reportes por Protocolo

Clic en Reportes por Protocolo, en el combo se escoge el tipo (ICMP, TCP o UDP) y clic en la opción consultar.



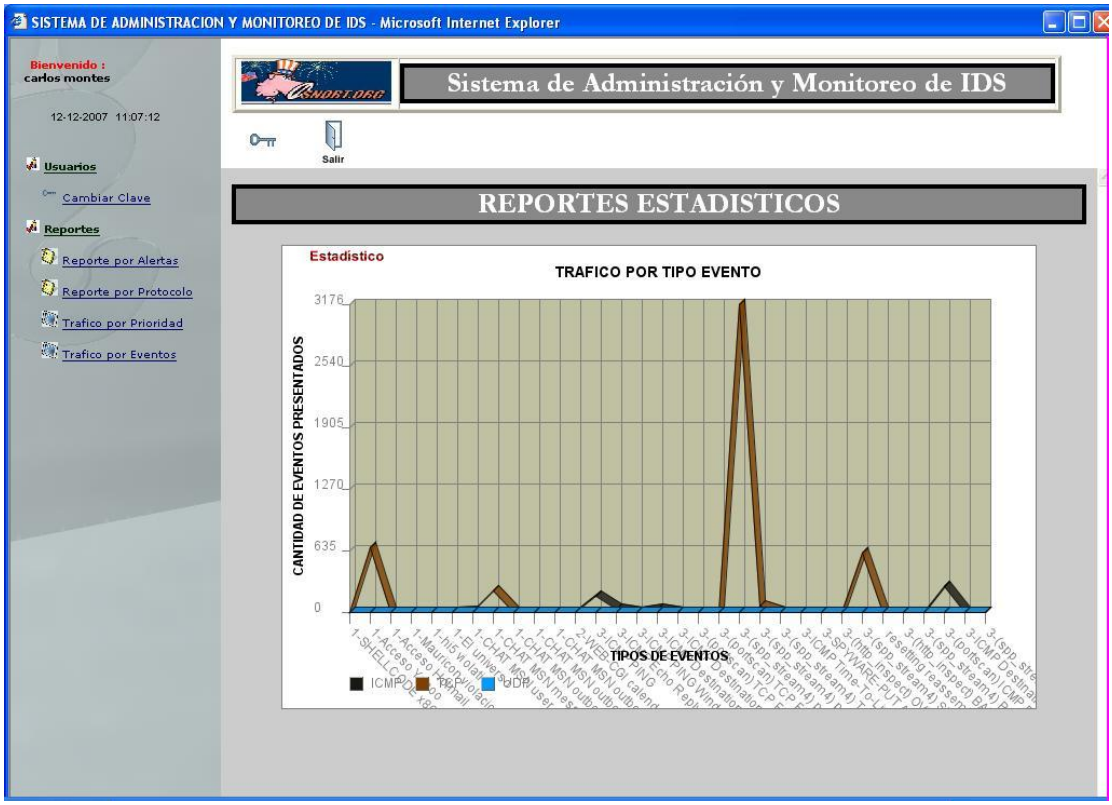
4.3. Trafico por Prioridad

Clic en Reportes por Prioridad, muestra de forma de gráfico estadístico el tráfico que circula por la red.



4.4. Tráfico por Evento

Clic en Reportes por Eventos, muestra de forma de gráfico estadístico el tráfico de eventos que ocurren en la red.



Manual del Usuario Administrador

5. Introducción al Administrador.

El presente documento tiene como objetivo principal capacitar a los administradores encargados del uso de la aplicación GuardianSnort, es necesario que el administrador en mención siga explícitamente los pasos y consejos escritos en este documento.

6. Iniciar Servicio

Levantados los servicios en nuestro servidor:

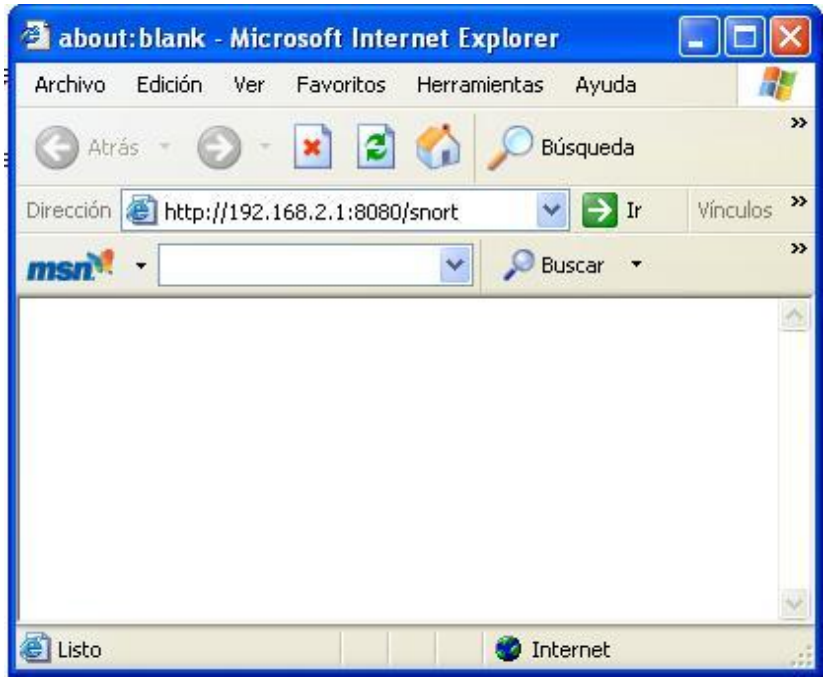
De red:	Service network Start
De correo:	Service sendmail start
	Service dovecot start
De base de datos:	Service postgresql start

NOTA: Estos servicios ya se encuentran levantados por defecto.

7. Accesos al Aplicativo

7.1. Formulario para Ingreso

Escribimos en el navegador la siguiente dirección: http://192.168.2.1:8080/snort para así poder acceder al la página principal de nuestro sistema.



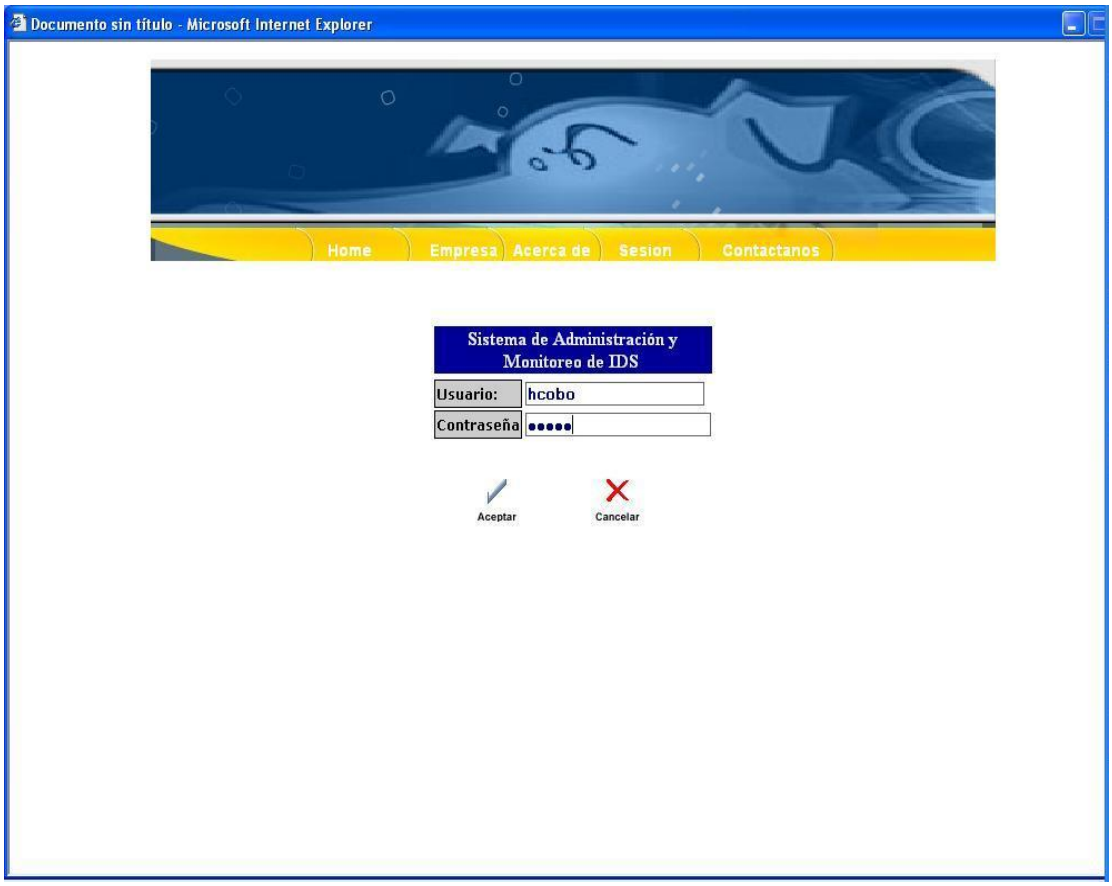
Aquí se levanta nuestra aplicación.

7.2. Iniciando Sesión

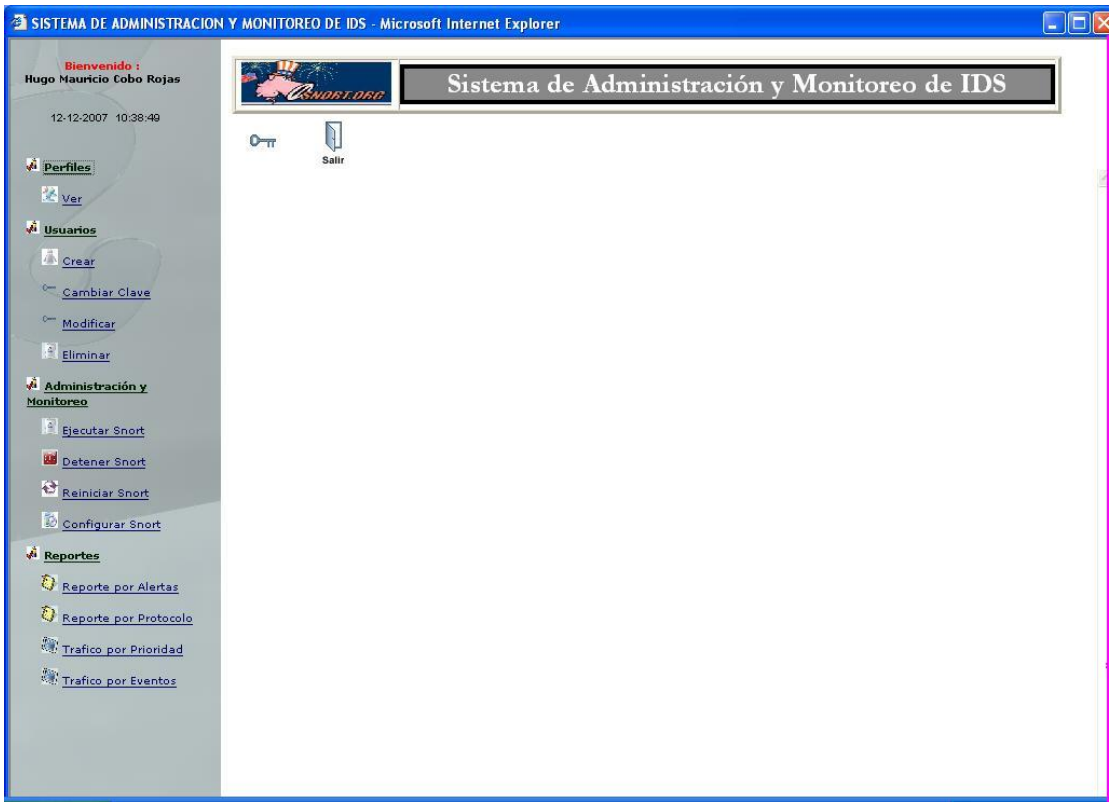
Haciendo Clic en **Sesión**:

Dependiendo del administrador que lo va a usar, ya sea Administrador (con todos los privilegios de usuario).

Ingresamos el **usuario** con su respectiva **contraseña**, clic en **aceptar**

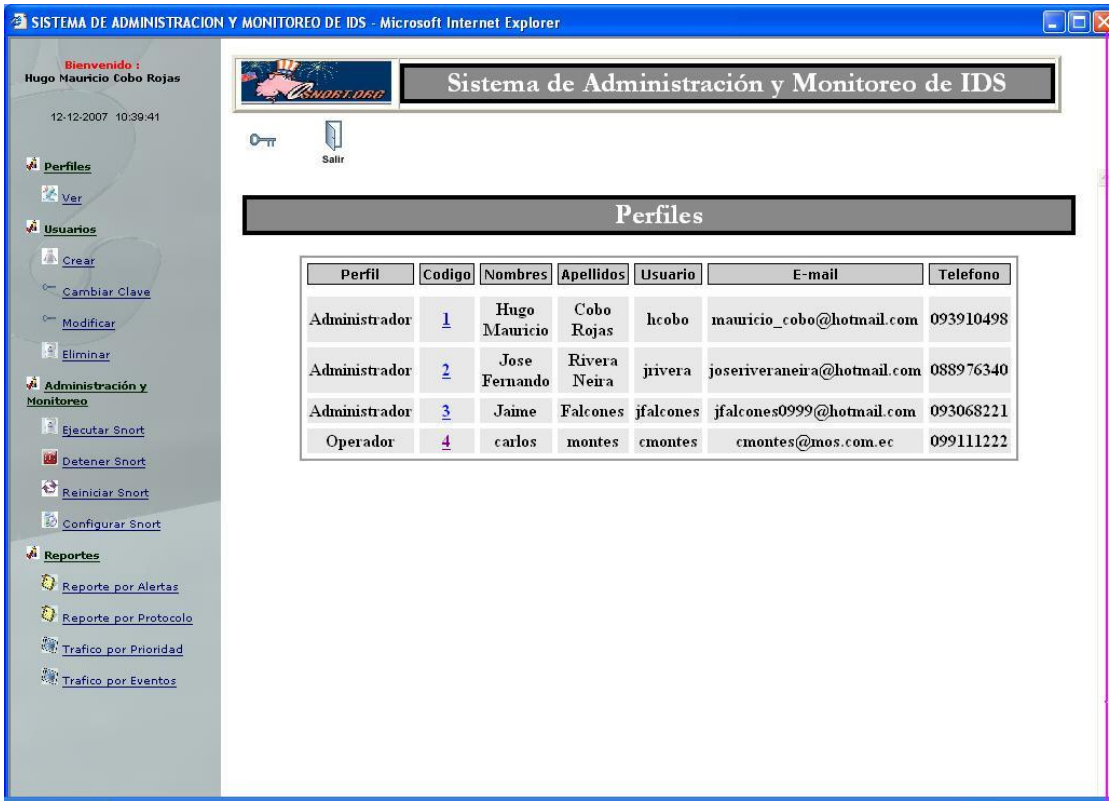


Se podrá visualizar el menú completo de nuestra aplicación con todas sus opciones:



8. Menú Perfiles

Haciendo clic en la opción **Perfiles**, clic en **Ver**. Se podrá visualizar todos los usuarios existentes para el uso del sistema con su respectiva información de cada uno de ellos.

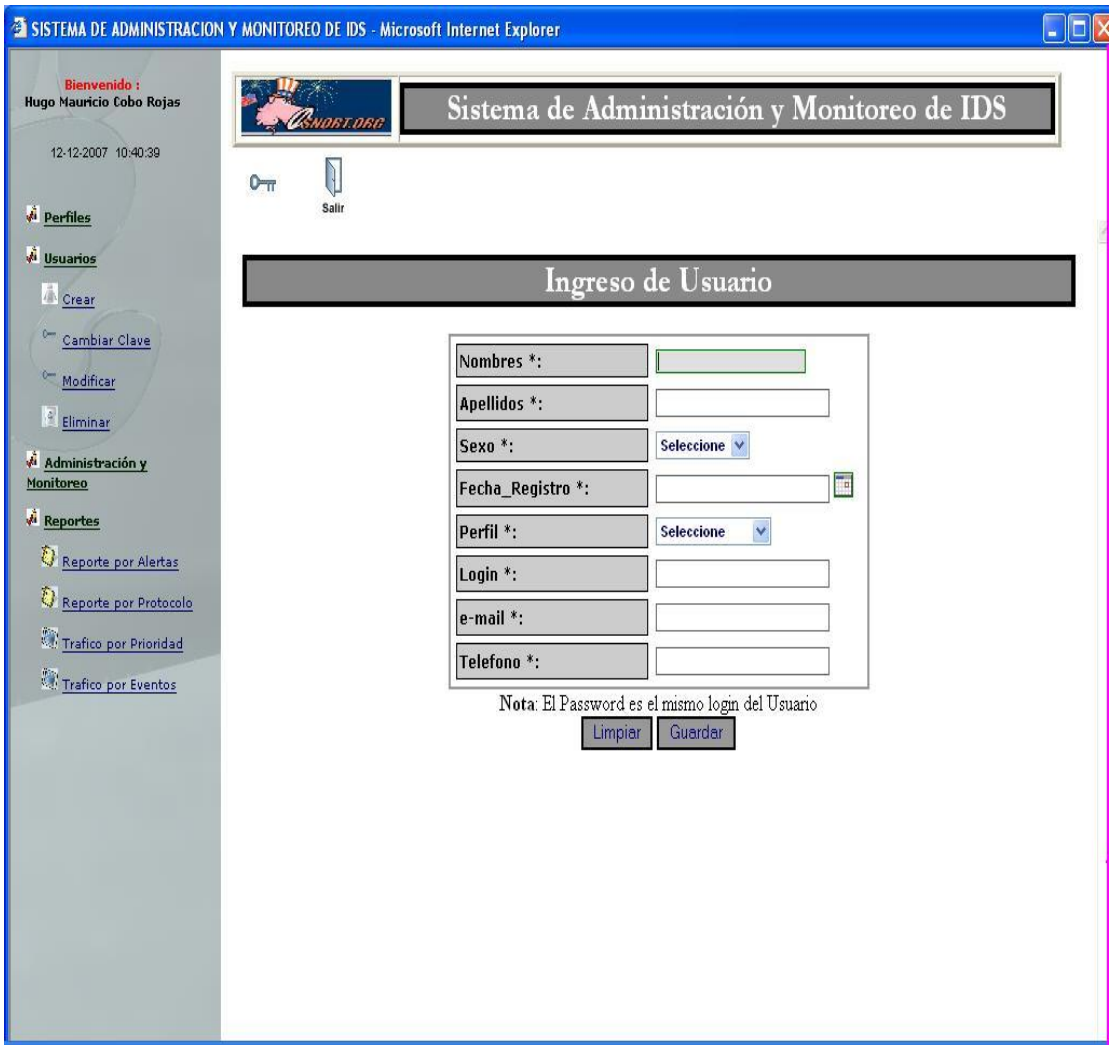


9. Menú Usuarios

Clic en opción Usuarios se tiene las opciones de configuración y administración de usuarios:

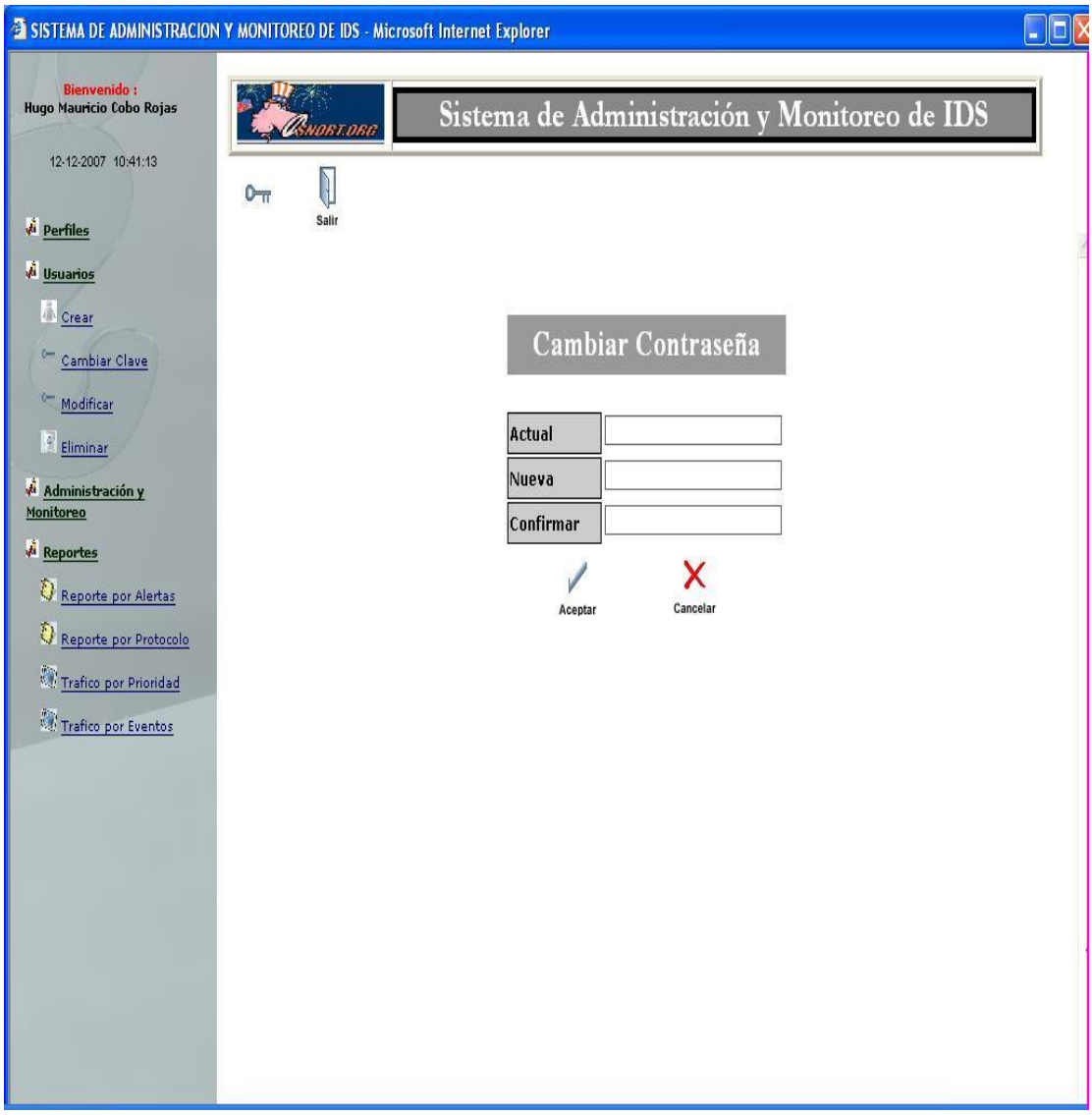
9.1. Crear usuario

Si se quiere ingresar un nuevo usuario al sistema, el cual se le asigna un perfil para el uso del aplicativo.



9.2. Cambiar Clave

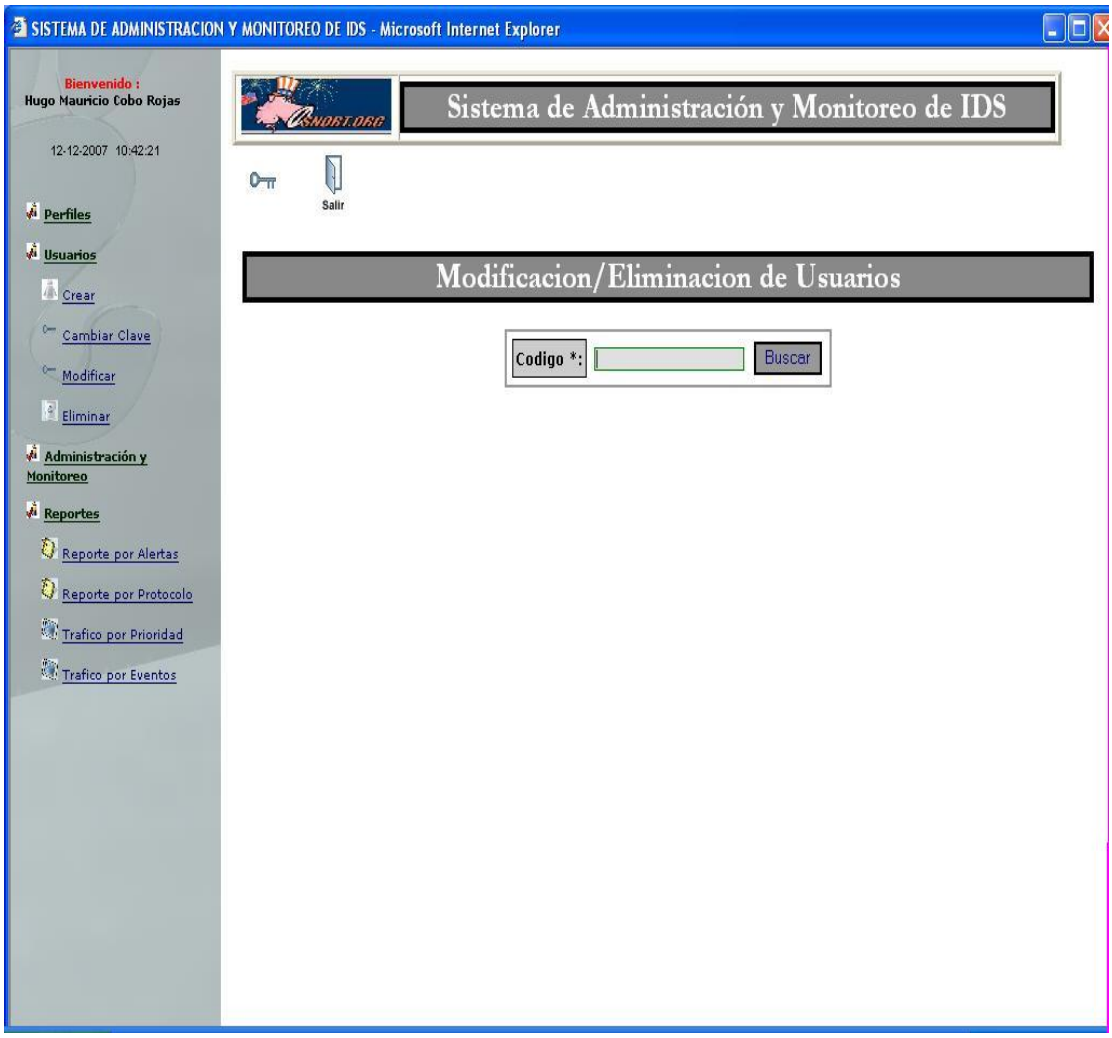
Cualquier usuario puede cambiar su contraseña de acceso, ingresará su código y procederá a cambiar su clave.



9.3. Modificar Usuario

Actualiza los datos de usuario seleccionado.

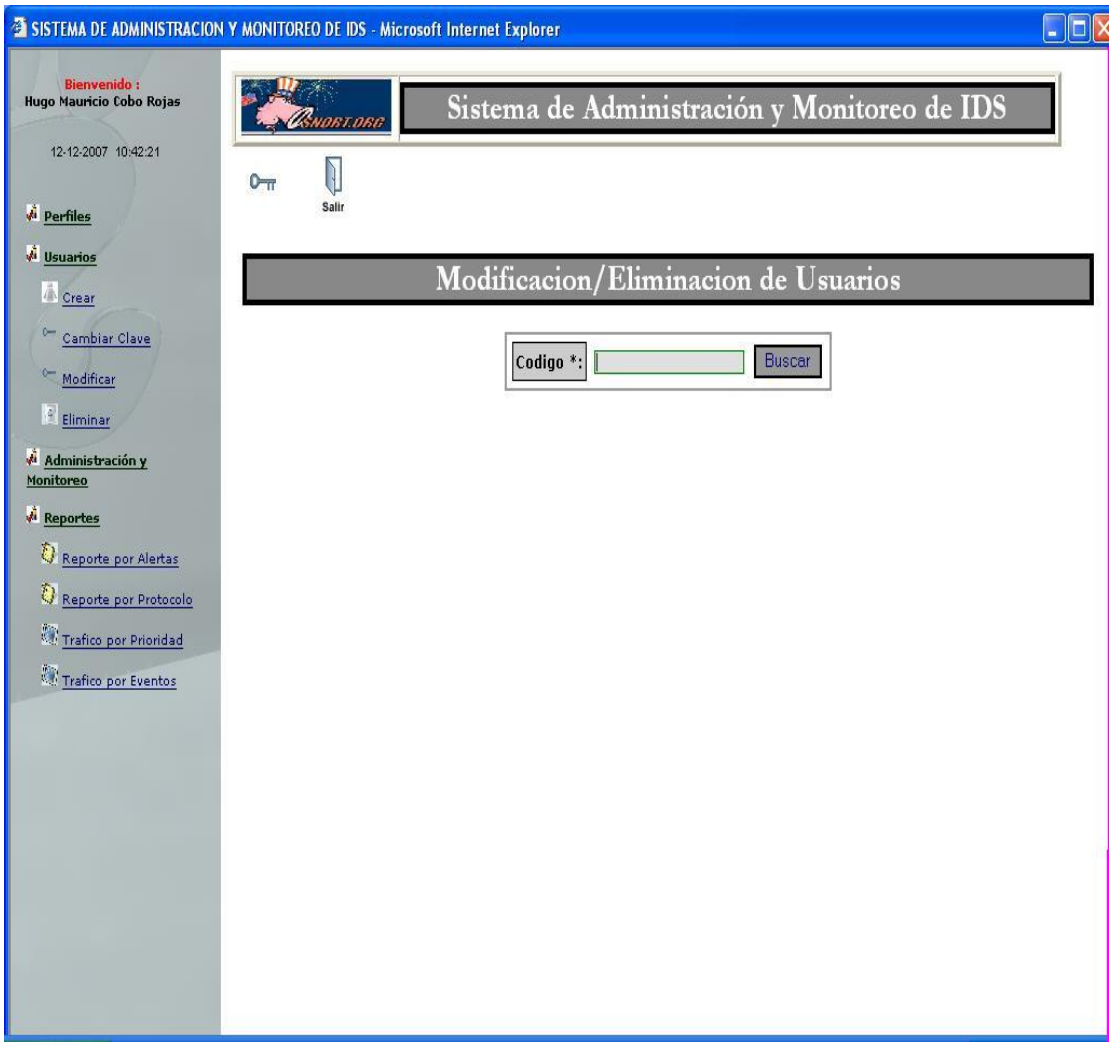
NOTA: Esta opción es solo permitida a usuarios con perfil de Administrador



9.4. Eliminar Usuario

Ingresando el código del usuario a eliminar.

NOTA: Esta opción es solo permitida a usuarios con perfil de Administrador



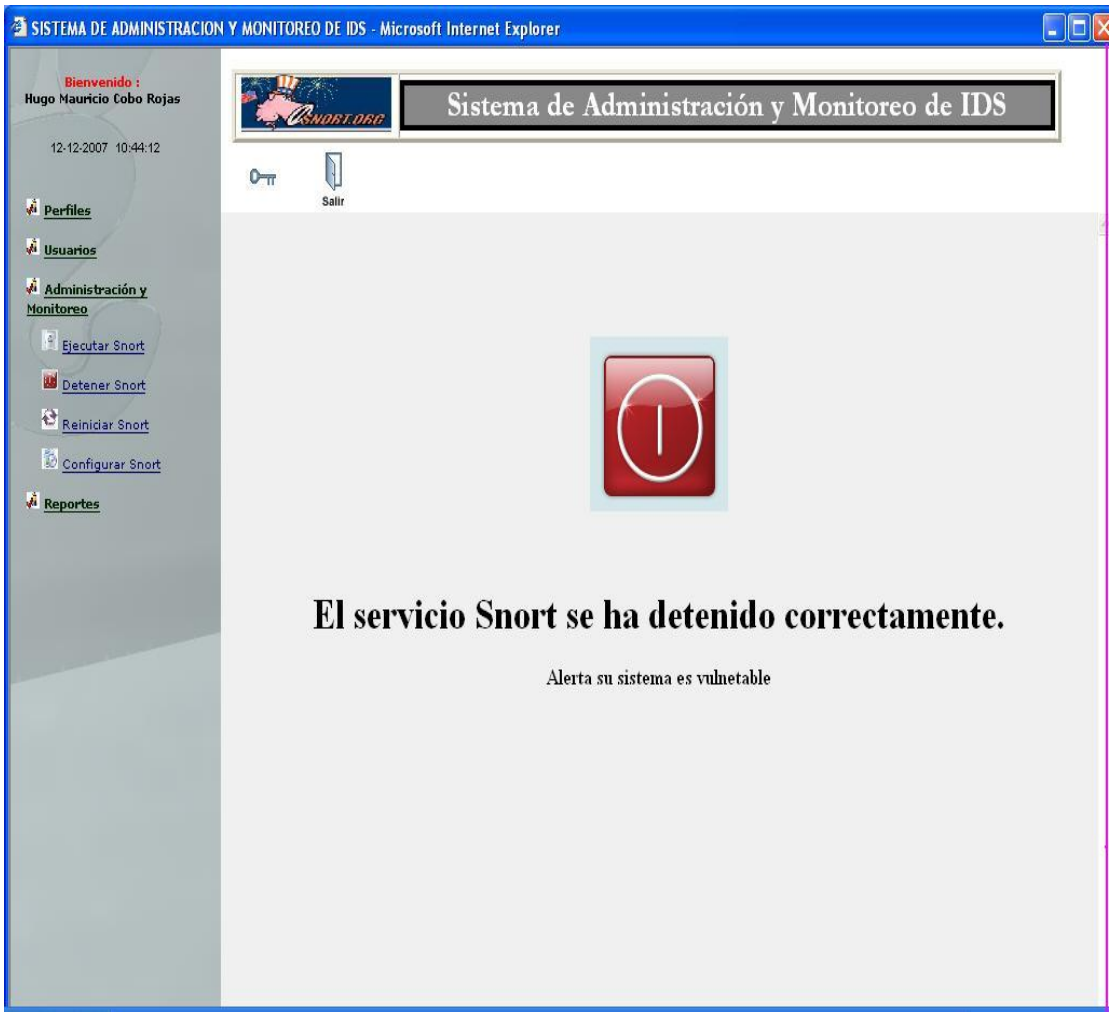
10. Administración y Monitoreo.

10.1. Ejecutar Snort

Haciendo clic en esta opción podrá iniciar el servicio del ids snort

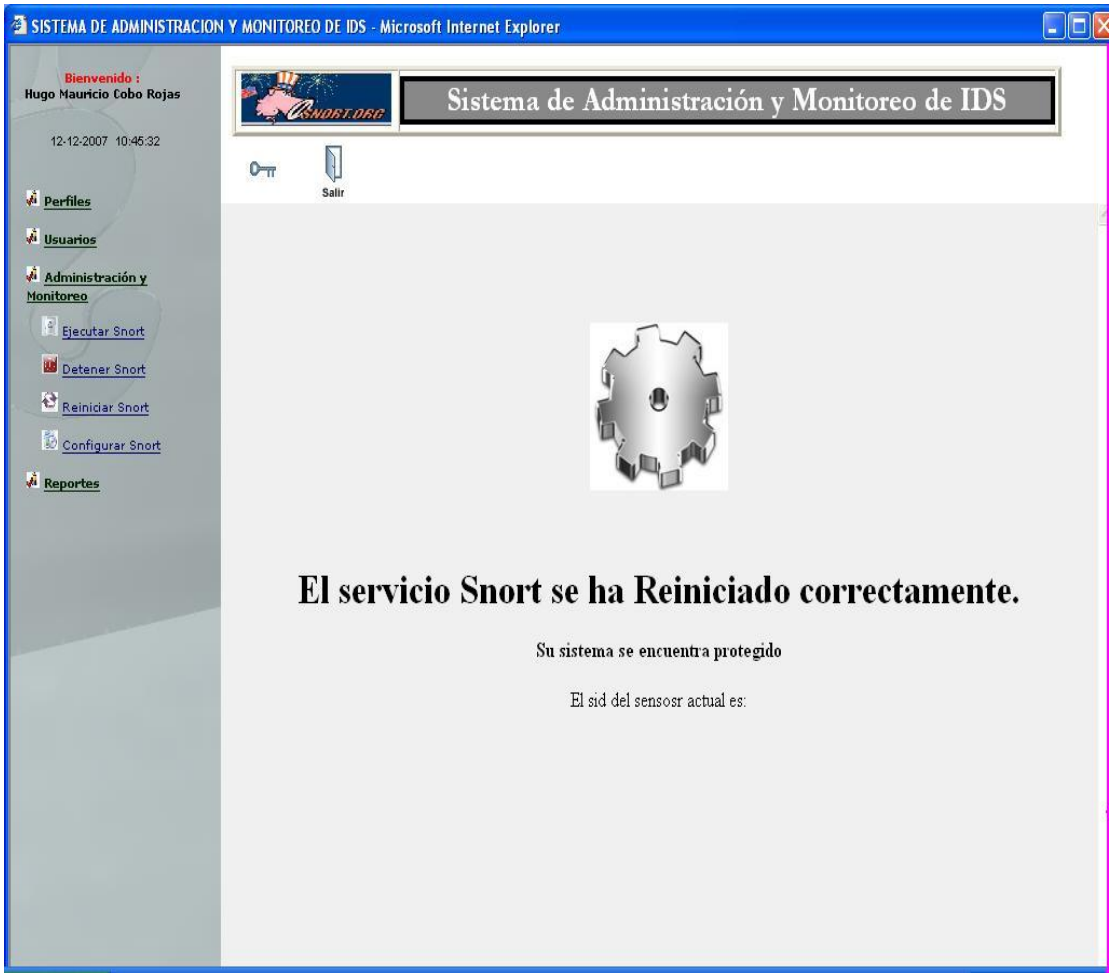
10.2. Detener Snort

Haciendo clic en esta opción podrá detener el servicio del ids snort



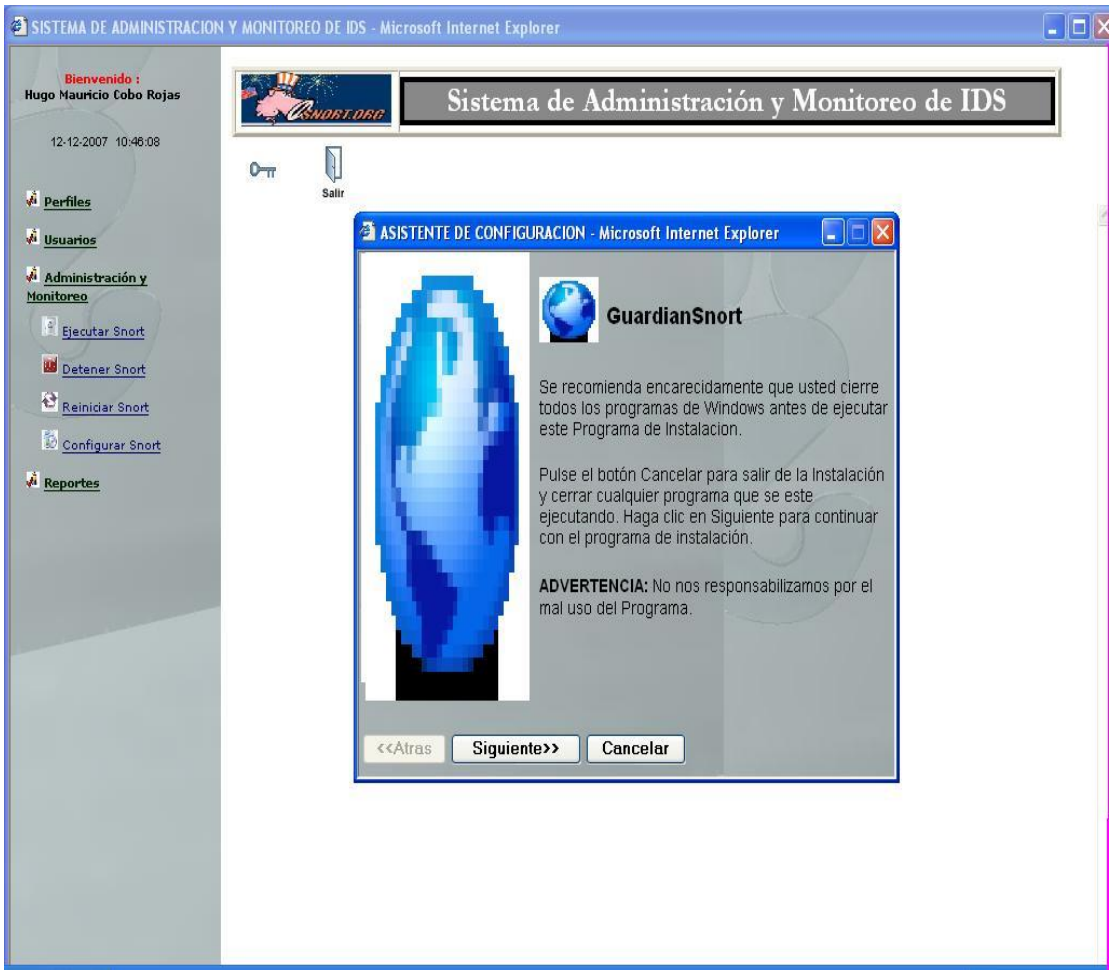
10.3. Reiniciar Snort

Haciendo clic en esta opción podrá reiniciar el servicio del ids snort



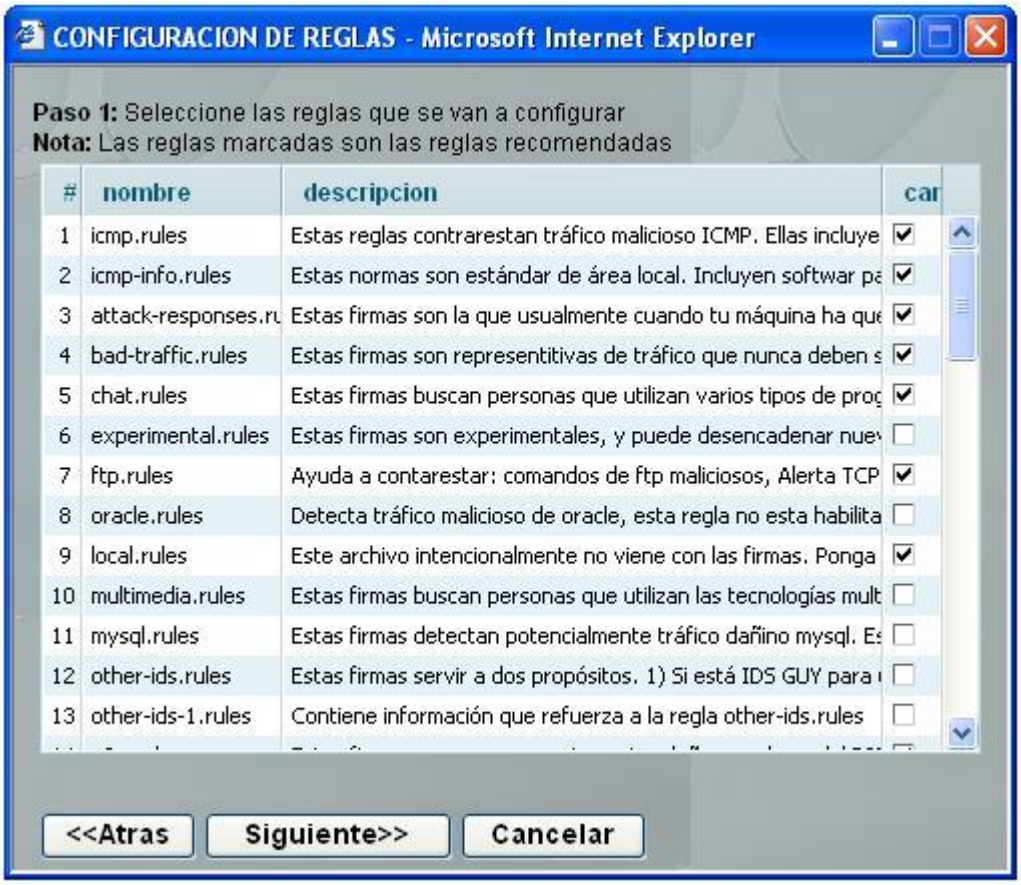
10.4. Configurar Snort

Haciendo clic en esta opción podrá configurar por medio de un wizard el servicio del ids snort.



Una vez levantado el wizard de configuración de, clic en el botón **siguiente**,

Aparecerá la lista con todas reglas se según sea la necesidad de configuración se podrá activar o desactivar mediante el uso de casillas de verificación.



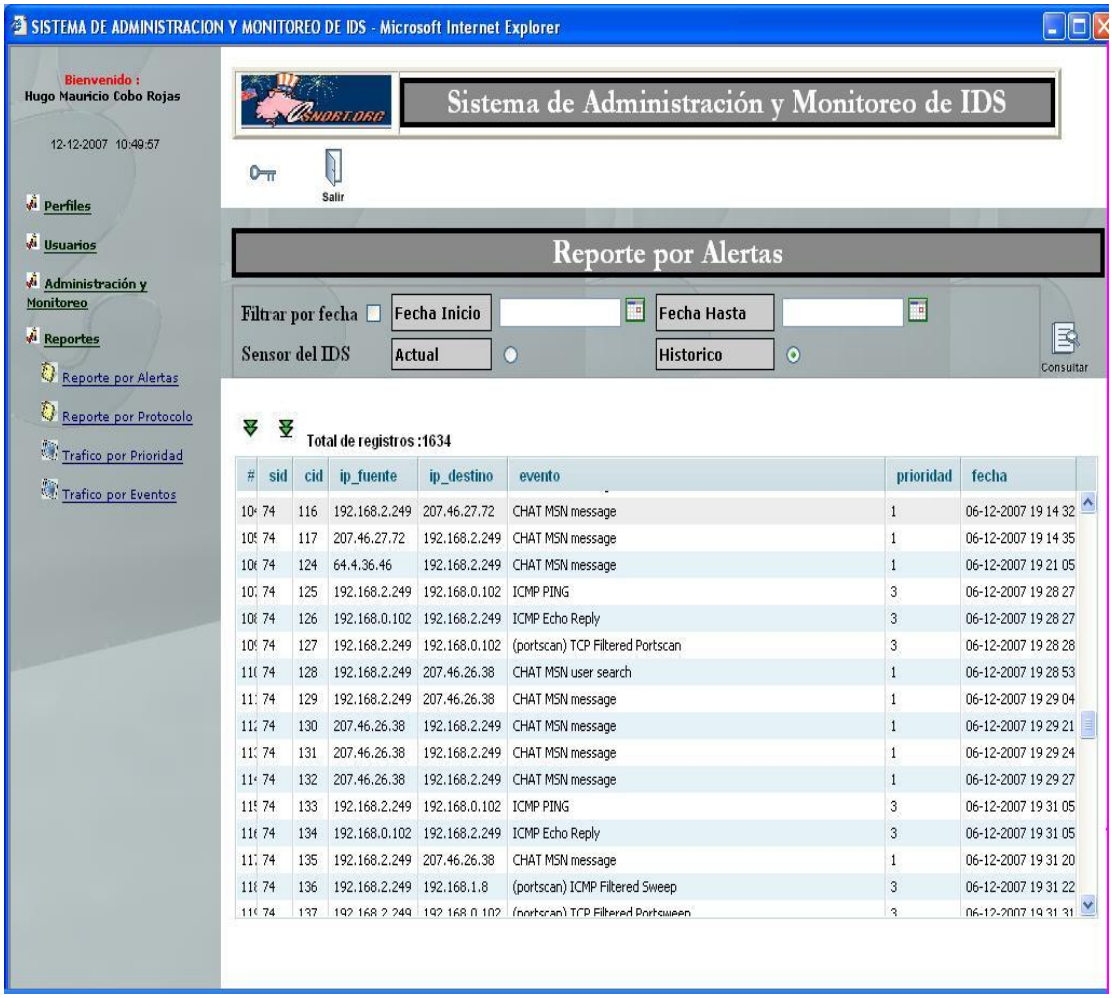
Una vez seleccionadas las reglas, clic en **siguiente** para seguir, caso contrario **atrás** para volver o **cancelar**.

Luego, se debe escoger la forma en que se enviarán las alertas al correo electrónico y al teléfono móvil, seleccionado la prioridad y a que perfiles de usuarios lo recibirán.

11. Reportes

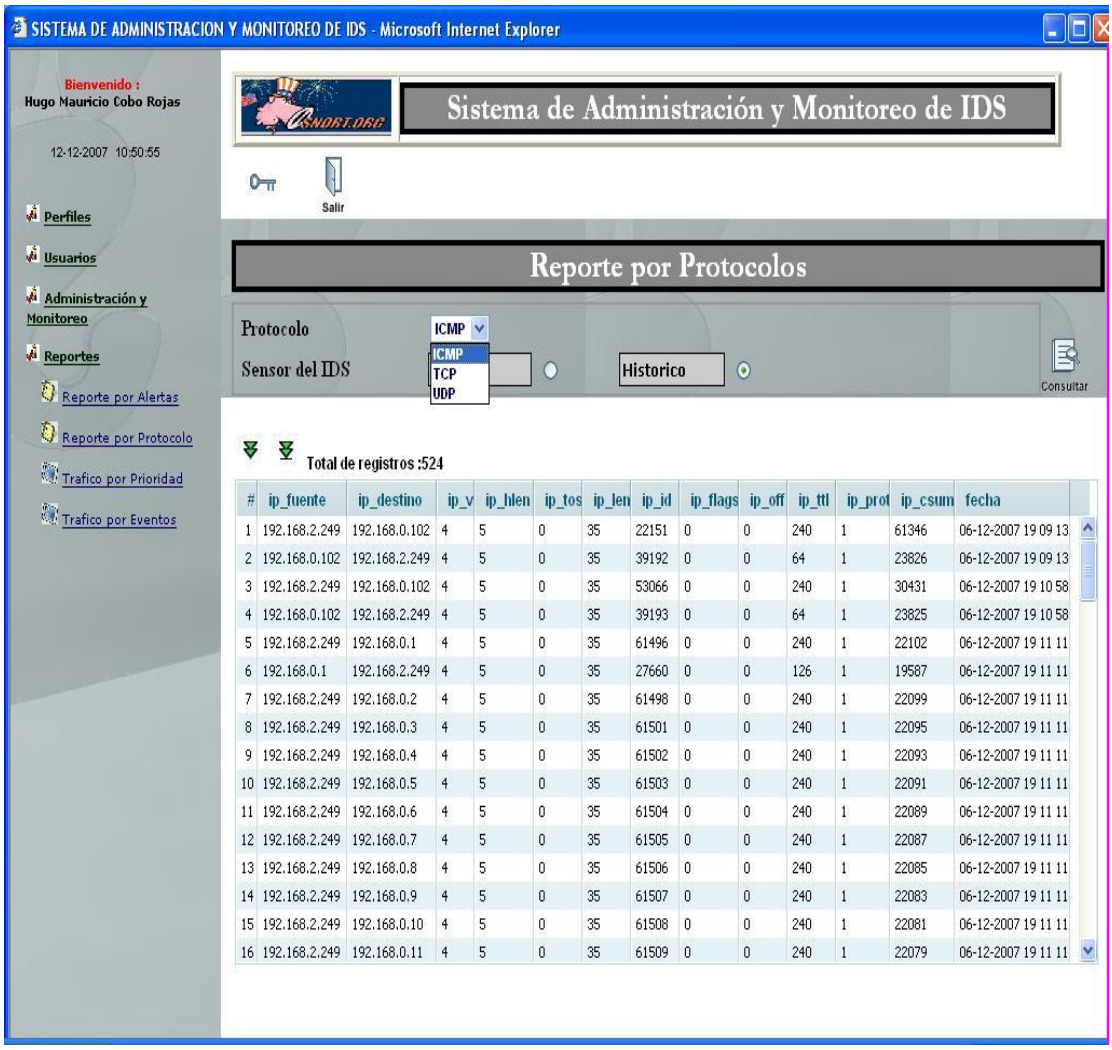
11.1. Reportes por Alertas

Clic en Reportes por Alertas, tiene que escoger un rango de fechas desde el icono del calendario tanto para **fecha inicio** como para **fecha hasta** y clic en la opción **consultar**.



11.2. Reportes por Protocolo

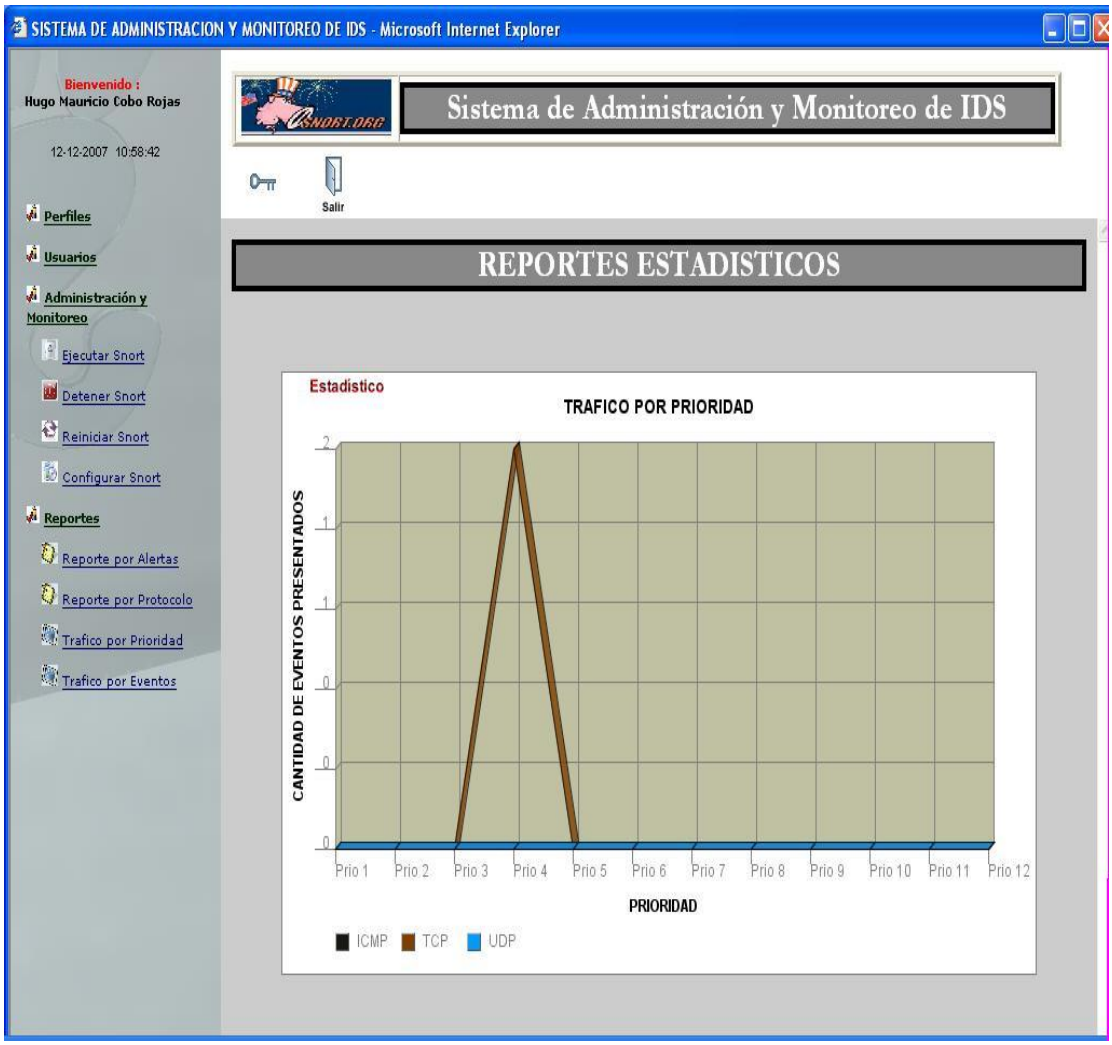
Clic en Reportes por Protocolo, en el combo se escoge el tipo (ICMP, TCP o UDP) y clic en la opción consultar.



#	ip_fuente	ip_destino	ip_v	ip_hlen	ip_tos	ip_len	ip_id	ip_flags	ip_off	ip_ttl	ip_prot	ip_csum	fecha
1	192.168.2.249	192.168.0.102	4	5	0	35	22151	0	0	240	1	61346	06-12-2007 19 09 13
2	192.168.0.102	192.168.2.249	4	5	0	35	39192	0	0	64	1	23826	06-12-2007 19 09 13
3	192.168.2.249	192.168.0.102	4	5	0	35	53066	0	0	240	1	30431	06-12-2007 19 10 58
4	192.168.0.102	192.168.2.249	4	5	0	35	39193	0	0	64	1	23825	06-12-2007 19 10 58
5	192.168.2.249	192.168.0.1	4	5	0	35	61496	0	0	240	1	22102	06-12-2007 19 11 11
6	192.168.0.1	192.168.2.249	4	5	0	35	27660	0	0	126	1	19587	06-12-2007 19 11 11
7	192.168.2.249	192.168.0.2	4	5	0	35	61498	0	0	240	1	22099	06-12-2007 19 11 11
8	192.168.2.249	192.168.0.3	4	5	0	35	61501	0	0	240	1	22095	06-12-2007 19 11 11
9	192.168.2.249	192.168.0.4	4	5	0	35	61502	0	0	240	1	22093	06-12-2007 19 11 11
10	192.168.2.249	192.168.0.5	4	5	0	35	61503	0	0	240	1	22091	06-12-2007 19 11 11
11	192.168.2.249	192.168.0.6	4	5	0	35	61504	0	0	240	1	22089	06-12-2007 19 11 11
12	192.168.2.249	192.168.0.7	4	5	0	35	61505	0	0	240	1	22087	06-12-2007 19 11 11
13	192.168.2.249	192.168.0.8	4	5	0	35	61506	0	0	240	1	22085	06-12-2007 19 11 11
14	192.168.2.249	192.168.0.9	4	5	0	35	61507	0	0	240	1	22083	06-12-2007 19 11 11
15	192.168.2.249	192.168.0.10	4	5	0	35	61508	0	0	240	1	22081	06-12-2007 19 11 11
16	192.168.2.249	192.168.0.11	4	5	0	35	61509	0	0	240	1	22079	06-12-2007 19 11 11

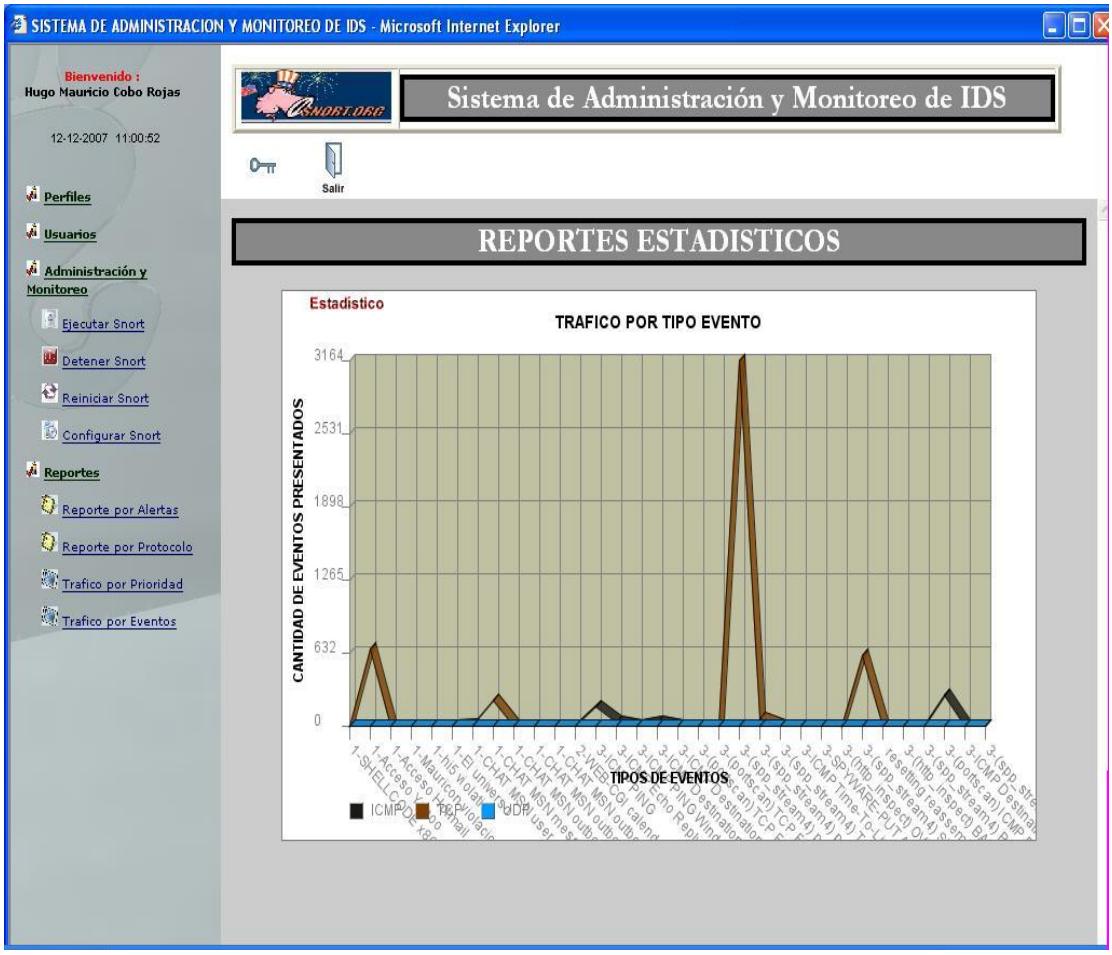
11.3. Trafico por Prioridad

Clic en Reportes por Prioridad, muestra de forma de gráfico estadístico el tráfico que circula por la red.



11.4. Tráfico por Evento

Clic en Reportes por Eventos, muestra de forma de gráfico estadístico el tráfico de eventos que ocurren en la red.



12. Manual Técnico

12.1. Configuración de Usuario Postgres

Configuración de usuario postgres para acceder a la base postgresql

Inicializa un grupo de bases de datos con los siguientes comandos:

```
mkdir -p /var/lib/pgsql/data &&
useradd -d /var/lib/pgsql/data postgres &&
chown postgres /var/lib/pgsql/data &&
su - postgres -c '/usr/bin/initdb -D /var/lib/pgsql/data'
```

12.2. Arranque del Servidor de Base de Datos

Arranca el servidor de bases de datos con el siguiente comando:

```
su - postgres -c '/usr/bin/postmaster -D /var/lib/pgsql/data > \
/var/lib/pgsql/data/logfile 2>&1 &'
```

12.3. Creación de Base de Datos Snort

```
# su - postgres
$ createdb snort          --base snort
$ createuser -P snort     --usuario snort
Enter password for new user: snort
```

Enter it again: snort

Shall the new user be a superuser? (y/n) n

Shall the new user be allowed to create databases? (y/n) n

Shall the new user be allowed to create more new users? (y/n) n

CREATE USER

12.4. Lenguajes Usados para las Bases de Datos

Language: plpgsql

CREATE TRUSTED PROCEDURAL LANGUAGE 'plpgsql'

HANDLER plpgsql_call_handler

VALIDATOR plpgsql_validator;

Language: plsh

CREATE PROCEDURAL LANGUAGE 'plsh'

HANDLER plsh_handler

VALIDATOR plsh_validator;

psql -d snort -f /usr/lib/pgsql/pgplsh-1.2/createlang_pgplsh.sql

psql -d olimpo -f /usr/lib/pgsql/pgplsh-1.2/createlang_pgplsh.sql

12.4.1. Base de Datos Snort (para las alertas y registro del ids)

Creación de la tabla Schema de la Base de Datos Snort

```
CREATE TABLE schema ( vseq       INT4     NOT NULL,
                      ctime      TIMESTAMP with time zone NOT NULL,
                      PRIMARY KEY (vseq));
INSERT INTO schema  (vseq, ctime) VALUES ('107', now());
```

Creación de la tabla Signature de la Base de Datos Snort

```
CREATE TABLE signature ( sig_id          SERIAL NOT NULL,
                         sig_name        TEXT NOT NULL,
                         sig_class_id    INT8,
                         sig_priority    INT8,
                         sig_rev         INT8,
                         sig_sid         INT8,
                         sig_gid         INT8,
                         PRIMARY KEY (sig_id));
CREATE INDEX sig_name_idx ON signature (sig_name);
CREATE INDEX sig_class_idx ON signature (sig_class_id);
```

Creación de la tabla Sig_Reference de la Base de Datos Snort

```
CREATE TABLE sig_reference (sig_id        INT4  NOT NULL,
                            ref_seq       INT4  NOT NULL,
                            ref_id        INT4  NOT NULL,
                            PRIMARY KEY(sig_id, ref_seq));
```

Creación de la tabla Reference de la Base de Datos Snort

```
CREATE TABLE reference (    ref_id            SERIAL,
                            ref_system_id     INT4 NOT NULL,
                            ref_tag           TEXT NOT NULL,
                            PRIMARY KEY (ref_id));
```

Creación de la tabla Referente_System de la Base de Datos Snort

```
CREATE TABLE reference_system (    ref_system_id      SERIAL,
                                   ref_system_name    TEXT,
                                   PRIMARY KEY (ref_system_id));
```

Creación de la tabla Sig_Class de la Base de Datos Snort

```
CREATE TABLE sig_class (    sig_class_id       SERIAL,
                            sig_class_name     TEXT NOT NULL,
```

```
                                PRIMARY KEY (sig_class_id) );
CREATE INDEX sig_class_name_idx ON sig_class (sig_class_name);
```

Creación de la tabla Event de la Base de Datos Snort

```
CREATE TABLE event  (  sid          INT4 NOT NULL,
                       cid          INT8 NOT NULL,
                       signature    INT4 NOT NULL,
                       timestamp    timestamp with time zone NOT NULL,
                       PRIMARY KEY (sid,cid));
CREATE INDEX signature_idx ON event (signature);
CREATE INDEX timestamp_idx ON event (timestamp);
```

Se detalla el sensor en uso.

Creación de la tabla Sensor de la Base de Datos Snort

```
CREATE TABLE sensor (  sid          SERIAL,
                       hostname     TEXT,
                       interface    TEXT,
                       filter       TEXT,
                       detail       INT2,
                       encoding     INT2,
                       last_cid     INT8 NOT NULL,
```

```
PRIMARY KEY (sid));
```

Todas las descripciones de cabeceras de IP

Creación de la tabla IPHDR de la Base de Datos Snort

```
CREATE TABLE iphdr  (   sid          INT4 NOT NULL,
                        cid          INT8 NOT NULL,
                        ip_src       INT8 NOT NULL,
                        ip_dst       INT8 NOT NULL,
                        ip_ver       INT2,
                        ip_hlen      INT2,
                        ip_tos       INT2,
                        ip_len       INT4,
                        ip_id        INT4,
                        ip_flags     INT2,
                        ip_off       INT4,
                        ip_ttl       INT2,
                        ip_proto     INT2 NOT NULL,
                        ip_csum      INT4,
                        PRIMARY KEY (sid,cid));
```

```
CREATE INDEX ip_src_idx ON iphdr (ip_src);
CREATE INDEX ip_dst_idx ON iphdr (ip_dst);
```

Todas las descripciones de caberas de tcp

Creación de la tabla TCPHDR de la Base de Datos Snort

```
CREATE TABLE tcphdr(  sid          INT4 NOT NULL,
                      cid          INT8 NOT NULL,
                      tcp_sport    INT4 NOT NULL,
                      tcp_dport    INT4 NOT NULL,
                      tcp_seq      INT8,
                      tcp_ack      INT8,
                      tcp_off      INT2,
                      tcp_res      INT2,
                      tcp_flags    INT2 NOT NULL,
                      tcp_win      INT4,
                      tcp_csum     INT4,
                      tcp_urp      INT4,
                      PRIMARY KEY (sid,cid));
```

```
CREATE INDEX tcp_sport_idx ON tcphdr (tcp_sport);
CREATE INDEX tcp_dport_idx ON tcphdr (tcp_dport);
CREATE INDEX tcp_flags_idx ON tcphdr (tcp_flags);
```

Todas las descripciones de caberas de udp

Creación de la tabla UDOHDRde la Base de Datos Snort

```
CREATE TABLE udphdr(  sid          INT4 NOT NULL,
                      cid          INT8 NOT NULL,
                      udp_sport    INT4 NOT NULL,
                      udp_dport    INT4 NOT NULL,
                      udp_len      INT4,
                      udp_csum     INT4,
                      PRIMARY KEY (sid,cid));

CREATE INDEX udp_sport_idx ON udphdr (udp_sport);
CREATE INDEX udp_dport_idx ON udphdr (udp_dport);
```

Todas las descripciones de caberas de icmp

Creación de la tabla ICMPHDR de la Base de Datos Snort

```
CREATE TABLE icmphdr( sid          INT4 NOT NULL,
                      cid          INT8 NOT NULL,
                      icmp_type    INT2 NOT NULL,
                      icmp_code    INT2 NOT NULL,
                      icmp_csum    INT4,
                      icmp_id      INT4,
                      icmp_seq     INT4,
                      PRIMARY KEY (sid,cid));

CREATE INDEX icmp_type_idx ON icmphdr (icmp_type);
```

Opciones de Protocolos

Creación de la tabla OPT de la Base de Datos Snort

```
CREATE TABLE opt   (  sid          INT4 NOT NULL,
                      cid          INT8 NOT NULL,
                      optid        INT2 NOT NULL,
                      opt_proto    INT2 NOT NULL,
```

```
opt_code      INT2 NOT NULL,
opt_len       INT4,
opt_data      TEXT,
PRIMARY KEY (sid,cid,optid));
```

Paquete payload

Creación de la tabla DATA de la Base de Datos Snort

```
CREATE TABLE data   (  sid            INT4 NOT NULL,
                       cid            INT8 NOT NULL,
                       data_payload   TEXT,
                       PRIMARY KEY (sid,cid));
```

Creación de la tabla ENCODING de la Base de Datos Snort

```
CREATE TABLE encoding(   encoding_type   INT2 NOT NULL,
                         encoding_text   TEXT NOT NULL,
                         PRIMARY KEY (encoding_type));
INSERT INTO encoding (encoding_type, encoding_text) VALUES (0, 'hex');
INSERT INTO encoding (encoding_type, encoding_text) VALUES (1,
'base64');
```

INSERT INTO encoding (encoding_type, encoding_text) VALUES (2, 'ascii');

Detalla los diferentes niveles de alertas que se generan

Creación de la tabla DETAIL de la Base de Datos Snort

```
CREATE TABLE detail  (   detail_type INT2 NOT NULL,
                         detail_text TEXT NOT NULL,
                         PRIMARY KEY (detail_type));
INSERT INTO detail (detail_type, detail_text) VALUES (0, 'fast');
INSERT INTO detail (detail_type, detail_text) VALUES (1, 'full');
```

12.4.2. Permisos de las Tablas para el Usuario Snort

- ✓ GRANT SELECT ON detail, encoding, event, reference_system, reference, schema,sensor, sig_class, sig_reference, signature TO snort;
- ✓ GRANT INSERT ON data, event, icmphdr, iphdr, opt, reference,reference_system, sensor, sig_class, sig_reference,signature, tcphdr, udphdr TO snort;

  GRANT UPDATE ON reference_ref_id_seq TO snort;
- ✓ GRANT UPDATE ON reference_system_ref_system_id_seq TO snort;
- ✓ GRANT UPDATE ON sensor_sid_seq TO snort;

- ✓ GRANT UPDATE ON sig_class_sig_class_id_seq TO snort;
- ✓ GRANT UPDATE ON signature_sig_id_seq TO snort;

12.4.3. Funciones Creadas para la Base Snort

CREATE FUNCTION plpgsql_call_handler () RETURNS OPAQUE AS '/usr/lib/pgsql/plpgsql.so' LANGUAGE 'C';

Note: remember to change the above path to 'plpgsql.so'

-Usando el lenguaje plpgsql se crea un procedimiento de llamado para la función.

CREATE TRUSTED PROCEDURAL LANGUAGE 'plpgsql' HANDLER plpgsql_call_handler
LANCOMPILER 'PL/pgSQL';

-Esta función me retorna la ip de manera entera y con formato ejemplo '192.168.2.1'

```
CREATE FUNCTION convertir_ip(INT8) RETURNS varchar AS $$
DECLARE    t varchar;
BEGIN
   t:= (($1>>24) & 255::INT8) || '.' ||
```

```
      (($1>>16) & 255::INT8) || '.' ||
      (($1>>8)  & 255::INT8) || '.' ||
      ($1     & 255::INT8);
  return  t; END;
$$ LANGUAGE plpgsql;
```

-Función que recibiendo una cadena hexadecimal me retorna código entero

```
CREATE FUNCTION hexa_decimal2(p_my_hex TEXT) RETURNS int4 AS
$$
DECLARE
        my_hex ALIAS for $1;
        my_hex TEXT;
        my_hex_str TEXT;
        my_digit TEXT;
         my_int INT4;
        ii INT4;
        RES INT4;
BEGIN
        my_int:=0;
        my_hex_str:= '';
        my_hex_str:= p_my_hex;
        my_int := 0;
```

```
            ii := 1;
   WHILE length(my_hex_str) > 0
    LOOP
            my_digit := substr(my_hex_str, length(my_hex_str));
         IF my_digit = 'A' THEN my_digit := '10';
         ELSE IF my_digit = 'B' THEN my_digit := '11';
         ELSE IF my_digit = 'C' THEN my_digit := '12';
         ELSE IF my_digit = 'D' THEN my_digit := '13';
         ELSE IF my_digit = 'E' THEN my_digit := '14';
         ELSE IF my_digit = 'F' THEN my_digit := '15';
         END IF; END IF; END IF; END IF; END IF; END IF;
         my_hex_str :=  substr(my_hex_str, 1, length(my_hex_str) - 1);
         my_int := my_int + (my_digit::INT4) * ii;
         ii := ii  * 16;
     END LOOP;
        if my_int=10 then
            my_int:=32;   --para evitar el salto de linea lo reemplazamos por
espacio ' '
           end if;
           RES:=my_int;
  RETURN RES;--my_int;
 END;
```

```
$$ LANGUAGE plpgsql
```

- Función que recibiendo una cadena me retorna código ASCII

```
CREATE FUNCTION convertir_ascii2(pv_var TEXT) RETURNS TEXT AS $$
DECLARE
        respuesta TEXT[];
          cadena TEXT;
          x int4:=0;
          y int4:=0;
          longuitud INT4:=0;
          z INT4:=0;
BEGIN
          longuitud:=length(pv_var)/2;
          x:=2;
          y:=1;
          z:=1;
          cadena:='';
--FOR n IN 1..longuitud  LOOP
WHILE z <=longuitud
   LOOP
   --respuesta[z]:= chr(hexa_decimal(substring(pv_var from y for x)));
```

```
   respuesta[z]:= chr(hexa_decimal2(substr(pv_var,y,2)));
   cadena:= cadena || respuesta[z];
    y:=y+2;
    x:=x+2;
    z:=z+1;
END LOOP;
RETURN cadena;
END;
$$ LANGUAGE plpgsql
```

12.4.4. Creación de Base de Datos Olimpo (para accesos de usuarios al sistema)

```
# su - postgres
$ createdb olimpo          --base olimpo
$ createuser -P olimpo      --usuario olimpo
Enter password for new user: olimpo
Enter it again: snort-password
Shall the new user be a superuser? (y/n) n
Shall the new user be allowed to create databases? (y/n) n
Shall the new user be allowed to create more new users? (y/n) s
CREATE USER
grant all privileges on database olimpo to olimpo;
```

-Tabla: Persona (las persona que podrán asignarles usuario y perfil)

```
CREATE TABLE persona
(
  id_persona            int4 NOT NULL,
  nombre                varchar(20) NOT NULL,
  apellido              varchar(20) NOT NULL,
  sexo                  char(1) NOT NULL,
  fecha_registro        date NOT NULL,
  fecha_modificacion    date,
  estado                char(1) NOT NULL,
  telefono              varchar(9),
  CONSTRAINT persona_pkey PRIMARY KEY (id_persona)
)
WITHOUT OIDS;
ALTER TABLE persona OWNER TO postgres;
GRANT ALL ON TABLE persona TO postgres;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE persona TO
olimpo;
```

-Tabla: Usuario (usuarios sujetos de a perfil)

```
CREATE TABLE usuario
(
  id_usuario                int4 NOT NULL,
  id_persona                int4 NOT NULL,
  id_perfil                 int4 NOT NULL,
  "login"                   varchar(50) NOT NULL,
  "password"                varchar(150) NOT NULL,
  email                     varchar(50) NOT NULL,
  clave_                    email varchar(50),
  fecha_registro            date NOT NULL,
  fecha_modificacion         date,
  estado                    char(1) NOT NULL,
 CONSTRAINT usuario_pkey PRIMARY KEY (id_usuario, id_persona, id_perfil),
 CONSTRAINT usuario_fk1 FOREIGN KEY (id_perfil)
 REFERENCES perfil (id_perfil) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT usuario_fk2 FOREIGN KEY (id_persona)
REFERENCES persona (id_persona) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
```

```
)
WITHOUT OIDS;
ALTER TABLE usuario OWNER TO postgres;
GRANT ALL ON TABLE usuario TO postgres;
GRANT SELECT, UPDATE, INSERT, DELETE ON TABLE usuario TO olimpo;
```

-Tabla: Perfil (perfiles que podrá validar el acceso a su uso)

```
CREATE TABLE perfil
(
  id_perfil                 int4 NOT NULL,
  nombre                    varchar(20) NOT NULL,
  descripcion               varchar(20),
  fecha_registro            date NOT NULL,
  fecha_modificacion        date,
  estado                    char(1) NOT NULL,
  CONSTRAINT perfil_pkey PRIMARY KEY (id_perfil)
)
WITHOUT OIDS;
ALTER TABLE perfil OWNER TO postgres;
GRANT ALL ON TABLE perfil TO postgres;
```

```
GRANT SELECT, UPDATE, INSERT ON TABLE perfil TO olimpo;
```

-Tabla: Modulo

```
CREATE TABLE modulo (tipos de módulos existentes en la aplicación)
(
  id_modulo                    serial NOT NULL,
  desc_modulo                  varchar(40) NOT NULL,
  desc_completa_modulo         varchar(80) NOT NULL,
  CONSTRAINT modulo_pk PRIMARY KEY (id_modulo)
)
WITHOUT OIDS;
ALTER TABLE modulo OWNER TO olimpo;
```

-Tabla: Modulo_Detalle (Detalle de cada módulo existente)

```
CREATE TABLE modulo_detalle
(
  id_modulo                int4 NOT NULL,
  id_modulo_detalle        int4 NOT NULL,
  descripcion              varchar(80) NOT NULL,
```

```
  ruta                          varchar(150) NOT NULL,
CONSTRAINT modulo_detalle_pk PRIMARY KEY (id_modulo,
id_modulo_detalle),
CONSTRAINT modulo_detalle_fk1 FOREIGN KEY (id_modulo)
REFERENCES modulo (id_modulo) MATCH SIMPLE
ON UPDATE NO ACTION ON DELETE NO ACTION
)
WITHOUT OIDS;
ALTER TABLE modulo_detalle OWNER TO olimpo;
```

-Table: Roles_Modulos (roles de perfil por módulo asignado)

```
CREATE TABLE roles_modulos
(
  id                    serial NOT NULL,
  id_modulo             int4 NOT NULL,
  id_modulo_detalle int4 NOT NULL,
  id_perfil              int4 NOT NULL
)
WITHOUT OIDS;
ALTER TABLE roles_modulos OWNER TO olimpo;
```

-Tabla: Reglas (reglas que van a ser usadas por el ids para configuración)

```
CREATE TABLE reglas
(
  codigo          serial NOT NULL,
  nombre          varchar(40) NOT NULL,
  descripcion     varchar(500) NOT NULL,
  cargar          int4,
  CONSTRAINT reglas_pk PRIMARY KEY (codigo, nombre)
)
WITHOUT OIDS;
ALTER TABLE reglas OWNER TO olimpo;
GRANT ALL ON TABLE reglas TO olimpo;
```

12.4.5. Reglas del Snort

insert into reglas (nombre,descripcion,cargar)values ('icmp.rules','Estas reglas contrarrestan tráfico malicioso ICMP. Ellas incluyen muestras del escaneo de herramientas ICMP y de otros tráficos malicio cd ICMP(Such as redirect host)Otras reglas ICMP son incluidas en icmp-info.rules',1);

insert into reglas (nombre,descripcion,cargar)values ('icmp-info.rules','Estas normas son estándar de área local. Incluyen software para hacer ping, así como la normalidad de enrutamiento realizada por ICMP. Hay una serie de captura todas las normas que se alerta sobre los tipos ICMP desconocidos',1);

insert into reglas (nombre,descripcion,cargar)values ('attack-responses.rules','Estas firmas son la que usualmente cuando tu máquina ha quedado en entredicho o especie como de compromiso a responder algo',1);

insert into reglas (nombre,descripcion,cargar)values ('bad-traffic.rules','Estas firmas son representativas de tráfico que nunca deben ser vistas en cualquier red. Ninguna de estas firmas incluyen data grama de contenido y son extremadamente firmas rápidas',1);

insert into reglas (nombre,descripcion,cargar)values ('chat.rules','Estas firmas buscan personas que utilizan varios tipos de programas de Chat (por ejemplo: AIM, ICQ, e IRC), que puede ser en contra de la política corporativa',1);

insert into reglas (nombre,descripcion,cargar)values ('experimental.rules','Estas firmas son experimentales, y puede desencadenar nuevas formas a menudo. Su forwarned, este es nuestro banco de pruebas. Ponemos nuevas firmas aquí para probar antes de incorporarlas en el conjunto de firmas por defecto. Esto es para el desbordamiento solamente',1);

insert into reglas (nombre,descripcion,cargar)values ('ftp.rules','Ayuda a contrarrestar: comandos de ftp maliciosos, Alerta TCP, Malos directorios, Vulnerabilidades específicas contra implementaciones de ftp, Archivos maliciosos, Sospechosos intentos de acceso, Protocolo de verificación',1);

insert into reglas (nombre,descripcion,cargar)values ('oracle.rules','Detecta tráfico malicioso de oracle, esta regla no esta habilitada por defecto, ya que pueden generar falsas alarmas en las redes q se desarrolla bajo plataforma oracle, si se usa una aplicación Web basada en oracle, debe configurar e puerto de destino para capturar a los atacantes que intentan explotar su aplicación.',1);

insert into reglas (nombre,descripcion,cargar)values ('local.rules','Este archivo intencionalmente no viene con las firmas. Ponga sus reglas locales adicionales aquí.',1);

insert into reglas (nombre,descripcion,cargar)values ('multimedia.rules','Estas firmas buscan personas que utilizan las tecnologías multimedia. Uso de medios puede ser una violación de las políticas corporativas',1);

insert into reglas (nombre,descripcion,cargar)values ('mysql.rules','Estas firmas detectan potencialmente tráfico dañino mysql. Estas firmas no están habilitadas por defecto, ya que pueden generar falsas alarmas en las redes que se desarrolla en mysql.',1);

insert into reglas (nombre,descripcion,cargar)values ('other-ids.rules','Estas firmas servir a dos propósitos. 1) Si está "IDS GUY" para una empresa, y la otra persona se establece un IDS sin permitir que usted sabe, éste es malo.

2) Si está "pen-tester", esta es una buena manera de descubrir qué sistemas IDS su objetivo es utilizar después de haber tenido acceso a su red.',1);

insert into reglas (nombre,descripcion,cargar)values ('other-ids-1.rules','Contiene información que refuerza a la regla other-ids.rules',1);

insert into reglas (nombre,descripcion,cargar)values ('p2p.rules','Estas firmas se encargan contrarrestar daño por el uso del P2P protocolos, que son por lo general en contra de la política corporativa',1);

insert into reglas (nombre,descripcion,cargar)values ('rpc.rules','Firmas para contrarestar daño en el llamado a procesamiento remoto',1);

insert into reglas (nombre,descripcion,cargar)values ('scan.rules','Estas firmas son para contrarrestar escáneres de red. Estos incluyen puerto de escaneo, la dirección ip de cartografía, y varios escáneres aplicación. NOTA: Esto NO incluye la Web, tales como escáneres bigote. Esos son en la Web',1);

insert into reglas (nombre,descripcion,cargar)values ('shellcode.rules','Estas firmas se basan en el shellcode que es común encontrar múltiples opciones de configuración a disposición del público.',1);

insert into reglas (nombre,descripcion,cargar)values ('telnet.rules','Estas firmas se basan en diversas opciones de telnet y protegidas por contraseñas de cuentas.',1);

insert into reglas (nombre,descripcion,cargar)values ('tftp.rules','Estas firmas se basan en el tráfico de TFTP. Contrarrestan archivos maliciosos que son distribuidos a través de TFTP.',1);

insert into reglas (nombre,descripcion,cargar)values ('virus.rules','Esta regla busca cualquiera de los siguientes tipos de archivo adjunto: ade, adp, asd, asf, asx, bat, chm, cli, cmd, com, cpp, diz, dll, dot, emf, eml, exe, hlp, hsq, hta, ini, js, jse, lnk, mda, mdb, mde, mdw, msi, msp, nws, ocx, pif, pl, pm, pot, pps, ppt, reg, rtf, scr, shs, swf, sys, vb, vbe, vbs, vcf, vxd, wmd, wmf, wms, wmz, wpd, wpm, wps, wpz, wsc, wsf, wsh, xlt, xlw.',1);

insert into reglas (nombre,descripcion,cargar)values ('virus-1.rules','Más definiciones que ayudan a reforzar a virus.rules',1);

insert into reglas (nombre,descripcion,cargar)values ('web-attacks.rules','Ellos se basan en que las firmas genéricas comunes de los comandos de captura emitida variable de la forma de explotar la

vulnerabilidad, pero es tan trivial para evadir estos que da al usuario una falsa sensación de seguridad en ataques via web.',1);

insert into reglas (nombre,descripcion,cargar)values ('web-client.rules','Estas firmas buscan dos cosas: Cosas malas vienen de nuestros usuarios y los ataques contra los usuarios de nuestra web',1);

insert into reglas (nombre,descripcion,cargar)values ('content-replace.rules','Contrarrestan definiciones que tienen la capacidad de cambiar o alterar el contenido de un archivo',1);

insert into reglas (nombre,descripcion,cargar)values ('ddos.rules','Contrarresta definiciones cuando existe eventos de generación de intentos por parte de un host para comunicarse con un Tribal Flood Network (TFN) DDoS cliente.',1);

insert into reglas (nombre,descripcion,cargar)values ('backdoor.rules','Combate las definiciones del acceso no permitido por otro sitio, es una secuencia especial dentro del código de programación mediante la cual el programador puede acceder o escapar de un programa en caso de emergencia o contingencia en algún problema.',1);

insert into reglas (nombre,descripcion,cargar)values ('deleted.rules','Contrarresta definiciones a eventos que se generan cuando la actividad de la Dagger troyano se detecta en el tráfico de la red.',1);

insert into reglas (nombre,descripcion,cargar)values ('dos.rules','Combate definiciones a eventos que se generan cuando un atacante remoto transmite paquetes fragmentados IGMP con errores de las cabeceras de la red interna, lo que indica un IGMP de Denegación de Servicio (DoS) de ataque.',1);

insert into reglas (nombre,descripcion,cargar)values ('dns.rules','Combate definiciones a eventos que se generan cuando se trata de solicitar una transferencia de zona de un servidor DNS',1);

insert into reglas (nombre,descripcion,cargar)values ('exploit.rules','Combate definiciones para las versiones del software de intercambio de archivos y anteriores contienen una condición de desbordamiento de búfer que puede ser explotado mediante el suministro de una intervención excesivamente larga la contraseña de servidores',1);

insert into reglas (nombre,descripcion,cargar)values ('finger.rules','Combate definiciones a eventos que se generan cuando el acceso a un conocido de backdoor desplegadas por atacantes se intenta. En este caso, puede ser una conexión con un Trojaned versión de fingerd.',1);

insert into reglas (nombre,descripcion,cargar)values ('imap.rules','Combate definiciones a eventos que se generan cuando un atacante remoto envía un argumento excesivamente largo en el AUTHENTICATE comando interno a un servidor IMAP, lo que indica un intento de explotar una vulnerabilidad de desbordamiento de búfer. Esto también puede afectar a otras implementaciones del servidor IMAP.',1);

insert into reglas (nombre,descripcion,cargar)values ('info.rules','Combate definiciones a eventos que se generan cuando una conexión está cerrada a partir de un recurso externo a la red protegida.',1);

insert into reglas (nombre,descripcion,cargar)values ('misc.rules','Combate definiciones a eventos que se generan cuando se descubrió un paquete con suelta de enrutamiento de origen establecidas en las opciones IP.',1);

insert into reglas (nombre,descripcion,cargar)values ('netbios.rules','Combate definiciones a eventos que se generan cuando se trata de emitir una denegación de servicio (DoS) en contra de un ataque de acogida utilizando el RFPoison herramienta.',1);

insert into reglas (nombre,descripcion,cargar)values ('nntp.rules','Combate definiciones a eventos que se generan cuando se hace un intento de explotar una conocida vulnerabilidad en el servidor.',1);

insert into reglas (nombre,descripcion,cargar)values ('policy.rules','Combate definiciones a eventos que se generan cuando el tráfico de la red indica la utilización de una aplicación o servicio que pueda violar la política de seguridad corporativa.',1);

insert into reglas (nombre,descripcion,cargar)values ('pop2.rules','Combate definiciones a eventos que se generan cuando se hace un intento de explotar un desbordamiento de búfer en el pop2 servicio.',1);

insert into reglas (nombre,descripcion,cargar)values ('pop3.rules','Combate definiciones a eventos que se generan cuando se hace un intento de explotar un desbordamiento de búfer en el POP3 qpopper servicio en sistemas BSD.',1);

insert into reglas (nombre,descripcion,cargar)values ('rservices.rules','Combate definiciones a eventos que se generan cuando se hace un intento de explotar una máquina utilizando Servicios de Información de Red (NIS).',1);

insert into reglas (nombre,descripcion,cargar)values ('smtp.rules','Combate definiciones a eventos que se generan un desbordamiento de búfer cuando se intenta en un Sendmail servidor.',1);

insert into reglas (nombre,descripcion,cargar)values ('snmp.rules','Combate definiciones a eventos que se generan cuando una conexión SNMP sobre UDP utilizando el método de "público" se hace comunidad.',1);

insert into reglas (nombre,descripcion,cargar)values ('snmp-1.rules','Refuerza las seguridades para smto.rules',1);

insert into reglas (nombre,descripcion,cargar)values ('sql.rules','Combate definiciones a eventos que se generan cuando se hace un intento de explotar una conocida vulnerabilidad en Microsoft SQL',1);

insert into reglas (nombre,descripcion,cargar)values ('voip.rules','Contrarrestan definiciones a eventos que se generan cuando el tráfico de la red indica que el cliente Gizmo VoIP se está utilizando.',1);

insert into reglas (nombre,descripcion,cargar)values ('web-cgi.rules','Combate definiciones a eventos que se generan cuando se hace un intento de explotar una conocida vulnerabilidad en una aplicación web CGI se ejecuta en un servidor',1);

insert into reglas (nombre,descripcion,cargar)values ('web-coldfusion.rules','Combate definiciones a eventos que se generan cuando se hace un intento de explotar una conocida vulnerabilidad en un servidor Web de ColdFusion.',1);

insert into reglas (nombre,descripcion,cargar)values ('web-frontpage.rules','Combate definiciones a eventos que se generan cuando se hace un intento de explotar una conocida vulnerabilidad en un servidor Web con las Extensiones de servidor de Microsoft FrontPage.',1);

insert into reglas (nombre,descripcion,cargar)values ('web-misc.rules','Combate definiciones a eventos que se generan cuando se intenta hacer referencia a una. Bate el archivo a ejecutar comandos arbitrarios en un Servicios de Internet Information Server (IIS) del servidor.',1);

insert into reglas (nombre,descripcion,cargar)values ('web-php.rules','Combate definiciones a eventos que se generan cuando se hace un intento de explotar una conocida vulnerabilidad en una aplicación Web corriendo PHP en un servidor.',1);

insert into reglas (nombre,descripcion,cargar)values ('web-iis.rules','Combate definiciones a eventos que se generan cuando se hace un intento de provocar una denegación de servicio del Servicio de Publicaciones y WWW IIS Administración de software',1);

insert into reglas (nombre,descripcion,cargar)values ('x11.rules','Combate definiciones a eventos que se generan cuando un intento de explotar una debilidad en el mecanismo de autenticación utilizado para conectarse a un servidor de Windows X',1);

insert into reglas (nombre,descripcion,cargar)values ('spyware-put.rules','Combate definiciones a eventos que se generan cuando la actividad relativa a una solicitud del spyware es detectado.',1);

insert into reglas (nombre,descripcion,cargar)values ('specific-threats.rules','Definiciones especificas para contrarrestar archivos maliciosos tales como: netsky.p, yarner.b, mydoom.e, mimail.a (e), lovgate.c, deborm.x (x,y,q,u,r), klez.d (e), etc ',1);

insert into reglas (nombre,descripcion,cargar)values ('porn.rules','Definiciones específicas que ayudan a controlar el acceso a lugares no autorizados establecidos en esta regla.',1);

12.4.6. Secuencias Creadas

-Secuencia: modulo_id_modulo_seq

```
CREATE SEQUENCE modulo_id_modulo_seq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 3
  CACHE 1;
ALTER TABLE modulo_id_modulo_seq OWNER TO olimpo;
```

-Secuencia: reglas_codigo_seq

```
CREATE SEQUENCE reglas_codigo_seq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 55
  CACHE 1;
ALTER TABLE reglas_codigo_seq OWNER TO olimpo;
```

-Secuencia: roles_modulos_id_seq

```
CREATE SEQUENCE roles_modulos_id_seq
  INCREMENT 1
  MINVALUE 1
  MAXVALUE 9223372036854775807
  START 15
  CACHE 1;
ALTER TABLE roles_modulos_id_seq OWNER TO olimpo;
```

12.5. Principales Clases

```
package administracion;
import java.io.*;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.Vector;
import javax.servlet.jsp.PageContext;
import biblioteca.Bean;
```

```
/*Clase que permite configurar opciones del IDS SNORT*/
public class AD_Configura_snort extends Bean{
    /*Constructor que indica que se va a acceder a la Base snort */
    public AD_Configura_snort(PageContext p_pageContext) throws
    Exception, SQLException
    {
        super(p_pageContext);
    }
    /*Constructor que indica que se va a acceder a la Base Olimpo */
    public AD_Configura_snort(PageContext p_pageContext,String base)
    throws Exception, SQLException
    {
        super(p_pageContext,"olimpo");
    }
    public AD_Configura_snort() throws Exception, SQLException
    {
        super();
    }
```

```
/*Funcion que se encarga de eliminar el sensor actual del IDS
 Esto es necesario en cada reinicio del IDS  */
public int registrarsensor() throws Throwable {
                String ls_sql ="delete from  sensor";
                PreparedStatement unPs =null;
                try{
                unPs = gpuc_conexion.prepareStatement(ls_sql);
                unPs.executeUpdate();
                gpuc_conexion.commit();
                }//fin del try
                catch(Exception ex){
                gpuc_conexion.rollback();
                System.out.println(ex.toString());
                throw(ex);


                }
        finally
        {
                if(unPs != null)unPs.close();
        }//fin del finally
        return 1;
}
```

```
/*Funcion que se encarga de leer las reglas que fueron
  configuradas por el asistente de configuracion.
  Retorno : Un Vector con las reglas configuradas */
 public Vector leerReglas() throws Throwable {
                Vector valorDevolver = new Vector();
                String ls_sql ="select nombre from reglas "+
                "where cargar=1";
                PreparedStatement unPs =null;
                ResultSet                unRs =null;
                try{
                unPs  = gpuc_conexion.prepareStatement(ls_sql);
                unRs  = unPs.executeQuery();
                while (unRs.next()){
                valorDevolver.addElement(unRs.getString(1).trim());
                        }
                        }//fin del try
                catch(Exception ex){
                        System.out.println(ex.toString());
                        throw(ex);
                        }
                finally
                {
```

```
                if(unRs != null)unRs.close();
                if(unPs != null)unPs.close();
            }//fin del finally
            return valorDevolver;
        }
```

/*Funcion que se encarga actualizarlas reglas que fueron disponibles para el asistente de configuracion. */

```
public void actualizarReglas(String pv_sql) throws Throwable {
            String ls_sql  =pv_sql;
            PreparedStatement  unPs =null;
            try{
                unPs  =  gpuc_conexion.prepareStatement(ls_sql);
                nPs.executeUpdate();
                gpuc_conexion.commit();
                }//fin del try
            catch(Exception ex){
                gpuc_conexion.rollback();
                System.out.println(ex.toString());
                throw(ex);
                }
```

```
            finally
            {
                if(unPs != null)unPs.close();
            }//fin del finally
        }
```

/*Funcion que permite leer un Archivo en Bytes

Retorna: Un String con el contenido del archivo. */

```
public   String regresaArchivo(String Archivo){
        try{
            // abrirlo
            RandomAccessFile raf=new
            RandomAccessFile(Archivo,"r");
            byte[] b=new byte[(int)raf.length()];
            // leerlo en un array de bytes
            raf.readFully(b);
            // eliminar los enters por un codigo propio
            for(int i=0;i<raf.length();i++){
                if((int)b[i] == 13)
                    b[i]=(byte)'~';
                else if((int)b[i] == 10)
```

```
                    b[i]=(byte)'%';
            }
            raf.close();
            //convertir el array de bytes a cadena
            return new String(b);
        }
        catch (IOException e){
        System.out.println(e+"->" +e.getMessage());
            return "";
        }
}
```

/*Funcion que Construye el archivo :snort_reglas en base a las reglas configuradas, archivo que luego reemplaza al snort.conf */

```
public void cargarReglas(String ls_cabecera,String ls_fin,Vector lv_reglas){
BufferedWriter bw = null ;
        try{
            bw = new BufferedWriter(new
            FileWriter("/etc/snort/snort_reglas.conf"));
```

```
}
catch (IOException e){
        System.out.println(e+"->" +e.getMessage());
}
        if(ls_cabecera.length() <1){
        System.out.println("El archivo no existe en el
        servidor o esta vacio\n");
        return;
}
byte arch[] = ls_cabecera.getBytes();
// regresar los retorno de carro y los avance de linea
        for(int i=0;i<arch.length;i++){
                if((int)arch[i] == (byte)'~')
                        arch[i]=(byte)13;
                else if((int)arch[i] == (byte)'%')
                        arch[i]=(byte)10;
        }
        byte archfin[] = ls_fin.getBytes();
        // regresar los retorno de carro y los avance de
        linea
        for(int i=0;i<archfin.length;i++){
                if((int)archfin[i] == (byte)'~')
```

```
                                        archfin[i]=(byte)13;
                                    else if((int)archfin[i] == (byte)'%')
                                        archfin[i]=(byte)10;
                                }
            try{
                bw.write(new String (arch));

                for (int i=0; i<lv_reglas.size(); i++)
                {
                    bw.write("include
                    $RULE_PATH/"+lv_reglas.elementAt(i)+"\n");
                }
                bw.write("\n");
                bw.write(new String (archfin));
                bw.flush();
                bw.close();
                }
                catch (IOException e){
                    System.out.println("No Se creo el archivo");
                }
}
```

/*Funcion que Configura el script sendmail para los tipos de alertas que se enviaran al administrador de la red */

```
public void archivoAlarmas(Vector telefonos,String opt_mail,String opt_telefonos){
        try{
                FileReader fr=new FileReader("/sendmail");
                FileWriter fw=new FileWriter("/sendmail_alarmas");
                BufferedReader entrada=new BufferedReader(fr);
                BufferedWriter bw =new BufferedWriter(fw);
                PrintWriter salida=new PrintWriter(bw);
                String s="";
                String ls_email="guardian@memorex.com.ec";
                String ls_telefonos="";
                String ls_fromcel="";
                while(!(s=entrada.readLine()).equals("# mensajes
                internos"))
                {
                        salida.println(s);
                        //System.out.println(s);
                }
                salida.println("# mensajes internos");
                if(opt_mail.equals("1"))
```

```
{
        salida.println("if [ $prioridad -gt 2 ]; then");
        salida.println("mail -s \"Alerta\"  "+ls_email+" <
        /mensajemail");
        salida.println("fi");
            }
else
        salida.println("mail -s \"Alerta\"  "+ls_email+" <
        /mensajemail");
if(telefonos!=null)
{
for(int i=0;i<telefonos.size();i++)
{
        ls_telefonos=(String)telefonos.elementAt(i);
        if(ls_telefonos.substring(1,2).equals("9"))
        ls_telefonos=ls_telefonos.substring(1);
ls_telefonos=ls_telefonos.substring(1)+"@portafree.com";
ls_fromcel=ls_fromcel+"mail -s \"Alerta\" "+ls_telefonos+" <
/mensajecel"+"\n";
}
}
salida.println("# mensajes celular");
```

```
                if(opt_telefonos.equals("1"))
                {
                        salida.println("if [ $prioridad -gt 2 ]; then");
                        salida.println(ls_fromcel);
                        salida.println("fi");
                }
                else
                salida.println(ls_fromcel);
                entrada.close();
                salida.close();
                fw.close();
        }
        catch(Exception e)
        {
        System.out.println("JAVA::AD_Configura_snort::"+e.toString());
        }
}

    public void cerrarBeanConexion() {
        try{
                closeConexion();
        }catch(Throwable e){
```

```
                    //System.out.println(e);
            }
}


/*Metodo main utilizado para pruebas rapidas*/
public  static void main(String[] args) {
        /*Vector reglas=new Vector();
        reglas.addElement("regla 1");
        reglas.addElement("regla 2");
        reglas.addElement("regla 3");
        try{
        AD_Configura_snort snort =new AD_Configura_snort();
        snort.cargarReglas(snort.regresaArchivo("/etc/snort/snort_cab.conf"),s
nort.regresaArchivo("/etc/snort/snort_fin.conf"),reglas);
        }catch(Exception e)
        {}*/
        try{
                Vector telefonos=new Vector();
                telefonos.addElement("088976340");
                telefonos.addElement("088759893");
                telefonos.addElement("091234567");
```

```
        AD_Configura_snort snort =new AD_Configura_snort();
        snort.archivoAlarmas(telefonos,"1","1");
        }catch(Exception e)
        {}
        }

package biblioteca;
import java.sql.*;
import javax.servlet.jsp.*;
import javax.servlet.http.*;
import biblioteca.JspLib;
/*import javax.naming.Context;
import javax.naming.InitialContext;
import javax.sql.DataSource;
*/

/*Clase que contiene el Bean de Conexion a las bases de datos */
public class Bean
{
  public JspWriter m_out;
  public PageContext m_pageContext;
  public HttpServletResponse m_response;
```

```
public Connection gpuc_conexion=null;;
public HttpServletRequest m_request;
public String conexion_error ="";
protected JspLib m_jl;
public Bean()throws Exception, SQLException

{
}
```

/*Para utilizar un DataSource*/

```
private DataSource dataSource;
private Context iniconte;
public Bean(PageContext p_pageContext,String base)throws Exception,
SQLException
{
        try{
        iniconte=new InitialContext();
    dataSource=(DataSource)iniconte.lookup("java:comp/env/jdbc/snort");
      gpuc_conexion=dataSource.getConnection();
      gpuc_conexion.setAutoCommit(false);
      }
        catch(Throwable e)
```

```
        {
                        conexion_error = e.toString();
                        System.out.println("JAVA BEAN : "+e);
                        //m_conn=null;

         }
   m_pageContext=p_pageContext;
   m_out=m_pageContext.getOut();
   m_response=(HttpServletResponse)p_pageContext.getResponse();
   m_request = (HttpServletRequest)m_pageContext.getRequest();
 }*/
```

/*Constructor para acceder a la Base de datos snort*/

```
 public Bean(PageContext p_pageContext)throws Exception, SQLException
  {
          String driver = "org.postgresql.Driver" ;
         Class.forName (driver).newInstance();
          try{
                  //ip,base,usuario
                 gpuc_conexion=
DriverManager.getConnection("jdbc:postgresql://192.168.2.1/snort","postgres
","postgres");
```

```
                gpuc_conexion.setAutoCommit(false);
                m_pageContext=p_pageContext;
                m_out=m_pageContext.getOut();
        m_response=(HttpServletResponse)p_pageContext.getResponse();
                m_request                                                =
(HttpServletRequest)m_pageContext.getRequest();
                m_jl=new JspLib(p_pageContext);
         }
        catch (Exception e)
        {
                System.out.println(e);
        }
 }
```

/*Constructor para acceder a la Base de datos Olimpo*/

```
 public Bean(PageContext p_pageContext, String unaCon)throws Exception,
SQLException
 {
         String driver = "org.postgresql.Driver" ;
        Class.forName (driver).newInstance();
         try{
                //ip,base,usuario
```

```
                gpuc_conexion=
DriverManager.getConnection("jdbc:postgresql://192.168.2.1/olimpo","olimpo"
,"olimpo");
                gpuc_conexion.setAutoCommit(false);
                m_pageContext=p_pageContext;
                m_out=m_pageContext.getOut();

m_response=(HttpServletResponse)p_pageContext.getResponse();
                m_request                                                =
(HttpServletRequest)m_pageContext.getRequest();
                m_jl=new JspLib(p_pageContext);
              }
              catch (Exception e)
              {
                    System.out.println(e);
              }
 }

 /*Metodo que sirve para cerrar la conexion con la base de datos*/
 public void closeConexion()  throws SQLException{
        if(gpuc_conexion!=null)gpuc_conexion.close();
 }//close_coneccion
```

```
}

package biblioteca;
import java.io.IOException;
import javax.servlet.jsp.PageContext;

public class BBB_Cerrar_Conexion extends Bean {

    /** Creates a new instance of BBB_Cerrar_Conexion */
    public BBB_Cerrar_Conexion(PageContext p_pc) throws IOException,
Exception{
    }
    public void close_conexion()throws  Exception
    {
        closeConexion();
    }
    }
```