PRACTICA EMPRESARIAL EN LA ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

JUAN CARLOS LAVERDE CARREÑO

DIRECTOR
MSc. MANUEL GUILLERMO FLOREZ

FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
UNIVERSIDAD INDUSTRIAL DE SANTANDER
BUCARAMANGA, JULIO 2008

PRACTICA EMPRESARIAL EN LA ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

JUAN CARLOS LAVERDE CARREÑO

Tesis de grado presentada como requisito parcial
para optar al título de Ingeniero de Sistemas.

DIRECTOR
MSc. MANUEL GUILLERMO FLOREZ

FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
UNIVERSIDAD INDUSTRIAL DE SANTANDER
BUCARAMANGA, JULIO 2008

DEDICATORIA

A mis abuelos
Cecília Martínez y Campo Elías Carreño,
por su apoyo, infinita colaboración y
la paciencia durante todos estos años
sin ustedes, no habría sido posible este trabajo,
Gracias!.

AGRADECIMIENTOS

Agradecimientos a...

Mis abuelos Cecília Martínez y Campo Elías Carreño por ofrecerme los recursos que me permitieron desarrollar mi proceso de formación en Ingeniería.

A mis padres, hermano y familiares por el apoyo incondicional en los momentos difíciles.

A mis amigos y compañeros de la UIS, por los momentos que jamás olvidare.

El Software Libre, por mostrarme el sentido humano de la tecnología.

La Universidad Industrial de Santander por su formación integral, ser pública y así sea por siempre.

Profesor Manuel Guillermo Florez Becerra, por la asistencia, dirección de este proyecto de grado y la confianza depositada en el.

INDICE DE CONTENIDOS

1. INTRODUCCIÓN....13
2. ESPECIFICACIONES DEL PROYECTO....14
2.1. TITULO. 14
2.2. DIRECTOR. 14
2.3. AUTOR. 14
3. DESCRIPCIÓN DE LAS CARACTERISTICAS DE LA EMPRESA....15
3.1. NOMBRE DE LA EMPRESA....15
3.2. MISIÓN DE LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA. 15
3.3. VISIÓN DE LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA. 15
3.4. ORGANIGRAMA. 16
4. PLANTEAMIENTO DEL PROBLEMA....17
5. OBJETIVOS 19
5.1. OBJETIVO GENERAL. 19
5.2. OBJETIVOS ESPECIFICOS. 19
6. JUSTIFICACIÓN 23
7. ANALISIS PRELIMINIAR DEL SERVIDOR CORMORÁN....25
7.1. Sistema Operativo (Linux RedHat Enterprise AS 3.0 taroon)....25
7.2. SERVIDORES WEB 28
7.2.1. ANÁLISIS DE SERVIDOR WEB APACHE. 28
7.2.2. Análisis del servidor WEB JAKARTA TOMCAT. 30
7.3. SISTEMAS GESTORES DE BASES DE DATOS 32
7.3.1. Análisis del sistema manejador de base de datos MYSQL....32
7.3.2. Análisis del manejador de base de POSTGRES. 34
7.4. PLATAFORMAS DE PROGRAMACION (PHP Y JAVA). 37
7.4.1. Análisis de lenguaje de programación PHP. 37
7.4.2. Análisis de JVM (JAVA VIRTUAL MACHINE – MAQUINA VIRTUAL DE JAVA). 38
7.5. ANÁLISIS DEL AGENTE DE TRANSPORTE DE MAIL SENDMAIL. 40
7.6. ANALISIS DEL FIREWALL 42

8. ASEGURANDO EL SISTEMA 43
8.1. SERVIDORES WEB. 44
8.1.1. Elección y algunas características de los servidores WEB. 45
8.1.2. Configuración del servidor WEB. 46
8.1.3. Configurando SECURE SOCKET LAYER, SSL (CAPA DE SEGURIDAD DE SOCKET). 48
8.1.4. Encapsulando los servicios WEB. 52
8.1.5. Configuración de módulos de seguridad en el servidor WEB. 52
8.1.6. Configuración del servidor web para paginas JSP. 54
8.2. SISTEMAS GESTORES DE BASES DE DATOS (SGBD). 56
8.2.1. Asegurando MYSQL. 57
8.2.2. Asegurando POSTGRESQL. 57
8.3. LENGUAJES DE PROGRAMACION 58
8.3.1. Aseguranado PHP. 58
8.4. CONFIGURANDO OTROS SERVICIOS 61
8.4.1. Asegurando SSH. 61
8.4.2. Configuracion del cortafuegos. 61
8.5. SISTEMAS DE REPORTE DE INFORMACION DEL SISTEMA 61
8.5.1. Sistemas de reporte WEB. 62
8.5.2. Configurando un analizador de sistema. 63
8.6. CONFIGURANDO UN SISTEMA DE DETECCION DE INTRUSOS (IDS) 63
8.7. CONFIGURANDO UN SCANNER DE VULNERABILIDADES 64
8.8. SISTEMA DE COPIAS DE RESPALDO DEL SERVIDOR (BACKUPS) 65
9. DEFINICION DE POLITICAS DE SEGURIDAD DEL SERVIDOR CORMORAN 67
9.1. ACCESO A RECURSOS FISICOS DEL SERVIDOR Y DEL ENTORNO 67
9.2. APLICACIONES WEB A RESIDENTES EN EL SERVIDOR 67
9.2.1. Aplicaciones hechas en PHP. 68
9.3. ADMINISTRACION DE LAS COMUNICACIONES Y OPERACIONES. 68
9.4. SEGURIDAD DE LOS RECURSOS HUMANOS 70
9.5. ADMINISTRACION DE INSIDENTES DE SEGURIDAD 70
9.6. POLITICAS DE LEGALIDAD DEL SERVIDOR 72
9.7. PROCESOS DE AUDITORIA EN EL SERVIDOR 72
9.8. Cumplimiento (LEGALES, DE ESTANDARES TECNICAS Y AUDITORIAS) 74

CONCLUSIONES 76
RECOMENDACIONES 77
BIBLIOGRAFÍA 78
LIBROS Y ARTICULOS 78
ENLACES 79

INDICE DE FIGURAS

Figura 1: Organigrama de la Escuela de Ingeniería de Sistemas e Informática. ...16
Figura 2: Cuota de mercado de Top de servidores en todos los dominios agosto de 1995 - Junio 2008 (Fuente www.netcraft.com).....29
Figura 3: Proceso de establecimiento de comunicación con SSL. (Fuente www.securityfocus.com) 50

RESUMEN

TITULO: PRACTICA EMPRESARIAL EN LA ESCUELA DE INGENIERIA DE SISTEMAS EN INFORMATICA DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER*

AUTOR: LAVERDE CARREÑO, Juan Carlos**

PALABRAS CLAVE: ADMINISTRACION SERVIDORES, EISI, SISTEMAS UIS

DESCRIPCIÓN DEL CONTENIDO:

La Escuela de Ingeniería de Sistemas e Informática de la Universidad de Industrial de Santander dispone entre sus recursos del servidor Cormorán, este ha sido empleado con diferentes propósitos, entre ellos, sitios orientados a la web para el respaldo a la gestión académica de proyectos, la publicación proyectos de investigación, simulación, publicación de información general orientada a la comunidad académica entre otros. La correcta administración de este recurso representa ofrecer a la comunidad académica servicios disponibles cuando se necesiten, que el tiempo de respuesta sea el adecuado y sobre todo que se conserve la integridad de la información.

Este trabajo de grado consiste en la administración del servidor Cormorán de la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander en la modalidad practica empresarial, orientada al desarrollo de los procesos operativos que permiten el correcto funcionamiento del servidor disponible a la comunidad académica además, la investigación de procesos que permitan elevar el grado seguridad y hacer del servidor un recurso informático que aloje algunos sitios que permiten procesos de gestión de la comunidad académica de forma confiable en la que se implementa normativas, mantenimiento y aplicaciones que permiten medir el estado del servidor.

El desarrollo de esta practica permitió la implementación de políticas de seguridad que desarrollan en el contexto del servidor y representa un recurso bibliográfico para futuras generaciones que se involucren con el desarrollo del servidor cormorán.

---

* Proyecto de grado

** Universidad Industrial de Santander, Facultad de Ingenierías Físico – Mecánicas, Escuela de Ingeniería de Sistemas e Informática. Director. MANUEL GUILLERMO FLÓREZ BECERRA.

ABSTRACT

TITTLE PRACTICA EMPRESARIAL EN LA ESCUELA DE INGENIERIA DE SISTEMAS EN INFORMATICA DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER*

AUTHOR LAVERDE CARREÑO, Juan Carlos**

KEYWORDS SERVER ADMINISTRATION, EISI, UIS

CONTENTS DESCRIPTION:

The Escuela de Ingenieria de Sistemas e Informatica of The Universidad Industrial de Santander owns the Cormorán Server resource, this server have been used with different purposes as academic web sites projects server, support to academic manage, simulation environment, investigation, and general projects. The correct administration of this resource means to give enabled services whenever is needed in the right time, keeping the integrity and security of the information.

This thesis work is about the manage of the Cormorán server as a “Practica Empresarial” oriented to support some operatives process to let the right server functionality to the academic community, moreover, the investigation of process to increase the securities levels and the confiability in process and data with a normative and administration server politics, maintenance and software implementation that let get measures of the server state.

The develop of this practice let the implementation of security politics in the server context and give a bibliography to future generations involved on the cormorán server.

---

* Degree work

** Universidad Industrial de Santander, Faculty of Engineerings Physical Mechanics, School of Systems engineering. Director: MANUEL GUILLERMO FLÓREZ BECERRA

1. INTRODUCCIÓN

La Escuela de Ingeniería de Sistemas e Informática de la Universidad de Industrial de Santander dispone entre sus recursos del servidor Cormorán, este ha sido empleado con diferentes propósitos, entre ellos, sitios orientados a la web para el respaldo a la gestión académica de proyectos, la publicación proyectos de investigación, simulación, publicación de información general orientada a la comunidad académica entre otros. La correcta administración de este recurso representa ofrecer a la comunidad académica servicios disponibles cuando se necesiten, que el tiempo de respuesta sea el adecuado y sobre todo que se conserve la integridad de la información.

La administración de servidores es una práctica constante que implica la ejecución de diferentes procesos. Estos procesos se pueden ser tan exhaustivos como se requieran en función de las necesidades los usuarios del servidor en si. Los servicios ofrecidos, su eficiencia, el grupo de usuarios al que se dirige, la topología de la red, los recursos computacionales del servidor, son algunos de los elementos que permiten definir u orientar los procesos y practicas que se debe aplicar a un servidor con el fin de garantizar ciertos estados a los usuarios. Es por esto que la administración de servidores puede llegar a ser muy diversa y componerse tanto de procesos operativos como análisis periódico de los estados del servidor con el fin de identificar una situación no deseada para ser corregida.

La metodología de desarrollo propuesta consiste en implementar los cambios que se consideren necesario en ambientes de prueba en los cuales se prueba el comportamiento del servidor, una vez se tiene conocimiento del funcionamiento, los cambios son implementados en producción.

2. ESPECIFICACIONES DEL PROYECTO

2.1. TITULO.

PRACTICA EMPRESARIAL EN LA ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

2.2. DIRECTOR.

_____

MSc. Manuel Guillermo Florez

Universidad Industrial de Santander, Bucaramanga Colombia.

2.3. AUTOR.

_____

Juan Carlos Laverde Carreño.

Estudiante de Ingeniería de Sistemas.

Código. 1991134

3. DESCRIPCIÓN DE LAS CARACTERISTICAS DE LA EMPRESA

3.1. NOMBRE DE LA EMPRESA

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA (EISI) DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER

3.2. MISIÓN DE LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA.

La Escuela de Ingeniería de Sistemas e Informática (EISI) comprometida con la misión institucional, tiene como propósitos: la formación de personas autónomas, creativas, que actúen según principios éticos universalmente aceptados, de alta calidad ciudadana y comprometidos con el desarrollo regional y nacional; y la construcción, innovación y mejoramiento del conocimientos, que permitan disponer de la fundamentación teórica, tecnológica e instrumental para administrar y tratar los sistemas de información, las comunicaciones y la automatización industrial.

La EISI forma, actualiza y proyecta el recurso humano en áreas de pregrado, postgrado y de educación continuada, soportadas en el respeto de los valores humanos, logrando profesionales competentes. La EISI define, establece, desarrolla y evalúa su proceso administrativo, pedagógico e investigativo, apoyándose en el enfoque sistémico y el reconocimiento propio y ajeno. Fundamenta su labor en el liderazgo, la pertenencia, la tolerancia y el trabajo unificado de profesores, estudiantes y demás colaboradores.

3.3. VISIÓN DE LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA.

La Escuela de Ingeniería de Sistemas e Informática (EISI) se proyecta como una Unidad académica y administrativa, respaldada por la calidad humana de su personal administrativo, académico e investigativo, la formación científica de sus

docentes, el nivel académico de sus estudiantes y su integración con las políticas institucionales y la sociedad para la generación, proyección y aplicación del conocimiento, poniéndolos en sus procesos de docencia, investigación e integración con la comunidad.

3.4. ORGANIGRAMA.

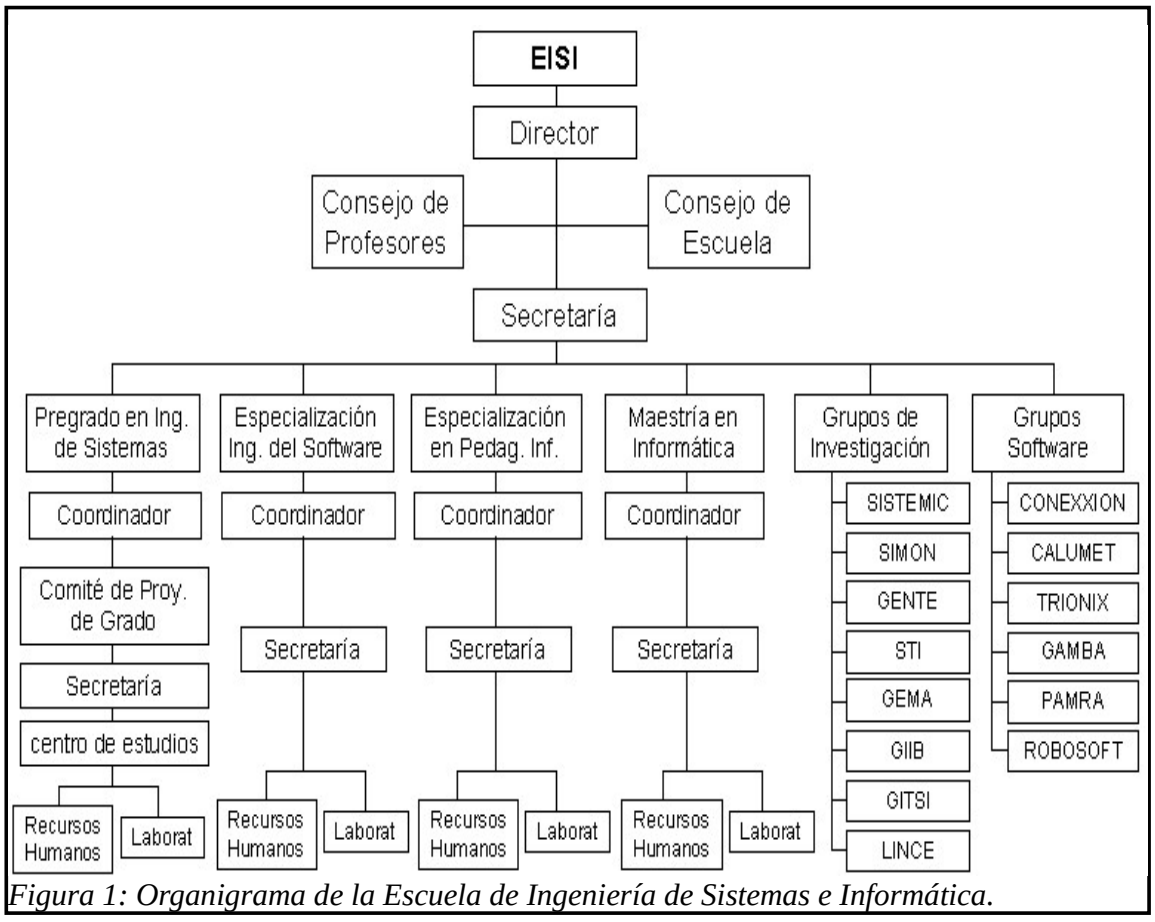

Figura 1: Organigrama de la Escuela de Ingeniería de Sistemas e Informática.

4. PLANTEAMIENTO DEL PROBLEMA

El servidor Cormorán de la EISI, ofrece servicios orientados a la web de apoyo a la gestión académica, soporte a algunas materias y aplicaciones con propósitos particulares. Estas aplicaciones son el resultado del desarrollo de grupos de investigación de la Escuela de Ingeniería de Sistemas e Informática.

Estas aplicaciones son escritas en lenguajes de programación, usan sistemas gestores de bases de datos, y estos a su vez están soportados en un sistema operativo. Los grupos de desarrollo deben contar con un espacio en el servidor que le permita hacer el respectivo mantenimiento para cada aplicación, de igual forma se debe garantizar la integridad del acceso a cada espacio, ya que el uso malintencionado o equivoco de los espacios podría significar la perdida de la integridad de las aplicaciones o de la información contenida en cada sitio.

Las aplicaciones deben ser monitoreadas periódicamente ya que los servidores web generan registros de su actividad que notifican parte de la actividad de las aplicaciones, estos registros ocupan espacio y se debe hacer el respectivo proceso de depuración para no saturar los medios de almacenamiento del servidor.

La comunidad académica y usuarios en general introducen información sensible en el servidor y se deben crear los mecanismos que permitan que la información y servicios sean ofrecidos y usados de la forma para los que fueron construidos y por los usuarios para los que fueron destinados. Este elemento es importante ya que permite a los sistemas de información contenidos en el servidor ofrecer datos que representan información correcta.

Los recursos computacionales del servidor se comportan según su propia naturaleza, es decir, cada recurso fue desarrollado con ciertos requerimientos y características especificas, por ejemplo, el hardware no puede funcionar sin energía eléctrica, la arquitectura del servidor demanda tipos de sistemas operativos específicos, el sistema operativo es funcional bajo ciertos requerimientos de memoria primaria y procesador, etc. en conclusión se puede decir que cada componente del sistema tiene requerimientos así como el sistema completo genera otros requerimientos deben ser atendidos si se quiere que el

sistema (en este caso el servidor) opere de la forma deseada.

La comunidad académica en el proceso de publicación de aplicaciones debe ser asistida y contar con asesoramiento en los requerimientos de publicación.

De todo lo anterior emergen algunas situaciones no deseadas que las que se enfoca este trabajo de grado con el fin de ser corregidas, asistidas o mejoradas según cada caso.

Es necesario mantener la disponibilidad del servidor el mayor tiempo posible, el servidor debe ser un medio que permita el uso de servicios y que estos sean usados según el propósito y usuarios para los que fue creado, el servidor debe contar con asistencia técnica y asesoramiento en la instalación de aplicaciones orientadas a al web. El servidor debe contar con medios que permitan aumentar la seguridad del mismo.

5. OBJETIVOS

5.1. OBJETIVO GENERAL.

- Monitorear y mantener el Servidor Cormorán de la Escuela de Ingeniería de Sistemas en estado operativo durante el periodo de la práctica.

- Administrar el servidor Cormorán EISI y asistir a la comunidad académica en el montaje de sitios así como los procesos que conciernen con el servidor.

- Analizar el rendimiento y el estado de seguridad actual del servidor para detectar vulnerabilidades e implementar las respectivas soluciones.

- Hacer el montaje de un servidor de pruebas que permita estudiar las tecnologías que se implementaran en el servidor cormorán con el propósito de tener plena definición de los procesos antes de ser implementados.

5.2. OBJETIVOS ESPECIFICOS.

- Definir los espacio de memoria en el disco duro de las particiones necesarias para el montaje del sistema de archivos. Instalar el Sistema Operativo con los respectivos paquetes requeridos según la documentación establecida en practicas anteriores. Configurar los archivos de red y DNS para el servidor Cormorán. Instalar los servidores de páginas Web Apache y Jakarta-tomcat con sus respectivos conectores para compatibilidad con los servicios de bases de datos. Instalar los manejadores de base datos mysql, postgres y los paquetes de administración phpMyAdmin, pgacces y mysql-admin. Instalación de la maquina virtual de java y la jdk (java development kit) para paginas que usan ambiente java y jsp. Instalar el Interpretador de lenguaje PHP. Instalación de Moodle. Instalar software que permita copiado de backups en dvd (k3b).

- Crear las cuentas de usuario a los grupos que necesitan contar con espacios en el servidor y asignarles los respectivos privilegios de acceso al sistema.

- Modificar los archivos de configuración de los servicios Web cuando se crean

nuevos sitios y modificar los permisos de acceso en la configuración de cada sitio, así como los enlaces simbólicos que se requieran.

- Reducir el tiempo de toma de copias de seguridad (backups) del sistema, de los sitios y de las bases de datos respecto a las versiones actuales, además, definir el tiempo de toma de imágenes en el servidor para contar con un respaldo del estado del servidor en un determinado tiempo.

- Actualizar y mantener los scripts de backups y borrado de archivos temporales para los sitios que están configurados actualmente en el servidor.

- Realizar tareas de mantenimiento en el sistema operativo (Backups del sistema, Estudio de archivos de registro del sistema, crear sitios, administrar usuarios, atender solicitudes de los grupos de desarrollo y la comunidad académica en general) .

Actividades de Estudio.

- Montar un servidor de respaldo que soporte las tecnologías y aplicaciones usadas por el servidor Cormorán para experimentar los cambios que puede presentar el servidor después actualizaciones en las versiones de los paquetes y sus configuraciones.

- Poner a prueba y funcionamiento (etapa productiva) el servidor de respaldo paralelo al servidor principal y estudiar el comportamiento.

- Estudiar la versión del kernel actual así como la distribución del Sistema Operativo Linux utilizada en el Servidor Cormorán para hacer un análisis de seguridad, compatibilidad e instalación de paquetes en el sistema de ser necesario se estimara cambiar el sistema operativo, sistema de paquetes y todos aquellos elementos que contribuyan al mejoramiento del mantenimiento y los servicios del servidor. Se presentará un informe del estudio para facilitar la toma de decisiones.

- Verificar las versiones de las herramientas instaladas en el servidor Cormorán y su posible actualización dependiendo del requerimiento de los sitios alojados

en el servidor.

- Comprobar y Organizar la documentación existente actual con el estado del servidor para conocer el soporte con el que se cuenta.

- Analizar el desempeño y la capacidad del servidor en el uso de memorias (física, disco duro, medios de almacenamiento externos) y procesamiento en función de carga.

- Estudiar el sistema de backups del servidor con el propósito de reducir el tiempo generado entre las copias de respaldo y las versiones mas recientes en el servidor.

- Estudiar la seguridad actual del sistema:

    - Estudiar los accesos y permisos a los diferentes directorios instalados en el servidor con el fin de redefinir el acceso a los sitios web.
    - Estudio de las políticas administrativas y seguridad actual en la forma definir perfiles de usuarios y control de acceso de recursos al sistema.
    - Analizar las vulnerabilidades en el sistema y corregir las configuraciones o fallas en el sistema que puedan comprometer la seguridad e integridad de la información y del sistema, para esto se ejecutaran:

        - Análisis de exploración de puertos.
        - Análisis de detección de sistema operativo.
        - Análisis de la información DNS en el sistema.
        - Análisis de enumeración.
        - Análisis de la arquitectura de red del sistema.
        - Análisis en la detección y bloqueo de intrusos.
        - Estudio de reportes de errores en los paquetes implementados.
        - Análisis de algunos exploits (programas que explotan vulnerabilidades).

- Analizar los archivos de actividades y registro del servidor para tener control de las actividades en el sistema con el propósito de conocer el comportamiento de los servicios ofrecidos por el servidor.

- Implementar un sistema de control de acceso de servicios (Firewall) .

- Documentar
    - Revisar la documentación actual
    - Elaborar el manual del administrador del sistema.
    - Documentar los procesos y actividades ejecutadas.

- Hacer el empalme y entrenamiento con el nuevo practicante.

Con el fin de ejecutar algunos procesos de administración como una tarea operativa accesible a un usuario determinado, se asignaran privilegios en el uso de algunos comandos a dicho usuario a partir de la definición en la configuración de la aplicación SUDO (nombre de la sigla en ingles, usuario substituto de súper-usuario).

6. JUSTIFICACIÓN

La escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander en su proceso de crecimiento ha implementado ambientes y servicios orientados a la Web que se ofrecen a la comunidad académica, estos, son desarrollados y orientados con diferentes propósitos.

Entre los recursos de la escuela de Ingeniería de sistemas e Informática, se cuenta con el servidor Cormorán, este, hospeda los sitios y aplicaciones que son el resultado del desarrollo de diferentes grupos de investigación, además, estos sitios aunque están enfocados a un tipo de arquitectura común (cliente-servidor) operan con diferentes tipos de plataformas web. La diversidad de las herramientas usadas en la creación de los sitios ah significado en el servidor el estudio e implementación de herramientas apropiadas como servidores web, manejadores de bases de datos, interpretadores, servidores virtuales entre otros. El estado actual del servidor es funcional y se puede considerar en una etapa estable en la que son ejecutadas las operaciones que son requeridas.

Sin embargo, el buen funcionamiento de un servidor no solo depende de ejecutar los procesos requeridos, también es necesario implementar mecanismos que aumenten la integridad de la información tanto en la transmisión como en el almacenamiento de la misma, de tal forma que los usuarios cuenten única y exclusivamente con los recursos que por naturaleza deben poder acceder, además la integridad de la información representa adoptar políticas y mecanismos que permitan al servidor aumentar cada vez mas la tendencia a la seguridad.

En este contexto, la seguridad es una tendencia a la que se debe recurrir para aumentar el estado de integridad de la información que esperamos, esta tendencia entre otras cosas implica la capacidad del sistema de cambiar según sea necesario para asumir los estados que representen determinada calidad en la integridad de la información, esta característica se resume en escalabilidad. Un sistema es escalable en la medida que es flexible al cambio y es capaz de crecer sin afectar dramáticamente en términos de implementación el estado del sistema, la escalabilidad es vital en la medida que permite al servidor cambiar a nuevas versiones, permitir actualizaciones, aplicar parches de actualización a las aplicaciones y todos aquellos procesos que impliquen cambiar los recursos

existentes en el servidor, ya que las sitios alojados en el servidor cambian continuamente y requieren el respectivo soporte. Es por esto que dentro del proceso de investigación es necesario estimar la seguridad, escalabilidad y rendimiento así como la implementación de políticas que hagan de Cormorán un servidor que además de ser funcional alcance un estado de seguridad que permita elevar la integridad de la información.

7. ANALISIS PRELIMINIAR DEL SERVIDOR CORMORÁN

En este análisis se estimara la plataforma funcional del servidor Cormorán y algunos servicios ofrecidos a la comunidad académica según la demanda de los mismo, el criterio de evaluación esta sujeto a cada aplicación y al desarrollo que han tenido respectivamente.

El propósito de este análisis es conocer el estado de las aplicaciones que representan servicios a la comunidad académica en términos de tiempo y estado de desarrollo en la actualidad.

Las conclusiones de este análisis permitirán definir que tipos de cambios son necesarios en el servidor a nivel de software.

7.1. SISTEMA OPERATIVO (LINUX REDHAT ENTERPRISE AS 3.0 TAROON)

El sistema Operativo Linux Red Hat es producido por Red Hat Software Inc fundada en 1994, año en el que salio la primera versión, Red Hat una de las distribuciones mas populares que fue ganando rápidamente adeptos con desarrolladores en diferentes lugares del mundo que se caracterizo por introducir los sistemas de paquetes RPM (Red Hat Package Manager), soporte a caracteres no latinos y kit de aplicaciones orientadas a usuarios inexpertos. Red Hat creció en gran medida en el sector empresarial a tal punto que en el año 1999 en un hecho sin precedentes en wall street las acciones de Red Hat alcanzaron la octava ganancia de su valor en un dia, hasta que cuatro años después las crisis de las denominadas empresas “punto com” (.com) afectaron las acciones de Red Hat de forma igualmente dramática hasta quebrar.

En abril del 2004 mientras que el proyecto Red Hat termina, se re-orienta el desarrollo a una distribución orientada a los negocios Red Hat Enterprise.

Actualmente, el servidor Cormorán usa como sistema operativo Linux Red Hat Enterprise AS 3.0 (taroon) en plataforma de 32 bits. Esta versión salió oficialmente el 22 de Octubre de 2003. A la fecha han salido 2 distribuciones mas, Red Hat Enterprise Nahant (15 Febrero 2005) y Red Hat Enterprise Tikanga (14 Marzo

2007), lo que en terminos de tiempo es una diferencia grande al estado actual del servidor.

Red Hat Enterprise entre sus repositorios aunque ofrece los código fuentes, no ofrece distribuciones en binarios y para obtener la distribución se debe compilar los fuentes de toda la distribución o hacer la respectiva gestión en la obtención de esta y el pago de su licencia, pero esto no ha sido un impedimento para que nazcan otros proyecto (forks o bifurcaciones) que han usado los archivos fuente de Red Hat Enterprise y hayan nacido nuevas distribuciones como: Centos, Lineox, Scientific-Linux, White box Enterprise Linux entre otras.

Red Hat Enterprise Linux es una distribución orientado al sector empresarial con fines económicos y para obtener el respectivo soporte de Red Hat es necesario contratar y pagar por este servicio.

Red Hat Enterprise ofrece entre sus inscritos, un sistema de soporte para actualizaciones y corrección de errores o bugs[1] en la medida que se van identificando, este sistema de paquetes se conoce como up2date, propio de los RHEL (Red Hat Enterprise Linux) en los que se contrata el soporte.

Como conclusiones respecto al sistema operativo implementado en el servidor cormorán se puede decir que es una distribución que ya tiene mas de 5 años de haber salido, lo que muy probablemente lo hace obsoleta en términos de compatibilidad e implementar nuevas versiones de paquetes, en esta medida no es necesario un análisis exhaustivo para suponer que existen nuevas versiones de los paquetes que aprovechan mejor los recursos con los que se cuenta y que principalmente son menos propensos a bugs o errores que comprometan la seguridad y estabilidad del servidor.

La actualización del sistema no se ha hecho debido a que no se ha pagado por la suscripción o soporte y a que el costo es elevado y la Escuela de Ingeniería de Sistemas e Informática esta limitada en rubros para este tipo de proyectos , lo que significa que los paquetes que operan actualmente pueden contener errores que no se han corregido y hacen susceptible el servidor a fallas de seguridad. Desde este punto no representa ningún beneficio contar con un servidor Red Hat

1 Comportamientos inesperados en las aplicaciones, normalmente son errores.

Enterprise Linux debido a que la fortaleza de este radica en el soporte empresarial que este puede brindar pero no se usa debido a su gestión, de hecho, el proceso de actualización a partir de un sistema de paquetes binarios o fuentes en linea no existe, situación que pone en desventaja a RHLE en comparación a un homónimo por ejemplo como la distribución GNU/Linux Centos que posee un sistema gestor de paquetes de carácter libre y con repositorios actualizados. La escalabilidad en términos de software está en un estado que se podría considerar bajo teniendo en cuenta que si bien la distribución en este caso RHLE es actualizable a partir de su sistema de paquetes RPM (Red Hat Package Manager) esto no resulta un sistema proceso si se tiene en cuenta que actualmente la gran mayoría de distribuciones que existen usan sistemas manejadores de paquetes en linea (ej. “apt-get” para GNU/Linux Debian, “yum” para GNU/Linux Centos entre otros) lo que permiten hacer el trabajo de actualización y mantenimiento de forma sencilla a partir de repositorios en internet (RHEL posee sistema manejador de paquetes “up2date” pero este servicio se debe contratar).

Actualizar el sistema a partir de paquetes RPM, significa contar con uno o varios paquetes que cumplen un determinado propósito y que estos se pueden instalar en la medida que se cumplan sus dependencias (requerimientos para cada paquete), es decir, un paquete se puede instalar siempre y cuando los paquetes de los que depende estén instaladas, y no solo eso, también se debe tener en cuenta que existen conflictos entre dependencias, esto quiere decir que puede que un paquete tenga problemas de dependencias con otro por lo tanto se debe desinstalar los paquetes que generan el problema. Suponiendo que se necesite instalar el paquete A en el que A depende del paquete B, entonces es necesario instalar el paquete B y después instalar el paquete A, no obstante si queremos instalar el paquete C y entre las dependencias existe conflictos de compatibilidad con el paquete B, debemos decidir si queremos dejar de instalar C y conservar A y B o por el contrario es imprescindible contar con el paquete C y eliminar los paquetes B y A, o apelar a una solución diferente, de esta forma se puede ilustrar el gran problema que puede representar para un administrador de sistema tener en cuenta los problemas de dependencias y establecer la posible solución con lo que esto representa, buscar “a mano” los paquetes en repositorios de internet y a su vez estimar la dependencia de cada uno de los paquetes que se instalan, los sistemas manejadores de paquete hacen este trabajo y permiten definir cuales son las dependencias, cuales generan conflictos, cuales se instalaran, actualizaran y

borraran de tal forma que el administrador tiene pleno conocimiento de las actividades de gestión de paquetes y solo debe decidir cual es el estado deseado del servidor y no como generar ese estado.

El kernel usado en la distribución es “2.4.21” ya hace algún tiempo de la salida de este kernel y el sitio http://www.kernel.org/pub/linux/kernel/v2.4/ChangeLog-2.4.21 ofrece referencias muy precisas y se puede consultar cambios que se hicieron en reparaciones de bugs (errores) y mejoras en la compatibilidad con el hardware.

7.2. SERVIDORES WEB

7.2.1. ANÁLISIS DE SERVIDOR WEB APACHE.

El proyecto Apache Http Server empieza cuando se reescribe en totalidad los parches hechos para NCSA HTTPd 1.3 que era un servidor web desarrollado por Robert McColl y una lista de colaboradores en la National Center of Supercomputing Applications el proyecto de la NCSA termina en 1998 aunque el código se mantuvo y migro a lo que hoy se conoce como el proyecto Apache.
Actualmente este servidor web tiene la cuota mas alta de servidores del mercado con cerca del 80% (según datos de Netcraft en Octubre del 2005) pero que en los últimos se redujo a un 56% (hasta el 2007) http://news.netcraft.com/.

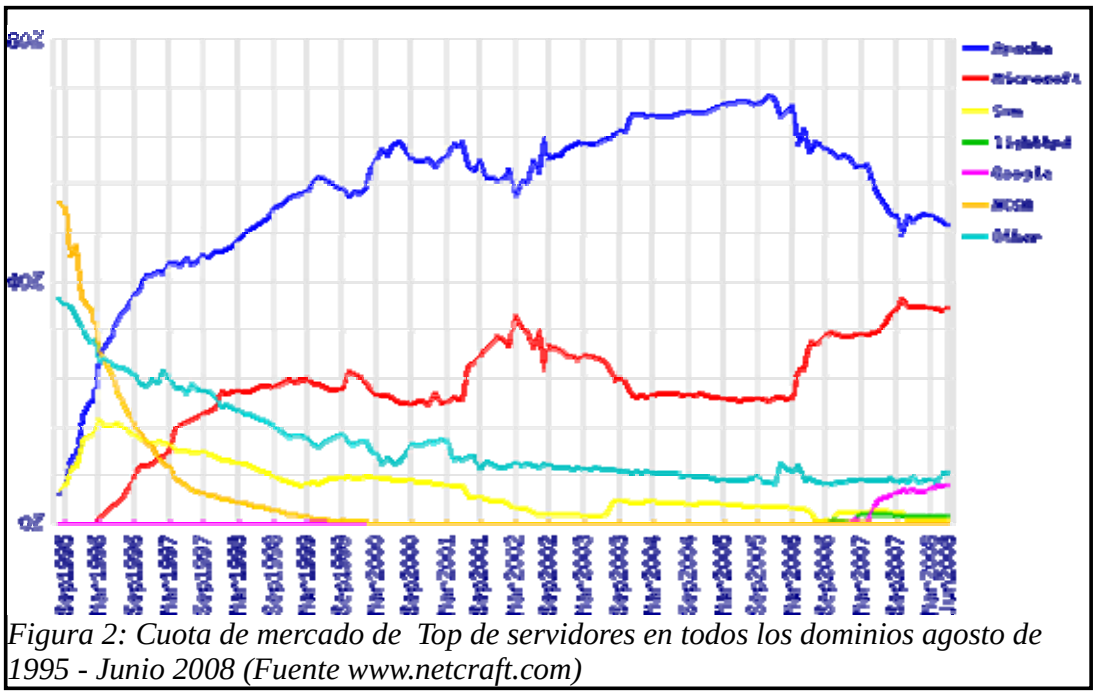

Figura 2: Cuota de mercado de Top de servidores en todos los dominios agosto de 1995 - Junio 2008 (Fuente www.netcraft.com)

El servidor web Apache se caracteriza por poseer una arquitectura modular, esta estructura permite que sus funcionalidades sean adaptadas en la medida que se necesitan y no estar envedidas en el core (núcleo) de la aplicación lo que permite un manejo de memoria mas eficiente.

Una estructura modular a nivel de desarrollo permite enfocar los esfuerzos a un determinado tipo de tecnología, es decir, existen diferente módulos orientados a seguridad como mod_ssl (comunicación segura a partir de Certificados de seguridad SSL), mod_auth_Ldap (Autentificaron de usuarios LDAP) también módulos que permiten usar al servidor Apache como front-end de otro servidor como Tomcat (mod_proxy_ajp).

http://httpd.apache.org/docs/2.2/mod/http://httpd.apache.org/docs/2.2/mod/ Este enlace muestra los módulos de la versión 2.2 de apache (apróx. 60 modulos) que van desde Procesos hibridos multihilo hasta hosting virtual masivo.

Actualmente el servidor cormorán cuenta con la versión 2.0.46 versión que fue instalada con la paquetería del sistema operativo Red Hat Enterprise 3 taroon. Esta versión viene de la evolución de la versión 2.0 de apache y que en la actualidad llega hasta la versión 2.0.61 http://www.apache.org/dist/httpd/CHANGES_2.0 el enlace muestra los cambios que se han hecho a la versión 2.0.

El comportamiento de Apache 2.0.46 en Cormorán ha sido estable, aunque registros de administraciones anteriores muestran un periodo en que el servidor estuvo fuera de servicio cerca de 6 semanas pero esta situación se debió a procesos de configuración y adecuación del servidor en general y no Apache explícitamente.

En la planificación del proceso de administración se estimo según administraciones anteriores implementar un sistema concurrente de versiones (en este caso Subversión), este sistema esta orientado el desarrollo a nivel de software de los grupos de investigación y pretende ser un soporte en el desarrollo del software a partir de un sistema de repositorios. En la actualidad esta implementación no se efectuado, entre las razones, prima la versión actual del núcleo del servidor web apache, la versión propuesta por el proyecto Subversion no es compatible con la actual de Apache, actualmente esta a consideración las posibles soluciones a este evento.

Actualmente el servidor no cuenta con una herramienta que permita medir la actividad del servidor web en términos de uso de paginas, y medidas que permitan obtener estadísticas de la actividad web y uso de los servicios prestados por apache por lo tanto se desconoce el uso de cada aplicación y los registros de las actividades orientadas a la web se deben estudiar desde el sistema de logs (registros) del sistema operativo.

7.2.2. Análisis del servidor WEB JAKARTA TOMCAT.

Jakarta Tomcat es un Servidor Web resultado de la iniciativa de James Duncan Davidson, en ese entonces trabajaba como arquitecto de Sun Microsystems y decidió donar el proyecto a Apache Software Foundation, proyecto que se hizo

pensado como contenedor de tecnologias servlets* (colocar pie de pagina de servlet) y JavaServer Page (JSP) bajo especificaciones de Sun y la Java Community Process (JCP). Actualmente el proyecto es desarrollado con un gran grupo de programadores al rededor del mundo y goza de gran soporte.

El servidor Cormorán implementa tomcat 4.1 como servidor de paginas web de servlets y JSP. A la fecha de esta versión se han presentado algunos cambios.
http://tomcat.apache.org/tomcat-5.5-doc/changelog.html

En este enlace se puede consultar los cambios que se se efectuaron en la funcionalidad de tomcat5.5 (version estable actual) con respecto a las anteriores. Se debe tener en cuenta que cada versión posterior a la 4.1 (versión actual en el servidor Cormorán) hizo sus respectivas correcciones a bugs e implementaron nuevas funcionalidades, así que la lista de variaciones de tomcat5.5 se hace con respecto a la inmediatamente anterior en este caso tomcat5.

La versión actual no ha representado problemas y según registros no han ocurrido en el pasado. Desde este punto de vista no es imprescindible actualizar el servidor, sin embargo, en la medida que hay nuevas versiones se corrigen problemas de seguridad por ejemplo http://tomcat.apache.org/security-5.html muestra los problemas de seguridad que se han corregido en la versión 5 de tomcat, lo que significa que efectivamente en el desarrollo de las versiones se han encontrado vulnerabilidades que pueden representar un riesgo para el buen funcionamiento de un servidor as las que deben el respectivo parche (corrección), además el desarrollo de nuevas utilidades en el servidor web puede representar un cuello de botella a nivel de desarrollo, es decir, en la medida que se implementan nuevas utilidades en el servidor, los desarrolladores tienden a usar estas nuevas utilidades por lo que se hace necesario contar con un servidor de aplicaciones web para servlets y JSP con soporte a estas nuevas utilidades, de lo contrario, los desarrolladores de sitios en el servidor cormorán serán forzados a seguir desarrollando sitios que usen las utilidades y estándares que ofrece el servidor Tomcat actual (4.1).

En el proceso de tuberculización el servidor web Jakarta Tomcat ha introducido nuevas reglas y formatos para la publicación de los sitios, por tanto, si se desea actualizar las aplicaciones actuales de Tomcat a las formatos actuales, se debe

hacer la respectiva modificación para los sitios.

7.3. SISTEMAS GESTORES DE BASES DE DATOS

7.3.1. Análisis del sistema manejador de base de datos MYSQL

En sus inicios, SQL se remonta a 1970 cuando en los laboratorios de IBM se desarrollaba un sublenguaje de acceso a los datos basado en el calculo de predicados, pero fue en 1977 cuando SEQUEL (Structured English QUEry Language) fue implementado por el SGBD (Sistema Gestor de Base de Datos) System R, también desarrollado por IBM. Sin embargo es Oracle quien introduce la herramienta por primera vez en un programa comercial en el que nace el predecesor de SEQUEL conocido como SQL.

En 1981 IBM comercializa y propone el Lenguaje de Consulta Estructurado (SQL), un lenguaje declarativo orientado al manejo de conjuntos de registros que permite especificar varias operaciones sobre las mismas valiéndose del álgebra y el calculo relacional, SQL fue presentado a la ANSI (Instituto Nacional Estadounidense de Estándares) y desde entonces fue aceptado como estándar para las bases de datos relacionales. Desde 1986 el estándar ha tenido variaciones por lo que se han desarrollado diferentes versiones como SQL:86 (primera publicación aprobada por ISO), SQL:89, SQL:92, SQL:99, SQL:2003 y SQL:2006, estas versiones son revisiones e inclusiones de elementos como triggers en la versión SQL:99 o estandarizaron de XML SQL:2003.

MySQL nace como un proyecto de la empresa Opensource MySQL AB establecida en sus inicios en Suecia en 1995, los fundadores del proyecto fueron David Axmark, Allan Larsson, y Michael "Monty" Widenius. En principio, la idea del proyecto era crear un sistema que cumpliera con el estándar SQL en el que no se sacrificara velocidad, fiabilidad y usabilidad.

MySQL esta escrito en C y C++ y por ser un proyecto con licenciamiento GPL es posible ver la documentación de su estructura interna y la forma como se desarrolla. El proyecto esta basado en un grupo de desarrolladores contratados

por MySQl AB quienes son los encargados de dar soporte a los cliente comerciales y solucionar los errores reportados por la comunidad de usuarios. Existen listas de correo y documentación de historia de los errores(bugs) corregidos así como la historia de los cambios, y documentación. La dirección y el patrocinio de la empresa esta a cargo de MySQL AB quienes poseen el copyright de MySQL.

Actualmente la empresa maneja suites de productos como MySQL Enterprise que no es software libre.

El servidor Cormorán de la EISI implementa MySQL 4.1.9 (i686) esta versión tiene motores de almacenamiento como InnoDB (Integrada a partir de la version 4) para integridad referencial así como Merge, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en las versiones de la 5 en adelante.

En el enlace se encuentran los cambios que se han hecho para la versión 5 (versión siguiente a la actual implementada en el servidor)
http://dev.mysql.com/doc/refman/5.0/en/news-5-0-x.html

Este enlace muestra los bugs detectados que corren en ambiente linux y que corresponden a la version 4.1.9 (son cerca de 37)
http://bugs.mysql.com/search.php?search_for=&bug_type%5B%5D=Server&status%5B5D=All&severity=all&limit=All&order_by=&cmd=display&phpver=4.1.9&os=5&os_details=&bug_age=0&tags=&similar=&target=

En el servidor Cormorán existe un gran porcentaje de bases de datos soportadas en MySQL el comportamiento de estas ha sido bastante estable, hasta el día de hoy no se ha registrado algún problema en el comportamiento y el rendimiento en términos de memoria es bueno debido a que las peticiones generan un porcentaje de consumo de memoria bajo, aunque en este análisis se deben estimar los periodos en los que se registra mayor actividad en el servidor para contar con información mas acertada de los recursos.

7.3.2. Análisis del manejador de base de POSTGRES.

Postgres es un potente sistema gestor de bases de datos relacionales (Relational Data Base Management System RDBMS), nació en la Universidad de Berkeley a partir del proyecto Ingres a la cabeza de Michael Stonebraker y que se caracterizó por ser una de las primeras propuestas en los Motores de de bases de datos relacionales, después de un periodo de ausencia gracias al resultado comercial de Ingres, Michael regresa a la Universidad para trabajar en un nuevo proyecto secuela de Ingres que fue denominado Post-Ingres y mas adelante Postgres.

El nuevo proyecto pretendía implementar conceptos aclarados en 1980 a cerca del modelo de datos relacional (uno de los principales problemas fue la incapacidad del modelo de reconocer “tipos” es decir, una unidad vista como el conjunto de diferentes datos.

En 1986 empezó la implementación del proyecto Postgres, desde entonces tuvo cambios significativos, en 1987 el sistema de pruebas se hace operacional y fue mostrado en la Conferencia ACM-SIGMOD de 1988. La primera versión sale en junio de 1989 y por una serie de criticas fue rediseñado, la versión 2 sale en 1990 y para el año 1993 había duplicado la cantidad de colaboradores externos al proyecto. Con el tiempo, el proyecto se convirtió en un trabajado de soporte mas que de investigación por lo que el proyecto termina. Un año mas tarde en 1994 Andrew Yun y Yolly Chen retoman el proyecto y añadieron un interprete de lenguaje SQL a Postgres, antes Postgres poseía su propio lenguaje de consultas y a partir de esta versión cambio a Postgres95, por esa misma época Postgres fue publicado en la red y se hizo de dominio publico y código abierto. Postgres95 fue adoptado por ANSI c, se optimizó el espacio y el código mejorando el rendimiento y el mantenimiento.

Postgres95 v1.0.x corría cerca de un 30-50% más rápido en el Wisconsin Benchmark comparado con Postgres v4.2. Además de corrección de errores, a continuación las mejora mas relevantes:

- El lenguaje de consultas Postquel fue reemplazado con SQL (implementado en el servidor). Las subconsultas no fueron soportadas hasta PostgreSQL (ver más abajo), pero podían ser emuladas en Postgres95 con funciones SQL definidas por el usuario. Las funciones agregadas fueron reimplementadas. También se añadió una implementación de la cláusula

GROUP BY. La interfaz libpq permaneció disponible para programas escritos en C.

- Además del programa de monitorización, se incluyó un nuevo programa (psql) para realizar consultas SQL interactivas usando la librería GNU readline.
- Una nueva librería de interfaz, libpgtcl, soportaba clientes basados en Tcl. Un shell de ejemplo, pgtclsh, aportaba nuevas órdenes Tcl para interactuar con el motor Postgres95 desde programas tcl
- Se revisó la interfaz con objetos grandes. Los objetos grandes de Inversión fueron el único mecanismo para almacenar objetos grandes (el sistema de archivos de Inversión fue eliminado).
- Se eliminó también el sistema de reglas a nivel de instancia, si bien las reglas siguieron disponibles como reglas de reescritura.
- Se distribuyó con el código fuente un breve tutorial introduciendo las características comunes de SQL y de Postgres95.
- Se utilizó GNU make (en vez de BSD make) para la compilación. Postgres95 también podía ser compilado con un gcc sin parches (al haberse corregido el problema de alineación de variables de longitud doble).

La enciclopedia libre Wikipedia (www.wikipedia.com) señala una serie de eventos “los mas relevantes” en cuanto al desarrollo de postgres.

1986 - se publicaron varios papers que describían las bases del sistema.
1988 - ya se contaba con una versión utilizable.
1989 - el grupo liberaba la versión 1 para una pequeña comunidad de usuarios.
1990 - se liberaba la versión 2 la cual tenía prácticamente reescrito el sistema de reglas.
1991 - liberación de la versión 3, esta añadía la capacidad de múltiples motores de almacenamiento
1993 - crecimiento importante de la comunidad de usuarios, la cual demandaba más características
1994 - antes de la liberación de la versión 4, el proyecto termina y el grupo se disuelve.
En 1996 Postgres95 parecía un nombre poco apropiado por lo que se retoma el nombre de PostgreSQL en alusión al nuevo soporte del estándar SQL, de igual

forma, se retoma la numeración de las versiones que ya se habían desarrollado (continuo en la versión 6).

Si bien Postgres95 enfatizo el fortalecimiento del código del motor de datos, la nueva versión se preocupo por los aspecto funcionales y el aumento de las características aunque de el desarrollo es continuo en todas la áreas.

Las principales mejoras en PostgreSQL incluyen:

- Los bloqueos de tabla han sido sustituidos por el control de concurrencia multi-versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante la actualización de registros, y permite copias de seguridad en caliente desde pg_dump mientras la base de datos permanece disponible para consultas.
- Se han implementado importantes características del motor de datos, incluyendo subconsultas, valores por defecto, restricciones a valores en los campos (constraints) y disparadores (triggers).
- Se han añadido funcionalidades en línea con el estándar SQL92, incluyendo claves primarias, identificadores entrecomillados, forzado de tipos cadena literales, conversión de tipos y entrada de enteros binarios y hexadecimales.
- Los tipos internos han sido mejorados, incluyendo nuevos tipos de fecha/hora de rango amplio y soporte para tipos geométricos adicionales.
- La velocidad del código del motor de datos ha sido incrementada aproximadamente en un 20-40%, y su tiempo de arranque ha bajado el 80% desde que la versión 6.0 fue lanzada.

http://www.postgresql.org/docs/8.0/static/release.html

Muestra enlaces a paginas de las versiones (release) que han salido y sus características.

En el servidor Cormorán fue implementado PostgreSQL en la versión 8.0 el comportamiento de esta herramienta ha sido bueno y estable, sin embargo no se ha hecho la configuración de postgres que permita la recolección de estadísticas o no existe una aplicación implementada que permita hacer un seguimiento medible del comportamiento del manejador de base de datos (uso de memoria, cantidad de registros manejados o registros de actividad), esto con el fin de tener datos

claros que permitan generar políticas de administración mas acertadas.

7.4. PLATAFORMAS DE PROGRAMACION (PHP Y JAVA).

7.4.1. Análisis de lenguaje de programación PHP.

Php es un Lenguaje de programación interpretado orientado a la web escrito en Perl (Lenguaje de programación Interpretado) por Rasmus Lerdorf en 1994, mas tarde 2 Programadores Israelies Zeev Suraski y Andi Gutmans redefinieron el analizador sintáctico de PHP y sentaron las bases de PHP3, versión que se hizo publica en 1998.

En el año 2000 se lanzo la versión 4 que usaba como motor Zend Engine 1.0 y que duro hasta el 2007 en su versión 4.4.7. PHP5 se lanza el 13 de Julio del 2004 y la ultima versión estable hasta la fecha (diciembre del 2007) es 5.2.5 en este mismo año php.net (web site oficial de php) anuncio la discontinuidad de este versión 4.

A continuación una lista de las características incluidas en PHP5.

- Soporte sólido y REAL para Programación Orientada a Objetos (OOP) con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte para MySQL con extensión completamente reescrita.
- Mejor soporte a XML (XPath, DOM...).
- Soporte nativo para SQLite.
- Soporte integrado para SOAP.
- Iteradores de datos.
- Excepciones de errores.

El lenguaje interpretado PHP se ha caracterizado por una fácil instalación y configuración. Los sistema gestores de base de datos como MySQL y Postgres gozan de un gran soporte en PHP y cuentan con librerías, así como la fácil integración con sistemas servidores de paginas web como Apache, que ofrecen diversos recursos permitiendo la interoperabilidad y un ambiente de desarrollo

integrado orientado a la web.

Las aplicaciones y sitios web desarrollados en PHP y que residen en el servidor Cormorán han tenido un comportamiento estable aunque la versión de PHP en el servidor es la 4.0, este hecho hace que cada vez los programadores estén limitados al uso de los nuevos recursos ofrecidos por las nuevas versiones (PHP5).

7.4.2. Análisis de JVM (JAVA VIRTUAL MACHINE – MAQUINA VIRTUAL DE JAVA).

"La Java Virtual Machine (Maquina Virtual de Java) es un conjunto de aplicaciones software y estructura de datos que implementan un modelo un modelo de maquina virtual".

La Maquina Virtual de Java nace de la necesidad de permitir la ejecución de aplicaciones en diferentes plataformas a partir de un único lenguaje de programación, debido a que el cambio entre plataformas implicaba la reelaboración de la aplicación a partir del código fuente Sun Microsystems a finales de los años 80 empezó el desarrollo de una plataforma orientada a dispositivos electrónicos diversos.

Inicialmente la intensión de Sun, era abarcar el mercado de masas de electrodomésticos en los que desarrollo algunos proyecto que no tuvieron la acogida esperada, algún tiempo en 1991 de la mano de James Gosblin y Mike Sheridan se inicia un proyecto (Green Proyect) se escribe el primer compilador junto con un grupo de desarrolladores junto con un decodificador al que denominaron mas tarde Java. Con el auge y crecimiento de internet Sun Microsystems apuesta al desarrollo orientado a la web e integra la maquina virtual elaborada, que aunque primitiva por ese entonces, cumplía con algunos estándares que permitían acoplarla ya que estaban diseñadas en un código independiente de la plataforma, eso permitió acoplarlo con los navegadores web. Los Java Applets (elementos java que permiten ser descargados en el navegador y ser usados a nivel local) ganaron un espacio importante en el mercado y fue pionero en los recursos dinámicos en la web, en este proceso grandes empresas como Novell, IBM, Symantec, Toshiba, Microsoft, SPARK entre otras

implementaron y adquirieron los permisos para el desarrollo de la maquina virtual de Java en sus respectivas plataformas así como el desarrollo de aplicaciones orientadas a las tecnologías JAVA.

La Maquina Virtual de Java y las tecnologías Java en general han tenido y en la actualidad conservan un protagonismo innegable en el desarrollo de aplicaciones orientadas a la web y sistemas multiplataforma y teniendo en cuenta que Sun Microsytems liberó el código bajo licencia GPL (General Public License) se impulsa al desarrollo de nuevas comunidades de desarrollo y motiva la permanencia de las tecnologías Java en el mercado.

La Escuela de Ingeniería de Sistemas e Informática ha madurado grupos de investigación y desarrollo, algunos grupos han incursionado en el campo de ambientes y tecnologías orientadas a la web con diferentes propósitos, en el desarrollo de estos proyectos en algunos casos se están usando tecnologías Java específicamente tecnologías JSP (Java Server Page), servlets y beans que son recursos tecnológicos basados en la arquitectura Java.

El servidor cormorán soporta algunas aplicaciones web entre ellas la web oficial de la EISI, este sitio esta desarrollado en su mayoría en tecnologías Java. Actualmente el servidor cuenta para el soporte de estas tecnologías con la Java Runtime Environment (Ambiente de ejecución de Java, JRE) Standard Edition 1.42.2-b28, Java versión 1.4.2 (Compilador de Java), Java HotSpot Client VM versión 1.4.2-b28.

El comportamiento de la plataforma Java ha sido estable y ha permitido un proceso de desarrollo que se puede considerar normal debido a que no ha presentado problemas, sin embargo, la plataforma establecida para servir paginas JSP (Java server page) esta configurada con un servidor web tomcat versión 4 desactualizado que si bien ha sido suficiente en términos de recursos ofrecidos también puede llegar a ser una limitante en el aprovechamiento de las nuevas características que implementan las nuevas versiones o la correcciones de errores.

Como conclusión se puede decir que en la medida que se cuenta con una plataforma Java actualizada y estable, los nuevos grupos podrán aprovechar los

nuevos recursos de la plataforma.
A la fecha, la plataforma Java estable esta en la versión 6 (1.6) update3.

7.5. ANÁLISIS DEL AGENTE DE TRANSPORTE DE MAIL SENDMAIL.

Sendmail es uno de los Agentes de Transporte de Correo (MTA, Mail Transport Agent) que mas goza de popularidad entre las comunidades de administradores de sistemas, grupos de software libre, código abierto y usuarios de plataformas UNIX en general.

Los MTA, son aplicaciones que permiten el envío de correos electrónicos a través de una red especifica hasta un destino final. A través de los MTA se puede establecer un servicios de mensajería adaptado a las necesidades de cada red, es decir, se puede adaptar la funcionalidad del servicio de correos de una determinada red a partir de la configuración local de un servidor que permita el envío de mensajes electrónicos desde y hacia la red dependiendo de las necesidades de la misma sin depender de terceros (servicios de mensajería ofrecidos por algún componente externo).

Sendmail desciende de la aplicación ARPANET delivermail (una aplicación que se valía de ftp para la transmisión de mensajería cuando por ese tiempo se usaba el protocolo ARPANET) aplicación que fue integrada a 4.0BSD por el año 1979 que usaba NCP como protocolo (Network Control Program, Programa de Control de Red), La primera version de Sendmail fue escrita por Eric Allman a comienzo de 1980 en la universidad de Berkeley quien también previamente escribió delivermail, aunque fue en 1983 cuando Sendmail fue integrado con 4.1cBSD (La primera versión BSD en incluir TCP/IP).

Una vez el paradigma de comunicación cambio de ARPANET a TCP/IP el mercado rápidamente se lleno de aplicaciones MTA (Mail Transport Agent) entre las cuales Sendmail podríamos considerar tuvo años de gloria, pero que en la actualidad se han disminuido ante la gran oferta y opciones de MTA disponible en el mercado. En noviembre del 2001 Sendmail contaba con una cuota de mercado del 42%, en agosto del 2007 una cuota de 29.4% (según un estudios de E-Soft).

Sendmail puede ser visto por algunos administradores inexpertos como complejo

debido a la cantidad de funcionalidades que tiene y a que la documentación de Sendmail y las herramientas no sale con la misma velocidad que se hacen cambios en la herramienta, es por esto que una u otra forma sea complejo para algunos administradores llevar a punto el servicio de mensajería con sendmail.

En los inicios de sendmail, la seguridad no era un elemento que fuera en sus inicios estimado como se debía y esto le ha costado a sendmail grandes problemas de reputación en cuanto a su confiabilidad, en su historia existen ya varios problemas críticos de seguridad que han creado (y con justa razón) desconfianza ante los administradores de sistemas, por lo que muchos han optado por opciones como qmail (MTA que se ha caracterizado por su robustez casi ningún problema de seguridad registrado, aunque este no cumple con el tipo de licencia de software libre, es license-free software).

Sendmail es el MTA de cormorán, actualmente, en el Servidor Cormorán no esta funcionando la ultima versión de sendmail.
http://www.obssys.com/index.php?id=2264

Esta dirección muestra una lista de algunos de los bugs en sendmail con la respectiva fecha y versión.

La pagina oficial de sendmail es: http://www.sendmail.org/

El servicio de correo resulta útil para algunos sitios que entre sus servicios ofrecen notificaciones a través de correo electrónico. El comportamiento del servidor de correos se estudia a partir de los logs (archivos de registro del sistema) del sistema y de las notificaciones hechas por el sistema operativo. Hasta el momento no existen registros de algún comportamiento en especifico de sendmail y tampoco estadísticas del envío de correos.

El servicio de correo no cuenta con el apoyo de herramientas antivírus para la verificación de archivos transmitidos pero tiene definidas reglas de no-relaying (falsificación de dominio en el envío de correos).

7.6. ANALISIS DEL FIREWALL

Linux en su versión 1.1 contaba con sistema de filtrado de paquetes pero fue en 1994 cuando Alan Cox cuando en un debatido proceso debido a los tipos de licencias hizo potable la versión del paquete ipfw de BSD a Linux, debido a al revuelo que causo esto, el paquete fue reescrito y el código reemplazado. Para la versión 2.1 del kernel Linux por el año 1998 fue incluida en el kernel la utilidad ipchain debido a la incompatibilidad del antiguo kernel de Linux de manipular fragmentos de paquetes, finalmente en 1999 aparece la cuarta generación firewall en el kernel 2.4 de Linux, iptables.

Iptables es una aplicación contenida en el framework Netfilter que está enbedido en el kernel de Linux. Este framework ofrece un conjunto de aplicaciones orientadas a las conexiones seguras y al manejo y filtrado de paquetes. Iptables es una herramienta que permite el control de flujo de paquetes Ipv4 y filtrado NAT en una red, con esta herramienta podemos establecer reglas de seguridad en el acceso de una red o sistema a los paquetes de la red, es decir, que a partir de reglas establecidas por el administrador del sistema podemos decidir cual debe ser el tratamiento que se le debe dar a cada paquete que entra o sale desde el sistema (servidor) o una red interna (intranet) en la que el servidor sea la entrada a la red.

Iptables se puede comportar como un firewall en el que según las reglas se permite el acceso a determinadas estaciones de trabajo y servicios específicos, por ejemplo podríamos negar el acceso al servicio de MySQL a todas las IP que no pertenezcan a la red interna, o también podríamos permitir el acceso a una dirección IP especifica al servicio de FTP (File Transfer Protocol, Protcolo de Transferencia de Archivos).

En conclusión, se puede decir que iptables es un herramienta libre que permite definir reglas precisas y configurables de acceso o restricción al servidor Cormorán, de esta forma se cuenta con una herramienta de filtrado que apoye el proceso de Administración desde la seguridad.

En el servidor no existen reglas definidas de seguridad en iptables.

8. ASEGURANDO EL SISTEMA

El proceso de configuración para asegurar el servidor Cormorán está definido en el tercer informe de avance, a continuación se obvian detalles de la configuración del servidor ya que representan información sensible y es de uso interno de Cormorán.

Hasta el momento, el servidor se ha configurado de modo que puede ser operativo, y puede pensarse que podría estar listo para entrar en fase de producción, sin embargo, no es cierto. Un servidor es un elemento susceptible a cambios ya que este ofrece ciertos servicios y estos servicios o aplicaciones son depurados, actualizados o mejorados diariamente por las casas desarrolladoras o grupos de soporte; estos cambios se deben asumir o no según las necesidades.

Un servidor representa un recurso importante y en algunos casos vital ya sea en procesos académicos, cálculos matemáticos, manejo de información, administrar otros servidores entre otras funcionalidades. Se puede decir que un servidor es una herramienta versátil que esta sujeto a cambios, ya sea por su disponibilidad, capacidad de ejecutar las tareas necesarias, escalabilidad, seguridad o un conjunto de estas.

Cormorán, es un servidor que se caracteriza por apoyar algunos procesos de administración y académicos en la EISI, y como la mayoría de servidores es sensible al tipo de información que se maneja, es decir, que la información contenida en el servidor solo debe ser manipulada por determinados usuarios o grupos.

El tercer informe tiene por objetivo mostrar algunos procesos de configuración que fueron ejecutados con el fin de aumentar grado de seguridad del Servidor Cormorán.

De igual forma el control sobre cada elemento representa esfuerzos mas grandes en el proceso de administración de servidores, pero este esfuerzo bien vale la pena en términos de garantizar la integridad de la información, además, se pueden usar recursos que permiten mecanizar procesos, pero esta es una practica que se

debe hacer consiente de que los aplicaciones informáticas cambian constantemente y las medidas o mecanismos que representan cierto estado de seguridad, pueden no ser útiles bajo ciertas/otras situaciones.

8.1. SERVIDORES WEB.

Los servidores web son herramientas orientadas a la transferencia de hipertexto (protocolo HTTP hypertext transfer protocol), permiten la transferencia texto complejo, imágenes, sonido, animaciones, etc. Los servidores web funcionan en una arquitectura cliente-servidor y se valen de un navegador web, que es una aplicación que permite el envío y recepción de hipertexto.

Estos servidores web se pueden obtener en diferentes formas bien sea el código fuente para ser configurado y compilado (según su naturaleza), o archivos binarios precompilados, estos últimos vienen con una configuración nativa dependiendo del lugar (repositorio fuente) donde sean obtenidos. En el caso de la versión oficial de Tomcat (desde los servidores oficiales del proyecto Tomcat) la configuración por defecto es mas permisiva en comparación GNU/Linux Debian por ejemplo, el paquete precompilado de Tomcat viene con una configuración al parecer orientada a desarrolladores, de igual forma pasa con otros paquetes como el servidor web Apache. En este caso, se analizarán aspectos bien importantes a la hora de contar con servidores web y una configuración apropiada que descarte ciertos problemas de seguridad conocidos. Cuando se instala una aplicación, se debe intentar garantizar que los paquetes, códigos fuentes, aplicaciones, binarios o cualquier otro elemento, no hayan sido modificados por terceros de forma indeseada. En el proceso de instalación de servidores web lo recomendable es descargar los códigos fuente (no archivos precompilados), hacer el proceso de configuración, es decir, definir los módulos y las opciones para instalación, compilar el código fuente y ejecutar el proceso de instalación, en estos procesos se deben estimar varios factores: Cuando se descargar una aplicación a través de una red o internet, se debe tener la plena seguridad de que los paquetes descargados son íntegros. Para esto , normalmente se usan sumas de chequeo md5, que permiten validar la integridad del paquete descargado con el ofrecido por algún servidor foráneo.

Las sumas MD5, son funciones de seguridad aplicadas a archivos que retornan un

valor especifico (llave), estas deben coincidir, es decir la suma MD5 aplicada en el archivo local con respecto a la suma MD5 del paquete descargado desde el servidor foráneo deben ser iguales. Este hecho de alguna forma garantiza que el paquete descargado concuerda con el paquete del servidor, en otras palabras es una copia idéntica.

En el proceso de administración de sistemas, es una practica frecuente tener el control de los elementos usados desde la instalación, configuración y todas las alteraciones que sean necesarias para aumentar el estado de seguridad de las aplicaciones, esto se debe a que entre mas elementos el administrador del sistema controle, menos situaciones se obviaran y por tanto la incertidumbre por la omisión de procesos disminuirá, hecho que aporta en gran medida a la seguridad de un sistema.

8.1.1. Elección y algunas características de los servidores WEB.

Aunque los servidores web están hechos para funcionar bajo un determinado protocolo, estos pueden ser desarrollados de forma diferente ya que el diseño, arquitectura, tecnología a la que están orientados, etc. puede cambiar, por esto es común encontrar que la forma de administrar estos servicios y la forma en la que opera cada uno puede variar de forma considerable. Algunos servidores web como Apache funcionan de forma modular y la configuración se hace respecto a cada modulo, Tomcat en cambio se vale de paquetes que conservan una estructura orientada a la plataforma de Java (JVM Java Virtual Machine), otros servidores están orientados a labores especificas como la depuración de aplicaciones CGIs (common gateway interface). Existe un gama amplia de servidores y cada uno con características particulares que pueden ser empleadas según la necesidad, algunos de estos son: Apache, Tomcat, AOLServer, Sun Java System WebServer, iTools, JigSaw, Hawkeye, RapidSite, Zeus Web Server.

En el montaje de un servidor web como en la mayoría de servicios, es importante definir requerimientos del tipo de solución deseada. En el caso de un servidor web, se pueden estimar aspectos como el tipo de licencia que cubre la aplicación software, el soporte de la herramienta (grupo de desarrollo que soporta la aplicación), soporte de otras tecnologías, rendimiento, robustez, seguridad, etc. de igual forma estos aspectos cobran mayor o menor relevancia según la necesidad especifica. Por ejemplo, en el caso de un desarrollador de aplicaciones web que

trabaja desde su casa o una estación de trabajo individual probablemente no se preocupara por factores como la seguridad ajenas a su aplicación o la tolerancia a fallos de su servidor web, por otra parte, podría ser importante la eficiencia del servidor o la facilidad de implementación para hacer pruebas locales.

En el caso de un servidor las características de los servicios web se deben analizar con mas detalle y por tanto la elección del tipo de servidor es un proceso relevante.

Entre las características mas relevantes tenidas en cuenta en la elección de servidores web en el desarrollo de la practica fueron:

Tipo de licencia: Un servidor web con licencia GPL, en otras palabras software libre. Soporte: Que la aplicación cuente con una comunidad amplia en el proceso de desarrollo, detección de errores y documentación.
Soporte de tecnologías: Un servidor web que permita implementar las aplicaciones web desarrolladas residentes en el servidor y que soporte gran parte de las tecnologías disponibles en el mercado orientadas a los servicios web como certificados, validación de usuarios, alta disponibilidad etc.
Eficiencia: Un servidor web que responda en tiempos adecuados según el volumen de peticiones (este requerimiento no se baso en una métrica estricta de comparación, la estimación se hizo de forma cualitativa.)
Seguridad: Este aspecto es resultado de un proceso de soporte continuo del grupo de desarrollo de la herramienta, además este proceso esta sujeto al correcto proceso de administración del servidor.

8.1.2. Configuración del servidor WEB.

Dado que la naturaleza de cada servidor web es diferente, también es de esperar el proceso de instalación y configuración en el caso de la practica se estimaron los servidores web para paginas estáticas y PHP y otro servidor para paginas JSP. Los detalles técnicos de la configuración hace parte de la documentación de cormorán, esta información no es de acceso publico.

Servidor web de paginas estáticas y PHP.
En la instalación de los servidores se debe validar la suma de chequeo md5 para verificar la integridad de los paquetes. Se debe analizar los módulos que son

usados de acuerdo a la necesidad por el servidor de forma nativa, de ser necesario es posible agregar módulos pero, el uso innecesario de estos representa cierto gasto de memoria y el aumento de la probabilidad de tener vulnerabilidad ya que los módulos también son software que puede ser vulnerable. Se debe identificar los módulo extra necesarios para hacer el servidor web compatible con aplicaciones externas como sistemas gestores de base de datos lenguajes de programación, módulos de seguridad etc.

Aunque normalmente los permisos de acceso de los archivos de las aplicaciones instaladas por defecto tiene las suficientes restricciones, no esta de mas verificar los permisos de los archivos de configuración del servidor y de los módulos.
Se debe revisar los archivos de configuración ya que estos traen configuraciones por defecto que son abiertas y pueden no ajustarse a necesidades especificas, entre estas configuraciones se encuentra:

Acceso al directorio raíz: Es el tipo de privilegios que se tiene para poder acceder esta partición, en el caso del servidor debe ser lo mas restrictiva posible
Raíz de documentos del servidor: Es la ruta en la que se encuentra la aplicación del servidor web.
Nombre Canónico: Es un prefijo identificador canónico del servidor web del cual cuelgan las aplicaciones montadas en el sistema.
Permisos de las aplicaciones instaladas: Estos tipo de privilegios no deben ser mas que los suficientes para que la aplicación sea operativa y segura (independiente a la seguridad inherente de la aplicación web instalada), en este caso debe haber políticas de administración y un canal de comunicación entre los desarrolladores y administradores para la notificación de permisos.
Limite de métodos: En el proceso de comunicación web podría ser necesario validar usuarios para permitir/restringir algún método especifico en alguna aplicación. El uso de estas directivas puede significar la implementación de algunos módulos para su funcionamiento.
Peticiones anidadas: Estas directiva se deben emplear cuando es necesario limitar la cantidad de direccionamientos anidados en las aplicaciones.
Cuerpo de la peticiones: La correcta definición de estas directivas evitará el mal desempeño del servidor web a peticiones grandes o malintencionadas con el propósito de mermar la actividad del servidor atendiendo una petición.
Tamaño de los campos: Los campos en las peticiones hechas en el servidor

también representan espacio de memoria que debe ser procesado, por tanto se deben establecer los parámetros según las necesidades del servidor.
Tamaño del cuerpo de peticiones xml: Los peticiones xml también disponen de una cantidad de memoria especifica y este tamaño debe ser tenido en cuenta.
Tiempo de espera: Dado que las peticiones tienen un tiempo estimado de respuesta, es prudente que este tiempo no se muy grande ya que puede disminuir la la eficiencia del servidor ocupando espacio de memoria mientras se espera por algún evento.
Procesos Hijo: Se debe definir una cantidad de procesos hijos en el arranque que permitan hacer de la recepción de peticiones un proceso mas eficiente, pero se debe estimar que estos procesos hijos implican memoria y puede agotar ciertos recursos.
Procesos Hijos inútiles: El uso de proceso hijos permite respuestas mas rápidas pero, de igual forma es importante tener una cantidad suficiente sin recargar el sistema, por tanto se debe definir la cantidad máxima y mínima de procesos hijos.
Peticiones por proceso hijo: Es la cantidad máxima peticiones que puede manejar cada proceso hijo, este parámetro se debe ajustar según la necesidad particular del servidor.
Tipo de registros a notificar: Debido a que los registros permiten hacer un buen análisis del comportamiento del servidor web, es útil contar con una buena cantidad de información sin que eso represente un proceso traumático en el análisis.

8.1.3. Configurando SECURE SOCKET LAYER, SSL (CAPA DE SEGURIDAD DE SOCKET).

SSL es un protocolo diseñado por Netscape Communications Corporation pretende aumentar la seguridad a partir del cifrado autenticado de datos intercambiados en las comunicaciones cliente-servidor en los niveles: Internet (IP), Transporte (TCP) y aplicación (http, ftp, telnet, etc).
El proceso se puede explicar según el siguiente esquema:

- Definir el algoritmo que se usará en la comunicación (cliente-servidor)
- Intercambio de claves publicas y autentificacion basada en certificados digitales.
- Cifrado del transito basado en cifrado simétrico.
- Aunque una definición mas detallada podría ser:
- El cliente envía un mensaje “HelloClient” especificando el protocolo TLS

(capa de seguridad de transporte) mas elevado que soporte, un numero aleatorio, una lista de cifrado y métodos de compresión.

- El servidor responde con un "ServerHello", contiene el protocolo seleccionado, un numero aleatorio y el método de cifrado escogido por el servidor de la lista del cliente.
- El servidor envía su certificado (dependiendo de la conjunto de cifrado), este certificado normalmente es X.509 (una infraestructura estándar de comunicación para llaves publicas), también hay certificados basados en OpenPGP.
- El servidor puede requerir el un certificado al cliente para que la comunicación se mutuamente autenticada. Nota: En el caso del servidor Cormorán, este hecho hasta el momento no es relevante y no es necesario que el cliente sea validado.
- El cliente y el servidor definen una clave secreta común denominada "Mater Secret" esto se puede con intercambio de Diffie-Hellman (un protocolo que permite el intercambio de llaves secretas) o cifrando una clave secreta con una clave publica que se descifrara con la clave privada de cada uno (cliente/servidor). Los datos claves restantes son generados a partir del "Mater Secret" y los valores generados por el cliente y el servidor, estos son comprobados a través de una función pseudoaleatoria escogida.

Una vez se tiene un idea del proceso de intercambio de información cifrado, es necesario tener una idea de la forma como se implementan los certificados.

Básicamente el proceso consiste en generar un certificado en el servidor, este, con el propósito de cifrar los datos que se comparten con el cliente, pero, no basta con cifrar la información para que el cliente esté satisfecho y de alguna forma tenga la certeza de estar intercambiando su datos con el servidor que desea. Para solucionar este problema, se debe acudir a un tercero, entidades certificadoras. Estas entidades tiene mecanismos que permiten establecer que un determinado certificado pertenece realmente a quien dice ser, es decir, las entidades certificadoras notifican al cliente acerca de la valides o no de un certificado, este mecanismos se hace a partir de firmas digitales, previamente el servidor solicita a una entidad reconocida mundialmente como certificadora su firma digital en el certificado, de esta forma cuando el cliente tiene el certificado del servidor, el cliente puede validar ante la entidad certificadora la identidad del servidor. Este

hecho hace que el cliente escoja muy bien cuales son las entidades certificadoras en las que realmente confía. Los certificados permiten cifrar la información entre el cliente y el servidor y aumentar la seguridad de la conexión sin embargo si los certificados no están firmados por una entidad certificadora (CA), los usuarios de las aplicaciones web no tendrán la seguridad de estar solicitar la información al servidor Cormorán.

El servidor Cormorán cuenta con cifrado de aplicaciones web a partir de certificados de seguridad y todas las aplicaciones web funcionan bajo protocolo https. La gráfica tomada de www.securityfocus.com permite ilustrar el proceso.

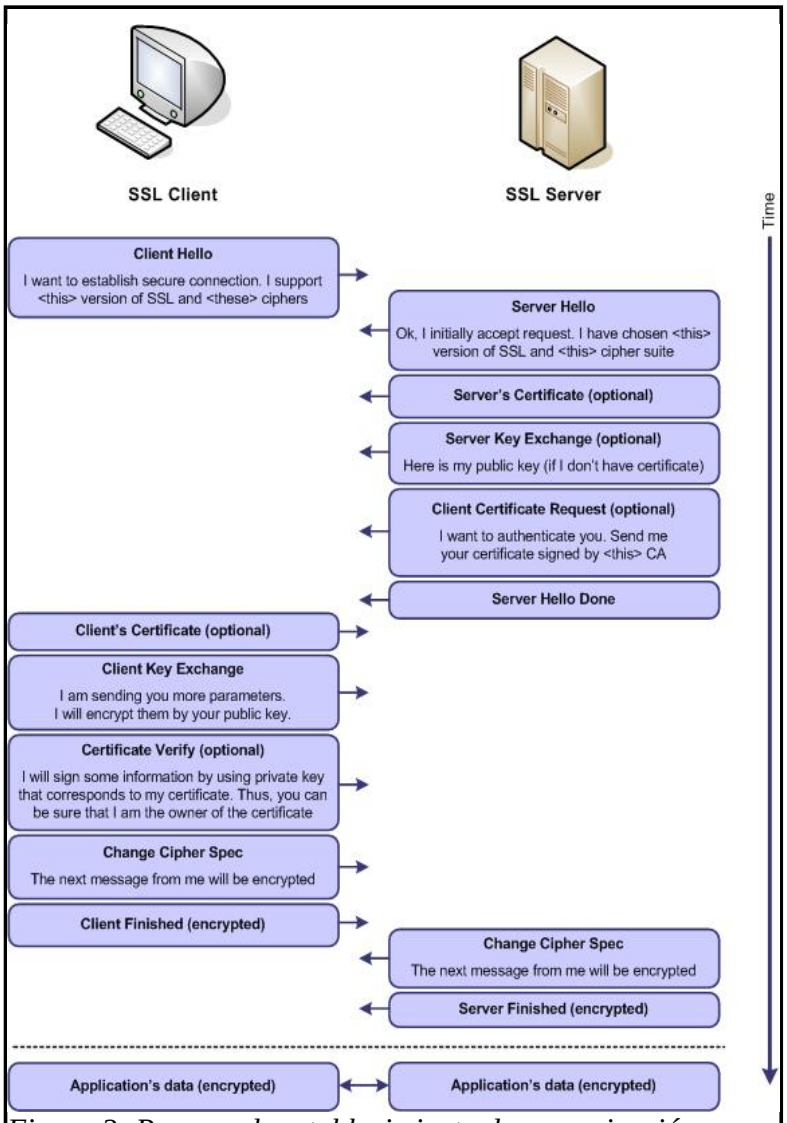

Figura 3: Proceso de establecimiento de comunicación con SSL. (Fuente www.securityfocus.com)

8.1.4. Encapsulando los servicios WEB.

En los sistemas tipo UNIX como GNU/Linux Debian es posible aplicar una técnica que encapsular un determinado proceso del sistema, esta permite cambiar el directorio raíz de una aplicación por un directorio raíz virtual, esto, con el propósito de mapear "encapsular" y definir un espacio de memoria para este proceso. En GNU/Linux esta característica es posible a través de la aplicación chroot.

El encapsulamiento de servicios representa un paso importante en la seguridad de un sistema porque permite definir un espacio de memoria no compartido especifico para un servicio determinado servicio, una de los problemas mas grande de seguridad por los que ha pasado la informática son los ataques por desbordamiento de memoria, a veces las aplicaciones no emplean técnicas correctas de programación o no se validad rigurosamente ciertos procesos, esto permite que sea posible la introducción de código de forma manual (con previo análisis de la aplicación), cuando estos eventos son posible algunas aplicaciones son susceptibles a ser manipuladas en la memoria, y de igual forma permitir instrucciones en esta que permitan redireccionar la memoria a otros sectores de memoria dejando a potestad del código insertado siguiente instrucción a ejecutar.

El encapsulado o enjaulado de una aplicación restringe el uso de la aplicación a si misma, es decir, no se puede referenciar elementos que estén fuera del encapsulado o jaula. Este mecanismo de protección es útil a ataques por desbordamiento de pila en el servidor web o direccionamiento de scripts a recursos de la aplicación web.

En el caso del servidor web Apache, es posible implementar módulos que permitan encapsular la aplicación, en el servidor Cormorán se implementa este mecanismos de protección, el proceso de configuración hace parte de la documentación interna de Cormorán.

8.1.5. Configuración de módulos de seguridad en el servidor WEB.

En el proceso de desarrollo de Cormorán no se han establecido políticas de seguridad ni una normativa que oriente el desarrollo de los sitios y aplicaciones residentes en el servidor con el propósito de garantizar la integridad de las aplicaciones, esto se debe básicamente a que el proceso de implementación de software en el servidor Cormorán no está definido y la implementación de este

implica un método de integración entre los desarrolladores y la administración del servidor. El requisito para la publicación web de contenidos consiste en una gestión de solicitud de espacio en el servidor y una aprobación.

Debido a que no se tiene control de: Las instrucción contenidas en las aplicaciones web, de la omisión de procesos de validación en las aplicaciones, ni de las peticiones hechas al servidor por parte del cliente, se genera incertidumbre y este es un elemento indeseable en el proceso de seguridad.

Los desarrolladores son propensos a cometer errores ya sea por técnicas incorrectas de programación, omisión de validaciones, bugs de las herramientas usadas, etc. además, lo usuarios representan un riesgo para si mismos en la medida que no practican políticas de seguridad comunes, como el cuidado de sus claves y la correcta administración de sus propios contenidos, a este evento se debe sumar que el sysadmin (administrador del sistema) ejecute practicas de actualización y correcta configuración de las herramientas disponibles en el servidor.

Es una practica común actualmente los ataques a las aplicaciones web desarrolladas mas que a las herramientas implementadas en el servidor, esto como resultado de los sencillos métodos de actualización de las aplicaciones que permiten a los administradores de sistema corregir un error, horas después de haberse notificado y por tanto el tiempo de riesgo del sistema disminuye considerablemente, sin embargo, las aplicaciones web son mas propensas a recibir instrucciones con propósitos ajenos a los que fueron construidos, por ejemplo, implementar un corta-fuegos (firewall) con reglas estrictas de acceso no garantiza integridad en la información, ya que si se ofrece un servicio a una determinada comunidad, este servicio aun es susceptible a recibir instrucciones que comprometan la seguridad, ahora, teniendo en cuenta que se puede cifrar la información entre el cliente y el servidor se puede pensar que la información sera de alguna forma protegida a terceros y de igual forma el corta-fuegos (firewall) no podrá aplicar sus reglas a una solicitud cifrada.

Por lo anterior, se hace necesario un mecanismo que permita validar las solicitudes al servidor web Apache.

Existen mecanismos basados en expresiones regulares que permiten validar el tipo de peticiones que se hacen al servidor web que permiten hacer del servidor una herramienta reactiva que pueda filtrar el tipo de peticiones a las que es expuesta. Aunque este mecanismo no representa total seguridad, de alguna forma ayudará a incrementar en gran medida la seguridad ya que valida gran cantidad de peticiones malintencionadas comunes basado en el motor de reglas.

Aunque el motor de reglas basado en expresiones regulares definido para filtrar las peticiones es generado por terceros, estas reglas se pueden considerar acertadas ante vulnerabilidades comunes y errores resultado de practicas indebidas en la programación de aplicaciones web, además, el núcleo de reglas es flexible porque permite tanto la modificación de reglas así como la creación de nuevas. Algunas pruebas en el servidor Cormorán demostraron la notificación y filtrado de peticiones ante ciertos tipos de ataques. A la fecha el mantenimiento del núcleo de reglas ha sido continuo y dada la naturaleza de un proyecto de esta naturaleza (paquete orientado al filtrado de reglas) es bastante probable que se mantenga durante buen tiempo ya que son pocos los módulos que son integrable a navegadores web de esta forma.

8.1.6. Configuración del servidor web para paginas JSP.

El servidor de paginas JSP, también debe pasar por un proceso de configuración para ajustarlo a las necesidades particulares de Cormorán. En este caso se estima el uso de un servidor web que se encarga de recibir peticiones JSP (Java Server Page) y un servidor web que reciba el resto de peticiones como documentos html y php, este proceso debe ser transparente al usuario de aplicaciones web en el servidor Cormorán. El servidor de paginas web de JSP también debe ser configurado. Nuevamente, los destalles del proceso de configuración del servidor hacen parte de la documentación interna del proceso de administración del servidor.

Antes de la instalación se debe verificar la integridad de los paquetes de instalación del servidor web con sumas md5, es importante estudiar la documentación y conocer el estado de desarrollo del servidor así como las nuevas características en la configuración debido a que suelen existir cambios importantes en el funcionamiento entre versiones.

En el proceso de instalación del servidor web para JSP se debe definir el directorio de instalación así como un nuevo usuario que tendrá privilegios sobre la ejecución de los servicios del servidor web. Los privilegios sobre los archivos de configuración deben ser estrictos y se deben revisar, en lo posible debe ser el usuario dueño del servidor (creado anteriormente) quien tenga acceso a los archivos de configuración del servidor y permitir determinados privilegios de lectura para el correcto funcionamiento del servidor. Las aplicaciones que residen en el servidor deben tener un usuario dueño quien sera encargado de dar el respectivo soporte, en este caso también se debe estar seguro de no permitir el acceso a usuarios ajenos a la aplicación en lectura, escritura o ejecución. De igual forma se debe definir el dueño de los archivos de registro de actividades así como los privilegios sobre estos archivos, se debe restringir el acceso a estos archivos de registro ya que estos pueden contener información sensible de las aplicaciones y representar posibles vulnerabilidades.

Debido a que el servidor debe contar con permisos de escritura en un directorio de archivos temporales, es importante incrementar la seguridad de estos archivos y restringir su uso solo al usuario creador, de esta forma se puede evitar la posible lectura desde terceros.

En el servidor web de paginas JSP como en la mayoría de aplicaciones, se debe implementar únicamente los recursos que sean necesario, es decir, no agregar módulos o funcionalidades que no representen una utilidad real y que por el contrario pueden aumentar la probabilidad de contar con bugs o problemas de seguridad.

Los servlets desarrollados en la arquitectura java, cuentan con estructura definida en la documentación de estos para que no respondan en caso de encontrar archivos de “bienvenida”, esto evita que las aplicaciones respondan bajo peticiones que no deben.

La publicación de las versiones de los servidores web a través de los navegadores o servicios prestados es un indicador muy utilizado en el escaneo de vulnerabilidades, entre menos información se ofrezca de las aplicaciones residentes en el servidor una posible petición malintencionada tomará mas tiempo en identificar posibles vulnerabilidades y tal vez el autor desista de su intento mas

rápido, por esto es importante dar la información que se necesita estrictamente ya que en ultimas los usuarios necesitan información de los servicios de las aplicaciones web y no de las aplicaciones del servidor. Sin embargo, esconder la información de las aplicaciones no representa una política estricta ya que las aplicaciones se caracterizan por un determinado comportamiento y para un experto estas características son identificables.

Cada grupo de desarrollo es responsable del correcto control de flujo de sus aplicaciones y por tanto de la correcta notificación en caso de error de las aplicaciones. Se convierte entonces en una política del servidor el correcto uso los mensajes de error con el fin de evitar notificar errores con código fuente al cliente e información sensible que puede representar perdida de la seguridad.

Dado que el servidor web de paginas JSP es también un servicio en el servidor, es necesario que se estudie como tal y verificar su comportamiento la forma en la que se inicia, apagar, reinicia etc. En este proceso algunos servicios pueden ser susceptibles a recibir peticiones en determinados puertos con determinados parámetros para el inicio del servicio o apagado, estos eventos se deben estimar y se deben configurar correctamente para su comportamiento sea el apropiado.

En el caso del servidor Cormorán es necesario hacer transparente a los usuario el uso de los servicios, por tanto se usa un servidor web como front-end de otro con el fin de recibir peticiones y reasignarlas según sea el caso, dado que existe un servidor especifico para recibir las peticiones JSP las peticiones de este tipo serán manejadas por este servidor, mientras que el resto de contenido sera manejado por servidor web de contenido estático y PHP. Una vez se define que el servidor web front-end reenvía las paginas JSP, es importante especificar al servidor de paginas JSP que solo recibirá peticiones de forma local, es decir, ya que existe un servidor que se puede considerar “primario” que reenvía paginas al servidor de JSP de forma local, no tiene sentido que el servidor de paginas JSP este abierto a peticiones externas.

8.2. SISTEMAS GESTORES DE BASES DE DATOS (SGBD).

Los sistemas gestores de base de datos son herramientas que facilitan el manejo organizado de la información, aunque no es objetivo de este trabajo un análisis

exhaustivo de los SGBD, se hará un análisis del proceso de configuración de estas herramientas en el servidor Cormorán. En este caso se enfatiza algunos aspectos en la seguridad de estas herramientas.

8.2.1. Asegurando MYSQL.

Una vez mas el proceso de instalación implica la revisión de la integridad de los paquetes a partir de la revisión de las sumas md5 con respecto a la fuente.

Una vez instalado MySQL es posible que contenga bases de datos de prueba, y usuarios de prueba, esto posiblemente para guiar a desarrolladores que empiezan el uso de la herramienta, en este caso el manejador de base de datos solo deben estar definidas las base de datos para el correcto funcionamiento, de igual forma los usuarios, esto implica eliminar lo que sobra.

El acceso remoto a las base de datos es de muy importante, en lo posible se debe restringir este tipo de acceso y se debe usar canales cifrados de información en la medida que sean necesarios, para ser estrictos en esta política se debe restringir el acceso a las bases de datos de forma remota desde la configuración del sistema gestor de base de datos y usar tuneles de información.

En el proceso de instalacion MySQL suele definir la clave de administrador como vacía, es importante definir esta clave dado los obvios riesgo que esto representa, la elección de la clave dependen de la administración de turno del servidor y es altamente recomendado usar claves robustas con no menos de ocho caracteres, mayúsculas-minúsculas y caracteres no tradicionales.

La mayoría de los sistemas gestores de base de datos en la actualidad permiten la lectura de archivos con el propósito de llenar posibles tablas a partir de archivos planos entre otros usos. MySQL no es la excepción, en este caso el uso de esta función no esta orientado al modelo de desarrollo de las aplicaciones contenidas en Cormorán y representan posibles salidas de información por tanto estas funciones son deshabilitadas.

8.2.2. Asegurando POSTGRESQL.

Postgresql es una herramienta que ha pasado por un proceso de desarrollo que se puede considerar amplio y esto le ha permitido convertirse en una herramienta

muy estable y robusta, sin embargo, como en el caso de MySQL es necesario hacer ajustes.

Es preciso verificar los usuarios contenidos en el manejador de base de datos y desechar los que no sean estrictamente necesarios para el correcto funcionamiento de la herramienta, de igual forma se debe verificar las bases de datos creadas en el proceso de instalación, aunque los usuario en este SGBD son tratados de forma diferente a MySQL también se hace necesario definir los privilegios de cada usuario de acuerdo al contexto. Los usuarios están directamente relacionados a los usuarios del sistema y según la configuración establecida estos tendrán permiso a funciones, procesos, tablas o bases de datos.

La seguridad en estos términos empieza desde la creación de las bases de datos; quienes son los dueños de estas y bajo que parámetros sera creada. Una vez definida, y restauradas las copias de seguridad, se debe definir los usuarios o grupos que tienen acceso a un recurso especifico, en Postgresql la definición de estas reglas se debe hacer de forma muy explicita, es decir, se conceden privilegios elemento por elemento.

Al igual que MySQL este Gestor de Bases de Datos permite el uso de algunas funciones que permiten el desarrollo y administración de las bases de datos de forma eficiente, sin embargo estas funciones son un arma de doble filo si se usan con propósitos malintencionados, dada esta situación, se debe restringir el uso de estas funciones a usuarios de aplicaciones del servidor y en general disminuir la probabilidad de vulnerabilidades a partir del uso de funciones que deben estar disponibles a los usuarios comunes.

Los permisos de acceso remoto a las bases de datos deben en los posible restringidos, sin embargo de ser necesarios, se pueden usar canales cifrados de comunicación, el cifrado de la validación de los usuarios debe ser fuerte, se recomienda md5.

8.3. LENGUAJES DE PROGRAMACION

8.3.1. Aseguranado PHP.

Php es un lenguaje de programación que ha ganado muchos adeptos por sus

grandes capacidades y por su corta curva de aprendizaje respecto a otros lenguajes de programación orientados a la web.

Desde el punto de vista del desarrollador la configuración de PHP puede no representar problemas, incluso muchos desarrolladores no conocen las características que se pueden manipular desde la configuración, esto, posiblemente como resultado de una configuración muy permisiva, no obstante la configuración de un lenguaje de programación es de vital importancia en el desarrollo de un servidor y mas aun si este lenguaje esta orientado a la web ya que de alguna forma esta expuesto a un medio externo y puede ser sensible al uso indebido de este. A continuación se definen algunos aspectos que contribuyen a la seguridad del sistema desde la correcta configuración de aplicaciones:

- PHP aunque es un lenguaje interpretado desde la óptica de servidor o sistema operativo, es un paquete o conjunto de estos, por tanto se debe validar su integridad a partir de la suma md5 con respecto a la fuente (la fuente debe ser oficial).

- Los registros globales en php han representado un punto de discusión grande a través de tiempo y que en la actualidad se haya notificado el no uso de registros globales en las futuras versiones puede dar una idea del uso de estos o no. En la configuración de php se estima el no uso de “Registros Globales”.

- Existen directivas de configuración que permiten la lectura de archivos remotos de forma transparente al usuario, aunque parece una ventaja para el usuario, desde la administración del servidor este hecho representa el posible uso de estas funcionalidades para mostrar información del servidor, aunque este hecho puede limitar las aplicaciones existen opciones que reemplazan esta funcionalidad.

- Es importante establecer las directivas que permiten definir los directorios del sistema que pueden ejecutar aplicaciones php, de esta forma se evita la ejecución de aplicaciones foráneas inyectadas en el servidor.

Las directivas que permitan el acceso a recursos de forma dinámica debe ser

revisado con detalle, las aplicaciones residentes en el servidor no tienen este servicio como una necesidad y por el contrario se evita el posible uso indebido de esta funcionalidad, por tanto son deshabilitadas estas directivas.

El control de limites es importante para establecer los alcances del lenguaje respecto al uso del cliente (usuario) y al manejo del servidor. De esta forma se puede controlar algunos parámetros de comunicación como el tiempo máximo de ejecución de aplicaciones, tiempo máximo de tratamiento de una entrada, la cantidad máxima me memoria a consumir por un script, limites de tamaño de subida de archivos y tamaño máximo de un post.

Las firmas de información de las cabeceras del servidor son útiles en la toma de estadísticas de servidores con php, sin embargo esto hace identificable las versión del sistema y aunque puede ser una practica paranoica su remoción contribuye al proceso de seguridad, por tanto se aplicara.

Php ofrece una gama de funciones y clases a los desarrolladores algunas de estas funciones pueden ser empleadas inapropiadamente por esto se debe verificar el uso de funciones que permiten la ejecución o manipulación de procesos o comandos , tiempo o apertura de archivos o directorios.

El uso de comillas en las aplicaciones es muy común y sobre todo en las que usan base de datos, en el uso de estas se debe tener especial cuidado de lo que esta permitido o no, una política incorrecta del uso de estas puede permitir la malformación de instrucciones con propósitos indebidos a la aplicación o al servidor, por tanto se debe control de uso de comillas y el contexto en el que es usado (métodos de petición, datos generados en tiempo de ejecución).

Las aplicaciones no deben notificar al cliente los errores o advertencias ocurridas en el sistema, esto es algo que le compete al grupo de desarrollo e indirectamente al administrador del sistema, por tanto se debe desactivar las directivas que muestran errores en las aplicaciones con código fuente, los errores serán estudiados en los archivos de registro del sistema.

8.4. CONFIGURANDO OTROS SERVICIOS

8.4.1. Asegurando SSH.

El servicio de Secure Shell permite establecer conexiones de terminales cifradas entre dos maquinas a través de una red local o internet. Aunque este servicio goza de un periodo de desarrollo amplio (desde 1995) existen aspectos que se pueden mejorar en la configuración.

Entre estas configuraciones esta:

- No permitir el inicio de sesión directamente del usuario root ya que es común intentos de conexión a esta cuenta desde aplicaciones en la red.
- Permitir únicamente conexiones con el protocolo ssh2
- El tiempo valido para establecer la conexión no supere los 30 segundos.

8.4.2. Configuracion del cortafuegos.

Esta aplicación en términos de seguridad representa una pieza importante en un sistema servidor, la protección de los servicios es uno de las practicas más comunes e importantes en la administración de servidores y con justa razón ya que estos son la entrada a posibles comportamientos indeseados y comprometer la integridad del sistema.

La configuración del servidor cormorán es bastante simple, solo permite conexiones a los protocolos http, https, y ssh, el servicio usa una política de rechazo por defecto.

8.5. SISTEMAS DE REPORTE DE INFORMACION DEL SISTEMA

En el proceso de administración de un servidor es necesario poder medir el estado de las variables en un determinado tiempo, aunque los sistemas operativos contienen aplicaciones especificas que permiten procesos de medición y visualizar determinados aspectos del sistema es posible contar con herramientas que permiten informes detallados del sistema en un determinado estado que se pueden generar de forma periódica.

8.5.1. Sistemas de reporte WEB.

El propósito de este tipo de herramientas es diverso según la necesidad y la naturaleza de la aplicación, en el caso de Cormorán, se implementó un sistema de reporte de las aplicaciones web residentes en el servidor, entre la información generada en los informes se encuentran servicio como servidores web, servicios de correo, ftp etc. Además permite el estudio de los archivos de registro del servidor web para generar reportes con:

*Cantidad de visitas y visitas únicas al servidor.
*Duración de visitas y ultimas visitas.
*Usuarios autenticados y ultimas visitas autenticadas.
*Estadísticas de días de se la semana y horas (paginas, peticiones, kb por hora y día de la semana).
* Dominios y países de los hosts visitantes (paginas, peticiones, KB, 269 dominios/países detectados, detección de GeoIp)
* Lista de hosts, ultimas visitas sin resolver lista de direcciones IP
* Paginas mas vistas, entradas y salidas
* Tipos de archivos
* Estadísticas de web comprimidas para los módulos mod_gzip o mod_deflate
* Sistemas Operativos usados (paginas, peticiones, KB por cada SO, 35 SO detectados),
* Cliente web usado (paginas, peticiones, KB por cliente web, cada versión (Web, Wap, clientes -media: 97 clientes web reconocidos, mas de 450 si se usa la librería browsers_phone.pm)
* Peticiones de robots (319 robots detectados)
* Ataques de gusanos (Son reconocidas 5 familias de gusanos)
* Reconoce peticiones de Motores de búsqueda, frases clave y palabras clave usadas para encontrar el sitio local (Reconoce 115 de los mas famosos motores de búsqueda como yahoo, google, altavista, etc...)
* Errores HTTP (Paginas no encontradas con la ultima referencia)
* Otros reportes personalizados basados en url
* Cantidad de veces que los sitios han sido adicionado a favoritos
* Tamaños de pantalla (para esto es necesario adicionar algunas etiquetas en el index del sitio).
* Radio de clientes web con soporte de: Java, Flash, RealG2 reader, Quicktime reader, WMA reader, PDF reader(es necesario adicionar etiquetas en en el index

del sitio)
* Reporte de cluster para radio de servidores de carga balanceada
La generación de estos reportes representa un recurso que muestra el comportamiento del servidor y de sus aplicaciones web durante un periodo de tiempo. A partir de este se puede definir políticas de administración que permitan optimizar los procesos del servidor dado un reporte de actividades, por ejemplo, si el reporte indica que hay una actividad muy fuerte durante las noches debido a que es en esta hora en la que mas gente tiene tiempo de ejercer sus actividades en el servidor y por el contrario al medio día la actividad es muy escasa, entonces se podría se programar backups o procesos de limpiado del sistema para no aumentar la carga computacional del servidor en una hora en la que esta hay mucha demanda.

8.5.2. Configurando un analizador de sistema.

Una vez el servidor adquiere cierto tamaño en aplicaciones y servicios, suele ocurrir que el sistema demanda mas tiempo en el proceso de administración porque cada vez se hace necesario el análisis de mas archivos de registro para verificar el comportamiento de las aplicaciones del servidor. Es por esto que suele ser común que los administradores de sistemas implementen scripts o aplicaciones para la recolección de información del sistema de forma eficiente. En este caso Cormorán implementó una aplicación que permite el tratamiento de algunos de los archivos de registro del sistema para organizarlos de forma ordenada en un informe, este puede ser configurado para ser enviado al correo del usuario administrador automáticamente y permitirle tomar decisiones respecto al estado especifico del sistema.

Para la implementación de estas aplicaciones es normalmente necesario la previa configuración o de paquetes terceros pero estos suelen venir en los sistemas tipo UNIX y definir la automatización de la aplicación manualmente a partir de los recursos del sistema.

8.6. CONFIGURANDO UN SISTEMA DE DETECCION DE INTRUSOS (IDS)

Los sistemas de detección de intrusos son aplicaciones que funcionan de forma diversa; algunos orientados a redes NIDS (analizando paquetes en la red),

maquinas locales HIDS (analizando paquetes locales) o sistemas distribuidos DIDS (analizando paquetes en al red y centralizándolos); pero el propósito es el mismo, identificar la integridad de archivos a partir del análisis pormenorizado de las mismas. Los IDS son herramientas que permiten el estudio de paquetes e identificar patrones de ataques según reglas definidas previamente, o la validación de la integridad de archivos a partir de métodos de comparación de sumas md5.

En el caso de Cormorán se implemento un sistema de detección de intrusos orientado a redes NIDS que actúa sobre la red interna e identifica y registra el trafico de la misma.

Una vez instalado el sistema de detección de intrusos orientado a redes, es necesario configurar el comportamiento de este, en este caso se especifica que corra como demonio, definir el modo máscara de creación de archivo, definir el usuario y el grupo según el que se ejecutara la aplicación y definir que configuración usar.

8.7. CONFIGURANDO UN SCANNER DE VULNERABILIDADES

En aras de mantener un sistema seguro, es importante conocer el estado de desarrollo de cada una de las aplicaciones que conforma el sistema. Este tipo de trabajo de documentación constante puede ser un trabajo exhaustivo si no se implementan recursos que automaticen las tareas. Por ejemplo la documentación del estado de la seguridad de un servidor web especifico implica el uso de internet y consultar (en caso de que exista) sitios web donde se maneja el estado de seguridad de dicho servidor web, en el mejor de los casos se puede estar informado a través de una lista de correos especifica que trata el tema. Sin embargo cuando el número de aplicaciones implementadas aumenta, esta tarea de documentación diaria demanda mas tiempo. Aunque estar suscrito a las listas de correo es una buena practica de administración, existen aplicaciones que se encargan de detectar ciertas vulnerabilidades (previamente identificadas) y que cuentan con varios tipo de pruebas.

En el servidor Cormorán se implementa una aplicación que cumple con este propósito, a partir de un banco de reglas que es actualizable a través de la red.

Esta medida de alguna forma representa un paso en la identificación posibles errores en la configuración ya sea por desconocimiento del administrador, omisión, etc.

Aunque esta herramienta permite detectar algunas vulnerabilidades en el sistema a partir de la realización de ataques, estos, perteneces a un grupo de reglas ya identificadas, lo que significa que una vulnerabilidad explotable no identificada sera pasada por alto, de igual forma si la aplicación no esta actualizada omitirá en el proceso de identificación dicha vulnerabilidad. Por otra parte el tiempo que tarde en crearse las reglas que permiten simular e identificar una vulnerabilidad puede ser alto en comparación con el tiempo de identificación. Es decir, primero ocurre la notificación de una vulnerabilidad y tiempo después una manera de como explotarla. Lo que representa un tiempo considerable de inseguridad en un servicio especifico (dependiendo la naturaleza de la vulnerabilidad). Por esto se debe mantener las listas de correo de las aplicaciones como una referencia de la seguridad de la respectiva aplicación. Además consultar fuentes de terceros, normalmente en la notificación de bugs o vulnerabilidades se hace por grupos externos al grupo de desarrollo involucrado a una determinada aplicación.

8.8. SISTEMA DE COPIAS DE RESPALDO DEL SERVIDOR (BACKUPS)

Las copias de respaldo representan una necesidad básica a los usuarios y un proceso requerido en la administración de servidores. Dada la naturaleza del servidor, la calidad de sus datos, el volumen de la información, los servicios prestados etc. se puede pensar en los métodos para hacer copias de respaldo que ofrezcan un grado de satisfacción aceptable a los usuarios, aunque esta es una medida muy subjetiva ya que cada usuario valora de diferente forma sus datos o estos están en función del tiempo y de la necesidad particular de cada usuario. Algunos ejemplos pueden ilustrar esta situación, en el caso en que un grupo de investigación de la escuela de ingeniería de sistemas desarrolla y mantiene aplicaciones residentes en el servidor Cormorán pero el desarrollo según el proceso de desarrollo de la aplicación implementa cambios significativos cada semana, para este grupo de desarrollo no será una gran pérdida en términos del ciclo de vida de la aplicación la perdida de 2 días de desarrollo, sin embargo, si los cambios hechos durante esos dos días representan el correcto funcionamiento de

un servicio que tiene una demanda alta en esos dos días, la perdida de estas capacidades de la aplicación, puede ser traumático. Siendo mas estrictos en la aplicación de sistemas de copias de respaldo se podría plantear a modo de ejemplo un sistema servidor de una entidad bancaria que registra cientos de procesos por hora en el que la perdida de una hora de datos puede representar mucho dinero y grandes perdidas se debe usar sistemas mas refinados de copias de seguridad que hagan copias de los datos almacenados con un periodo de tiempo mucho mas corto sobre los datos nuevos (los cambios en una hora) y no todos los datos. La explicación de estos sistemas de backups (copias de respaldo) se escapan a los propósitos de este trabajo de grado, sin embargo, en las etapas del proyecto relacionas con la implementación de sistemas de respaldo de información se evaluaron algunos modelos.

Cormorán implementa un sistema de copias de respaldo bastante simple dada la actividad de desarrollo de los grupos y el uso de los usuarios de las aplicaciones de estos servicios. El sistema de copias de respaldo de Cormorán es automatizado y se ejecuta de forma periódica, las bases de datos residentes son respaldadas en medios externos al servidor diariamente, de igual forma las aplicaciones.

El respaldo del sistema del servidor no es automatizado dado el consumo de recursos computacionales y la actividad bajo demanda del servidor es variable, estos respaldos se ejecutan a partir de scripts que facilitan el proceso pero deben ser ejecutados.

Las copias de respaldo son la piedra angular en la migración de una plataforma a otra, sin estos, no estaría completo un proceso de migración ya que el cambio de versiones formateo de particiones o movimiento de datos hace susceptible el servidor a perdidas de datos importantes, por esto, las copias de respaldo se deben revisar de forma periódica con el fin de verificar su integridad y deben ser implementados en ambientes de practica como servidores de prueba, solo así se puede tener plena seguridad de que las copias de respaldo son integras y cumplen con su cometido.

9. DEFINICION DE POLITICAS DE SEGURIDAD DEL SERVIDOR CORMORAN

9.1. ACCESO A RECURSOS FISICOS DEL SERVIDOR Y DEL ENTORNO

El desarrollo y definición de estas políticas permiten del control de acceso y uso de los recursos del sistema servidor.

El acceso al espacio físico de Cormorán es restringido al personal no autorizado.

El servidor Cormorán está ubicado en un espacio cerrado en la escuela de Ingeniería de Sistemas e Informática. El acceso a este se hace por parte del administrador del sistema, el director del proyecto y las personas autorizadas por estos a entrar. La distribución de llaves para el acceso a los recursos es exclusivo de los directamente relacionados al proyecto y gran parte de la responsabilidad de los recursos recae sobre estos.

Los recursos físicos del servidor no son compartidos con terceros a menos que los responsables de estos lo consideren necesario y se haga la respectiva gestión con planta física para cumplir la normativa de la Universidad en cuanto a inventarios.

El espacio dispuesto para almacenar parte de los recursos del servidor pertenece a la Universidad y este es aseado por empleados de la universidad previa autorización de acceso del director encargado del proyecto o el administrador del sistema, de considerarse necesario este proceso puede ser auditado para evitar problemas con desconexiones de cables y recomendaciones al ejecutante del aseo en el espacio de trabajo de Cormorán.

9.2. APLICACIONES WEB A RESIDENTES EN EL SERVIDOR

Esta normativa tiene por objeto establecer políticas en la implementación de aplicaciones web en el servidor Cormorán que aumenten el grado de seguridad del servidor estableciendo políticas en las aplicaciones.

9.2.1. Aplicaciones hechas en PHP.

- Restringir el uso de las variables de registro global.
- Restringir el uso del registro de contenido remoto (allow_url_fopen).
- Restringir la ejecución de archivos que este por fuera del directorio de cada aplicación, respectivamente.
- Restringir la carga dinámica de módulos.
- Existen limites de: Tiempo máximo de ejecución de aplicaciones, Tiempo máximo de tratamiento de entradas, Consumo de memoria máximo en la ejecución de un script, Tamaño máximo de subida de archivos y Tamaño máximo de post. Los valores deben ser consultados con el administrador de sistema.
- Restringir el uso de ciertas clases entre estas: system, exec, shell_exec, passthru, pcntl_exec, putenv, proc_close, proc_get_status, proc_nice, proc_open, proc_terminate, popen, pclose, set_time_limit, ini_alter, virtual, openlog, escapeshellcmd, escapeshellarg, dl, curl_exec, parse_ini_file, show_source. De ser necesario se puede consultar con la administración del sistema el uso de estas funciones y sus métodos de uso.
- Se debe consultar a la administración del servidor la configuración la política del uso de comillas.

9.3. ADMINISTRACION DE LAS COMUNICACIONES Y OPERACIONES

Debido a que el proceso de comunicación entre los desarrolladores y la administración es muy escasa se desarrollo un proyecto piloto orientado a la web en el que desarrolladores y administración contaran con un espacio mas formal de comunicación y establecer un carácter comunitario en cuanto a los grupos que involucra el servidor Cormorán.

Este proyecto consiste en una aplicación web administrada por Cormorán que contiene módulos un modulo de administración de usuario para garantizar el correcto orden y privilegios de los participantes en la comunidad; un modulo que permite la redacción de artículos para hacer notificaciones tanto a los grupos de desarrollo como a los administradores; Un modulo de foro, en el que cada grupo de desarrollo o aplicación web, dispone de un espacio de opinión y orientado a su propia aplicación. Un modulo de perfiles de usuario para la identificación ante la

comunidad de cada uno de estos. El proyecto incluye bitácoras de actividades y medios de notificación partir de correos.

Este proyecto pretende integrar mas a los grupos de mantenimiento y desarrollo de las aplicaciones del residentes en el servidor Cormorán, abrir canales de comunicación mas directos y que permitan una normativa mas clara en el desarrollo de procesos a nivel de comunidad.

El desarrollo del proyecto se encuentra en fase de pruebas aun no es productivo, por lo tanto como medio de contingencia se usa la lista de correos que opera en el sitio de la escuela a partir de la cual se hace las respectivas notificaciones a los desarrolladores y a la comunidad en general de los futuros cambios que se harán en el servidor, el inconveniente con este sistema es que no esta orientado a los grupos de desarrollo si no a la comunidad académica en general, como una alternativa se estima la solicitud de desarrollo de un modulo orientado a estos requerimientos en el sitio de la EISI.

Las operaciones internas de administración del servidor son definidas entre la administración del servidor y la dirección del proyecto encargado, en este espacio se definen las directivas de trabajo según los objetivos propuestos en el desarrollo del proyecto de administración de Cormorán.

Aunque no existe un conducto regular del proceso formal de gestión de procesos, se pueden definir una serie de políticas que permiten en determinado grado la gestión de algunos procesos.

Una vez establecidas las labores operativas y de investigación se trabaja según el cronograma de actividades, dado que la administración no es meramente un proceso de investigación si no un proceso de producción que implica terceros es difícil cumplir los requerimientos del cronograma ya que estos terceros no están involucrados en el cronograma de actividades, a modo de sugerencia para la continuación de este proyecto se debe crear invita a darle continuidad al proyecto orientado a la web que intenta integrar a las comunidades de soporte y desarrollo de la EISI, de lo contrario la creación de un mecanismo que reúna los procesos externos que involucren la gestión del servidor Cormorán y sean tenidos en cuenta en el cronograma de actividades para disminuir la incertidumbre de los procesos y

refinar los procesos en la administración del servidor ya que estos estarían plenamente definidos.

Debido a que en la administración del servidor es factible encontrarse con situaciones no deseadas o comportamientos que se salen de lo estimado, es necesario tomar decisiones para resolver o dicha situación. En estos casos el proceso a seguir es la notificación del estado del servidor al director encargado, definir las causas del estado y notificarlas al directorio del proyecto, una vez se tiene pleno conocimiento de la situación se discuten las posibles soluciones y se simulan las posibles soluciones en los servidores de prueba, una vez se tiene pleno conocimiento de la solución se implementa en el servidor de producción.

9.4. SEGURIDAD DE LOS RECURSOS HUMANOS

El proceso de administración del servidor Cormorán es una tarea que implica una cantidad baja de personas y estas son los encargados directos del acceso a los recursos, aunque debido a que estos recursos hacen parte de la Universidad Industrial de Santander, estos, son vigilados por el equipo de planta de seguridad de la universidad, además, el manejo de llaves para el acceso de recurso se reduce a los directamente encargados con el proyecto.
El acceso a los recursos desde terceras personas se hace con la previa autorización de la dirección del proyecto.

9.5. ADMINISTRACION DE INSIDENTES DE SEGURIDAD

La administración de Cormorán es autónoma en la definición de prestación de servicios, aunque los servicios están orientados al uso de la comunidad académica, estos pueden verse afectados si hacen mal uso de los servicios.

Algunas de las causas de inhabilitación de los servicios son:

Para desarrolladores:

- El no cumplimiento de requisitos en el desarrollo de aplicaciones residentes en el servidor.

- El no correcto mantenimiento de las aplicaciones según la notificación o políticas del servidor.
- La no actualización/migración de aplicaciones según el desarrollo de las herramientas en el tiempo. Previa notificación por parte de la administración.
- El uso inadecuado del espacio asignado para las aplicaciones y el consumo excesivo de recursos del sistema.
- Intentos de acceso o el acceso de información de terceros en el sistema.
- La violación de las políticas de terceros a partir de las aplicaciones desarrolladas o instaladas.
- Falsedad en la identidad de sitios o suplantación de terceros en el servidor.
- El uso de contenido inapropiado o con fines no académicos, así como el uso de contenido de terceros que violen licencias o derechos de autor.

Para usuarios:

- La violación de los acuerdos de uso de cada sitio contenido en el servidor Cormorán.
- La sobre carga de peticiones al servidor.
- La inyección de código o aplicaciones en las aplicaciones web o en el sistema.
- La descarga o subida desmesurada de contenido que no permita el correcto funcionamiento del servidor.
- El escaneo de puertos del sistema.
- Intentos de conexiones fallidas reiteradas veces.
- Cualquier uso de recursos, contenidos o datos de terceros en el servidor.
- Intentos de negación de servicio.

Además, de considerarlo necesario la administración del servidor notificará a las autoridades legales pertinentes el uso indebido que de los recursos que pueda afectar a terceros o a la comunidad académica.

Los incidentes de seguridad que se presenten serán valorados por la administración del servidor, se evaluaran dependiendo de cada caso, así mismo se tomaran los correctivos necesarios y de ser necesario se apelara a la instancias institucionales de la Universidad o legales Colombianas.

9.6. POLITICAS DE LEGALIDAD DEL SERVIDOR

El servidor Cormorán de la Escuela de Ingeniería de Sistemas de la Universidad Industrial de Santander es un recurso que apoya parte de la gestión académica, por tanto su uso está encaminado únicamente a los procesos internos de la Escuela de Ingeniería de Sistemas. La administración del servidor o la Escuela de Ingeniería de Sistemas de la Universidad Industrial de Santander no es/son responsables del uso indebido que pueda ejercer los usuarios del servidor Cormorán o las aplicaciones contenidas en este.

El uso indebido de los recursos del sistema detectados serán notificados a las instancias competentes de la Universidad o de ser necesario a las instancias legales Colombianas.

9.7. PROCESOS DE AUDITORIA EN EL SERVIDOR

La auditoría del servidor implica un esfuerzo constante o al menos periódico en el que se debe valorar el estado de ciertas variables para diagnosticar el estado del servidor en un determinado momento.

Los procesos de auditoría pueden ser tan complejos y exhaustivos como sea necesario dependiendo de la necesidad o del tipo de organización en el que se realizan y representan una medida de control importante. En el desarrollo de la practica empresarial se implementaron aplicaciones que permiten el estudio de los archivos de registro del servidor, sin embargo como se menciono anteriormente el proceso de auditoría puede llegar a un nivel de detalle alto que puede ir desde el montaje del servidor hasta la interacción del servidor con los usuarios.

Desde la migración del servidor se tomaron medidas de seguridad como el uso de paquetes íntegros verificados con sumas md5; el proceso de configuración de las herramientas se hizo teniendo en cuenta la documentación oficial y de terceros para aumentar la seguridad de cada servicio. El estado de configuración de cada herramienta tiende a cambiar dado el desarrollo de las aplicaciones así como la identificación de bugs o vulnerabilidades, por esto es necesario para la administración del servidor la documentación constante de cada servicio, así como

del sistema operativo en general.

Una vez verificada la integridad del sistema en la instalación es una buena practica hacer sumas md5 de la configuración del sistema antes de entrar en la etapa productiva, así, es posible consultar si uno o varios archivos del sistema han cambiado con respecto al inicial. Para aplicar este método de validación de integridad se debe tener en cuenta el cambio de las aplicaciones por migración o por configuración manual de nuevo recursos, lo que implica una labor constante de cálculos de sumas es recomendable hacer scripts que automaticen este proceso.

El análisis de historial de comandos de usuarios es una buena pista de las actividades del servidor, aunque estos pueden ser borrados, en caso tal, puede resultar interesante un seguimiento particular a dicho usuario. Los privilegios de usuario son una constante en la administración ya que aunque el servidor debe ser funcional solo de debe permitir funciones especificas a cada usuario. Las aplicaciones web residentes en el servidor son responsabilidad de cada grupo de soporte o desarrollo sin embargo es necesario verificar la actividad de cada uno de los sitios y analizar los archivos de registro de actividades así como las notificaciones hechas en los archivos de registro de cada servidor web. El uso de módulos de seguridad en los servidores web permite le proceso de notificación de posibles ataques que se comporten según el núcleo de reglas del modulo, además, puede ser usado como una herramienta reactiva que además de validar las peticiones puede negarlas según la configuración definida por el administrador de sistemas. La auditoría de archivos de registro de actividades es una actividad que puede tomar tiempo, debido a este hecho se implementaron aplicaciones que permiten el análisis de algunos archivos registro, procesados y organizados en un informe que es configurado para ser enviado a un correo, en este caso el administrador. El uso de sistemas de detección de intrusos orientados a la red permite identificar posibles ataques por exploración en la red, de igual forma esta herramientas han sido configurada para notificar la actividad diaria del servidor y la red para ser enviados al correo del administrador del sistema.

Además fueron implementadas aplicaciones que permiten el monitoreo de la actividad de red y los paquetes transmitidos en estas, existen paquetes que incluso permiten mostrar el contenido de los paquetes enviados y de ser necesario

un seguimiento entre las direcciones involucradas.

Nuevamente el proceso de auditoría puede ser una actividad tan exhaustiva como sea necesaria si se desea ratificar el proceso de seguridad del servidor, por ejemplo seria posible analizar el trazado de los recursos (librerías y aplicaciones) que son usadas cuando se ejecuta un comando con el fin de verificar la correcta ejecución de cada comando sin que ejecute algún elemento externo o ajeno a la naturaleza de este.

El uso de recursos o ambientes de prueba es importante en la validación del sistema sin comprometer la parte productiva, además, simular ataques para verificar ataques tanto conocidos como elaborados de manera particular en las aplicaciones en el servidor de prueba.

El proceso de auditoría de un servidor es dinámico, esto representa un análisis constante de los cambios y documentación con el fin de identificar posibles errores en el sistema, fallas por omisión o cualquier otro evento que implique el mal funcionamiento del servidor.

9.8. Cumplimiento (LEGALES, DE ESTANDARES TECNICAS Y AUDITORIAS)

En el desarrollo de la práctica se cumplieron las políticas de la universidad en cuanto al manejo de inventarios, se cumplieron los requerimientos en su totalidad en el uso adecuado de licencias ya que la mayoría del software implementado es software libre bajo licencia GPL, los otros tipos de licencia son de código abierto.

Este trabajo de grado empleó algunos mecanismos que permitían la obtención de resultados a partir de la ejecución de algunos procesos a modo de metodología, además se analizó algunos elementos del proyecto OWASP[2] que facilitaron el trabajo, sin embargo, no se hizo un análisis estricto de la implementación de un sistema de gestión de seguridad de la información basado en algún estándar. Un estándar como la serie ISO 27001 podría ser implementado ya que este es aplicable a pequeñas medianas o grandes organizaciones.

2 Owasp , comunidad mundial enfocada en el mejoramiento de la seguridad de aplicaciones software.

El uso de normativas como la ISO/IEC 27001 permite la plena definición de procedimientos y políticas que permiten la Gestión de un sistema de seguridad de la información, sin embargo la implementación de un estándar de este tipo se sale de los alcances de este trabajo. Dada la continuidad de este trabajo se recomienda el análisis y documentación de la posible implementación de un estándar que permita la gestión de seguridad de la información.

Las auditorías realizadas por terceros en el servidor fueron enfocadas al uso de aplicaciones correctamente licenciadas, hasta el momento no se han hecho auditorías orientadas a la seguridad por parte de terceros como entes institucionales de la Universidad, sin embargo, existen nuevos procesos implementados en función de la seguridad que pasan a ser operativos.

CONCLUSIONES

- El proceso de administración de servidores es una tarea enriquecedora que está sometida a cambios constantes y en los que es necesario tener procedimientos y políticas definidas. Es importante establecer metodologías de trabajo y establecer perfiles de todos los actores que están involucrados directa o indirectamente así como sus alcances, el empleo de estas practicas, permite la creación y desarrollo explicito de funciones y el desarrollo de actividades de forma ordenada.

- Manejar información desde la administración de un servidor implica una responsabilidad grande que se debe afrontar con los elementos necesarios, no es un proceso meramente técnico, por el contrario involucra una actitud holístaca en la que se deben asumir los diferentes tipos de roles que están involucrados en los procesos, para ser funcional, sin dejar de ser un recurso seguro.

- La seguridad de sistemas de información es un proceso dinámico e inacabado debido al uso de nuevos recursos o tecnologías, por tanto implica una tarea constante de documentación y de adaptación al rápido desarrollo de la informática incluso a la forma de pensar en cuanto a soluciones ya que existen formas diferentes de concebir los procesos tecnológicos y las diversas soluciones informáticas son prueba de ello.

- El Software Libre es un recurso bien importante que permite el uso sin restricciones de la tecnología, en el caso del servidor Cormorán es la base lógica del funcionamiento y que apoya parte de la gestión de la comunidad académica de la EISI. La Universidad Industrial de Santander debería estimar el Software Libre como un alternativa viable para el apoyo de su gestión.

- La interacción de grupos académicos, grupos de soporte y de las personas a través del servidor estimula el desarrollo grupal, este hecho puede ser empleado en la creación de comunidades de investigación y desarrollo colaborativas que permitan compartir el conocimiento abierto para incrementarlo en pro de la EISI y de las comunidades en general.

RECOMENDACIONES

- Establecer el uso formal de las políticas de seguridad y en general de toda la normativa del servidor Cormorán en sus procesos incluyendo lo grupos de soporte y desarrollo.

- Continuar el proceso de documentación de las tecnologías implementadas para conocer sus características, identificar posibles vulnerabilidades y definir las nuevas políticas que sean precisas para el correcto funcionamiento del servidor Cormorán, además, continuar el proceso de auditoría a partir de las herramientas establecidas en el servidor, de ser necesario, implementar nuevas alternativas que permiten mantener la funcionalidad del servidor e incrementar la seguridad del mismo.

- Implementar un estándar de carácter internacional en el proceso de gestión de seguridad de la información, a modo de sugerencia se propone el estándar ISO/IEC 27001.

- Hacer el respectivo proceso de actualización de las herramientas contenidas en el servidor Cormorán así como las reglas contenidas en las aplicaciones que permiten identificar errores, vulnerabilidades o bugs y mantener el carácter libre del software implementado en el servidor.

- Darle continuidad al proyecto de integración de la comunidad de desarrollo de aplicaciones residentes en el servidor Cormorán y la administración del mismo a través del proyecto piloto orientado a la web desarrollado con este propósito para contar con canales solidos de comunicación y tener claridad en las políticas y procesos establecidos.

BIBLIOGRAFÍA

LIBROS Y ARTICULOS

Aoki Asamu; Echarri Walter O.; Guia de Referencia de Debian. (Enero 2007).

Falkner Jayson; Galbraith Ben; Irany Romin; Kochmer Casey; Narayana Panduranga Sathya; Perrumal Krishnaraj; Timney John; Moidoo Kunnupurath Meeraj; Beginning JSP web Development Wrox Press (2001). Introducción, Capitulo 1.

Negus Christopher; Linux Bible 2005 Edition; Wiley Publishing, Inc. Este libro sirvió como referencia general de procesos en diferentes distribuciones de GNU/Linux.

Stanfield Vicki; Smith Roderick W.; Linux System Administration Second Edition. Craig Hunt-Linux Library Sybex. Este libro representó una referencia constante del proceso de administración.

Whittal Hamish; Shell Scripting (2005); Licencia CopyLeft. Este libro permitió el estudio y referencia de la codificación en shell.

ENLACES

http://es.wikipedia.org/wiki/Red_Hat
http://es.wikipedia.org/wiki/RHEL
(Análisis del sistema Operativo, Historia y desarrollo de RHEL)

http://apache.org
http://es.wikipedia.org/wiki/Servidor_HTTP_Apache
http://hoohoo.ncsa.uiuc.edu/
http://es.wikipedia.org/wiki/NCSA_HTTPd
(Historia del proyecto Apache, Servidor web Apache.)

http://es.wikipedia.org/wiki/Tomcat
(Servidor web Jakarta Tomcat, Historia del proyecto Jakarta Tomcat)

Mysql
http://dev.mysql.com/doc/
http://es.wikipedia.org/wiki/SQL
http://es.wikipedia.org/wiki/MySQL
(Sistemas Gestores de Bases de Datos, Historia y desarrollo)

PostgreSQL
http://www.postgresql.org
http://en.wikipedia.org/wiki/Postgresql
(Sistemas Gestores de Bases de Datos, Historia y desarrollo de PostgrSQL)

PHP
http://php.net
http://es.wikipedia.org/wiki/PHP
(Plataformas de Programación, Historia y desarrollo)

http://en.wikipedia.org/wiki/Java_Virtual_Machine
(Java y JVM, Definición de la maquina JVM)

http://en.wikipedia.org/wiki/Mail_transfer_agent
(Agente de Transferencia de Mensajes, Definición de MTA)

http://www.geekcomix.com/cgi-bin/classnotes/wiki.pl?UNIX03/Ipchains_And_Iptables_-_Small_History_Of_Filtering_Under_Linux
(IPTABLES, firewall a partir de reglas.)
GNU/Debian Linux.

www.debian.org
(Manual de Referencia de GNU/Linux Debian)