PROTOTIPO DE SERVIDOR DE ALTA DISPONIBILIDAD PARA LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

GISELLE GRAZT MEDINA
ALEJANDRO SANTA ARCINIEGAS

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTA DE FISICOMECANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2010

PROTOTIPO DE SERVIDOR DE ALTA DISPONIBILIDAD PARA LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

GISELLE GRAZT MEDINA
ALEJANDRO SANTA ARCINIEGAS

Proyecto de Grado para optar al título de Ingeniero de Sistemas

Director
M.Sc. MANUEL GUILLERMO FLOREZ BECERRA

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTA DE FISICOMECANICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA
2010

DEDICATORIAS

Tú eres lo que es el profundo deseo que te impulsa,
Tal como es tu deseo es tu voluntad,
Tal como es tu voluntad son tus actos,
Tal como son tus actos es tu destino.
- Brihadaranyaka Upanishad IV.4.5

El siguiente párrafo lo cito porque expresa para mí una guía de vida y estoy muy agradecida porque es Dios quién inspira ese profundo deseo que me impulsa, mi Fé, por ello quiero compartirlo al igual que el logro y éxito de este Proyecto de Grado especialmente con mis padres y con toda mi familia, que me transmiten día a día todo su cariño y constante compañía.

A M.Sc. Jorge E. Pinto Valderrama, por creer en mí y en el éxito de este proyecto.

A la Universidad Industrial de Santander por su carácter público y pluralista.

Giselle Grazt Medina.

Seamos realistas y hagamos lo imposible.

-Ernesto Guevara

A Dios por darme una segunda oportunidad, a mi papá y a mi mamá porque siempre ha estado presente su amor incondicional, a mi esposa por iluminarme el camino, a M.Sc. Javier Eduardo Arias por creer en mí, a mi compañera Giselle Grazt por llenarme de fé para continuar con el proyecto de grado y a nuestra universidad pública, porque gracias a ella pude estudiar.

Alejandro Santa Arciniegas

AGRADECIMIENTOS

A M.Sc. Manuel Guillermo Flórez Becerra, Director del proyecto de grado, por darnos la oportunidad de trabajar con los servidores de la Escuela de Ingeniería de Sistemas, permitiéndonos tener una gran experiencia técnica, académica e investigativa.
Por su buena disposición dándonos siempre lo mejor de sí.

A M.Sc. Jorge E. Pinto Valderrama, Director de la Escuela de Geología - UIS, por apoyar la investigación del proyecto facilitándonos sus instalaciones y creyendo en la implementación de nuevas tecnologías.

A M.Sc. Javier Eduardo Arias, Docente de la Escuela de Ingeniería Industrial – UIS, por apoyar el desarrollo del núcleo de la aplicación web y creer en el proyecto Vochica.

CONTENIDO

PÁG.

LISTA DE TABLAS

PÁG.

LISTA DE FIGURAS

PÁG.

LISTA DE ANEXOS

RESUMEN

TÍTULO: PROTOTIPO DE SERVIDOR DE ALTA DISPONIBILIDAD PARA LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA*

AUTORES: GRAZT MEDINA, Giselle, SANTA ARCINIEGAS, Alejandro.**

PALABRAS CLAVES: ALTA DISPONIBILIDAD, ADMINISTRACIÓN SERVIDORES, SEGURIDAD, PANEL DE CONTROL WEB.

DESCRIPCIÓN: El Prototipo de servidor de alta disponibilidad se implementó con la ultima distribución disponible del sistema operativo GNU / LINUX Debian 5 y la herramienta Heartbeat para ofrecer alta disponibilidad. Este prototipo permite conectar servidores (en este caso servidores web) para que interactúen entre sí como una sola máquina frente el usuario final, de manera que si alguno de las máquinas o nodos que conforman el clúster deja de funcionar, el servidor de respaldo asume la carga de trabajo del servidor principal, reduciendo así considerablemente las caídas de servicio y su respectivo tiempo de restauración.

Como parte del proyecto se realizó la migración del servidor de la Escuela de Ingeniería de Sistemas e Informática a la última versión estable del Sistema Operativo Debian, la configuración de un servidor de respaldo que asume la carga de trabajo del servidor principal por medio de scripts y manualmente, y el diseño, desarrollo e implementación de una aplicación Web utilizando software libre y el núcleo Vochica, para la interacción entre administrador y usuarios de los servidores.

Durante el desarrollo del proyecto, se realizó la administración de los servidores, al igual que el soporte a usuarios y desarrolladores de proyectos de grado que necesitaban alojamiento.

---

* Trabajo de Grado

** Facultad de Físico-Mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director. MANUEL GUILLERMO FLORÉZ BECERRA.

SUMMARY

TITLE: PROTOTYPE HIGH AVAILABILITY SERVER FOR SCHOOL OF ENGINEERING AND COMPUTER SYSTEMS. *

AUTHORS: GRAZT MEDINA, Giselle, SANTA ARCINIEGAS, Alejandro. **

KEY WORDS: HIGH AVAILABILITY, SERVER MANAGEMENT, SECURITY, WEB CONTROL PANEL.

DESCRIPTION: The prototype high-availability server was implemented with the latest distribution available operating system GNU / Linux Debian 5 heartbeat tool, to provide high availability. This prototype allows you to connect servers (web servers in this case) to interact with each other as a single machine against the end user, so if any of the machines or nodes that make up the cluster fails, the backup server assumes workload on the primary server, thereby significantly reducing falls and their respective service restoration time.

As part of the migration took place on the server of the School of Systems Engineering to the latest stable Debian operating system, configuring a backup server that takes over the workload of the main server through scripts and manually and the design, development and implementation of a Web application using free software and the core Vochica for interaction between administrator and users of the servers.

During the development of the project, conducted the administration of servers, as well as user support and project developers who needed accommodation grade.

---

* Work of degree

**Faculty of Physics-Mechanical engineerings. School of Engineer and Information Systems. Director. M.Sc. MANUEL GUILLERMO FLORÉZ BECERRA.

INTRODUCCIÓN

El presente proyecto manejo tres ejes de interés en investigación y desarrollo:

- La administración de los servidores de la Escuela de Sistemas e Informática (EISI) asignados por parte de la Universidad Industrial de Santander al proyecto y el soporte a sus respetivos usuarios.
- El desarrollo de una aplicación web para la comunicación de los usuarios de los servidores y los administradores.
- El planteamiento de un prototipo Servidor de Alta disponibilidad para la EISI.

Se integraron estos tres ejes para darle solución eficiente a la demanda de los usuarios y las necesidades de la EISI. Es nuestra meta con la ejecución de este proyecto coincidir con la misión de la EISI en la construcción, innovación y mejoramiento del conocimiento para promover la aplicación de nuevas tecnologías informáticas en el ámbito académico e industrial de Santander.

El impacto del proyecto en la comunidad, permitió que se le concediera a esta iniciativa, una donación por parte de la Fundación Raúl Ocazionez.

1. ASPECTOS GENERALES

1.1 OBJETIVOS

1.1.1 OBJETIVO GENERAL

- Administrar y proveer soporte a usuarios de los servidores de la Escuela de Ingeniería de Sistemas e Informática.
- Configurar un prototipo de servidor de alta disponibilidad con herramientas libres.

1.1.2 OBJETIVOS ESPECÍFICOS

- Automatizar las funciones más utilizadas en la administración del servidor.
- Desarrollar e implementar una aplicación Web para la interacción entre administrador y usuarios del servidor.
- Configurar e implementar un servidor de respaldo para el actual servidor de producción.
- Configurar un prototipo de servidor de alta disponibilidad que minimice el tiempo actual de restauración en caso de falla.
- Implementar nuevos mecanismos de seguridad en los servidores.

1.2 ENTIDADES INTERESADAS EN EL PROYECTO

Es de especial interés la implementación del proyecto en la EISI, debido a que le permite ofrecer una plataforma más robusta y escalable para los proyectos de desarrollo web que actualmente se alojan en el servidor, tener una infraestructura en software y hardware que le

permite ampliar sus servicios de hosting, aulas virtuales y laboratorios de clase.

Todas las empresas o instituciones educativas que posean o requieran servidores robustos, escalables, seguros para aumentar la calidad en los servicios que utilizan u ofrecen, estarán interesadas en conocer e invertir, en implementaciones bajo el concepto de alta disponibilidad.

La aplicación Web al ser totalmente configurable, puede ser utilizada en cualquier entorno donde se necesite tener control a las solicitudes que los usuarios realicen. Bajo la configuración actual es muy útil a cualquier organización que preste el servicio de Hosting.

Es importante mencionar que a este proyecto de grado se le concedió una donación por parte de la Fundación Raúl Ocazionez, quien se interesó en apoyar está gran iniciativa, debido al impacto del proyecto en la comunidad.

1.3 JUSTIFICACIÓN

Al automatizar las principales funciones de administración del servidor, se reducirá el tiempo destinado a la ejecución cotidiana de las mismas, lo cual permitirá al administrador invertir más tiempo en mejorar la configuración del servidor, investigar nuevas tecnologías y dar soporte a usuarios.

Para establecer una comunicación más efectiva entre los usuarios y el administrador de los servidores, se requiere un sistema más dinámico, especialmente para los desarrolladores de las aplicaciones y/o herramientas web, que constantemente están solicitando y realizando modificaciones. Teniendo en cuenta lo expuesto anteriormente, se desarrollará una aplicación web que brinde esta interacción y comunicación, permitiendo realizar, dar respuesta y llevar un control de

las solicitudes, informar a los usuarios y dar soporte técnico a los desarrolladores.

Anticipándose a la creciente demanda de recursos informáticos en la Escuela de Ingeniería de Sistemas, se configurará un prototipo de servidor que minimice el tiempo actual de restauración en caso de falla en la tarjeta de red o en los principales servicios.

1.4 IMPACTO

1.4.1 Impacto Técnico

Disponer de una tecnología innovadora, configurada con software libre y mínimos recursos de hardware, que ofrece el servicio de alta disponibilidad.

Contar con una aplicación web que permite ser configurada a las necesidades de comunicación entre los usuarios y los administradores.

1.4.2 Impacto Económico

Esta tecnología puede ser implementada en pequeñas, medianas y grandes empresas reduciendo significativamente los costos en software (debido a su licenciamiento libre) y hardware, lo cual implica, poder contar con un servidor de alta disponibilidad seguro, robusto y de gran desempeño sin tener que invertir gran cantidad de dinero, en hardware de respaldo especializado.

1.4.3 Impacto Social

Al contar con esta infraestructura estos servicios pueden ser ofrecidos a la comunidad académica santandereana, dándole especial énfasis a las aulas virtuales, beneficiando el proceso enseñanza-aprendizaje, gracias

a herramientas virtuales flexibles que permiten adaptarse a las necesidades de los estudiantes y profesores. En el área industrial permite implementar aplicaciones bajo la modalidad de alta disponibilidad, contando con una infraestructura estable, segura y robusta, a muy bajo precio permitiendo a las pymes adquirir esta clase de tecnología, lo cual promueve el desarrollo tecnológico regional y por ende la competitividad nacional. Esto permitiría a la escuela de Ingeniería de Sistemas e Informática ofrecer servicios tecnológicos y su respectivo soporte.

1.5 VIABILIDAD

Actualmente la EISI tiene un servidor que centraliza las aplicaciones, herramientas y páginas web, desarrolladas por los estudiantes como proyecto de grado y otras utilizadas por los profesores, como soporte a la enseñanza de sus materias. Esta dinámica origina una demanda de mayores recursos informáticos por parte de la comunidad, por ello, es importante realizar las actividades necesarias para que los servidores cumplan las expectativas de los usuarios.

Para ello se tienen disponibles los componentes de hardware y software necesarios para configurar un prototipo de servidor económico de alta disponibilidad, ajustándose al presupuesto de la escuela.

Se cuenta con personal capacitado para brindar soporte a los usuarios y la tecnología informática para llevar a cabo los objetivos propuestos.

En el aspecto del software se conto con el apoyo de la comunidad de desarrolladores y usuarios de software libre, quienes a través de internet difunden las herramientas, el conocimiento y experiencia de implementaciones afines a la expuesta en el proyecto, permitiendo contar con un amplio material de consulta, investigación e intercambio

de ideas.

2. MARCO TEÓRICO

2.1 SISTEMA OPERATIVO LINUX DEBIAN 5

Debian GNU/Linux es un sistema operativo libre, desarrollado por más de mil voluntarios alrededor del mundo, que colaboran a través de Internet. La dedicación de Debian al software libre, su base de voluntarios, su naturaleza no comercial y su modelo de desarrollo abierto la distingue de otras distribuciones del sistema operativo GNU. Todos estos aspectos y más se recogen en el llamado Contrato Social de Debian.

Nació en el año 1993, de la mano del proyecto Debian, con la idea de crear un sistema GNU usando Linux como núcleo ya que el proyecto Debian, organización responsable de su mantenimiento en la actualidad, también desarrolla sistemas GNU basados en otros núcleos (Debian GNU/Hurd, Debian GNU/NetBSD y Debian GNU/kFreeBSD).

Uno de sus principales objetivos es separar en sus versiones el software libre del software no libre. El modelo de desarrollo es independiente a empresas, creado por los propios usuarios, sin depender de ninguna manera de necesidades comerciales. Debian no vende directamente su software, lo pone a disposición de cualquiera en Internet, aunque sí permite a personas o empresas distribuir comercialmente este software mientras se respete su licencia.

Debian GNU/Linux puede instalarse utilizando distintos mecanismos de instalación, como DVD, CD, Blu-Ray, memorias USB y diskettes, e incluso directamente desde la red.

2.2 SERVIDORES WEB

2.2.1 Apache

Este servidor se encarga de resolver las peticiones de páginas de Internet de los clientes utilizando el protocolo de transferencia de Hipertexto HTTP o el protocolo seguro de transferencia de hipertexto, HTTPS. Desde su origen a evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad.

La historia de Apache se remonta a febrero de 1995, donde comienza el proyecto del grupo Apache, en inglés *A PAtCHy sErver* (un servidor "parcheado"), el cual se baso en un conjunto de parches del popular NCSA HTTPd 1.3, que más tarde fue reescrito por completo. Fueron Brian Behlendorf y Cliff Skolnick quienes a través de una lista de correo, lograron coordinar el trabajo estableciendo en espacio compartido de libre acceso para que los Web Master que venían creando parches para sus servidores web, pudieran compartir su trabajo. Fue aquí donde se conformo el grupo Apache, que en 1999 se convertiría en Apache Software Foundation.

Su nombre se debe a que Brian Behlendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que actualmente es el gobierno de los EEUU. La preocupación inicial del grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros

ingenieros de internet.

Apache tiene amplia aceptación en la red: desde 1996, es el servidor HTTP más usado, alcanzando su máxima cuota de mercado en 2005 convirtiéndose en el servidor empleado en más del 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

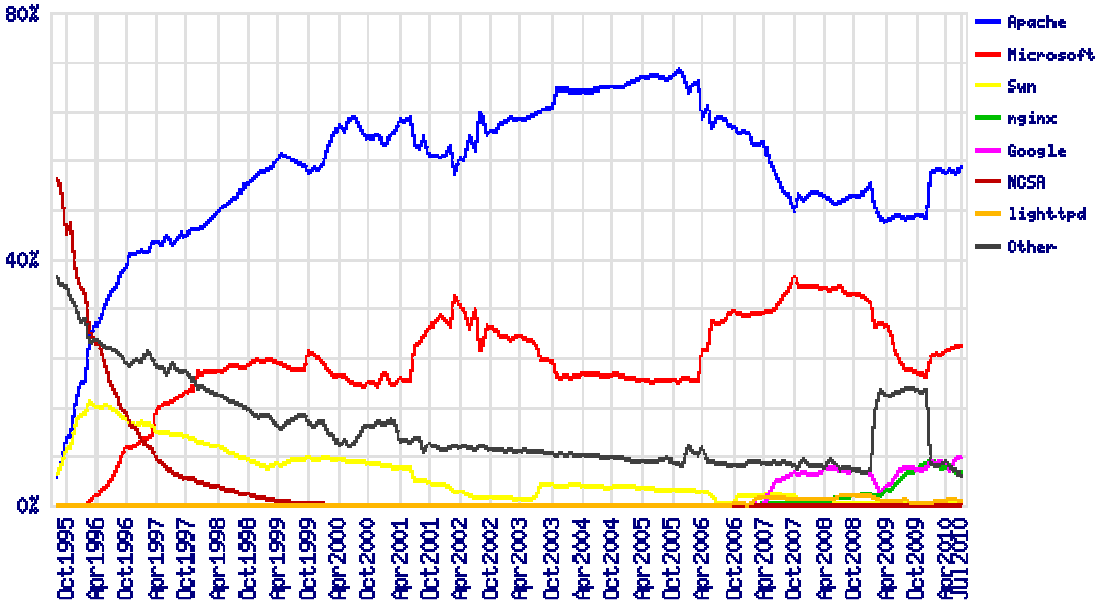


Fig. 1 Estadística del uso de servidores. Fuente: http://news.netcraft.com

Sus principales características:

- Popularidad: Facilitando su aprendizaje por la gran cantidad de material, ayuda y soporte.
- Modularidad: Altamente configurable, muy sencillo para ampliar sus capacidades.

- Código Abierto: Facilita la rápida corrección de errores y la velocidad de desarrollo.
- Multi-Plataforma: Al correr sobre multitud de sistemas operativos, lo convierte prácticamente en un servidor universal.
- Trabaja con gran cantidad de lenguajes como PHP, Perl, Python, Rexx, Ruby, etc.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor.

2.2.2 TomCat

También llamado Jakarta Tomcat o Apache Tomcat, funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.

Tomcat es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones, como JBoss o JOnAS. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets. El motor de servlets de Tomcat a menudo se presenta en combinación con el servidor web Apache.

Tomcat puede funcionar como servidor web por sí mismo. En sus inicios existió la percepción de que el uso de Tomcat de forma autónoma era sólo recomendable para entornos de desarrollo y entornos con requisitos mínimos de velocidad y gestión de transacciones. Hoy en día ya no existe esa percepción y Tomcat es usado como servidor web autónomo en entornos con alto nivel de

tráfico y alta disponibilidad. Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java.

2.3 MOTORES BASES DE DATOS

2.3.1 MySQL

Es un sistema de gestión de bases de datos relacional. Licenciado bajo la GPL de GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, desde Enero de 2008 es una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde Abril de 2009.

MySQL surgió como un intento de conectar el gestor mSQL a las tablas propias de MYSQL AB, usando sus propias rutinas de bajo nivel. Tras unas primeras pruebas, vieron que mSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

Mo se sabe con certeza de donde proviene su nombre. Por un lado dicen que sus librerías han llevado el prefijo ‘my’ durante los últimos 10 años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de

uso. Esta gran aceptación es debida, en parte, a que exigen infinidad de librerías y otras herramientas qye permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Entre sus principales características, se encuentran:

- Amplio subconjunto del lenguaje SQL, con algunas extensiones incluidas.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- Transacciones y claves foráneas, conectividad segura, replicación.

2.3.2 PostgreSQL

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 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 Ingre que fue denominado Post-Ingres y más 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, en1987 el sistema de pruebas se hace operacional y fue mostrado en la Conferencia ACM-SIGMOED de 1988. La primera versión sale en junio de 1989 y por una serie de críticas 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 trabajo de soporte más que de investigación por lo que el proyecto termina. Un año más tarde en 1994, Andrew Yun y Yolly Chen retoman el proyecto y añaden un intérprete de lenguaje SQL a Postgres, debido a que este contaba con su propio lenguaje de consultas y a partir de esta versión fue publicado en la red y se hizo de dominio público y código abierto.

Entre sus principales ventajas se encuentran:

- La velocidad del motor de datos.
- Control de concurrencia multi-versión, el cual permite a los accesos de sólo lectura continuar leyendo datos consistentes durante las actualizaciones de registros.
- Copias de seguridad en caliente (mientras la base de datos permanece disponible para consultas).
- Amplia variedad de tipos nativos.
- Claves foráneas, disparadores(triggers), integridad transaccional, herencia de tablas.

2.4 LENGUAJES DE PROGRAMACIÓN

2.4.1 PHP

Acrónimo de “PHP: Hypertext Preprocessor” es un lenguaje de código

abierto interpretado, un lenguaje de script incrustado dentro de HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características propias. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas.

PHP es un lenguaje que ha sido creado gracias a un gran número de contribuciones de la comunidad. Su desarrollo tiene origen en 1994, cuando Rasmus Lerdorf escribió en C un CGI que permitía la interpretación de un número limitado de comandos, al cual denominó Personal Home Page Tool; gracias a su gran aceptación, su creador diseño un sistema para procesar formularios al que le atribuyo el nombre de FI (Form Interpreter), formando asi lo que sería la primera versión del lenguaje: PHP/FI.

La siguiente gran contribución al lenguaje se realizó a mediados del 97 cuando se reprogramo el analizador sintáctico, se incluyo soporte a nuevos protocolos de internet, al igual que a la mayoría de motores de bases de datos comerciales.
Actualmente PHP utiliza el motor Zend, el cual ofrece mayor rapidez, mayor independencia del servidor Web y un API mucho más potente.

PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, es utilizado para desarrollar aplicaciones como Facebook, Joomla, OsCommerce, WordPress, MediaWiki (creado para Wikipedia), Moodle, PhpMyAdmin y es utilizado por organizaciones como Mitsubishi, RedHat, Der Spiegel, Ericsson y la Nasa.

Debido a su amplia utilización PHP es soportado por una gran comunidad de desarrolladores, permitiendo que los fallos de

funcionamiento se encuentren y reparen rápidamente.

Algunas de las ventajas de PHP son:

- Es un lenguaje multiplataforma.
- Soporte de gran cantidad de motores de bases de datos, como InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras.
- Integración de gran cantidad de bibliotecas externas, que permiten generar documentos en pdf, crear gráficos, analizar código XML, etc.
- Permite la técnica de programación Orientada a Objetos.
- Es código abierto, lo que lo convierte en una alternativa de fácil acceso.

2.4.2 Java

La Java Virtual Machine (Máquina Virtual de Java)- JVM, es un conjunto de aplicaciones software y estructura de datos que implementan un modelo de máquina virtual.

La JVM 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 electrodomésticos en el cual, no tuvo la acogida esperada. Con el auge y el crecimiento de internet Sun Microsystem 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 integrarla, ya que estaba diseñada en un código independiente de la plataforma, lo cual permitió acoplarla 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 y 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 Microsystems liberó el código bajo la licencia GPL (General Public Licence) se implulsa el desarrollo de nuevas comunidades de desarrollo y motiva la permanencia de las tecnologías Java en el mercado.

2.5 Administración de Servidores en GNU/LINUX

La persona responsable de establecer y mantener el sistema se le denomina administrador de sistemas o sysadmin.

Los administradores del sistema pueden ser miembros de un departamento de tecnología de la información. La mayor parte se aplica también a los administradores de redes y del sistema de Windows.

Sysadmins suelen ser los encargados de la instalación, soporte y el mantenimiento de los servidores u otros sistemas informáticos, así como la planificación y respuesta a interrupciones del servicio y otros problemas. Otras tareas pueden incluir la programación de secuencias de comandos o programación ligera, para los sistemas de gestión relacionados con los proyectos.

El administrador del sistema es responsable de lo siguiente entre otros:

- Administración de usuarios (*instalación y mantenimiento de cuentas*)
- El mantenimiento de sistema.
- Comprobar que los periféricos funcionan correctamente.
- En caso de fallo de hardware, el designa los horarios de reparación.
- Monitor de rendimiento del sistema.
- Crear sistemas de ficheros.
- Instalar el software.
- Crear la política de copias de seguridad y recuperación.
- Monitor de la comunicación de red.
- Actualizar los sistemas según sean accesibles nuevas versiones de Sistemas Operativos y software aplicativo.
- Aplicar las políticas para el uso del sistema informático y de red.
- Configuración de las políticas de seguridad para los usuarios. Un administrador de sistemas debe contar con una sólida comprensión de la seguridad informática.
- Administrar las bases de datos.

Para el ejercicio de la administración de los servidores se requiere del entendimiento y manejo de diferentes herramientas y software a continuación se resumen algunas de ellas:

2.5.1 Interprete de comandos utilizado SHELL BASH

Para comenzar se debe definir que es una Shell, intérprete de comandos o consola. Como su nombre lo dice es un programa que interpreta lo que el usuario por medio del lenguaje de cada Shell puede manipular o interactuar con el sistema operativo.

El Shell original disponible con los sistemas UNIX era el Shell Bourne (“sh”). Después dos shells que se volvieron populares fueron la Shell Korn (“ksh”) y el Shell C (“csh”), cada una con sus características propias, ventajas y desventajas.

Los Shell Bourne y el C fueron reescritos, como resultado ahora se tiene el “Bourne again Shell (Sehll Bourne nuevamente)” o “bash”, y el Shell T (“tcsh”). Los tres shells están disponibles en casi todas las distribuciones de GNU/Linux. Bash es probablemente el Shell más utilizado actualmente, y es el que viene por defecto en la mayoría de los distribuciones de GNU/LINUX.

La “línea de comandos” es la entrada del usuario para el Shell. El Shell examina la línea de comandos, verifica que lo que se ha escrito es correcto, determina si se ha digitado el nombre de un programa (un programa binario o compilado), y de ser así envía dicho programa al núcleo (kernel) para su ejecución.

Todos los comandos Shell utilizan el siguiente formato:

Comando opcion1 opcion2 opcion3 … opción argumento1 argumento2 … argumentoN

En GNU/LINUX los comandos son sensibles al uso de mayúsculas y minúsculas. El comando se termina pulsando la tecla Enter; aunque se

puede continuar el comando en una nueva línea usando backslash (\). Además se pueden concatenar varios comandos, separándolos con punto y coma (;).

Cuando se ejecuta un programa en Linux se abre automáticamente tres archivos (flujos) de E/S para ellos. Estos son: la entrada estándar, la salida estándar y el error estándar. Por defecto la salida estándar está conectada a la pantalla, la entrada estándar al teclado y el error estándar a la pantalla. Es posible reasignar estos destinos antes de ejecutar el programa, lo que se conoce como redirección de E/S.

2.5.2 Scripts

Una de las razones de ser de los Shell, es que son usados para la creación de scripts. Un script es un guión, es decir una serie de instrucciones para la consola guardado en un fichero de texto de extensión:.sh.

2.5.2.1 Comandos

touch *fichero* # crea fichero vacío.
ls -la # lista todos los ficheros detalladamente.
-lat # lista todos los ficheros por fechas.
-latr # invierte ordenación**.**
-l # Detalles.
mkdir # crea directorio.
rmdir # borra directorio.
cp # copia directorio.
mv # renombra y mueve simultaneamente**.**
rm # **borra** directorios vacios.
rm -r # borra directorios llenos.
cat fichero **#** /head presenta fichero. (con head las 15 primeras**)**

more fichero # paginar fichero.

tail -f fichero # ver finál de fichero y los cambios que pueda haber en tiempo real.

echo # devuelve lo que se escribe.

ls / wc # se usa -c (caracteres) -w (palabras) -l (lineas).
lista / y muestra palabras y caracteres.

cat # cuenta. Con **-head** cuenta la cabecera

Con **-grep** + cadena de caracteres, filtra.

grep + cadena de caracteres # filtra la cadena de caracteres indicada.

grep -r "ServerName" /etc # busca recursivamente **ServerName** en: /etc

grep -i cadena de caracteres # busca cadena de caracteres y no tengas en
cuenta mayúsculas y minúsculas (-i).

who # quién es el usuario (conectado).

w # quien es el usuario.

last # los últimos usuarios (estén o no conectados).

lastlog # listado de usuarios y fechas de las última
entrada de cada uno.

date # fecha.

top # procesos y estado de la máquina.

ps # procesos activos.

ps aux # todos los procesos

ps aux | grep "apache" | grep -v "grep" |wc
muestra todos los procesos | filtra apache | selecciona todos menos apache | muestra palabra y
número.

kill+Nº proceso # mata proceso.

kill -9 Nº proceso # mata procesos resistentes.

file # tipo de fichero

tar *donde que* **#** empaqueta fichero.

tar -cf *fichero*.tar /etc **#** desempaqueta fichero.tar en: /etc

gzip -9 f*ichero fichero*.gz **#** comprimir. -9 máxima compresión.

gunzip # descomprimir.

gzip -d # descomprimir.

tar cvzf *destino*.tgz *origen* **#** crear un targz (tarball).

tar xvzf *origen*.tgz *destino* **#** descomprimir.

2.5.2.2 Creación de Scripts

Generalmente, las secuencias de comandos empiezan con ¡ /bin/sh. Los primeros caracteres indican al sistema que dicho archivo es una secuencia de comandos, y /bin/sh inicia el Shell bash. Esto es de esa manera para permitir indicar que el programa se ejecute con otro Shell (tcsh, ksh, etc) o programa interprete. Por ejemplo, si se crea un programa en perl es necesario iniciar con la línea: #!/usr/bin/perl. Una vez iniciado el Shell indicado, éste ejecuta una a una las líneas del archivo.

Estos scripts deben tener encendido su bit de permiso de “ejecución” a la hora

de ejecutarlos. En caso de no tenerlo activado se puede activar con el comando:

chmod a+x nobrearchivo.

En GNU/LINUX esta técnica de scripts ha alcanzado el nivel de lenguaje de programación. Para los administradores de Servidores, los scripts son herramientas muy útiles, pues le permiten hacer tareas en ellos de forma muy productiva, rápida y que se pueden ejecutar con la programación que se desee, mediante el demonio crotab. Incluso el

sistema operativo como tal se inicia a causa de un script llamado init.d.
Ejemplo de script:

#! / bin / bash

hora =`date +%H:%M` # hora en formato: 20:29

conectados=`who | wc -l`

echo "A las" $hora "hay" $conectados

2.5.2.3 Asignación de permisos

Un aspecto importante es asignar los permisos adecuados al scripts para que los puedan ejecutar solo los usuarios que estén autorizados, se puede definir por:

- |rw- |r-- |r--|

tipo dueño grupo todos

Dos formas de cambiar los permisos: numerica y alfabetica. Se hace con un numero de 3 cifras. la primera es el dueno, la
segunda el grupo y la tercera son todos. Va de 1 a 7. 4 es lectura (r), 2 es es escritura(w) y 1 es ejecucion (x). 750 es:
maximo para el dueno (7, resultado de 4+2+1, 5 para el grupo, y 0 para todos los demas) El comando para dar los máximos
permisos al usuario es:

chmod 750 prueba.sh

chmod - R o-rwx prueba.sh #Quito los permisos a otros de lectura, escritura y ejecución.

chown pepe fichero # cambia los permisos de un fichero al
usuario pepe.

chown pepd:hileras fichero # cambia un fichero a un usuario y además lo cambia de grupo.

chown -R dueño. trabajo directorio #asigna al directorio como dueño a

dueño y como grupo a trabajo, y con el parámetro -R lo hace recursivamente.

2.5.2.4 Redireccionamientos

Con el comando ls puedo listar, puede ser los ficheros que hay dentro de un directorio, el resultado lo muestra por defecto por pantalla, se puede redireccionar a un fichero o impresora.

Esto es, no imprimir en la pantalla sino en la impresora, o en un fichero.

ls 1> lista.txt

salida estandard

ls 1> lista.txt 2>error.txt

pasa el fichero lista.txt con el nombre error.txt donde se desee.

ls 1> lista.txt 2>>error.txt

pasa el fichero lista.txt con el nombre error.txt donde se desee y concatenando la salida al fichero.

grep "ServerName" /etc 2>/dev/null

Busca con el filtro que coincida con "ServerName" y lo redirecciona a un dato nulo**.**

/dev/null es un directorio donde todo lo que se envía desaparece.

Por eso, lo llaman el agujero negro.

2.5.3 El demonio CRONTAB

Crontab es una herramienta indispensable para el administrador de sistemas, su principal función radica en la automatización de tareas dentro de nuestro servidor, ya que permite su ejecución con periocidad.

Cada usuario puede tener y gestionar su propio fichero de configuración para cron. Existen los ficheros /etc/cron.allo y /etc/cron.deny donde se

pueden poner restricciones. En caso de existir el fichero /etc/crona.allow, sólo los usuarios incluidos en este fichero podrán disponer de un crontab propio. Si no existe /etc/cron.allow pero sí existe un fichero /etc/cron.deny, cualquier usuario incluido en este último fichero no podrá disponer de fichero crontab propio.

Los ficheros de configuración de crontab no están diseñados para editarse directamente por el usuario, aunque se podría pues son ficheros de texto, los cuales se pueden modificar mediante el comando crontab. El formato para utilizar crontab es:

crontab [-u usuario] fichero

crontab [-u usuario] {-l | -r |-e}

La opción -u se utiliza para indicar el usuario cuyo crontab se desea gestionar. Sólo root podrá usar la orden crontab con esta opción. La ausencia de esta opción supone que es el usuario que ejecuta la orden el que gestiona su propio crontab.

La opción -l muestra el crontab activo en la salida estándar

La opción -r se usa para crear y editar el crontab activo mediante el editor especificado en las variables de entorno EDITOR. El crontab modificado se instala automáticamente al salir del editor guardando los cambios.

Cron funciona continuamente. Se puede parar o arrancar pero no es necesario.

El script de arranque se encuentra en: **/etc/init.** También existe un script para

cron.

La línea de programación de tareas sigue un formato estándar formado por cinco campos que indican un instante de ejecución y la ruta del fichero que hay que ejecutar.

Estructura del fichero de configuración de cron

CUANDO > ruta>programa, script, comando...

min Horas Día Mes Día de la semana

0-59 0-23 1-31 12 0-7

(7 y 0 es lo mismo)

ejemplo 0 12 3 * *

ejemplo 0,3 0,4,5 * * *

* significa cualquiera o mejor dicho todos.

Los campos que describen el instante de ejecución son por orden:

m: El valor del minuto que se desea fijar entre 0-59 minutos.

h: El valor de la hora a la cual se ejecutará el script, fijo entre 0-23 horas.

dom: Día del mes, fijar entre 1 - 31.

mon: Mes del año fijado entre 0 - 12 (o su nombre con las tres primeras letras en inglés).

dow: Día de la semana, fijar entre 0 - 7, (0 o 7 indica domingo, o su nombre con las primeras letras en inglés).

Un campo puede contener:

Un asterisco (*) para indicar todos los posibles valores.

Un valor fijo para indicar un minuto, hora, día o mes.

Un rango de valores, dos números separados por guiones. Un rango puede terminar en /numero para indicar el incremento.

Una lista de valores separados por comas.

Un valor */número para indicar todos los valores con incremento de "número".

Los otros campos son:

user:Usuario que ejecuta el script.

command: El comando a ejecutar.

Hay unos croptab predeterminados por el sistema, para facilitar la

programación de tareas de administración, el crontab estándar también permite ejecutar tareas cada hora, cada día, cada semana o cada mes; se crea un directorio para cada una de estas tareas y todos los ficheros ejecutables que se pongan allí, normalmente guiones de Shell, se ejecutarán automáticamente. Los directorios en cuestión son /etc/cron.hourly, /etc/cron.dail, /etc/cron.weekly y /etc/cron.monthly.

2.5.4 Secure Sockey Layer (SSL)

El protocolo de capa de conexión segura (SSL) es un sistema diseñado y propuesto por Netscape Communications Corporation en 1994, que proporciona comunicaciones seguras por red.

Se encuentra en la pila OSI entre los niveles deTCP/IP y de los protocolos HTTP, FTP, SMTP, etc. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4 o IDEA, y cifrando la clave de sesión de RC4 o IDEA mediante un algoritmo de cifrado de clave pública, típicamente el RSA. La clave de sesión es la que se utiliza para cifrar los datos que vienen del y van al servidor seguro. Se genera una clave de sesión distinta para cada transacción, lo cual permite que aunque sea reventada por un atacante en una transacción dada, no sirva para descifrar futuras transacciones. MD5 se usa como algoritmo de hash.

Proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y, opcionalmente, autenticación de cliente para conexiones TCP/IP.

Cuando el cliente pide al servidor seguro una comunicación segura, el servidor

abre un puerto cifrado, gestionado por un software llamado Protocolo SSL
Record, situado encima de TCP. Será el software de alto nivel, Protocolo SSL
Handshake, quien utilice el Protocolo SSL Record y el puerto abierto para comunicarse de forma segura con el cliente.

Hay algunos términos por definir:

RSA Private Keys: fichero digital que se puede usar para descifras mensajes que nos mandan. Tiene una parte pública (que se distribuye con nuestro certificado), que permite a la gente cifrar los mensajes que nos manda. Este mecanismo de clave asimétrica nos asegura que los mensajes cifrados con la clave pública (que se distribuyo a mucha gente) sólo pueden ser descifrados con la clave privada (que sólo la conoce el administrador).

Certificate Signing Request (CSR): es un fichero digital que contiene nuestra clave pública y nuestro nombre.

Certification Authority (CA): entidad de confianza encargada de firmar certificados (CSR).

Certificate (CRT): Una vez la CA ha firmado el CSR, se obtiene un CRT. Este fichero contiene nuestra clave pública, nuestro nombre, el nombre de al CA, y está firmado digitalmente por la CA. De esta forma otras entidades pueden verificar esta firma para comprobar la veracidad del certificado. Es decir, se obtiene un certificado que está firmado por una CA que se considera de confianza, se puede confiar también en la autenticidad del certificado.

SSL se puede configurar de diferentes formas.

Por ejemplo:

- Cualquier cliente puede conectarse a una URL determinada, usando https. En este caso el servidor enviará su certificado al cliente para que este pueda descifrar la información que le llega del servidor y cifrar la que envía hacia el servidor.
- También se puede hacer que sólo los clientes que tengan un determinado certificado puedan conectarse a una determinada URL.
- Otra posibilidad es combinar las técnicas de autenticación. De forma que cuando se intenta acceder a una determinada URL usando https se debe autenticar primero.

El protocolo SSL Handshake

Durante el protocolo SSL Handshake, el cliente y el servidor intercambian una serie de mensajes para negociar las mejores de seguridad. Este protocolo sigue las siguientes seis fases:

La fase Hola, usada para ponerse de acuerdo sobre el conjunto de algoritmos para mantener la intimidad y para la autenticación.

La fase de intercambio de claves, en la que intercambia información sobre las claves, modo que al final ambas partes comparten una clave maestra o pública.

La fase de producción de clave de sesión, que será usada para cifrar los datos intercambiados.

La fase de producción de clave de sesión, que será la usada para cifrar los datos intercambiados.

La fase de verificación del servidor, presente sólo cuando se usa RSA

como algoritmo de intercambio de claves, y sirve para que el clientes autentique al servidor.

La fase de autenticación del cliente, en la que el servidor solicita al cliente un certificado X.509 (si es necesaria la autenticación de cliente).

Por último, la fase de fin, que indica que ya se puede comenzar la sesión segura.

El protocolo SSL Record

Este protocolo especifica la forma de encapsular los datos transmitidos y recibidos. La porción de datos del protocolo tiene tres componentes:

MAC-DATA, el código de autenticación del mensaje.

ACTUAL -DATA, los datos de aplicación a transmitir.

PADDING-DATA, los datos requeridos para rellenar el mensaje cuando se usa cifrado en bloque.

2.5.5 Secure Shell (SSH)

Hay una manera todavía segura de administrar sistemas UNIX remotamente: se denomina Secure Shell (SSH). Permite una comunicación a través de Internet. SSH es un compendio de herramientas que se corresponden aproximadamente a los comandos rsh, rcp y rlogin de Sun, pero con una diferencia importante: la paranoia. SSH permite hacer lo mismo que estos comandos utilizando un cierto grado de libertad de encriptación y métodos de autentificación. OpenSSH, un producto 100% código abierto del proyecto OpenBSD, una versión de OpenSSH está incorporada en muchas de las últimas versiones de Red Hat, Debian y SUSE Linux.Una versión libre de SSH llamada OpenSSH se encuentra disponible en el paquete Debian ssh.

Open Secure Shell (OpenSSH)

Hace algunos años, el programador finlándes Tatu Ylönen creó una herramienta terrorifícamente útil llamanda Secure Shell o SSH. Es una alternativa de código abierto, con licencia BSD, hacia la implementación propietaria y de código cerrado **SSH** creada por Tatu Ylönen. **OpenSSH** es un proyecto creado por el equipo de desarrollo de OpenBSD y actualmente dirigido por Theo de Raadt. Se considera es más segura que su contraparte propietaria debido a la constante auditoría que se realiza sobre el código fuente por parte de una gran comunidad de desarrolladores, una ventaja que brinda al tratarse de un proyecto de fuente abierta.

OpenSSH incluye servicio y clientes para los protocolos **SSH**, **SFTP** y **SCP**.

URL: http://www.openssh.org/.

SSH tiene dos protocolos de autenticación:

Protocolo SSH versión 1:

1. La versión que viene con Potato admite únicamente este protocolo.
2. Métodos de autenticación disponibles:
    - RSAAuthentication: autenticación del usuario basada en una clave RSA
    - RhostsAuthentication: autenticación basada en .rhosts (insegura, desactivada)
    - RhostsRSAAuthentication: autenticación basada en .rhosts combinada con una clave RSA (desactivada)

- ChallengeResponseAuthentication: autenticación basada en challenge-response RSA
- PasswordAuthentication: autenticación basada en contraseña

Protocolo SSH versión 2:

1. Versiones posteriores a Woody usan este protocolo como protocolo principal.
2. Métodos de autenticación disponibles:
    - PubkeyAuthentication: autenticación del usuario basada en una clave pública
    - HostbasedAuthentication: autenticación basada en.rhosts o /etc/hosts.equiv combinada con la autenticación de la clave pública de la máquina cliente (desactivada)
    - ChallengeResponseAuthentication: autenticación basada en challenge-response
    - PasswordAuthentication: autenticación basada en contraseña

Herramientas de SSH:

sshd: El demonio que actuá como un servidor para todos los demás comandos de SSH.

ssh: La primera herramienta de usuario final. Se utiliza para shell remota, comando remoto y envío de sesiones.

scp: (**S**ecure **C**o**p**y, o Copia Segura) es una protcolo seguro para transferir ficheros entre un anfitrión local y otro remoto, a través de

SSH. Básicamente, es idéntico a **RCP** (**R**emote **C**o**p**y, o Copia Remota), con la diferencia de que los datos son cifrados durante la transferencia para evitar la extracción potencial de información a través de programas de captura de las tramas de red (**packet sniffers**). **SCP** solo implementa la transferencia de ficheros, pues la autenticación requerida es realizada a través de **SSH**.

sftp: (**S**SH **F**ile **T**ransfer **P**rotocol) es un protocolo que provee funcionalidad de transferencia y manipulación de ficheros a través de un flujo confiable de datos. Comúnmente se utiliza con **SSH** para proveer a éste de transferencia segura de ficheros.

ssh-keygen: Genera los pares de clave privada/pública para utilizar en autentificación RSA y DSA (incluyendo claves de host).

ssh-agent: Un demonio utilizado para automatizar las autentificaciones en el cliente RSA/DSA.

ssh-add: Carga la clave privada en un proceso del agente ssh-agent.

ssh-askpass: Proporciona una interfaz X Window al ssh-add.

2.5.6 Comando MYSQLDUMP

El gestor de Bases de datos MySQLtiene soporte para copias de seguridad, incluyendo varias herramientas para ello. Mediante ellas se puede tener un respaldo de los datos, para que, en el eventual caso de que se pierdan, se puedan recuperar. .

En el momento de hacer una copia de seguridad, lo primero que se hay que tener en cuenta es la integridad de los datos que se estén guardando. En todos los casos es necesario que haya integridad en los datos de una tabla, con esto se quiere decir que todos los datos de la tabla deberán estar escritos en la misma, esto porque como todos los

gestores de bases de datos, Mysql dispone de diversas "caches" en las que se almacenan datos temporalmente con el objetivo de mejorar en rendimiento, por ejemplo, una vez hecha una modificación en una tabla, puede ser que los datos no se guarden inmediatamente en disco, hasta que termine la transacción, es el caso de una consulta que se estaba ejecutando. Por ello, es necesario "forzar" a Mysql a escribir todos los datos en el disco, mediante la sentencia "Flush Tables".

Es necesario que no se escriba en las tablas mientras se esta haciendo la copia de seguridad de la base de datos, que se consigue con el comando "lock tables", seguido del nombre de la tabla. Puede haber bases de datos en las que sea necesario bloquear todas las tablas al mismo tiempo antes de hacer la copia de seguridad.

Existen varias opciones para realizar la copia de seguridad de una base de datos de Mysql,

Se podría utilizar alguna herramienta comercial que gestione todo el proceso, esto tiene como ventaja la simplicidad del método, y como inconveniente, que no suelen ser gratis, sino que hay que pagar licencia por la utilización de los mismas.

A partir de la versión 3.23.25 y posteriores de MySQL, existe la posibilidad de realizar una copia de seguridad a través de la sentencia sql "backup table".

Como tercera opción, es posible realizar copias de seguridad a través de las herramientas que nos proporciona el propio gestor de base de datos de MySQL, como pueden ser mysqldump ó mysqlhotcopy.

Comando mysqldump

El comando mysqldump del sistema gestor de base de datos MySQL sirve para hacer copias de seguridad. Este comando permite hacer la copia de seguridad de una o múltiples bases de datos. Además permite que estas copias de seguridad se puedan restaurar en distintos tipos de gestores de bases de datos, sin la necesidad de que se trate de un gestor de mysql. Esto lo consigue creando unos ficheros, que contienen todas las sentencias sql necesarias para poder restaurar la tabla, que incluyen desde la sentencia de creación de la tabla, hasta una sentencia insert por cada uno de los registros que forman parte de la misma.

El comando dispone de una amplia variedad de opciones que nos permitirá realizar la copia de la forma más conveniente para el propósito de la misma.

Para poder restaurar la copia de seguridad, se ejecutan todas las sentencias sql que se encuentran dentro del fichero, bien desde la línea de comandos de mysql, o desde la pantalla de creación de sentencias sql de cualquier entorno gráfico como puede ser el Mysql Control Center.

Las limitaciones de la restauración dependerán de las opciones que se han especificado a la hora de hacer la copia de seguridad, por ejemplo, si se incluye la opción --add-drop-table al hacer la copia de seguridad, se podrán restauran tablas que existen actualmente en el servidor (borrándolas primero). Por lo que es necesario estudiar primero los procedimientos que se utilizarán tanto en la copia como en la restauración, para que tenga el comportamiento correcto que se desea.

Algunas de las opciones que tiene son:

--add-locks

Añade LOCK TABLES antes, y UNLOCK TABLE después de la copia de cada tabla.

--add-drop-table

Añade un drop table antes de cada sentencia create

-A, --all-databases

Copia todas las bases de datos. Es lo mismo que utilizar --databases seleccionando todas.

-a, --all

Incluye todas las opciones de creación específicas de Mysql.

--allow-keywords

Permite la creación de nombes de columnas que son palabras clave, esto se realiza poniendo de prefijo a cada nombre de columna, el nombre de la tabla

-c, --complete-insert

Utiliza inserts incluyendo los nombres de columna en cada sentencia (incrementa bastante el tamaño del fichero)

-C, --compress

Comprime la información entre el cliente y el servidor, si ambos soportan compresión.

-B, --databases

Para copiar varias bases de datos. En este caso, no se especifican tablas. El nombre de los argumentos se refiere a los nombres de las bases de datos. Se incluirá USE db_name en la salida antes de cada base de datos.

--delayed

Inserta las filas con el comando INSERT DELAYED.

-e, --extended-insert

Utiliza la sintaxis de INSERT multilinea. (Proporciona sentencias de insert más compactas y rápidas.)

-#, --debug[=option_string]

Utilización de la traza del programa (para depuración).

--help

Muestra mensaje de ayuda y termina.

--fields-terminated-by=...

--fields-enclosed-by=...

--fields-optionally-enclosed-by=...

--fields-escaped-by=...

--lines-terminated-by=...

Estas opciones se utilizan con la opción -T y tienen el mismo significado que la correspondiente cláusula LOAD DATA INFILE.

-F, --flush-logs

Escribe en disco todos los logs antes de comenzar con la copia

-f, --force,

Continúa aunque se produzca un error de SQL durante la copia.

-h, --host=..

Copia los datos del servidor de Mysql especificado. El servidor por defecto es localhost.

-l, --lock-tables.

Bloquea todas las tablas antes de comenzar con la copia. Las tablas se bloquean con READ LOCAL para permitir inserts concurrentes en caso

de las tablas MyISAM. Cuando se realiza la copia de múltiples bases de datos, --lock-tables bloqueará la copia de cada base de datos por separado. De forma que esta opción no garantiza que las tables serán consistentes lógicamente entre distintas bases de datos. Las tablas en diferentes bases de datos se copiarán en estados completamente distintos.

-K, --disable-keys

Se inluirá en la salida /*!40000 ALTER TABLE tb_name DISABLE KEYS */; y /*!40000 ALTER TABLE tb_name ENABLE KEYS */; Esto hará que carga de datos en un servidor MySQL 4.0 se realice más rápido debido a que los índices se crearán después de que todos los datos hayan sido restaurados.

-n, --no-create-db

No se incluirá en la salida CREATE DATABASE /*!32312 IF NOT EXISTS*/ db_name; Esta línea se incluye si la opción --databases o --all-databases fue seleccionada.

-t, --no-create-info

No incluirá la información de creación de la tabla (sentencia CREATE TABLE).

-d, --no-data

No incluirá ninguna información sobre los registros de la tabla. Esta opción sirve para crear una copia de sólo la estructura de la base de datos.

--opt

Lo mismo que --quick --add-drop-table --add-locks --extended-insert --lock-tables. Esta opción le debería permitir realizar la copia de seguridad de la base de datos de la forma más rápida y efectiva.

-pyour_pass, --password[=your_pass]

Contraseña utilizada cuando se conecta con el servidor. Si no se especifica, `=your_pass', mysqldump preguntará la contraseña.

-P, --port=...

Puerto utilizado para las conexiones TCP/IP

--protocol=(TCP | SOCKET | PIPE | MEMORY)

Especifica el protocolo de conexión que se utilizará.

-q, --quick

No almacena en el buffer la sentencia, la copia directamente a la salida. Utiliza mysql_use_result() para realizarlo.

-Q, --quote-names

Entrecomilla las tablas y nombres de columna con los caracteres ``'.

-r, --result-file=...

Redirecciona la salida al fichero especificado. Esta opción se debería utilizar en MSDOS, porque previene la conversión de nueva línea `\n' en nueva línea y retorno de carro`\n\r'.

--single-transaction

Utiliza el comando BEGIN antes de realizar la copia desde el servidor. Es muy útil con las tables InnoDB y el nivel de transacción READ_COMMITTED, porque en este modo realizará la copia de seguridad en un estado consistente sin necesidad de bloquear las aplicaciones.

-S /path/to/socket, --socket=/path/to/socket

El fichero de sockets que se especifica al conectar al localhost (que es el host predeterminado).

--tables

sobreescribe la opción --databases (-B).

-T, --tab=path-to-some-directory

Crea un fichero table_name.sql, que contiene la sentencia de creación de SQL, y un fichero table_name.txt, que contiene los datos de cada tabla. El formato del fichero `.txt' se realiza de acuerdo con las opciones --fields-xxx y --lines--xxx options. Nota: Esta opción sólo funciona si el comando mysqldump se ejecuta en la misma máquina que el demonio mysqld, el usuario deberá tener permisos para crear y escribir el fichero en la ubicación especificada.

-u nombre_usuario, --user=nombre_usuario

El nombre de usuario que se utilizará cuando se conecte con el servidor, el valor predeterminado es el del usuario actual.

-v, --verbose

Va mostrando información sobre las acciones que se van realizando (más lento)

-w, --where='cláusula where'

Sirve para realizar la copia de determinados registros

-X, --xml

Realiza la copia de seguridad en un documento xml

-x, --first-slave

Bloquea todas las tablas de todas las bases de datos

Ejemplos de comandos mysqldump:

Para realizar la copia se seguridad de la base de datos mibasededatos al fichero copia_seguridad.sql

mysqldump --opt mibasededatos > copia_seguridad.sql

Otro ejemplo más complejo de comando mysqldump para hacer el backup de una base de datos es el siguiente:

mysqldump --opt --password=miclave --user=miuser mibasededatos > archivo.sql

En este último caso se indica un nombre de usuario y una clave para acceder a la base de datos sobre la que se está haciendo el backup: mibasededatos. Las sentencias SQL para reconstruir esa base de datos se volcarán en el fichero archivo.sql.

Restaurar la base de datos

Si se desea recuperar la información de un fichero para restaurar una copia de seguridad de la base de datos se hace con el comando mysql. Utilizando una sintaxis como esta:

mysql mibasededatos < archivo.sql

En este ejemplo se restauraría la base de de datos mibasededatos con el backup almacenado en el fichero archivo.sql.

Otro ejemplo más complejo de comando para restaurar una base de datos es el siguiente:

mysql --password=miclave --user=miuser mibasededatos < archivo.sql

Es el mismo ejemplo que el anterior, pero indicando un nombre de usuario y una clave con las que acceder a la base de datos mibasededatos.

2.6 SEGURIDAD EN SERVIDORES GNU/LINUX

La seguridad dentro de los servidores puede ser frustrante y liberadora a la vez.

Una vez se comprenda que la seguridad es un proceso constante y continuo, se puede aproximar a la seguridad del sistema y reducir las oportunidades en las que su sistema esté comprometido, por lo menos se puede minimizar la duración y daños provocados por los atacantes .

El proceso de asegurar el servidor puede ser llamado hardering o endurecimiento o blindaje del servidor, blindaje que se hace en el marco del objetivo del servidor, es decir, de acuerdo a los servicios que proporciona. En pocas palabras blindar un servidor es realizar una configuración cuidadosa, un host o servidor que proporcione un restringido pero a la vez servicio público accesible a usuarios y sistemas no conocidos. En este aspecto GNU/LINUX ha sido un éxito como plataforma para servidores, incluso en organizaciones que contrariamente poseen un fuerte entorno de costo comercial con sistemas operativos como Windows, Linux se sitúa dentro de la infraestructura como puerta de enlace SMTP y servidores DNS por su fiabilidad, bajo costo, y calidad de sus aplicaciones de servidor. Además Linux y TCP/IP, son la lengua franca de Internet, van unidos. Cualquier cosa que se pude a hacer sobre una red TCP/IP puede ser trasladada a Linux, y prácticamente con muy pocas excepciones.

2.6.1 Componentes de los riesgos

Un riesgo es la relación entre sus activos y las vulnerabilidades que

cualquier atacante aprovecharía para adueñarse o interferir sobre estos. Sobre los tres factores principales que interactúan en el escenario de seguridad del servidor, activos, vulnerabilidades y atacantes, se puede tener cierto grado de control sobre sus activos y vulnerabilidades, pero casi nunca se tendrá control sobre los atacantes.
Entre los componentes que comprende el análisis de riesgos están:

2.6.1.1 Activos

Son la respuesta a la siguiente pregunta: ¿Qué es lo que se esta intentando proteger? Probablemente sea la información que se maneja en el servidor y no se desee sea pública. Pero este sólo es el comienzo. Si alguien compromete un sistema, ¿qué clases de riesgos conllevará a otros sistemas dentro de la red?¿Qué tipo de datos son almacenados o manejados por estos otros sistemas, o parte de esta información es confidencial?¿Cuáles de estas ramificaciones pueden ser manipuladas por alguien o directamente robada?¿Y cómo afectará el robo de esta información sobre nuestra reputación si es noticia?.
Generalmente, se quiere proteger datos y sistemas, ambos individualmente. Los computadores, la red y los datos son los activos de información que directamente se ven afectados por un ataque, pero indirectamente a través de estos se ven afectados otros activos. Algunos ejemplos de ellos, serían la confidencialidad de los datos de los clientes, su reputación y la protección contra la responsabilidad de perder información sustancial de los clientes y por supuestos las pérdidas que se produzcan por las víctimas de los ataques originados en los sistemas comprometidos.

2.6.1.2 Objetivos de seguridad

Una vez determinado lo que necesita proteger, debe decidir qué niveles y tipos de protección requiere cada activo. Estos pueden definir los objetivos de seguridad y ser clasificados en categorías interrelacionadas: integridad y confidencialidad de datos, integridad de sistema y disponibilidad del sistema/red.

- Confidencialidad de datos: Algunos tipos de datos necesitan ser protegidos contra escuchas y otras revelaciones inapropiadas.
- Integridad de datos: Sin tener en cuenta la necesidad de mantener una parte o datos secretos, necesita asegurar que no se alterarán inadecuadamente por ningún medio. No sólo hay que pensar en un contexto de transmisión de datos segura, sino también deben ser protegidos contra la manipulación incluso si no se transmiten.
- Integridad del sistema: Hace referencia a que el sistema no esté comprometido y no sea alterable inadecuadamente, es decir está siendo utilizado como sus administradores han decidió (siendo utilizado sólo por los usuarios autorizados, sin más privilegios de los que hayan sido asignados).

  La integridad del sistema puede ser afectada tanto por los usuarios remotos (conexión a través de una red) como por los usuarios locales escalando en su propio nivel de privilegios en el sistema.

  Para afrontar los ataques se debe pensar anticipadamente que hacer en caso de:

  - Los datos almacenados en el sistema o disponible mediante relaciones de confianza podrían estar también comprometidos; es decir, tales datos dejarían de considerarse como confidenciales o inalterables.

- Los ejecutables del sistema por sí mismos deberían también estar comprometidos.
- Si ejecuta el comando ps auxw para ver todos los procesos que se están ejecutando en un sistema comprometido, ¿está realmente viendo todo?, o ¿podría reemplazar el binario ps por uno que esté conveniente blindado contra los procesos de posibles atacantes?, como ocurre cuando se utilizan rootkits en los ataques.

  Los rootkits son colecciones de binarios modificados de los comandos más comunes como ps, ls y who.

- Disponibilidad del sistema/red: Es una forma de decir que el sistema esté disponible para los usuarios. Una red o sistema que no responde a las peticiones de los usuarios se dice que no está disponible. Una de las formas más comunes de comprometer un servidor atacando el objetivo de disponibilidad es el llamado distribución de denegación de servicios (DDoS, Distributed Denial of Service). Su función es lanzar un gran número de agentes que se ejecuten y comprometan el sistema llegando a saturar al host. Lo único esperanzador de este tipo de ataques de disponibilidad del sistema es que una vez finalizado el mismo, el sistema o red normalmente se recuperan rápidamente. Es más, excepto que esté combinado con otros ataques, DDoS raramente ataca directamente a los datos confidenciales o a la integridad de datos/sistema.

  La realidad es que muchos de estos DDoS son imposibles de prevenir debido a la dificultad de diferenciarlos del gran volumen de tráfico “legítimo”. Para contrarestar esto se pude disuadir (intentar identificar y perseguir a los atacantes) y la redundancia

es uno de los diseños de sistemas/red viable como defensa de estos ataques. Pero incluso así, la redundancia no imposibilita los ataques; simplemente incrementa el número de sistemas a atacar simultáneamente.

2.6.1.3 Amenazas

¿Quién podría atacar su sistema, red o datos? Estos atacantes incluyen los rutinarios (personas con acceso a información confidencial, vándalos, personas de mantenimiento y naturaleza), lo sensacionalistas (redes criminales organizadas y extorsionistas) y todos los que se encuentran en todos los niveles de acceso.

2.6.1.4 Motivos

Muchas amenazas son bastante obvias y fáciles de comprender. Es de conocimiento general que los competidores en los negocios desean hacer más dinero u obtener más ventajas competitivas por intereses personales, comerciales o académicos y también pueden estar involucrados empleados descontentos o anteriores administradores del servidor que a menudo quieren vengarse por haber sido despedidos obrando mal. Otros motivos no son tan fáciles de fijar.

Motivos financieros

Una de las razones más comprensibles y poderosas de los delitos informáticos es el dinero. Los empresarios pagan a los espías industriales para romper los sistemas de sus competidores y apropiarse de los datos.

Motivos políticos

En los últimos años se han incrementado los ataques por motivos

políticos, dado que los delitos informáticos son mucho más que una parte de los conflictos humanos modernos; no es sorprendente que incluya conflictos militares y políticos a nivel de gobiernos de estado.
Este motivo político también se presenta a nivel de empresas e instituciones donde se atacan los sitios web para afectar políticas administrativas, por diferencias de opiniones, o a nivel de grupos formados por comunidades afines.

Motivos personales/sicológicos
La baja autoestima, el deseo de impresionar a otros, venganza contra la sociedad en general o compañía en particular y organización, curiosidad desacertada, un romanticismo equivocado de la "clandestinidad informática" , búsqueda de emociones y una vieja misantropía son los motivadores comunes, a menudo combinados.

2.6.1.5 Vulnerabilidades y ataques
Un riesgo no es pertinente solo a activos y atacantes; si un activo no tiene vulnerabilidades (lo que es imposible en la práctica), no hay riesgos y no importa cuántos atacantes haya.
Observe que la vulnerabilidad sólo representa un ataque potencial, y así se mantendrá hasta que alguien sepa cómo explotar dicha vulnerabilidad en un ataque con éxito. En la mayoría de los casos, es peligroso no tener en cuenta una vulnerabilidad conocida porque todavía no haya sido atacada. Es como ignorar una bomba porque no se oiga el tic-tac. El punto a tener en cuenta no es si una vulnerabilidad puede ser explotada, sino si es previsible que pueda ser explotada para adoptar las medidas adecuadas.

Para iniciar un análisis de riesgos respecto a los miles de ataques posibles contra un sistema dado, necesita identificar y orientarse a identificar:

- Vulnerabilidades que claramente son aplicables en su sistema y deben ser atenuadas inmediatamente.
- Aquellas que probablemente se puedan aplicar en el futuro y contra las que se deben crear planes de choque.
- Todas las que tienen mucha probabilidad de dar problemas más tarde pero son fáciles de atenuar.

2.7 SERVIDOR DE ALTA DISPONIBILIDAD

2.7.1 Cualidades del servidor

En los últimos años con el continuo y gran crecimiento de Internet, es común que los servidores reciban miles o millones de conexiones diarias, teniendo que atender concurrentemente quizás a cientos o miles. Esto conlleva a contar con una enorme cantidad de recursos computacionales para enfrentar las demandas crecientes por parte de los usuarios de los diferentes servicios ofrecidos por la web.

Los requerimientos para los sistemas que puedan hacer frente a estas demandas deben incluir:

Escalabilidad: Característica que permite al sistema extenderse y crecer para poder atender el aumento de las solicitudes de un servicio web sin que los tiempos de respuesta se vean afectados.

Alta Disponibilidad: El servicio debe estar disponible en todo momento, esto se ve limitado por la tolerancia a fallos del sistema.

Costos aceptables: La inversión inicial y los costos de mantenimiento y expansión del sistema deben ser accesibles.

Para implementar soluciones que cumplan con estas exigencias del sistema se han desarrollado varios modelos basados en el procesamiento paralelo. Estos modelos se dividen en dos grandes enfoques: el fuertemente acoplado y, como contrapartida, el débilmente acoplado.

En le primer enfoque se utilizan equipos con múltiples procesadores que pueden ejecutar instrucciones concurrentemente y comparte el acceso a los datos locales, mientras que en el segundo se utilizan varios equipos independientes que colaboran entre sí para cumplir con un objetivo.

El enfoque de computación paralela fuertemente acoplada ha sido utilizado para todo tipo de tareas, sin embargo, esta solución, aunque efectiva y ampliamente aceptada, tiene serias desventajas principalmente en cuanto a escalabilidad y costos.

2.7.2 Antecedentes

Para introducirse en el concepto de alta disponibidad se puede pasar por su unidad básica conformada por 2 computadores que se pueden unir bajo la configuración de cluster o para este caso servidor de alta disponibilidad.

.

- El comienzo del término cluster y del uso de este tipo de tecnología es desconocido pero se puede considerar que comenzó a finales de los años 50 y principios de los años 60.
- La base formal en la Ingeniería informática de hacer trabajos paralelos de cualquier tipo fue posiblemente inventado por Gene Amdahl de IBM, que en 1967 publicó lo que ha llegado a ser considerado como el inicio del procesamiento paralelo: la Ley de

Amdahl que describe matemáticamente lo que se puede esperar paralelizando cualquier otra serie de tareas realizadas en una arquitectura paralela.

- La historia de los primeros grupos de computadoras es más o menos directamente ligado a la historia de principios de las redes, como una de las principales motivaciones para el desarrollo de una red para enlazar los recursos de computación, de hecho la creación de un clúster de computadoras. Las redes de conmutación de paquetes fueron conceptualmente inventados por la corporación RAND en 1962.

  Utilizando el concepto de una red de conmutación de paquetes, el proyecto ARPANET logró crear en 1969 lo que fue posiblemente la primera red de computadoras, basadas en el clúster de computadoras por cuatro tipos de centros informáticos (cada una de las cuales fue algo similar a un “clúster”, pero no un “comodity cluster” como hoy en día se entiende.
- El primer producto comercial de tipo cluster fue ARCnet, desarrollado en 1977 por Datapoint, pero no obtuvo un éxito comercial y los clusteres no consiguieron tener éxito hasta que en 1984 VAXcluster produjeran el sistema operativo VAX/VMS.

2.7.3 Definición de clúster

Es un grupo de dos o más computadoras que están interconectadas y funcionan como una sola unidad de proceso de información o de servicio.

2.7.4 Clasificación

Determinada según el uso del clúster y los servicios que ofrecen.

- **High Performance:** Son clusters en los cuales se ejecutan tareas que requieren de gran capacidad computacional por sus exigencias en el procesamiento, o grandes cantidades de memoria, o ambos a la vez. Llevando estas exigencias a comprometer el uso del cluster por largos períodos de tiempo.
- **High Availability:** Son clusters cuyo objetivo de diseño es el de proveer disponibilidad y confiabilidad. Estos clusters brindan la máxima disponibilidad de los servicios que ofrecen, ya sean servidores web o de base de datos entre otros. La confiabilidad se destaca mediante software que detecta fallos y permite recuperarse frente a los mismo, ya sea porque otros nodos suplen el funcionamiento del nodo que fallo o por hardware al estar replicados algunos elementos del sistemas para evitar tener puntos únicos de fallo.
- High Throughput: Son clusters cuyo objetivo de diseño es el de ejecutar la mayor cantidad de tareas en el menor tiempo posible. Existe independencia de datos entre las tareas individuales. El retardo entre los nodos del cluster no es considerado un gran problema.

3. ADMINISTRACIÓN DE LOS SERVIDORES DE LA ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA Y SOPORTE A USUARIOS

3.1 Automatización de las funciones más utilizadas en la administración del servidor

Se utilizó la Shell Bash para automatizar algunas funciones de la administración del servidor, ejecutando diferentes scripts e integrándolos en un script principal dispuesto a manera de menú.
Las funciones que el administrador puede ejecutar en el menú son:

Nombre	**Descripción**
administracion	Script que integra todos los script en un menú
iniciarred	Reiniciar la interfaz de red
iniciarmysql	Reiniciar el servicio de MySQL
iniciarpostgres	Reiniciar el servicio de PostgreSQL
iniciartomcat	Reiniciar el servidor Tomcat
iniciarapache	Reinicia el servidor Apache
bd_copyesclavo	Hace backups de las bases de datos con copia en el servidor de respaldo y en la usb externa
sitios	Hace backups de los sitios web con copia en el servidor de respaldo y en la usb externa
Enjaula	Crea usuarios enjaulados
Enjaulameiweb	Crea usuarios enjaulados para desarrolladores del meiweb

3.2 Configurar e implementar un servidor de respaldo

Se instaló el servidor de respaldo con el sistema operativo GNU/LINUX Debian 5 Lenny, determinando el sistema de particiones de acuerdo a las necesidades del servidor, la actualización de los últimos paquetes y parches de seguridad.

Los paquetes servidores de apache y tomcat, los motores de bases de datos MySQL y Postgres, los lenguages PHP y Java, los respectivos conectores entre los paquetes, el paquete ntp para sincronización de hora y fecha, ssh, configuración de ssl, la configuración de llaves públicas y privadas, las restricciones en permisos para directorios y ficheros, el montaje de todos los sitios web y sus respectivas bases de datos, los plugins necesarios y las debidas restricciones en cada software instalado.

Entre otras actividades de soporte a usuarios, tanto a desarrolladores de los sitios web alojados en los servidores como a los usuarios finales, para el manejo de los sitios web e instalación y adecuación de los mismos.

Organización de los equipos de desarrollo y la integración y migración de software en desarrollo y producción.

La debida documentación de instalación y procedimientos llevados a cabo quedan documentados en el servidor como documentos reservados y bajo la dirección del profesor a cargo de servidor.

3.3 Implementar nuevos mecanismos de seguridad en los servidores

3.3.1 Aspectos básicos para endurecer el sistema

Se puede seguir la siguiente lista de chequeo:

Pre-instalación

Restringir el acceso físico.

Aislar la red.

Deshabilitar los puertos y dispositivos innecesarios.

Seleccionar de dispositivo de arranque.

Configurar password de modificación de BIOS.

Instalación

Particionado.

Aislar la red.

Ajustar las opciones de inicio en el BIOS.

Creación de usuario/s no privilegiado/s.

Limpieza

Servicios de red preinstalados.

Paquetes no utilizados.

Módulos de kernel.

Ajustar /etc/fstab.

Usuarios y permisos.

Ajustar las propiedades de los usuarios (shell).

Eliminar los suid bit de los ejecutables.

Restringir el uso del cron.

Ajustar las opciones de sudo.

Limitar el acceso de root.

Crear el grupo wheel y agregar los usuarios correspondientes.

Configurar que consolas seguras (/etc/securetty).

Administración remota (SSH).

Permitir sólo versión 2.

Configurar tcpwrapper (hosts.deny y hosts.allow).

Restringir login de root.

Ajustar parámetros del kernel (/etc/sysctl.conf).

Instalar paquetes (servicio/s que brindará el servidor).

Configurar el firewall de host.

Instalar cliente para la sincronización horaria.

Actualizar los paquetes instalados (apt-get upgrade).

Planear y configurar esquema de back-up.

3.3.2 Configuración de ssl

Trabajar con SSL nos permite que todos los datos que se transfieren entre el cliente y el servidor vayan cifrados.

Se encontro estos archivos al instalar el paquete openssl:

servidor:/usr/lib/ssl# ls -l

total 20

lrwxrwx--- 1 root root 14 abr 27 2009 certs -> /etc/ssl/certs

drwxr-x--- 2 root root 4096 abr 27 2009 engines

drwxr-x--- 2 root root 4096 abr 27 2009 misc

lrwxrwxrwx 1 root root 20 abr 27 2009 openssl.cnf -> /etc/ssl/openssl.cnf

lrwxrwxrwx 1 root root 16 abr 27 2009 private -> /etc/ssl/private

Se modifica /usr/lib/ssl/openssl.cnf

default_days   = 730   # how long to certify for

default_bits   = 2048

Los scripts para generar el certificado se encuentra en /usr/lib/ssl/misc

servidor:/usr/lib/ssl/misc# ls -l

total 28

-rwxr-x--- 1 root root 5875 ene 7 2009 CA.pl

-rwxr-x--- 1 root root 3784 ene 7 2009 CA.sh

-rwxr-x--- 1 root root 119 ene 7 2009 c_hash

-rwxr-x--- 1 root root 152 ene 7 2009 c_info

-rwxr-x--- 1 root root 112 ene 7 2009 c_issuer

-rwxr-x--- 1 root root 110 ene 7 2009 c_name

Modificar el script para crear nuestra propia CA , /etc/usr/lib/misc/CA.pl

$DAYS="-days 365";   # 1 year

$CADAYS="-days 1095";   # 3 years

print "Making CA certificate ...\n";
reemplace esto:

```
system ("$REQ -new -keyout " .
```

por:

```
system ("$REQ -newkey rsa:2048 -keyout " .
```

Crear la CA

delfin:/usr/lib/ssl/misc# ./CA.pl -newca

CA certificate filename (or enter to create)

Making CA certificate ...

Generating a 2048 bit RSA private key
.....+++.....+++

writing new private key to './demoCA/private/cakey.pem'

Enter PEM pass phrase:

Verifying - Enter PEM pass phrase:-----

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:CO

State or Province Name (full name) [Some-State]:Santader

Locality Name (eg, city) []:Bucaramanga

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Escuela de Sistemas_UIS

Organizational Unit Name (eg, section) []:Adm. Servidores

Common Name (eg, YOUR name) []:Certification Authority

Email Address []: sistemas.servidor@gmail.com

Please enter the following 'extra' attributes

to be sent with your certificate request
A challenge password []:
An optional company name []:

Using configuration from /usr/lib/ssl/openssl.cnf

Enter pass phrase for ./demoCA/private/cakey.pem:

Check that the request matches the signature

Signature ok

Certificate Details:

Serial Number:

89:f9:49:99:3c:0e:19:39

Validity

Not Before: Nov 25 15:27:26 2009 GMT

Not After : Nov 23 15:27:26 2019 GMT

Subject:

countryName = CO

stateOrProvinceName = Santader

organizationName = Escuela de Sistemas_UIS
organizationalUnitName = Adm. Servidores

commonName = Certification Authority

emailAddress = sistemas.servidor@gmail.com

X509v3 extensions:

X509v3 Subject Key Identifier:

92:91:FE:E1:9B:19:D6:6E:AF:4B:A9:65:FA:FE:89:09:62:83:D8:BE

X509v3 Authority Key Identifier:

keyid:92:91:FE:E1:9B:19:D6:6E:AF:4B:A9:65:FA:FE:89:09:62:83:D8:BE

DirName:/C=CO/ST=Santader/O=Escuela de Sistemas_UIS/OU=Adm. Servidores/CN=Certification Authority/emailAddress=sistemas.servidor@gmail.com

serial:89:F9:49:99:3C:0E:19:39

X509v3 Basic Constraints:

CA:TRUE

Certificate is to be certified until Nov 23 15:27:26 2019 GMT (3650 days)

Write out database with 1 new entries

Data Base Updated

Ya se es autoridad certificadora. Se genera el directorio demoCA y dentro de él cacert.pem y cakey.pem.

servidor:/misc# ls -l

total 44

drwxr-x--- 3 root root 4096 nov 25 09:37 archivos_generados20091119

-rwxr-x--- 1 root root 5888 nov 18 12:09 CA.pl

-rwxr-x--- 1 root root 5875 nov 18 11:53 CA.pl20091118

-rwxr-x--- 1 root root 3784 ene 7 2009 CA.sh

-rwxr-x--- 1 root root 119 ene 7 2009 c_hash

-rwxr-x--- 1 root root 152 ene 7 2009 c_info

-rwxr-x--- 1 root root 112 ene 7 2009 c_issuer

-rwxr-x--- 1 root root 110 ene 7 2009 c_name

drwxr-xr-x 6 root root 4096 nov 25 10:27 **demoCA**

servidor:/ssl/misc# cd demoCA/

servidor:/ssl/misc/demoCA# ls -l

total 44

-rw-r--r-- 1 root root 5341 nov 25 10:27 **cacert.pem**

-rw-r--r-- 1 root root 1155 nov 25 10:27 careq.pem

drwxr-xr-x 2 root root 4096 nov 25 10:12 certs

drwxr-xr-x 2 root root 4096 nov 25 10:12 crl

-rw-r--r-- 1 root root 3 nov 25 10:12 crlnumber

-rw-r--r-- 1 root root 196 nov 25 10:27 index.txt

-rw-r--r-- 1 root root 21 nov 25 10:27 index.txt.attr

-rw-r--r-- 1 root root 0 nov 25 10:12 index.txt.old

drwxr-xr-x 2 root root 4096 nov 25 10:27 newcerts

drwxr-xr-x 2 root root 4096 nov 25 10:12 private

-rw-r--r-- 1 root root 17 nov 25 10:27 serial

Crear el CSR

Ahora se puede a generar una clave privada y una petición de certificado para el
sitio web delfin.uis.edu.co, con la clave sin encriptar

servidor:/usr/lib/ssl/misc# ./CA.pl -newreq-nodes

Generating a 2048 bit RSA private key
.....+++
.....+++

writing new private key to 'newkey.pem'

You are about to be asked to enter information that will be incorporated

into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [AU]:CO

State or Province Name (full name) [Some-State]:Santander
Locality Name (eg, city) []:Bucaramanga

Organization Name (eg, company) [Internet Widgits Pty Ltd]:Servidores y Clusters Sistemas
Organizational Unit Name (eg, section) []:Administracion de Servidores

Common Name (eg, YOUR name) []:delfin.uis.edu.co

Email Address []:sistemas.servidor@gmail.com

Please enter the following 'extra' attributes

to be sent with your certificate request

A challenge password []:

An optional company name []:

Request is in newreq.pem, private key is in newkey.pem

Se crearon los archivos newkey.pem y newreq.pem:
servidor:/ssl/misc# ls -l

total 52

-rwxr-x--- 1 root root 5888 nov 18 12:09 CA.pl

-rwxr-x--- 1 root root 3784 ene 7 2009 CA.sh
-rwxr-x--- 1 root root 119 ene 7 2009 c_hash

-rwxr-x--- 1 root root 152 ene 7 2009 c_info

-rwxr-x--- 1 root root 112 ene 7 2009 c_issuer

-rwxr-x--- 1 root root 110 ene 7 2009 c_name

drwxr-xr-x 6 root root 4096 nov 25 10:27 demoCA

-rw-r--r-- 1 root root 1679 nov 25 10:34 **newkey.pem**

-rw-r--r-- 1 root root 1155 nov 25 10:34 **newreq.pem**

Crear el CRT

Ahora, con la clave privada del certificado raíz, se firma la petición de certificado, para que éste ya esté completo:

delfin:/usr/lib/ssl/misc# ./CA.pl -sign

Using configuration from /usr/lib/ssl/openssl.cnf

Enter pass phrase for ./demoCA/private/cakey.pem:

Check that the request matches the signature

Signature ok

Certificate Details:

Serial Number:

89:f9:49:99:3c:0e:19:39

Validity

Not Before: Nov 25 15:37:15 2009 GMT

Not After : Nov 25 15:37:15 2011 GMT

Subject:

countryName = CO

stateOrProvinceName = Santander

localityName = Bucaramanga

organizationName = Servidores y Clusters Sistemas - UIS

organizationalUnitName = Administracion de Servidores

commonName = servidor.uis.edu.co

emailAddress = sistemas.servidor@gmail.com

X509v3 extensions:

X509v3 Basic Constraints:

CA:FALSE

Netscape Comment:

OpenSSL Generated Certificate

X509v3 Subject Key Identifier:

C8:74:53:B0:97:5F:BC:34:DE:56:49:19:9B:EA:C0:26:79:62:4F:F5

X509v3 Authority Key Identifier:

keyid:92:92:FE:E1:9B:22:D6:6E:AF:3B:A9:65:FA:FE:89:09:62:83:D8:BE

Certificate is to be certified until Nov 25 15:37:15 2011 GMT (730 days)

Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y

Write out database with 1 new entries

Data Base Updated

Signed certificate is in newcert.pem

Se creo el archivo newcert.pem:
servidor:/ssl/misc# ls -l

total 60

-rwxr-x--- 1 root root 5888 nov 18 12:09 CA.pl

-rwxr-x--- 1 root root 3784 ene 7 2009 CA.sh

-rwxr-x--- 1 root root 119 ene 7 2009 c_hash

-rwxr-x--- 1 root root 152 ene 7 2009 c_info

-rwxr-x--- 1 root root 112 ene 7 2009 c_issuer

-rwxr-x--- 1 root root 110 ene 7 2009 c_name

drwxr-xr-x 6 root root 4096 nov 25 10:37 demoCA

-rw-r--r-- 1 root root 4996 nov 25 10:37 newcert.pem

-rw-r--r-- 1 root root 1679 nov 25 10:34 newkey.pem

-rw-r--r-- 1 root root 1155 nov 25 10:34 newreq.pem

Los archivos generados

drwxr-xr-x 6 root root 4096 2009-02-18 20:06 demoCA/ ← Directorio que contiene el certificado raíz y su clave privada (demoCA/cacert.pem es el certificado y demoCA/private/cakey.pem es la clave privada)
-rw-r--r-- 1 root root 3532 2009-02-18 20:06 newcert.pem ← Certificado de delfin.uis.edu.co firmado por la autoridad certificadora

-rw-r--r-- 1 root root 963 2009-02-18 20:05 newkey.pem ← Clave privada del certificado
-rw-r--r-- 1 root root 781 2009-02-18 20:05 newreq.pem ← Petición de certificado para firmar

Ahora se puede mover el certificado y su clave a /etc/ssl/:
mv newkey.pem /etc/ssl/private/delfin.key
mv newcert.pem /etc/ssl/certs/delfin.crt

y cambiar la configuración del sitio para que lo use en /etc/apache2/sites-available/default-ssl

SSLCertificateFile /etc/ssl/certs/delfin.crt
SSLCertificateKeyFile /etc/ssl/private/delfin.key

Tras lo que se deberá que reiniciar el servidor web.

El fichero demoCA/cacert.pem es el certificado raíz de la nueva autoridad certificadora. Es un certificado X.509 con formato PEM. Ese fichero debe ser copiado a los sistemas que vayan a acceder a nuestro sistema e importar el certificado en sus navegadores. Es muy importante que a la hora de importar el certificado, se le indique al navegador que el certificado sirve para identificar servidores web.

3.3.3 Configuración y aseguramiento de SSH

Secure Sheell (SSH), es un conjunto de estándares y protocolo de red que permite establecer una comunicación a través de un canal seguro entre un cliente local y un servidor remoto. Utiliza una clave pública cifrada para autenticar el servidor remoto y, opcionalmente, permitir al servidor remoto autenticar al usuario. SSH provee confidencialidad e

integridad en la transferencia de los datos utilizando criptografía y **MAC** (**M**essage **A**uthentication **C**odes, o Códigos de Autenticación de Mensaje). De modo predeterminado, escucha peticiones a través del puerto 22 por TCP.

Es la herramienta de conexión segura más usada en el mundo Linux, no hay nada como ssh para conectarse a servidores remotos Linux, ya sea desde Internet o dentro de una Lan. Todo el tráfico se encripta de punto a punto haciendo la conexión sumamente segura. Pero aún así siempre hay riesgos en el salvaje Internet, hackers black hat, script kiddies, crackers, mafias cibernéticas, etc. que en cuanto detecten un servidor ssh trataron de atacarlo por todos los medios posibles. Además, dentro de una LAN relativamente grande también se corren riesgos como el famoso tipo de ataque "man in the middle".

Tipo de ataque man-in-the-middle
Suponiendo que la IP de nuestro servidor es 192.168.2.3. Se realiza una conexión vía SSH desde otro equipo, la primera vez tarda un poco más porque tiene que escribir en el archivo */home/[nuestro_usuario]/.ssh/known_host* una entrada para esa IP.

Las subsiguientes veces que se realice la conexión compara la clave gpg de la IP con la que tiene guardada en el archivo *known_host*, en caso de variar supone que el ordenador es distinto -por ejemplo, porque el ordenador servidor ssh ha sido formateado o porque se ha apagado y sustituido por otro al que se le ha asignado la misma IP-. Ese es el error **man-in-the-middle** y significa: *"el ordenador cuya IP es esa, no es el mismo que yo conocía. Puede ser que alguien se esté haciendo pasar por él"* y no permite volver a conectar salvo que se borre la entrada de

esa IP en el archivo *known_host*. Tienes más información al respecto en el artículo de la Wikipedia Ataque Man-in-the-middle.

Básicamente, los ataques a ssh están basados en una situación (muy frecuente) de un servidor o demonio sshd mal configurado o que no esté actualizado. Entonces, el objetivo es atacarlo mediante alguna vulnerabilidad descubierta a través de un escaneo de puertos o de ataques de login mediante fuerza bruta.
Por ejemplo, una mala configuración sería permitir que el todopoderoso usuario root tuviera permiso de acceso al servidor ssh, esto será relativamente fácil de descubrir y la siguiente parte es lanzar un ataque de fuerza bruta con el objeto de adivinar la contraseña, esto claro mediante un script automático que realize esta función. La siguiente parte de la mala configuración sería no tener un número máximo de intentos de conexión pudiendo el atacante entonces lanzar hasta cientos de sesiones simultaneas de solicitud de ingreso y en cada una lanzar el script de fuerza bruta. Otro mal elemento de configuración sería el no limitar el número de intentos fallidos por conexión, etc. Y si a todo esto se agrega una contraseña débil de root (por ejemplo menos de 8 caracteres y solo minúsculas) será cuestión de posiblemente solo minutos o unas cuantas horas para lograr el objetivo.

3.3.3.1 Instalación

Se utilizó la configuración de ssh para transmitir datos de forma segura entre servidores.
Para instalar el paquete puede hacerse directamente de los repositorios de Debian o puede descargarlo del sitio web oficial de OpenSSH en http://www.openssh.com. Aquí encontrará la versión más moderna de OpenSSh, tanto en formato de código fuente como en formato RPM.

También encontrará OpenSSL, necesario para OpenSSH. Si necesita zlib, se encuentra disponible en http://www.zlib.net y si requiere compilarlo se asume que el sistema debe tener la librería gcc.
/etc/ssh/sshd_no_debe_estar_ejecutandose; no debe estar presente si desea ejecutar el servidor OpenSSH.
Descargar el .tar.gz para su distribución
Descomprimir: tar -xzvf openssh-xxxx.tar.gz
cd openssh-xxxx
./configure –sysconfdir=/etc/ssh
make
make install

O como en nuestro caso en debian se descarga de los repositorios
aptitude install ssh

Puede reiniciar el servicio cuando lo requiera ejecutando:
/etc/init.d/ssh stop
/etc/init.d/ssh start

3.3.3.2 Comandos utilizados en SSH

Para conectarse por acceso remoto

ssh host.remoto

Le solicitará una contraseña (ssh asume que desea utilizar el mismo nombre de usuario activo en el sistema remoto) si no hay error, ya estará logueado remotamente al host que deseaba, sólo que más seguro que Telnet.

Si quiere conectarse con otro usuario en el sistema remoto, necesita añadir delante del nombre de host como si fuese una dirección de correo electrónico. Por ejemplo, está en su máquina local con el usuario pepe y desea conectarse al host remoto con el usuario carlos, puede ejecutar el comando:

ssh carlos@host.remoto

Parece que no hay diferencia con la conexión que se hacía con Telnet, antes de SSH se solicita una contraseña o palabra de paso, ahora ya está negociada una sesión encriptada con el servidor remoto.
Cuando se escribe un usuario y contraseña, lo enviará cifrados por la red mediante la sesión encriptada, no en texto inteligible como con Telnet. Por lo tanto, todos los datos posteriores también serán cifrados.

Utilizando sftp

Con la versión 2.0 de SSH, Tatu Ylönen introdujo una característica: sftp.
El soporte del lado del servidor para sftp está dentro de sshd. Si el host remoto no lo soporta tendrá que utilizar scp.
Utilizar el cliente sftp es tan simple como usar ssh.
Un ejemplo:Conectado con usuario pepe en la máquina cliente hostlocal. Se ejecuta:
[pepe@host.local]#sftp carlos@host.remoto

sftp> get archivo1

sftp> put archivo 2

sftp> quit

Utilizando scp

El comando scp, es equivalente a la antigua utilidad rcp, que se usa

para copiar un archivo o directorio de un host otro. De hecho scp se basa en el código fuente de rcp.

La sintaxis básica del comando scp es:

scp [opciones] cadena-archivo-origen usuarioenhostremoto@host.remoto:/directoriodondevoyguardar/

Configurar OpenSSH no es complicado. Para controlar el comportamiento del cliente y servidor de SSH, sólo hay que editar dos archivos: ssh_config y sshd_config, respectivamente. Dependiendo del paquete que haya instalado o compilación que haya realizado, estos archivos pueden localizarse en /etc/ o algún otro lugar que haya especificado en ./configure –sysconfdir. El archivo de configuración global para las sesiones iniciadas desde el host local es ssh_config.

3.3.3.3 Aseguramiento y personalización de ssh

Se edita el archivo de configuración de ssh ubicado en /etc/ssh/sshd_config (puede variar según la distribución de GNU/Linux.

Algunos de los más importantes para tener en cuenta son:

Port 432 (o el que asigne que sea menor a 1024)

Protocolo 2

LoginGraceTime 30

PermitRootLogin no

MaxAuthTries 2

MaxStartups 3

AllowUsers usuarioautorizado1 o también pude especificar AllowUsers usuarioautorizado1@ip

Port: Por default el demonio ssh funciona en el puerto 22, y precisamente muchos scripts de ataques están dirigidos a este puerto, el cambiar de puerto no garantiza que el servicio ya no será localizable,

de hecho con herramientas como nmap o amap es sumamente fácil descubrir que un servicio ssh esta a la escucha en otro puerto distinto al 22, pero al menos no será localizable por varios scripts que de manera automática escanean redes y en cuanto a ssh se enfocan solo al puerto 22. A este tipo de técnicas se le conoce como Seguridad por Oscuridad. Puede elegirse cualquier otro puerto entre el 1025 y 65535.

Protocol 2: Hay dos versiones de ssh en cuanto a su protocolo de comunicación, la versión 1 y la versión 2. La 1 está en desuso pero todavía se incluye por compatibilidad, tiene varias vulnerabilidades conocidas y su uso no es ya recomendable. Un error frecuente es dejar al demoinio ssh que permita el uso de las dos versiones (Protocol 2,1). Para evitar el uso del protocolo 1 y sus posibles ataques a este, basta con indicar en esta línea que solo admita comunicaciones de ssh basadas en el protocolo 2.

LoginGraceTime 30: El número indica la cantidad de segundos en que la pantalla de login estará disponible para que el usuario capture su nombre de usuario y contraseña, si no lo hace el login se cerrará, evitando así dejar por tiempo indeterminado pantallas de login sin que nadie las use, o peor aún, que alguien esté intentando mediante un script varias veces el adivinar un usuario y contraseña. Aquí conviene identificar en nuestros usuarios el tiempo promedio que tardan en ingresar su usuario y contraseña y darles unos cuantos segundos más de margen por los usuarios lentos para que ingresen sus credenciales. Si se es el único usuario del sistema se considera que con 20 o 30 segundos es más que suficiente.

PermitRootLogin: Establece si se va a permitir el acceso directo del usuario root al servidor SSH. Si se va a permitir el acceso hacia el

servidor desde redes públicas, resultará prudente utilizar este parámetro con el valor **no**.

MaxAuthTries 2: El número indica la cantidad de veces que se puede equivocar en ingresar el usuario y/o contraseña, en este caso después de dos intentos, se perderá o cerrará la conexión. Claro, es totalmente posible volver a intentarlo, pero como son dos intentos por vez, se evita de esta forma ataques basados en la persistencia de la conexión, como se perderá al tercer intento de conexión, el ataque cesará.

MaxStartups 3: El número indica la cantidad de pantallas de login, o cantidad de conexiones simultaneas de login que permitirá el sshd por ip que intente conectarse. Hay ataques muy efectivos que dividen el ataque en decenas y puede ser que en cientos (si el sistema atacado lo permite) de conexiones de login. Es decir, el ataque divide en una gran cantidad de logins los intentos por ingresar, aumentando sus posibilidades de más rápidamente adivinar al usuario y contraseña. Con esta directiva se limita a tan solo 3 pantallas de login. Que quede claro, una vez logueados en el sistema, es posible tener más de 3 terminales de ssh, se refiere exclusivamente a pantallas de login.

AllowUsers: En sistemas donde se tiene varios usuarios, quizás existan varios que solo pueden acceder desde la LAN por ejemplo, o quizás solo desde ciertos equipos. O incluso que solo desde su PC puedan trabajar en Linux por lo que no hay razón para que se conecten remotamente via ssh. Con esta directiva se puede indicar los usuarios que pueden ingresar via ssh. Si solo se indica al usuario:

AllowUsers Usuario 1

Usuario 1 podrá ingresar desde cualquier PC en cualquier lugar, no se está validando el host.

Si se quiere más seguridad, es posible indicar también el host mediante el símbolo @

AllowUsers Usuario1@192.168.0.25 (solo desde la IP indicada)
AllowUsers Usuario1@192.168.0.* (Toda la red indicada)
AllowUsers Usuario1@*.uis.com Usuario2@ventas.pato.com Usuario3@192.168.0.23
(Usuario1 desde cualquier equipo del dominio indicado, Usuario2 solo desde el equipo indicado y Usuario3 desde esa Ip)

Hay otros parámetros que puede configurar como:

Parámetro ListenAddress.
Por defecto, el servicio de SSH responderá peticiones a través de todas las interfaces del sistema. En algunos casos es posible que no se desee esto y se prefiera limitar el acceso sólo a través de una interfaz a la que sólo se pueda acceder desde la red local. Para tal fin puede establecerse lo siguiente, considerando que el servidor a configurar tiene la IP **192.168.34.12**:

ListenAddress 192.168.34.12

Parámetro X11Forwarding.
Establece si se permite o no la ejecución remota de aplicaciones gráficas. Si se va a acceder hacia el servidor desde red local, este parámetro puede quedarse con el valor **yes**. Si se va a permitir el acceso hacia el servidor desde redes públicas, resultará prudente utilizar este parámetro con el valor **no**.

X11Forwarding yes

Aplicando los cambios.

El servicio de **SSH** puede iniciar, detenerse o reiniciar a través de un guión similar a los del resto del sistema. De tal modo, podrá iniciar, detenerse o reiniciar a través del mandato **service** y añadirse al arranque del sistema en un nivel o niveles de corrida en particular con el mandato **chkconfig**.

Para ejecutar por primera vez el servicio, utilice:

service sshd start

Para hacer que los cambios hechos a la configuración surtan efecto, utilice:

service sshd restart

Para detener el servicio, utilice:

service sshd stop

De forma predeterminada, el servicio **SSH** está incluido en todos los niveles de corrida con servicio de red. Para desactivar el servicio Sshd de los niveles de corrida 2, 3, 4 y 5, ejecute:

chkconfig --level 2345 sshd off

3.3.3.4 Autentificación transparente por clave pública/privada con OpenSSH

El protocolo SSH está preparado para que poder autentificarse de

forma transparente (sin introducir una contraseña manualmente). Para ello, lo que se hace es generar una pareja de claves pública/privada (se puede tener varias, una por protocolo) en el cliente de SSH y a continuación, al servidor de SSH se especifica una serie de claves públicas de clientes que, si acceden con la clave privada asociada, pueden entrar sin especificar una contraseña.

Uno puede evitar recordar la contraseña para cada sistema remoto usando RSAAuthentication (protocolo SSH1) o PubkeyAuthentication (protocolo SSH2).

En el /etc/ssh/sshd_config del sistema remoto debe figurar "RSAAuthentication yes" o "PubkeyAuthentication yes".

NAME

ssh-keygen - authentication key generation, management and conversion

SYNOPSIS

ssh-keygen [**-q**] [**-b** *bits*] **-t** *type* [**-N** *new_passphrase*] [**-C** *comment*] [**-f** *output_keyfile*]
ssh-keygen -p [-P *old_passphrase*] [-N *new_passphrase*] [-f *keyfile*]
ssh-keygen -i [-m *key_format*] [-f *input_keyfile*]
ssh-keygen -e [-m *key_format*] [-f *input_keyfile*]
ssh-keygen -y [-f *input_keyfile*]
ssh-keygen -c [-P *passphrase*] [-C *comment*] [-f *keyfile*]
ssh-keygen -l [-f *input_keyfile*]
ssh-keygen -B [-f *input_keyfile*]
ssh-keygen -D *pkcs11*
ssh-keygen -F *hostname* [-f *known_hosts_file*] [-l]

ssh-keygen -H [-f *known_hosts_file*]
ssh-keygen -R *hostname* [-f *known_hosts_file*]
ssh-keygen -r *hostname* [-f *input_keyfile*] [-g]
ssh-keygen -G *output_file* [-v] [-b *bits*] [-M *memory*] [-S *start_point*]
ssh-keygen -T *output_file* -f *input_file* [-v] [-a *num_trials*]
[**-W** *generator*]
ssh-keygen -s *ca_key* -I *certificate_identity* [-h] [-n *principals*]
[**-O** *option*] [**-V** *validity_interval*] [**-z** *serial_number*] *file ...*
ssh-keygen -L [-f *input_keyfile*]

Se puede generar las claves de autentificación en forma local y copiar la clave pública en el sistema remoto:

```
$ ssh-keygen          # RSAAuthentication: clave RSA1 para SSH1
$ cat .ssh/identity.pub | ssh user1@remote \
      "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t rsa   # PubkeyAuthentication: clave RSA para SSH2
$ cat .ssh/id_rsa.pub | ssh user1@remote \
      "cat - >>.ssh/authorized_keys"
...
$ ssh-keygen -t dsa   # PubkeyAuthentication: clave DSA para SSH2
$ cat .ssh/id_dsa.pub | ssh user1@remote \
      "cat - >>.ssh/authorized_keys"
```

Observe que SSH2 realiza una autenticación del tipo HostbasedAuthentication. Para que esto funcione, debe asignar a la entrada HostbasedAuthentication el valor yes tanto en el /etc/ssh/sshd_config de la máquina servidor como en el

/etc/ssh/ssh_config o el $HOME/.ssh/config de la máquina cliente.
Uno puede cambiar la frase de contraseña posteriormente haciendo "ssh-keygen -p". Asegúrese de verificar la configuración probando la conexión. En caso de problemas, haga "ssh -v".

Puede añadir opciones a las entradas en authorized_keys para limitar los hosts y ejecutar comandos específicos.

En la máquina desde donde se quiera conectar sin poner password debe crear un juego de llaves (publica y privada), para hacer esto:
$ ssh-keygen -t rsa -C usuarioconqueseconecta@host.remoto

La opción -C le permite colocar un comentario para identificar la llave.

Tu llave pública se ha creado en /home/username/.ssh/id_rsa.pub y tu llave privada en /home/username/.ssh/id_rsa

Si no se especifican opciones, el comando crea una clave RSA de 2048 bits, pero se puede cambiar este comportamiento. Por ejemplo, para clave DSA de 1024 bits, sería:

$ ssh-keygen -t dsa -b 1024

Hay que tener en cuenta que si le especifica una *passphrase*, se tiene que introducirla manualmente cada vez que se quiera usar la clave privada, por lo que la autentificación no será transparente. Por ello, aunque no sea lo más seguro, para nuestro propósito es necesario dejarla en blanco.

Y ahora se debe copiar tu llave publica id_dsa.pub o del id_rsa.pub en

el fichero authorized_keys del directorio $HOME/.ssh/ del usuario del sistema con servidor SSH al que se desea conectar. Este directorio (.ssh) y el archivo authorized_keys deben tener permisos de lectura, escritura y acceso, únicamente por el dueño, de otra forma sshd puede rechazar la conexión argumentando *bad ownership or modes*

Puede pasarla id_dsa.pub o id_r sa:pub con los siguientes comandos:
scp ./id_dsa.pub hostremoto:/home/usuarioconqueseconectara

En el servidor SSH, edite /home/usuarioconquesconectara/.ssh/authorized_keys2/ ya sea que copie la llave publica aquí o puede concatenarla al archivo con el comando:

cd home/usuarioconqueseconectara/

cat id_dsa.pub >> .ssh/autorized_keys2

Se reinicia el servicio ssh
Autorized_keys2 si conecta con ssh -2 o sino concatenelo a autorized_keys.
Por ejemplo:

ssh -2 host.remoto.com

Cuando se utiliza el parámetro -2; esto indica a SSH que sólo intente usar la versión de protocolo SSH v2. Por defecto utiliza v1, pero este sólo soporta claves RSA y se ha copiado una clave DSA. Hay que tener en cuenta que la clave está referida por el nombre de su archivo local: esto es para recordar que cuando se utiliza autentificación con RSA o DSA, la palabra de paso que se teclea sólo se utiliza para desbloquear

nuestra clave privada almacenada localmente y no viaja de ninguna forma por la red.
Con el comando anteriormente ejecutado se asumió que tiene el host remoto tiene el mismo usuario que estoy utilizando en mi sistema local y que el servidor remoto reconoce el protocolo SSH v2.

Si como administrador, se quiere como mecanismo de seguridad decide quitar la autentificación usuario/contraseña y mantener solo una política estricto de acceso RSA/DSA únicamente, cambie el parámetro PasswordAuthentication a No en el archivo de configuración sshd_config de cada host remoto que esté ejecutando sshd.

Copia de seguridad
Si vas a migrar la configuración de tu usuario a otra máquina tienes que conservar el directorio *$HOME/.ssh*. Ahí es donde se guarda los ficheros de la clave pública y privada generadas en el punto anterior:

$ tar czf ssh.tgz $HOME/.ssh

3.3.4 Enjaulado o aseguramiento de usuarios del sistema.

Es una gran responsabilidad para el administrador del servidor y riesgo para la seguridad del sistemas habilitar usuarios del sistemas a personas externas a la administración; por ello hay que asegurarnos que el usuario quede restringido en un directorio, de manera que no pueda salir de ese PATH y curiosear archivos que no les corresponde ver o ejecutar código que pueda afectar al sistema anfitrión. Para ello se enjaula a los usurios, en otras palabras el punto esencial del método es cambiar el directorio raíz del sistema de ficheros para ese usuario.

Esto se puede hacer con el comando chroot, por ejemplo:

chroot /home/usuario

Al realizar esto, la raíz [/] para este usuario sería lo que hay en /home/usuario.

Parece sencillo, pero no lo es porque ahora, esa raíz debe tener todo lo necesario para comportarse como un Sistema Operativo funcional, es decir, que contenga las librerías y programas de sistema necesarios para que el entorno funcione. Afortunadamente la comunidad Open Source existe y no hay que reinventar la rueda, lo ingenioso es aprovechar los recursos disponibles y adaptarlos a nuestras necesidades.

En el servidor de SISTEMAS se requieren crear usuarios que tengan el rol de desarrolladores de los sitios web para ello necesitan servidor sftp o scp (sin cuenta Shell) para subir sus archivos y hacer modificaciones en ellos , para lograr lo anterior se utilizó el paquete scponly, que recopila todo todas las librerías y un programa que restringe el uso de comandos a los estrictos usados por scp y sftp. Este paquete viene en los repositorios de debían o se puede descargar una versión más actualizada en http://www.sublimation.org/scponly.

Nombre del paquete: *scponly-4.8*

Se puede descargar de: http://www.sublimation.org/scponly

Instalación:

Descargar el paquete

Descomprimir el paquete scponly-4.8.tgz

Como usario root pase el paquete descargado al directorio de root/

Dentro del directorio descomprimido los scripts deben tener permisos de lectura y ejecución en especial el script setup_chroot.sh que es el que se modifica según los intereses y luego se ejecuta

Procedimiento:

#./configure –enable-chrooted-binary

#make

#make install

Esto generará un ejecutable que se instalará en /usr/sbin/scponlyc .

En este caso no se utiliza más opciones en la compilación, pero se puede personalizar como se quiera, algunas posibilidades son:

--disable-wildcards

--disable-sftp

--disable-winscp-compat

Se ejecuta el script jail, que nos ahorra la tarea de incluir todas las librerías que harán falta para el entorno chroot, este script tal como se descarga de internet tiene algunos fallos, no crea algunos directorios y falla, por ello se realizaron las modificaciones necesarias en el script setup_chroot.sh para que creara directamente la raíz de los usuarios en el directorio donde están todos los sitios web .

Por ejemplo, para hacer un usuario enjaulado en el directorio destinado para los sitios web llamado usuario, con su directorio raíz en /sitiosweb/sitio y dentro con un directorio llamado subdirectorio con permisos para escribir cuando se conecte por sftp o winscp con el password xxxxx, se realizaría de la siguiente manera.

Ejecutar el comando make jail dentro del directorio donde están los archivos de scponly.

#make jail

El script le pide el nombre del usuario, el home del usuario (para

nuestro caso, el directorio que se tiene destinado para los sitios web), el subdirectorio en el cual el usuario tendrá acceso por winscp o sftp para escribir y por último el password para el usuario. En la Shell verá algo así:

Username to install [scponly]usuario

home directory you wish to set for this user [/home/usuario/]/sitiosweb/sitio

name of the writeable subdirectory [incoming]subdirectorio

please set the password for usuario:

Introduzca la nueva contraseña de UNIX:xxxxx

Passwd:contraseña actualizada correctamente

if you experience a warning with winscp regarding groups, please install the provided hacked out fake groups program into your chroot, like so:

cp groups /sitiosweb/sitio/bin/groups

Se puede mencionar que el usuario se creó correctamente, donde su home asignado es /sitiosweb/usuario/ y la shell es scponlyc y no bash.

#less /etc/passwd

Se puede visualizar algo parecido a:

usuario:x:1003:1003::/sitiosweb/usuario/subdirectorio:/usr/local/sbin/scponlyc

Ver que el scponlyc figura como Shell , sino agregarla con:

#which scponly >> /etc/shells

Fije SUID_Bit para scponlyc

#chmod u+s /usr/sbin/scponlyc

Cree el directorio dev en el directorio raíz del usuario, para nuestro ejemplo:

#cd /sitiosweb/usuario/

mkdir dev

cd dev

mknod –m 666 null c 1 3

Se puede probar por interfaz gráfica desde un equipo con Windows con winscp u otro software que permita conexión al servidor sftp o por consola desde cualquier sistema operativo que el usuario se puede loguear con sftp o scp y este no podrá salir de su home o directorio raíz y solo puede escribir en el subdirectorio asignado.

4. APLICACION WEB PARA LA INTERACCIÓN ENTRE ADMINISTRADOR Y USUARIOS DE LOS SERVIDORES

4.1 Introducción

Uno de los pilares para la correcta prestación de los servicios en los servidores, es la efectiva y eficiente comunicación entre el administrador y los usuarios. Esta se venía realizando mediante correos electrónicos o personalmente, lo cual dificultaba la realización de las solicitudes y su correcto seguimiento. Con el objetivo de mejorar los tiempos de respuesta, la efectividad en el desarrollo de las actividades y poder llevar un seguimiento de las solicitudes, se desarrollo la aplicación web.

4.2 Modelo de Desarrollo

Para obtener un producto de calidad que cumpla satisfactoriamente los objetivos propuestos, es fundamental seguir una metodología de acuerdo a las características del proyecto.

El análisis del ciclo de vida constituye el orden del proceso del desarrollo del producto software que se va a construir, proporcionando un entendimiento y compresión del mismo, esta fase es fundamental para obtener el éxito del producto final.

De acuerdo a las características y los requerimientos de la aplicación, se realizo un análisis de los ciclos de vida más frecuentemente utilizados en el marco de las metodologías de ingeniería de software, y teniendo en cuenta aspectos de desarrollo y viabilidad, se eligió como modelo a seguir el espiral; dado los cambios que podría tener el modelo de administración de los servidores, este modelo nos permitiría manejar

el menor índice de riesgo.

Fig. 2 Modelo de Desarrollo en Espiral

4.3 Descripción de la aplicación

La aplicación contiene una presentación inicial pública y un sistema de información con acceso restringido.

4.3.1 Presentación y Estructura Inicial

Fig. 3 Interfaz de entrada a la aplicación web

La presentación inicial de la aplicación fue diseñada pensando en crear un espacio agradable y ergonómico de consulta, manteniendo la identidad con la escuela de Ingeniería de Sistemas e Informática. El espacio de trabajo está dividido en tres secciones:

- Sección de Menú: Clasifica la información a la cual se desea acceder. Se encuentra dividido en cinco bases:

Nosotros: Destinado a la información básica de la escuela de Ingeniería de Sistemas, como misión, visión, historia, ubicación, etc.

Estudiantes: Información relevante para estudiantes de pregrado, maestría y si llegase a presentarse especialización.

Docentes: Hoja de vida de docentes tiempo completo y cátedra, estudios, área de investigación, artículos, congresos, etc.

Investigación: Información significativa grupos de investigación y desarrollo presentes en la escuela.

Industria: Relación de la escuela con la industria, trabajos desarrollados o en desarrollo por egresados, experiencias laborales.

- Sección de Eventos: Actividades relevantes de interés general próximas a llevarse a cabo.
- Sección de Información:. Resumen de contenidos autorizados por el administrador, publicados por estudiantes o docentes, cuyo interés se considera general y relevante con los servicios que se prestan.

4.3.2 Sistema de Información

Para ingresar en la aplicación se debe seleccionar la opción de Estudiantes o Docentes, y en el formulario de identificación ingresar el usuario y contraseña.

4.3.2.1 Perfiles de Usuario

Usuario General: Este usuario puede ver en la presentación inicial los eventos, contenidos y secciones.

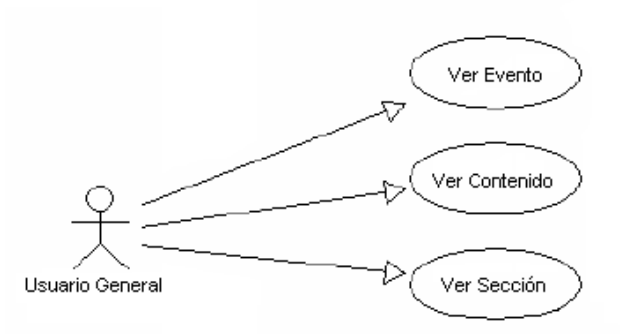


Fig. 4 Diagrama de Caso de Usuario General

Usuario Registrado: Este usuario puede ingresar en el sistema de información, administrar su cuenta, crear solicitudes de servicio y crear solicitudes de soportes.

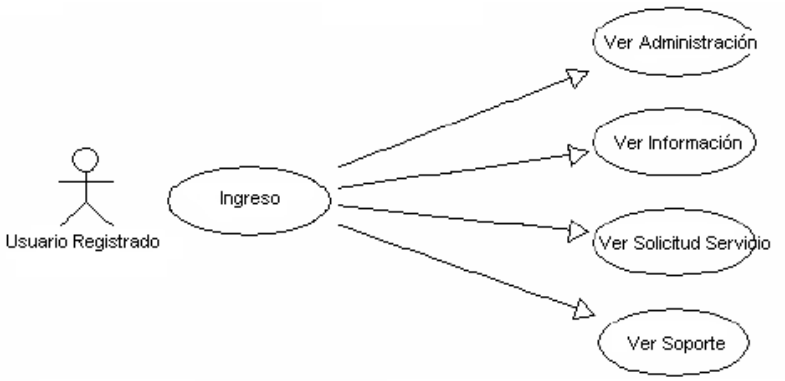


Fig. 5 Diagrama de Caso de Uso de Usuario Registrado

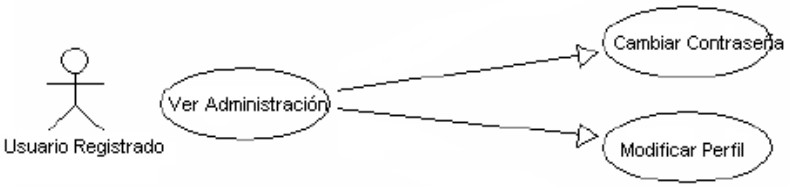


Fig. 6 Diagrama de Caso de Uso de Usuario Registrado (Ver administración)

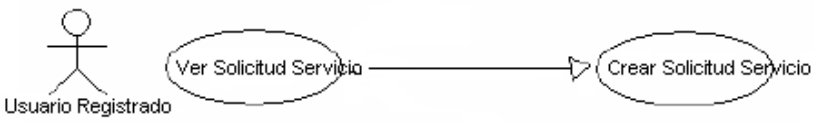


Fig. 7 Diagrama de Caso de Uso de Usuario Registrado (Ver solicitud Servicio)

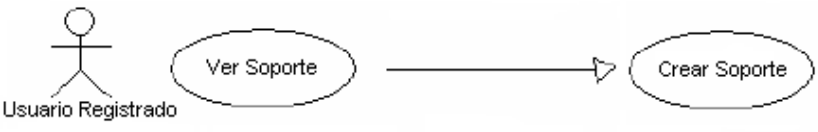


Fig. 8 Diagrama de Caso de Uso de Usuario Registrado (Ver Crear Soporte)

Usuario Administrador: Este usuario puede ingresar en el sistema de

información, administrar la configuración de la aplicación, usuarios, grupos y revisar la auditoria. Dar respuesta a solicitudes de servicio y solicitudes de soporte.

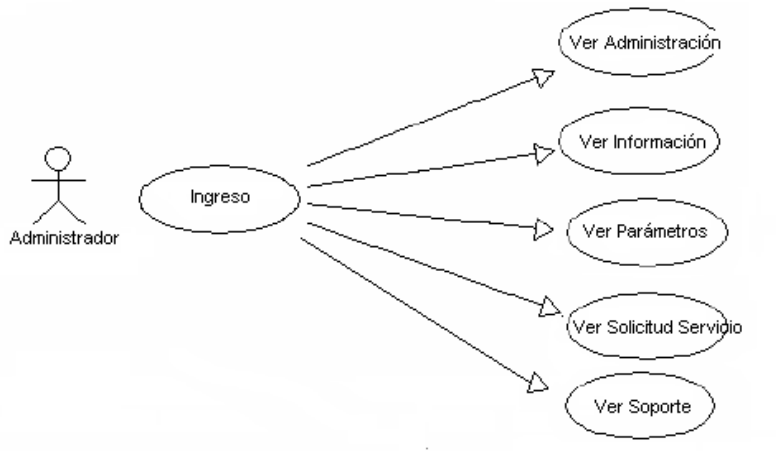


Fig. 9 Diagrama de Caso de Uso de Usuario Administrador

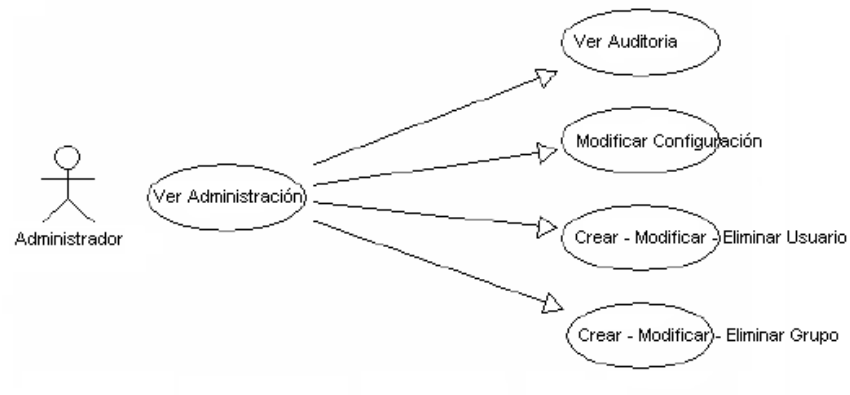


Fig. 10 Diagrama de Caso de Uso de Usuario Administrador (Ver Administración)

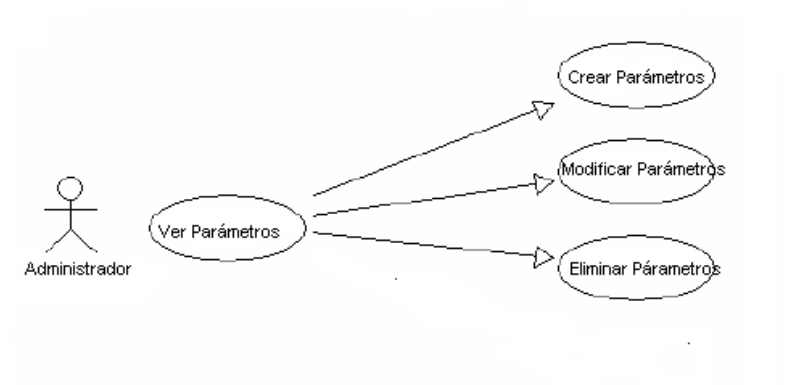


Fig. 11 Diagrama de Caso de Uso de Usuario Administrador (Ver Parámetros)

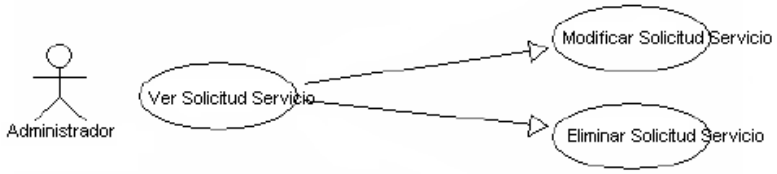


Fig. 12 Diagrama de Caso de Uso de Usuario Administrador (Ver Solicitud Servicio)

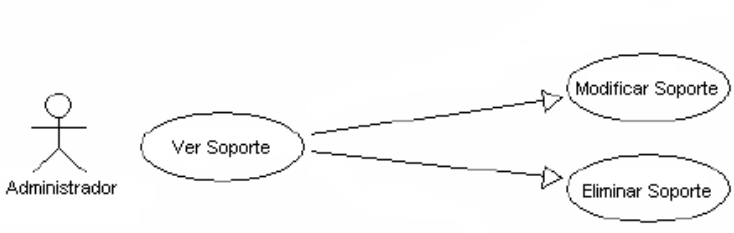


Fig. 13 Diagrama de Caso de Uso de Usuario Administrador (Ver Soporte)

4.3.2.2 Módulos

Módulo de Administración

En este módulo el usuario registrado puede administrar su cuenta, modificando su contraseña y la información de su perfil. El administrador puede gestionar la aplicación, modificando su configuración, supervisando la auditoria y creando usuarios y grupos.

Módulo de Parámetros

A este módulo solo tiene acceso el administrador y permite la manipulación de los datos de donde los formularios se alimentan, para su respectiva parametrización. Estos parámetros se dividen en generales como: País, Departamento, Municipio, Tipo de Identificación y la Nomenclatura de Dirección, y específicos como: Estado de Solicitud, Servicios, Tipo de Servicio y Tipo de Solicitud.

Módulo de Solicitud de Servicio

El usuario para solicitar el servicio de alojamiento, especifica el espacio en disco duro en MB que desea, el lenguaje de desarrollo a utilizar, el usuario, nombre y motor de base de datos al igual que el usuario FTP. El sistema halla y almacena la fecha en que se realizo la solicitud y la fecha en la cual, el administrador cambio el estado de la petición. Este estado es seleccionado, según el protocolo para la prestación de servicios que se halla establecido.

Módulo de Soporte

El usuario para requerir soporte, selecciona el tipo de solicitud, el tipo de servicio al cual va destinado la petición y redacta lo que desea solicitar. Este proceso permite clasificar la prioridad que debe ser asignada a cada petición y el respetivo procedimiento que debe realizarse. El sistema halla y almacena la fecha en que se realizo la

solicitud y la fecha en la cual, el administrador le dio respuesta y cambio el estado de la petición. Este estado es seleccionado, según el protocolo para la prestación de soporte que se halla establecido.

5. PROTOTIPO DE CLÚSTER DE ALTA DISPONIBILIDAD

Esta implementación se llevo a cabo con el sistema operativo Debian 5, con la y con la herramienta Heartbeat para ofrecer alta disponibilidad.

SERVIDOR DE ALTA DISPONIBILIDAD	**Máquina 1 Principal**	**Máquina 2 De Respaldo**
Modelo	Dell Precision T3400	Dell Precision T3400
Sistema Operativo	GNU/LINUX	GNU/LINUX Debian 5 (Lenny) Kernel 2.6
Distribución	Debian 5 (Lenny) Kernel 2.6	Debian 5 (Lenny) Kernel 2.6
Procesador	Intel Core 2 Quad 2.5 Ghz	Intel Core 2 Quad 2.5 Ghz
Memoria	4 GB	4 GB
Disco Duro	250 GB	250 GB
Número de tarjetas de red	1 Ethernet base 10/100 1 Broadcom Netxtreme Gigabit	1 Ethernet base 10/100 1 Broadcom Netxtreme Gigabit

Tabla 1. Equipos de Cómputo utilizados

5.1 HEARTBEAT

HEARTBEAT es un paquete de software creado por LINUX-HA, que

permite ofrecer alta disponibilidad al pasar los servicios de un nodo que deja de funcionar a otro nodo disponible, basandose en la señal que recibe al ejecutar un ping entre tarjetas de red o por un cable conectado entre puerto serial, como sistema de monitoreo para detectar la dispionibilidad de los nodos.
Utiliza una técnica llamada STONITH (que son las sigles de “Shoot The Other Node In The Head”, disparele al otro nodo en la cabeza) para asegurarse que el nodo que se detecta fuera de servicio no interfiera con el funcionamiento del cluster (en nuestro caso el servidor de alta disponibilidad).
En resumen la técnica STONITH consiste en eque le servidor secundario o de respaldo identifica que el servidor principal no funciona y, este hace un DDOS al principal para asegurarse que ha sido un falso positivo y tomaría el nodo secundario el control.

Se puede descargar el paquete de heartbeat de los repositorios oficiales de Debian o de http://linux-ha.org, y los paquetes de los servicios que se quiere ofrecer como apache y mysql.

Se puede instalar con el manejador de paquetes de Debian, pero antes de instalar heartbeat hay que tener en cuenta que la fecha y hora de los servidores estén sincronizadas, esto se consigue instalando el paquete ntp y descargar el paquete adecuado si se tiene arquitectura de 64 bits.

5.2 Instalación
Para la implementación se instaló:
aptitude install ntp ssh

Se deben tener dos interfaces de red, el sistema las identificó como

eth0 y eth3.

Se configura una interfaz para que se las máquinas puedan navegar y se comuniquen por la subred puede ser la interfaz eth0, y la eth3 se deja exclusivamente dedicada para el monitoreo por parte de heartbeat entre los nodos.

Se agregan los nodos del cluster en /etc/hosts

```
x.x.x.x    nodo1.dominio.com    nodo1
x.x.x.x    nodo2.dominio.com    nodo2
```

Se instala el paquete de acuerdo a nuestra arquitectura del sistema operativo, en este caso de 64 bits.

```
aptitue  install heartbeat-2
```

Se copian los archivos de configuración del paquete heartbeat al dirrectorio por defecto que crea heartbeat en /etc/ha.d

```
cp /usr/share/doc/heartbeat/ha.cf /etc/ha.d/
cp /usr/share/doc/heartbeat/haresources /etc/ha.d/
cp /usr/share/doc/heartbeat/authkeys /etc/ha.d/
```

Al listarlos quedan en el directorio de ha.d

```
nodob1:/etc/ha.d# ls -l

total 56

-rw-r----- 1 root adm   645 jul 30 02:19 authkeys

drwxr-x--- 2 root adm  4096 ago  8  2009 conf
```

drwxr-x--- 2 root adm 4096 ago 8 2009 cts

-rw-r----- 1 root adm 10843 jul 30 02:18 ha.cf

-rwxr-x--- 1 root adm   745 ago 8 2009 harc

-rwxr-x---1 root.adm jul 30 02:19 haresources

Se configura los tres archivos anteriores:

Primero se edita ha.cf y se coloca lo parámetros de acuerdo al comportamiento que se quiera programar el cluster de alta disponibilidad.
nodo1:/etc/ha.d# vim ha.cf

Y se modifican los siguientes parámetros:
logfile /var/log/ha-log

logfacility   local0

keepalive 10
deadtime 30

warntime 10

initdead 120
bcast   eth0
ucast eth3 10.10.0.2

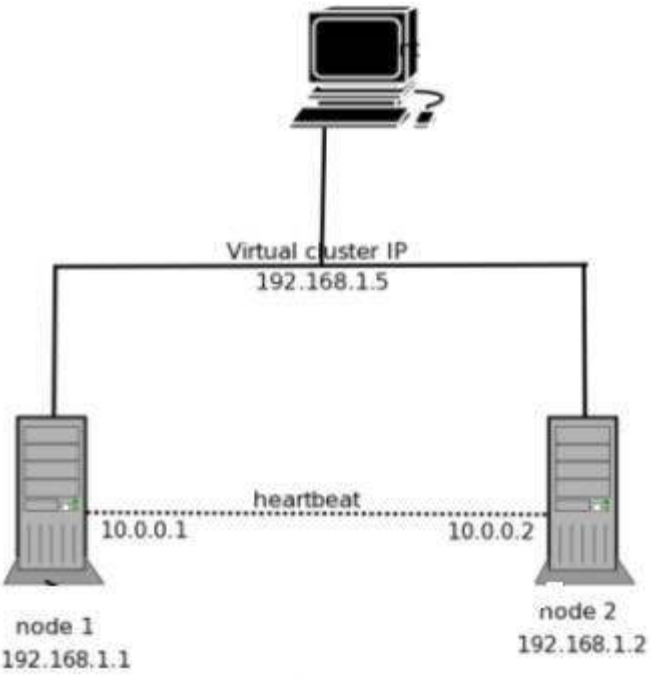
auto_failback on
node   nodo1

node   nodo2

#ping 10.10.10.254

La configuración anterior también se realiza para el nodo secundario con la diferencia de ucast que se define la dirección IP dedicada a la comunicación entre nodos.

Se instalan los servicios que se requieran en nuestro caso Apache y Mysql.
atitude install apache2
aptitud install mysql.-serrver-5.0 mysql-client-



5.0

Fig. 14 Modelo Prototipo de Servidor de Alta Disponibilidad

Se edita /etc/ha.d/haresources donde se define el nodo principal, el IP virtual y los recursos que voy a reiniciar.

vim haresources

Y ase agrega la siguiente línea:

nodo1.midominio.com 192.168.X.X apache2

Este archivo exactamente igual se copia en ambos nodos. Los script de reinicio de los servicos se encuentran en directorio /etc/init.d/. Es importante que el servicio de apache no se reinicie automaticamente al reiniciar las máquinas, debe ser reinciado, iniciado y detenido por heartbeat. Se debe desactivar el reinicio automatico. Pude ejecutar:

chkconfig -d apache2

Se puede ver en la salida por pantalla como quita de cada uno de los runlevel el inicio automático del servicio de Apache.

nodob1:/etc/ha.d# chkconfig -d apache2

insserv: warning: current start runlevel(s) (6) of script `reboot' overwrites defaults (empty).

insserv: warning: current start runlevel(s) (0 6) of script `sendsigs' overwrites defaults (empty).

insserv: warning: current start runlevel(s) (0) of script `halt' overwrites defaults (empty).

insserv: warning: current start runlevel(s) (0 6) of script `umountfs' overwrites defaults (empty).

insserv: warning: current start runlevel(s) (0 6) of script `umountroot' overwrites defaults (empty).

insserv: warning: current start runlevel(s) (0 6) of script `umountnfs.sh' overwrites defaults (empty).

insserv: warning: current start runlevel(s) (0 6) of script `wpa-ifupdown' overwrites defaults (empty).

apache2 0:off 1:off 2:off 3:off 4:off 5:off 6:off

nodob1:/etc/ha.d# /etc/init.d/heartbeat start

Starting High-Availability services:

2010/07/30_03:32:36 INFO: Resource is stopped

Done.

Se determina en el archivo de configuración de apache el Document Root
Listen x.x.x.x:80
DocumentRoot “/var/www”
<Directory “/var/www”>

Se reinicia apache.

Y ahora el tercer archivo de configuración de heartbeat es /etc/ha.d/authkeys. Este archivo también debe ser el mismo en todos los nodos, y que sea solo de lectura y escritura para el root del sistema. Si los permisos son diferentes heartbeat rechazará el inicio.

vim authkeys

Y se habilita la siguiente línea quitando el numeral al inicio de este parámetro.
auth 1
1 crc

Cambie permisos
chmod 600 /etc/ha.d/authkeys

Finalmente se puede iniciar el heartbeat en ambos nodos:
/etc/init.d/heartbeat start

Ahora puede probar el failover, reiniciando el nodo principal y observar que el nodo de respaldo asume el control manteniendo el servicio.

Se prueba cuando el nodo principal vuelve a estar disponible el nodo 2 libera la ip virtual y el nodo 1 toma de nuevo el control.

6. CONCLUSIONES

- La solución de herramientas libres propuestas en el presente proyecto cumple con las expectativas de permitir la disponibilidad de servidores estables, robustos y seguros, bajo los lineamientos que enmarcan las necesidades de los usuarios.

- La aplicación Web desarrollada es parametrizable, permitiendo un uso versátil de los módulos, ampliando su funcionalidad y con ello logrando una comunicación más dinámica rápida y flexible, adaptable a las necesidades de los usuarios y a la administración de los servidores.

7. RECOMENDACIONES

- Debido al continuo avance y crecimiento de Internet, y como consecuencia la aparición de nuevas herramientas y tecnologías, el servidor de la escuela de Ingeniería de Sistemas e Informática debe permanecer en una constante actualización y crecimiento, sin perder nunca los lineamientos de estabilidad, robustez y seguridad, dirigidos siempre a las necesidades de sus usuarios.

- La universidad debe impartir un conocimiento pluralista y la Escuela de Ingeniería de Sistemas como parte de ella, debe promover en el ejercicio de la enseñanza y la investigación, la diversidad en herramientas, tecnología y hardware, tanto libres, privativas y experimentales para enriquecer los criterios de conocimiento y competitividad de los egresados. Por consiguiente se recomienda la puesta en marcha de un servidor con tecnología privativa.

- Investigar e implementar el prototipo de clúster de alta disponibilidad para ofrecer un servicio más robusto y que se pueda ampliar aún más la comunidad de usuarios, ofreciendo servicio de hosting a los estudiantes de la eisi para que puedan hacer sus pruebas.

8. BIBLIOGRAFIA

8.1 MARCO GENERAL

En este marco se presentan fuentes que orientarán el desarrollo de la investigación, en cuanto a metodologías y lineamientos a seguir en el proceso de investigación.

- GOMEZ, L. Conferencias sobre planeación de proyectos Colombia, 2001.

- ZORRILLA, S, TORRES, M. Guía para elaborar la tesis, México, 1992.

8.2 MARCO TÉCNICO

- AULDS, C. Linux Apache web server administration. Sybex. , 2001.

- BAUER, MICHAEL D. Seguridad en Servidores Linux. Editorial Anaya Multimedia. Madrid, 2005

- EYLER, P. Redes Linux con TCP/IP. Editorial Prentice Hall / Pearson., 2001.

- HARE, C. Firewalls y la Seguridad en Internet. Editorial Prentice Hall/Pearson, 1997.

- KABIR, M. La Biblia del Servidor Apache. Editorial Anaya Multimedia, 1999.

- SÁNCHEZ, S. Unix y Linux: Guía Práctica. Editorial Alfa Omega. México D.F., 2002.

- WELLING, L, THOMSOM, L. Desarrollo Web con PHP y MySQL, Editorial Anaya Multimedia, 2003

Recursos Web

- API JSP de SuN. http://java.sun.com/products/jsp/
- MySQL, Manual de referencia. http://www.mysql.com/
- PostgreSQL, sistema gestor de bases de datos. http://www.postgresql.org/
- Página Oficial de GNU-Linux Debian http://www.debian.org
- Página Oficial de Tomcat http://tomcat.apache.org
- Página Oficial de Apache http://www.apache.org
- Página Oficial de PHP http://www.php.net
- Configuración de ssl: http://www.debian-administration.org/article/Certificate_Authority_CA_with_OpenSSL
- http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=apache_secure_debian
- http://www.vicente-navarro.com/blog/2009/02/22/crear-los-certificados-ssl-para-nuestro-servidor-web-https-con-apache-openssl-y-debian-lenny/

ANEXOS

ANEXO 1. Actividades Realizadas en el Servidor

Servidor de Respaldo

Migración de servidor actual Debian 4 a Debian 5

- Instalación del sistema operativo.
- Actualización de los últimos paquetes de seguridad, plugins.
- Instalación de ntp, ssh, ssl, nmap.
- Instalación y aseguramiento de Apache2, mysql, php5, tomcat, los módulos de conexión php mysql, php y postgres, php y apache, apache y tomcat.
- Instalación de Snort.

Creación de cuentas de usuario restringidas.

Montaje y configuración de los todos los sitios web que había en el servidor principal y los sitios web y aplicaciones de trabajos de grado de la Escuela de Sistemas para sus respectivas sustentaciones; entre ellas: grupo foco, televisión digital terrestre, U2U Project – v1.01 Beta, versiones del meiweb y modulo de scorm, moodles.

Establecer las políticas de backups y monitoreo del servidor.

Políticas de seguridad

Ajuste de permisos en el servidor.

Instalar Snort.

Seguridad modificando parámetros de la Bios

Políticas de copias de seguridad y recuperación.

Políticas de uso de los desarrolladores web y tesistas.

Configuración de logwatch

Chequeo de ips atacantes

Mantenimiento y mejoras

Asistencia como Sysadmin.

Revisión de rutina de los servidores, basado en los mensajes que envia el software Logwath a un email para visualizarlos cada día.

Conexión a Informix desde Windows.

Mejorar y crear nuevos script para el backups de las bases de datos, sitios y documentos de los servidores en su momento cormoran, sistemas y respaldo.

Limpieza física y lógica del servidor de respaldo.

Borrar bases de datos viejas, log y temporales.

Instalar las últimas versiones de apache, mysql, ssh, phpmyadmin, snort, logwatch.

Monitorear la comunicación con la red y que los servidores estuvieran

habilitados dentro y fuera de la subred de la uis.
Administrar las bases de datos.
Instalación del software phpmyadmin
Configuración de scripts en el crontab y reconfiguración de los tiempos de los scripts que habían.
Configuración del sitio eisi web en el servidor de respaldo.
Gestión de activación de 3 puntos de red en la sala de servidores y supervisión de la disponibilidad de los mismos

Soporte a usuarios
Montaje de los sitios web para sustentación de trabajos de grado de tesistas de la Escuela de Sistemas algunos de ellos son:
Asesoría y orientación en el aseguramiento de algunos de estos sitios y montaje de servidores locales en las máquinas de los estudiantes para sus pruebas.
Soporte a profesores para el manejo del Aula Virtual Meiweb.
Instalar plugins flash
Solucionar cambio de root mysql
Solucionar cambio jugo de caracteres
Soporte, organización de los equipos de desarrollo del meiweb (meweb3, meiweb4, meiweb5, meiweb5a)
Soporte a los desarrolladores del eisi cuando se tenía el sitio en el servidor cormoran.
Administración y creación de cuentas de usario
Modificaciones al meiweb para incluir cursos cisco, colaborar en validad el software y detectar vulnerabilidades.

Investigación
Cómo instalar y configurar un servidor de correo con zimbra.
Instalación de la plataforma joomla y sus respectivos componentes.
Instalación de la plataforma moodle y sus respectivos componentes.
Configuración de clusters de alta disponibilidad y las herramientas que hay disponibles en el mercado.
Sincronización automática de bases de datos.

Administración
Organización de la forma de trabajo y metodología a seguir con los desarrolladores del Meiweb.
Organización de horarios de trabajo con los desarrolladores.
Gestión de activación de 3 puntos de red en la sala de servidores y supervisión de la disponibilidad de los mismos
Gestión para la compra de nuevos servidores (cotizaciones, consulta de los procesos administrativos para la compra, elaboración de cartas y

solicitudes, seguimiento para generar la orden de compra y recibo de los equipos).
Elaboración del inventario de la sala de servidores.
Estar pendiente de que solucionarán el problema que había con los switches en el rack de los servidores de sistemas cuando se iba la luz y había que estar atento a informar y gestionar para que los ingenieros a cargo de la División de Servicios de Información pudieran solucionar.