UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO.

FACULTAD DE INGENIERÍA.

“POLÍTICAS Y BUENAS PRÁCTICAS DE SEGURIDAD EN SERVIDORES WEB DEL CDMIT”

T E S I S

QUE PARA OBTENER EL TÍTULO DE:

INGENIERO EN COMPUTACIÓN

PRESENTAN:

CLAUDIA YVETTE CASTRO JAIME
TOMÁS HERNÁNDEZ MUÑOZ

DIRECTORA DE TESIS:

ING. GABRIELA SUSANA CANCINO RAMÍREZ

CIUDAD UNIVERSITARIA FEBRERO 2010

Dedicatorias.

Este triunfo y este trabajo son para mis padres, mi hermana, mi abuelita y mi tío Gustavo por todo el apoyo que me han brindado desde el día que nací hasta verme convertida en una profesionista. Mil gracias por todo...

Claudia.

Este trabajo se lo dedico a todas aquellas personas que de alguna manera me brindaron su apoyo y su ayuda para conseguir algo tan importante en la vida.

Tomás.

Agradecimientos.

No tengo palabras para agradecerles a mis padres, a mi hermana, a mi abuelita y a mi tío por todo el apoyo que me han brindado en cada uno de los proyectos que he emprendido en mi vida siendo el más importante el desarrollo de la carrera de ingeniería y la terminación de esta tesis.

Agradezco a la Facultad de ingeniería y principalmente a los profesores por compartir en su momento su conocimiento y formarme como un profesional de calidad.

A cada una de las personas que contribuyó en el desarrollo de la tesis y que aportó sus conocimientos y su experiencia para corregir la tesis, de todo corazón les digo, mil gracias...

Sobran las palabras para agradecerle a mi querido amigo Tomás Hernández Muñoz y que ha llegado a ser más que eso, por realizar juntos la tesis y superar cada uno de los obstáculos que se presentaron durante la realización de ésta y superar también nuestras diferencias con profesionalismo, gracias por todo...

Claudia.

Gracias a la Universidad por haberme dado la oportunidad de formarme en sus aulas y de ser un orgulloso miembro de su comunidad. Agradezco a mis padres y hermanos por su constante e incondicional apoyo, a Claudia por su ayuda y compañía, gracias a la Ing. Gabriela Cancino y a la Ing. Socorro Armenta por su valiosa aportación en la realización de este trabajo.

Tomás.

TABLAS Y FIGURAS **PÁGINA**

Tablas

Figuras

Introducción

Contexto.

En la actualidad, el cómputo se ha extendido prácticamente a todas las actividades cotidianas, es por esto que la necesidad de proteger los recursos de cómputo (hardware, software e información) ha retomado importancia.

Para una organización la información y los recursos que ésta posee son de suma importancia, por lo que salvaguardarlos es una medida crítica que se debe aplicar para que la organización siga operando de una manera normal.

El Centro de Diseño Mecánico e Innovación Tecnológica (CDMIT) de la Facultad de Ingeniería de la UNAM, fue fundado en 1976, y desde su creación ha formado y capacitado a un gran número de profesores y alumnos de esta Facultad en el diseño de máquinas industriales. El centro ha sido pionero en el desarrollo de máquinas en nuestro país.

La protección de la información y de los recursos que la contienen es de gran importancia para el CDMIT por el valor que tales bienes representan. Dado el crecimiento y evolución que ha tenido el centro en los últimos 10 años, se ha visto en la necesidad de incorporar nuevos activos, entre ellos un servidor web y de bases de datos propio, el cual fue creado en un inicio sin ninguna medida de seguridad, por lo que el CDMIT decidió implementar medidas de seguridad en este servidor, para lo cual se formalizaron las siguientes tareas:

i. Realizar las configuraciones correspondientes para que este servidor sea seguro.
ii. Proponer las medidas de seguridad que se deban llevar a cabo por parte de los administradores para que la información que se almacena en este servidor esté segura, incluyendo un esquema de respaldos.
iii. Elaborar una guía básica de recomendaciones para que las aplicaciones web y de bases de datos que generan los usuarios cumplan con la seguridad mínima requerida.
iv. Elaborar una propuesta tecnológica para actualizar el equipo que actualmente funge como servidor.
v. Elaborar una propuesta de políticas de seguridad para el servidor Web del CDMIT.

Estructura de la tesis.

La presente sección describe la organización de este trabajo:

En el primer capítulo (marco teórico) se presentan los conceptos básicos referentes a la seguridad informática, análisis de riesgos, políticas de seguridad, sistemas operativos para servidor, bases de datos y servidores web.

En el segundo capítulo (vulnerabilidades en aplicaciones web) se presentan las vulnerabilidades más comunes que pueden tener las aplicaciones web y de bases de datos.

En el tercer capítulo (análisis de riesgos) se presentan los resultados del análisis de riesgos que se hizo para el servidor web y de bases de datos del CDMIT.

En el cuarto capítulo (recomendaciones para un servidor web y de bases de datos seguro) se presenta la propuesta tecnológica para contar con un servidor web y de bases de datos robusto en cuanto a seguridad se refiere.

En el quinto capítulo (propuesta de políticas de seguridad) se presenta la propuesta de políticas de seguridad para el servidor web y de bases de datos del CDMIT generadas con base en el análisis de riesgos realizado.

Finalmente se presentan las conclusiones y un anexo con las configuraciones de seguridad propuestas para los servidores web y de bases de datos del CDMIT, para que los administradores o responsables de cómputo del CDMIT cuenten con un documento de consulta y de referencia rápida referente al servidor. Así mismo, para que los usuarios que desarrollan sus aplicaciones web o de bases de datos sepan las medidas básicas de seguridad que deben cumplir sus aplicaciones.

Capítulo 1.- Marco teórico.

En este capítulo se hace una revisión de los conceptos básicos que son relevantes para la realización de este trabajo, además de hacer un análisis de plataformas de software y de hardware que ayuden a proponer una solución tecnológica para un servidor web y de bases de datos seguro en el CDMIT.

1.1.-Seguridad Informática.

La seguridad informática es un conjunto de protecciones (reglas, herramientas, recomendaciones, etc.) para salvaguardar la información, dispositivos y los procesos que forman parte de una red de datos.

Para poder brindar la seguridad deseada es necesario implementar herramientas y mecanismos de seguridad que ayuden a alcanzar el objetivo que es la seguridad de la información.

Se tienen que elegir las mejores herramientas y mecanismos para cumplir con el cometido de proporcionar robustez en cuanto a seguridad a la infraestructura de IT de la organización se refiere. Es necesario tener una serie de pasos que nos ayuden a contestar las siguientes preguntas:

a) ¿Qué es lo que se quiere proteger?

b) ¿De quiénes se van a proteger?

c) ¿Cómo se van a proteger?

Se deben seguir una serie de pasos que lleven a obtener las respuestas a las preguntas mencionadas y justamente en ese orden.

a) ¿Qué es lo que se quiere proteger?

A través de esta respuesta se identificarán los recursos a los que se requieren proteger y a lo que es denominado entorno de seguridad. En la implantación de un esquema de seguridad es necesario identificar los riesgos potenciales. Este proceso tiene que ser desarrollado formalmente por un grupo de todas las áreas de la organización, con el objeto de tener una visión más amplia sobre el valor de los activos y las consecuencias de que se vea comprometida la confidencialidad, integridad o disponibilidad. Para obtener está información es necesario contestar las siguientes preguntas:

- ¿Qué podría pasar?

  Esta pregunta tiene como objeto identificar los activos existentes para la compañía (software, hardware, datos, personas, etcétera) y los eventos amenazantes.

- ¿Si pasara, qué tan malo sería?

  Esta pregunta tiene como objeto cuantificar el impacto de las amenazas en todos los ámbitos (operativo, financiero, etcétera).

- ¿Qué tan frecuentemente podría pasar?

  Esta pregunta tiene como objeto cuantificar la frecuencia de ocurrencia potencial de los eventos amenazantes.

- ¿Qué tan correctas son las respuestas a las tres preguntas anteriores?

Con las respuestas anteriores se identifican claramente los activos sensibles para la empresa y las posibles consecuencias de que se vea comprometida la seguridad.

Con el estudio cuidadoso de las respuestas a estas preguntas, se podrá tener un nivel de conocimiento sobre los activos con que se cuenta y lo que representan para la organización, así como el conocer cuál sería el impacto en caso de que sufran algún incidente.

Para este trabajo, lo que se quiere proteger es el servidor web y de bases de datos del CDMIT.

b) ¿De quiénes se van a proteger?

Al resolver esta pregunta se identifican las amenazas, los riesgos y las vulnerabilidades a las que se encuentra expuesto el entorno identificado.

En una organización se garantiza la seguridad de la información que se maneja, determinando quienes son las personas autorizadas para manejar la misma, así como los recursos que necesita cada uno de los empleados para realizar las tareas que se le asignan.

La seguridad se enfoca en dar protección a los bienes que están expuestos a riesgos considerados como una potencial amenaza. Se debe prestar especial atención a las actividades maliciosas o a las actividades humanas, véase figura 1.1.

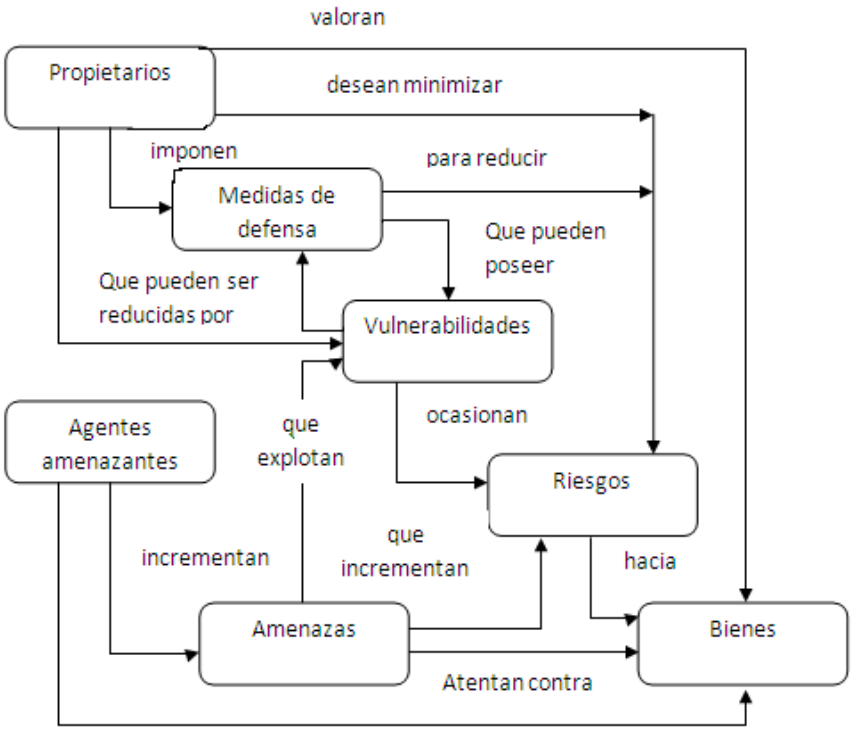

Figura 1.1.-Contexto y relaciones de la seguridad.

Los dueños, responsables o custodios son quienes estiman y valoran los bienes, que desean minimizar los riesgos informáticos implementando medidas de defensa para reducir las vulnerabilidades asociadas. Las amenazas también pueden estimar y valorar los bienes, además de buscar la forma de obtenerlos o abusar de ellos en forma contraria a los intereses de los propietarios.

Los dueños de los bienes, esto es, los propietarios, con ayuda de especialistas en seguridad informática, analizarán todas las amenazas que podrían presentarse para determinar únicamente cuáles son los que aplican a su entorno. Los resultados de este análisis se conocen como riesgos, y dicho análisis ayuda en la selección de las medidas de defensa para contrarrestarlos y reducir éstos a un nivel considerable.

Las medidas de defensa deben seleccionarse e implementarse para reducir puntos vulnerables y cumplir las políticas de seguridad de los dueños de los bienes. Después de la implantación de las medidas de defensa es posible que aún queden puntos vulnerables residuales, de manera que éstos pueden ser explotados por agentes amenazantes que representan un nivel mínimo de riesgo para los bienes; sin embargo es necesario que los dueños implementen restricciones adicionales para minimizar los riesgos.

Para este trabajo, la respuesta a esta pregunta se realizará con el análisis de riesgos de lo que se quiere proteger en el CDMIT.

c) ¿Cómo se van a proteger?

Las dos respuestas anteriores nos llevan a determinar las políticas de seguridad informática para el entorno analizado, ya que las normas ayudan a contrarrestar las amenazas y vulnerabilidades identificadas a fin de salvaguardar su entorno.

El plantear y dar respuestas a estas interrogantes será lo que nos dará la oportunidad de seleccionar de manera formal y segura las herramientas de seguridad necesarias para resguardar la información en riesgo.

Para este trabajo, la pregunta ¿qué es lo que se quiere proteger en el CDMIT? tiene como respuesta en este momento los servidores web y de bases de datos del CDMIT, la pregunta ¿de quiénes se van a proteger? se contestará con los resultados del análisis de riesgos que se explicará más adelante y por último, la pregunta ¿cómo se van a proteger? se resolverá con las políticas de seguridad.

A continuación se explicará el ciclo de la administración de la seguridad y la importancia que tiene para este trabajo.

1.2.- Administración de la seguridad.

Para que un esquema de seguridad quede completo, es necesario que se lleve a cabo la administración de la seguridad, véase figura 1.2. Para realizar esto se recomienda contar con un Departamento de Seguridad en Cómputo, el cual está conformado por personal que se encarga de cada área y por personal especializado en seguridad informática; así como la del equipo de trabajo que se encargará de administrar la seguridad informática.

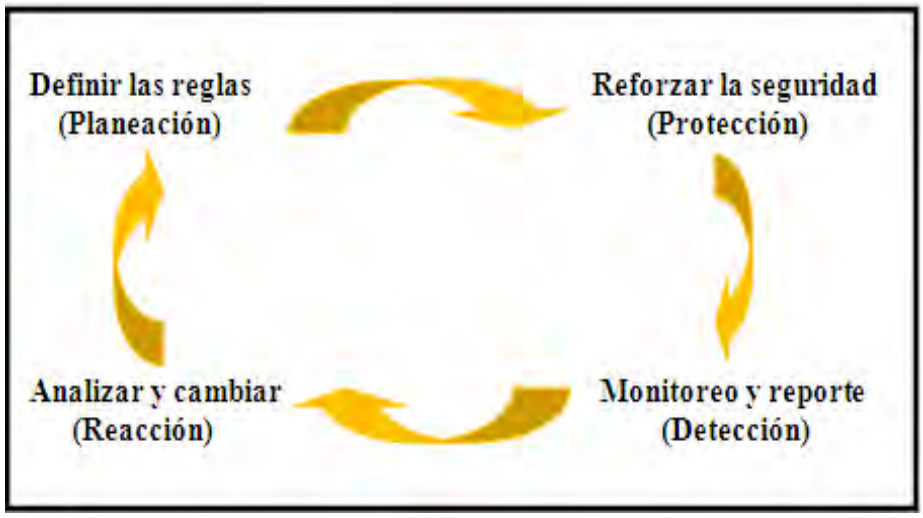

Figura 1.2.-Ciclo de la administración de la seguridad.

1.2.1.- Definición de administración de la seguridad.

La administración de la seguridad se refiere a gestionar y dirigir todas las acciones que se lleven a cabo con el fin de proteger la información, hacer uso lícito de ésta, así como de los recursos con los que cuenta la organización.

La administración consta de cuatro etapas que son:

Etapa 1: Planeación. Se debe llevar a cabo una revisión periódica de las políticas de seguridad, por lo que hay que revisar el esquema de seguridad desarrollado para identificar si se requiere actualizar, remover y modificar las que ya existen.

Etapa 2: Protección. Después de revisar y actualizar las políticas de seguridad del entorno, se debe de reforzar la seguridad con base en éstas y hacer uso de nuevas tecnologías, ya que estas nuevas formas de protección elevan el nivel de seguridad del entorno en cuestión.

Etapa 3: Detección. Es necesario contar con sistemas que permitan realizar actividades de monitoreo de forma continua y permanente en toda información, áreas y sistemas que sean considerados dentro de las políticas como de relevancia; y así mismo, generar reportes que permitan detectar alguna anomalía para tomar las medidas pertinentes, es decir, reaccionar a la anomalía.

Etapa 4: Reacción ante el incidente. En ésta etapa se toman las decisiones que dictan las acciones orientadas a salvaguardar los bienes informáticos de la empresa u organización, esto con base en la información obtenida de la etapa anterior, se realiza de manera continua un análisis de ésta para tomar una decisión de cambio de políticas o mecanismos. Estos cambios pueden ser desde actualizar la tecnología para llevar a cabo la protección de los bienes, modificar esquemas de seguridad o llevar a cabo una revisión extraordinaria de las políticas, entre otros.

Para que el esquema de seguridad del departamento de Cómputo del CDMIT este completo es necesario llevar a la práctica el ciclo de administración de la seguridad.

A continuación se explicará lo que es el análisis de riesgos, las metodologías que se tienen para realizar el análisis de riesgos y los tipos esenciales del análisis de riesgos para más adelante entender los resultados obtenidos de la realización del mismo.

1.3.- Definición de análisis de riesgos.

Se enuncian algunas definiciones de suma importancia para el análisis de riesgos:

- Activo. Es todo aquello que tiene valor para la organización y necesita protección.
- Riesgo. Todo aquello que representa la posibilidad de sufrir algún daño o pérdida.
- Aceptación del riesgo. Decisión para aceptar un riesgo.
- Análisis de riesgos. Uso sistemático de información disponible para identificar las fuentes y para estimar la frecuencia en la que determinados eventos no deseados pueden ocurrir y la magnitud de sus consecuencias.
- Manejo de riesgo. Proceso de identificación, control y minimización o eliminación de riesgos de seguridad – que pueden afectar a los sistemas de información – por un costo aceptable.
- Evaluación del riesgo. Comparación de los resultados de un análisis de riesgo con los criterios, estándares u otros criterios de decisión.
- Impacto. Pérdidas como resultado de la actividad de una amenaza (destrucción, modificación, revelación, denegación de servicios). El impacto generalmente se expresan en las áreas de impacto mencionadas.
- Pérdida esperada. El impacto anticipado y negativo a los activos debido a la manifestación de una amenaza.
- Amenaza. Todo aquello que puede, intenta o pretende destruir o dañar algo.

- Vulnerabilidad. Son las debilidades pertenecientes a algo.
- Ataque. La realización de una amenaza. Cuando una amenaza explota una vulnerabilidad y se logra el objetivo.
- Riesgo residual. Nivel de riesgo que queda después de la consideración de todas las medidas necesarias.
- Control. Protocolos y mecanismos de protección que permiten el cumplimiento de las políticas de seguridad de la organización.

1.3.1.-Tipos de Análisis de Riesgos.

Existen dos tipos de análisis de riesgos:

a) Cuantitativo.

Todos los activos, sus recursos y controles se identifican, y se evalúan en términos monetarios. Todas las amenazas potenciales se identifican y se estima la frecuencia de su ocurrencia, estas amenazas se comparan con las vulnerabilidades potenciales del sistema de tal forma que se identifiquen las áreas que son sensibles.

Esta metodología hace uso del término Expectativa de Pérdida Anual (ALE) o también llamado Costo Anual Estimado (EAC). La forma de calcularlo para un evento en concreto se realiza mediante la multiplicación de la ocurrencia de la amenaza por el valor del activo o clasificación del daño.

De esta forma se puede determinar si los controles existentes son adecuados o se requiere la implementación de otros, esto se observa cuando el producto obtenido tras multiplicar el valor del activo por la frecuencia de ocurrencia de la amenaza en un periodo de tiempo determinado por la duración del control es menos que el costo de dicho control.

Teóricamente es posible situar acontecimientos en el orden del riesgo ALE y posteriormente tomar las decisiones más convenientes. Los problemas de este tipo de análisis de riesgos se asocian generalmente a la falta de fiabilidad (probabilidad del buen funcionamiento de una cosa) y exactitud de los datos, debido a que es difícil lograr una figura representativa de la pérdida o daño que se tiene como resultado de las brechas de seguridad. La probabilidad raramente puede ser exacta y en algunos casos es capaz de promover la satisfacción personal. Además, los controles abordan acontecimientos potenciales que se correlacionan con frecuencia.

b) Cualitativo.

En esta otra metodología en lugar de establecer los valores exactos se asignan niveles de alto, bajo y medio que representan la frecuencia de ocurrencia y el valor de los activos. Este tipo de análisis es el consenso que debe realizarse para jerarquizar la información, los controles y decidir los valores, otra dificultad es la comparación de la pérdida potencial con el costo de implementación de controles para minimizarla, así como qué tan factible resulta aplicar los controles y en qué niveles de información.

Ambos análisis de riesgos hacen uso de tres elementos interrelacionados:

Amenazas

Son aquellos eventos que pueden causar daño a algo y están siempre presentes en cada sistema.

Vulnerabilidades

Son todos aquellos puntos de un sistema que están propensos a ser explotados por una amenaza y que pueden desencadenar en que dicha amenaza tenga mayor probabilidad de tener éxito.

Controles

Son las medidas precautorias que se toman para contrarrestar los ataques y reducir las vulnerabilidades. Existen cuatro tipos de controles:

- Los controles disuasivos reducen la probabilidad de un ataque deliberado.
- Los controles preventivos protegen vulnerabilidades haciendo que los ataques fracasen o que reduzcan su impacto.
- Los controles correctivos reducen el efecto de un ataque.
- Los controles detectores descubren ataques y disparan controles preventivos o correctivos.

Estos tres elementos pueden ser ilustrados mediante un modelo relacional simple que se aprecia en la figura 1.3.

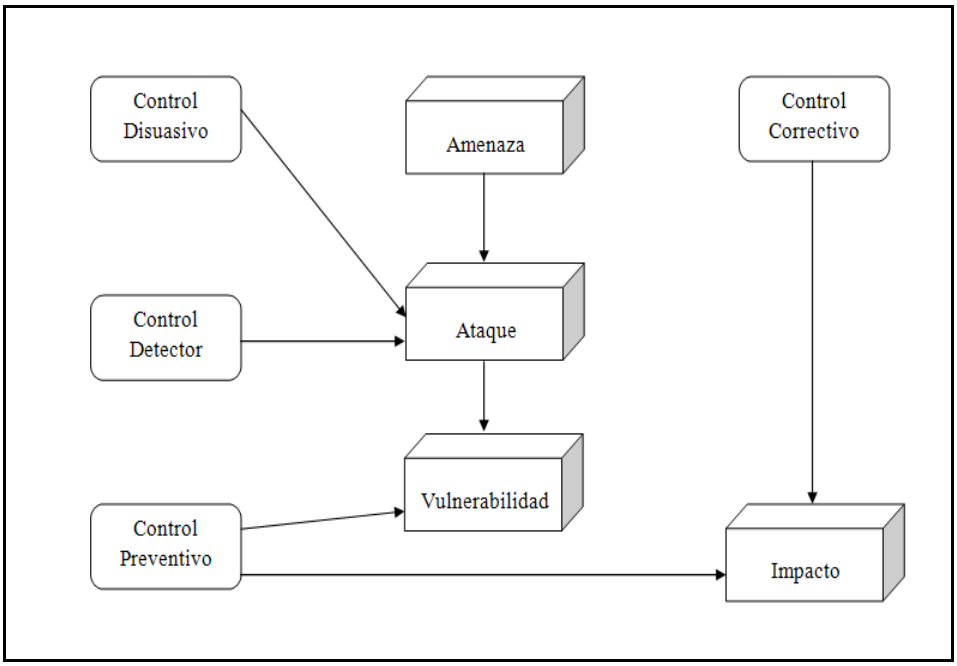

Figura 1.3.-Modelo relacional simple.

En el presente trabajo se hará el análisis cualitativo.

1.3.2.-Pasos a seguir para realizar un análisis de riesgo de tipo cualitativo.

El proceso del análisis de riesgos consiste en 8 pasos interrelacionados:

1. Identificar y evaluar los activos.

   El primer paso para todas las evaluaciones del riesgo es identificar los activos y asignar un valor a los activos que necesitan protección. El valor del activo es importante en la toma de decisiones para realizar cambios operacionales o incrementar la protección de los activos. El valor del activo se basa en su costo, sensibilidad, misión crítica o la combinación de estas propiedades. Así como determinar la importancia de los activos que tiene la organización.

2. Identificar las amenazas correspondientes.

   Después de identificar los activos que requieren protección, es necesario identificar y examinar las amenazas para determinar la posible pérdida si dichas amenazas se presentan.

3. Identificar y describir vulnerabilidades.

   El nivel de riesgo se determina analizando la relación entre las amenazas y las vulnerabilidades. Existen áreas de alta vulnerabilidad que no tienen consecuencias si

no se presentan amenazas. Un riesgo existe cuando una amenaza tiene una vulnerabilidad correspondiente.

4. Determinar el impacto de ocurrencia de una amenaza.

Cuando una amenaza explota una vulnerabilidad los activos sufren algún daño (cierto impacto). Las pérdidas son catalogadas en áreas de impacto llamadas:

- Revelación. Cuando la información es procesada y se pierde la confidencialidad.
- Modificación. El ataque cambia el estado original del archivo.
- Destrucción. Cuando el activo deja de funcionar completamente.
- Denegación de servicio. Pérdida temporal de los servicios.

5. Controles en el lugar.

Identificar los controles. Existen dos tipos que son:

- Controles requeridos. Todos los controles están basados en reglas y procedimientos. La clasificación de los datos almacenados y procesados en un sistema o red y su operación determinan que reglas aplicar, y éstas indican cuales son los controles.
- Controles discrecionales. Elegido por alguien comúnmente los administradores. Muchos de los controles requeridos no reducen el nivel de vulnerabilidad a un nivel aceptable.

6. Determinar los riesgos residuales.

Determinar cuál es el riesgo residual, si es aceptable o no. El riesgo residual toma la forma de las conclusiones alcanzadas en el proceso de evaluación. Las conclusiones deben identificar las áreas que tienen alta vulnerabilidad junto con la probabilidad de ocurrencia de una amenaza y todos los controles que no están dentro del lugar.

7. Identificar los controles adicionales.

Se identifica la forma más efectiva y menos costosa para reducir el riesgo a un nivel aceptable.

8. Preparar un informe del análisis de riesgos. En este informe se detallan los resultados obtenidos del análisis de riesgos así como las recomendaciones para evitar los riesgos.

Una vez que se ha realizado el análisis de riesgos el siguiente paso a desarrollar son las políticas de seguridad para tener una mejor administración de los recursos y del personal de la organización para evitar riesgos.

1.4.-Políticas de Seguridad.

Toda organización, tiene la necesidad y porqué no, la obligación de definir políticas de seguridad.

La necesidad surge porque existe un fallo o deficiencia en la seguridad de la información, la cual pone en riesgo a la misma y a la seguridad a la hora de proteger los datos, que implican significantes sumas de dinero y tiempo invertido.

Con buenas políticas se informa con mayor nivel de detalle a los usuarios, empleados y gerentes de las normas y mecanismos que deben cumplir y utilizar para proteger los componentes de los sistemas de la organización.

El estándar ISO 17799 contiene diez secciones de seguridad. Cada sección cubre un asunto o área, las cuales son utilizadas como base para la determinación de los riesgos de seguridad y la aplicación de controles de seguridad para el manejo de la seguridad de la información. La que se menciona a continuación es laque corresponde a las políticas de seguridad.

Las políticas de seguridad son un conjunto de leyes, reglas y prácticas que regulan la manera de dirigir, proteger y distribuir recursos en una organización para llevar a cabo los objetivos de seguridad de la información dentro de la misma.

Las políticas definen la seguridad de la información en el sistema central de la organización, por lo tanto, un sistema central es seguro si cumple con las políticas de seguridad impuestas para esa organización. Las políticas de seguridad especifica qué propiedades de seguridad el sistema debe de proveer. De manera similar, las políticas definen la seguridad informática para una organización, especificando tanto las propiedades del sistema como las responsabilidades de seguridad de las personas.

Esta primera sección trata la administración, el compromiso y la dirección para lograr las metas de seguridad de la información. El objetivo de esta sección es:

- Proporcionar a la dirección o administración ayuda para la seguridad de la información.

a) Documento de la política de la información: una política de seguridad debe ser especificada en un documento especial para el propósito de ser cumplida, redactada en un lenguaje natural, claramente y sin ambigüedades posibles. El documento debe especificar cuáles son las metas de seguridad de la organización, qué propiedades de seguridad se pretenden cubrir con la aplicación de las políticas y la manera de usarlas. Este documento, junto con una jerarquía de estándares, principios y procedimientos, ayuda a implementar y reforzar los enunciados de la política, además de aprobarse por la administración, publicarse y comunicarse, de manera apropiada a todos los empleados, también debe expresar el compromiso y la aproximación de la organización para manejar la seguridad de la información.

b) Propiedad y análisis: el compromiso de administración de seguridad de la información se establece al asignar planes de propiedad y análisis del documento de

la política de seguridad de la información. La política debe tener un propietario, éste es el responsable de su mantenimiento y revisión periódica de acuerdo a un proceso definido, dicho proceso debe asegurar una revisión periódica debido a los cambios que afectan la evaluación original del riesgo, por ejemplo, incidentes de seguridad, nuevas vulnerabilidades o cambios a la infraestructura técnica o de la organización.

Para este trabajo, las políticas de seguridad ayudarán a contrarrestar las amenazas y vulnerabilidades de los servidores web del CDMIT.

A continuación se explican los principios fundamentales que deben ser reflejados en las políticas de seguridad para que cumplan el fin para lo que fueron hechas.

1.4.1.-Principios fundamentales.

Las políticas de seguridad deben reflejar fielmente la misión, la visión de la organización y los principios fundamentales que se explican a continuación.

1. Responsabilidad individual. Las personas son responsables de sus actos.
2. Autorización. Se establecen las reglas de quién y de qué manera puede utilizar los recursos.
3. Mínimo privilegio. La gente debe estar autorizada única y exclusivamente para acceder a los recursos que necesita para hacer su trabajo.
4. Separación de obligaciones. Las funciones deben estar divididas entre las diferentes personas relacionadas a la misma función o actividad.
5. Auditoría. El trabajo y los resultados deben de monitorearse desde el inicio y hasta después de haber terminado.
6. Redundancia. La redundancia puede afectar el trabajo y la información porque se tienen múltiples copias guardadas con importantes registros y dichas copias frecuentemente son guardadas en diferentes lugares.
7. Reducción del riesgo. El costo de la aplicación debe ser proporcional al riesgo.

Una vez que se sabe lo que tienen que reflejar las políticas de seguridad, el paso que sigue es redactarlas.

1.4.2.-Ciclo de vida de las Políticas de Seguridad.

Las políticas de seguridad dentro de una empresa tienen un ciclo de vida. A continuación se explican los pasos de este ciclo; así como las recomendaciones para su redacción.

1. Definición de las políticas de seguridad.

Para la definición de las políticas se consideran las siguientes recomendaciones:

i) Conocer los activos que se quieren proteger en la organización.

ii) Se elije una filosofía básica de las dos existentes que son la prohibitiva y la permisiva, la primera dice que todo esta prohibido a excepción de lo que específicamente está permitido y la segunda, todo esta permitido a excepción de lo que específicamente está prohibido.

iii) Se redacten como estándares o como recomendaciones, de manera positiva, ser generales, que no sean ambiguas y difíciles de entender, que las políticas no lleven a malos entendidos, hostigamientos, discriminación, abusos, etcétera y que no cambien mucho con el tiempo.

iv) Establecer responsabilidades de control y se tiene que asignar un dueño a los recursos y a la información que debe protegerse.

Una vez que han sido definidas las políticas de seguridad:

2. Implementación de las políticas de seguridad.

   Poner en funcionamiento las políticas de seguridad que se redactaron.

3. Verificación del cumplimiento de las políticas de seguridad.

   Se verifica con ayuda de la capacitación inicial y continua de todos los usuarios sobre la concientización de la seguridad y su importancia, además del cumpliendo de las políticas.

4. Revocación de la política de seguridad.

   Se verifica si las políticas tienen que actualizarse, si se están cumpliendo o si deben eliminarse o si no se cumplen. El tiempo de verificación de las políticas lo propone la organización dependiendo de los cambios de la tecnología, de los procesos, de las personas y de la misma organización.

Una vez que se redactaron las políticas de seguridad el siguiente paso es elaborar un plan de contingencias, que para este trabajo, se dejará como una recomendación para el CDMIT, ya que involucran muchos aspectos donde se necesita la colaboración de todos los empleados y encargados del CDMIT.

1.5.- Plan de contingencias.

1.5.1.-Definición de plan de contingencias.

Todas las instituciones deberían contar con un plan de contingencias actualizado, ya que es una herramienta que elimina, transfiere, mitiga o acepta los riesgos. Este plan de contingencias se basa en el análisis de riesgos que realiza la empresa previamente.

El plan de contingencias es una estrategia que se constituye de un conjunto de recursos ideados que tienen el propósito de servir de respaldo para conseguir una restauración progresiva y oportuna de los servicios de una organización.

A su vez, un plan de contingencia cuenta con las medidas necesarias para garantizar que la organización continúe con sus operaciones y se trata de un programa de tipo preventivo y correctivo que indica las acciones que deben tomarse inmediatamente ante una eventualidad de incidentes, accidentes y/o estados de emergencias que pudieran ocurrir tanto en las instalaciones como fuera de ella.

Los objetivos del plan de contingencia son el de planificar y describir la capacidad para respuestas rápidas, requerida para el control de emergencias.

A continuación se hará una comparativa sobre las características de diversos sistemas operativos, servidores web, manejadores de bases de datos y plataformas de hardware para poder elegir cada una de estás opciones para instalar un servidor web de acuerdo a los recursos económicos disponibles y el uso que se le va a dar al servidor web y de bases de datos.

1.6.-Sistemas Operativos para Servidores.

En el mercado existen diversos sistemas operativos especializados para funcionar como servidores, los más utilizados son los de las familias de Microsoft Windows o alguna distribución de Linux.

1.6.1.-Sistemas Operativos de la familia Microsoft.

La corporación estadounidense Microsoft es conocida mundialmente por el desarrollo del popular sistema operativo Microsoft Windows. Este sistema operativo ha evolucionado mucho desde la aparición de su primera versión en 1985 hasta tener toda una gama de productos orientados a cubrir las necesidades de los diferentes tipos de usuarios.

Actualmente Microsoft Windows cuenta con versiones para usuarios domésticos, para empresas y para dispositivos móviles.

1.6.1.1.-Microsoft Windows Server.

El sistema operativo Microsoft Windows Server cuenta con tres versiones recientes que se pueden encontrar en muchas organizaciones, nos referimos a las versiones 2000, 2003 y 2008.

En la tabla que se muestra abajo (tabla 1), podemos apreciar algunas características importantes de estas tres versiones de Microsoft Windows Server.

Característica.	Microsoft Windows 2000 Server.	Microsoft Windows Server 2003.	Microsoft Windows Server 2008.
Sistema de Archivos.	FAT 32 y NTFS.	NTFS.	NTFS.
Soporte HTTP.	Si.	Sí	Sí.
Soporte DNS.	Sí.	Sí.	Sí.
Soporte FTP.	Sí.	Sí.	Sí.
Soporte HTTPS.	Sí.	Sí.	Sí.
Soporte SSH.	Sí.	Sí.	Sí.
Soporte DHCP	Sí.	Sí.	Sí.
Soporte RAID.	Sí.	Sí.	Sí.
Fecha de lanzamiento.	Febrero 2000.	2003.	Febrero 2008.
Versión estable.	SP4.	R2.	SP2.
Precio promedio.	$199-$5,999 USD	$199-$7,999 USD	$199-$9,750 USD
Soporte.	Sólo Actualizaciones de Seguridad.	Sí.	Sí.
Licencia.	Propietaria.	Propietaria.	Propietaria

Tabla 1. Comparativa de Sistemas Operativos de la Familia Microsoft Windows Server.

Microsoft Windows 2000 Server es un sistema operativo ya obsoleto con respecto a las versiones 2003 y 2008. Ya no cuenta con soporte por parte de Microsoft, aunque aún cuenta con actualizaciones de seguridad.

El sistema operativo Microsoft Windows Server 2003 ha disminuido su precio con la salida al mercado de la versión 2008 de este sistema operativo, aunque su precio aún sigue siendo alto con respecto a otros sistemas operativos en el mercado, ya que oscila entre los 199 y los 7,999 USD, según la versión que se requiera, pero cuenta con nuevas herramientas con las que no contaba Microsoft Windows 2000 Server, como son IIS 6, Microsoft Identity Integration Server 2003 (MIIS), implementación de estándares abiertos (IEEE 802.1X), entre otras.

Microsoft Windows Server 2008 es el sistema operativo más reciente para servidores que ha sacado al mercado Microsoft, por lo que cuenta con todo el soporte y respaldo de Microsoft. Si se compara con otros sistemas operativos, su costo es muy elevado, ya que según la versión que se quiera, va desde los 199 hasta los 9,750 USD. Entre las novedades que se presentan respecto a la versión 2003 tenemos IIS 7, Windows Communication Foundation, Windows SharePoints Services, Security Configuration Wizard (SCW) y Network Access Protection, entre otras.

1.6.2.-Sistemas Operativos de la familia Linux.

El sistema operativo Linux cuenta con muchas distribuciones, algunas de las cuales son: Mandriva, Debian, Red Hat, SuSE, Gentoo, Ubuntu, Fedora, BSD, entre otras. Las distribuciones que cuentan con versiones especializadas en servidores y que son más utilizadas son Red Hat, SuSE, Ubuntu (basado en Debian) y Fedora (basado en Red Hat).

1.6.2.1.-Comparativa entre distribuciones del Sistema Operativo Linux.

Fedora es un Sistema Operativo Linux que cuenta con una amplia aceptación por parte de la industria, con lo último y lo más nuevo en software libre y de código abierto. Su versión actual a Diciembre de 2009 es la versión 12.

Ubuntu Server también es una distribución Linux que va ganando adeptos como Sistema Operativo para servidores. Cuenta con una gran cantidad de repositorios de software y actualmente se encuentra en su versión 9.10.

Por su parte Red Hat (la distribución en la que se basa Fedora), es una distribución de Linux ya probada y con un amplio mercado en la industria, cuenta con un soporte continuo y amplio que permite a las empresas implementar las soluciones que requieren de una manera robusta. La versión actual de Red Hat Enterprise Linux es la 5.3.

Finalmente otra distribución de Linux muy utilizada para el servicio web y de bases de datos es SuSe Linux, que es una distribución desarrollada por la compañía Novell, cuenta con herramientas muy poderosas que facilitan la gestión de paquetes, además de una gran capacidad de integración con tecnologías de Microsoft Windows, su versión actual es SuSE Linux Enterprise Server 11.

En la tabla 2 hacemos una comparativa de las características más importantes de las distribuciones de Linux más utilizadas para servidores.

Característica.	Fedora Core.	Ubuntu Server.	Red Hat Enterprise	SuSe Linux Enterprise Server
Sistema de Archivos.	Ext3.	Ext3.	Ext3.	Ext3.
Versión Actual.	12	9.10	5.3	11
Licencia	GPL.	GPL.	Propietaria	Propietaria
Precio Promedio.	Gratuito.	Gratuito.	$349.00 USD - $18,000.00 USD	$349.00 USD - $4,050.00 USD
Principales Servicios.	HTTP, FTP, SSH, DNS, DHCP,	HTTP, FTP, SSH, DNS, DHCP,	HTTP, FTP, SSH, DNS, DHCP,	HTTP, FTP, SSH, DNS, DHCP,

	HTTPS.	HTTPS.	HTTPS.	HTTPS.
Versión Estable.	11	9.04	5.3	11

Tabla 2. Comparación de los sistemas operativos de la familia Linux.

La distribución Fedora Core es completamente gratuita, está basada en la distribución Red Hat, pero a diferencia de ésta, es soportada por la comunidad, lo que puede ser una desventaja, ya que las actualizaciones, modificaciones y correcciones a fallos suelen tardar mas tiempo en aparecer que en un sistema operativo soportado por una corporación. Otra desventaja es que su manejo no es tan sencillo para usuarios con un nivel de conocimiento básico en Linux.

Ubuntu Server es otra distribución de Linux basada en Debian, completamente gratuita y que cuenta con un gran repositorio de software. Por defecto no incluye entorno gráfico, aunque es posible instalarlo y configurarlo. Este sistema operativo es soportado por la comunidad. Tiene gran aceptación a nivel mundial por su gran facilidad de uso, incluso para personas con conocimientos básicos en Linux. Es bastante rápido y se puede ejecutar en computadoras con bajos recursos. Además de ofrecer actualizaciones automáticas de manera constante.

El sistema operativo Red Hat Enterprise es una distribución de Linux de código abierto, que sin embargo no es gratuita, pero cuenta con actualizaciones continuas y un gran soporte por parte de la corporación que lo desarrolla. Es ampliamente utilizado en un sin fin de empresas medianas y grandes, ya que ofrece soluciones robustas y acordes a las necesidades de la mayoría de las empresas. Cuenta además con una integración muy buena con sistemas de las familias Windows y Unix.

SuSE Linux Enterprise Server es otra distribución de Linux que también es de código abierto y al igual que Red Hat Enterprise no es gratuita, pero cuenta con el soporte de Novell y con una amplia integración con sistemas operativos de Microsoft, ya que Novell y Microsoft han trabajado para ello. Esto representa una gran ventaja para las empresas que combinan ambas tecnologías, ya que hace más fácil, más rápido y menos costoso este proceso.

El conocer los tipos de sistemas operativos que hay en el mercado nos da la pauta para elegir la mejor plataforma para el servidor dependiendo de las necesidades del CDMIT y los recursos con los que cuenta.

1.7.-Servidores web.

En la actualidad existen una gran cantidad de servidores web, pero los más populares y utilizados son sin duda alguna Apache HTTP Server de Apache Software Foundation y Microsoft IIS de Microsoft Corporation.

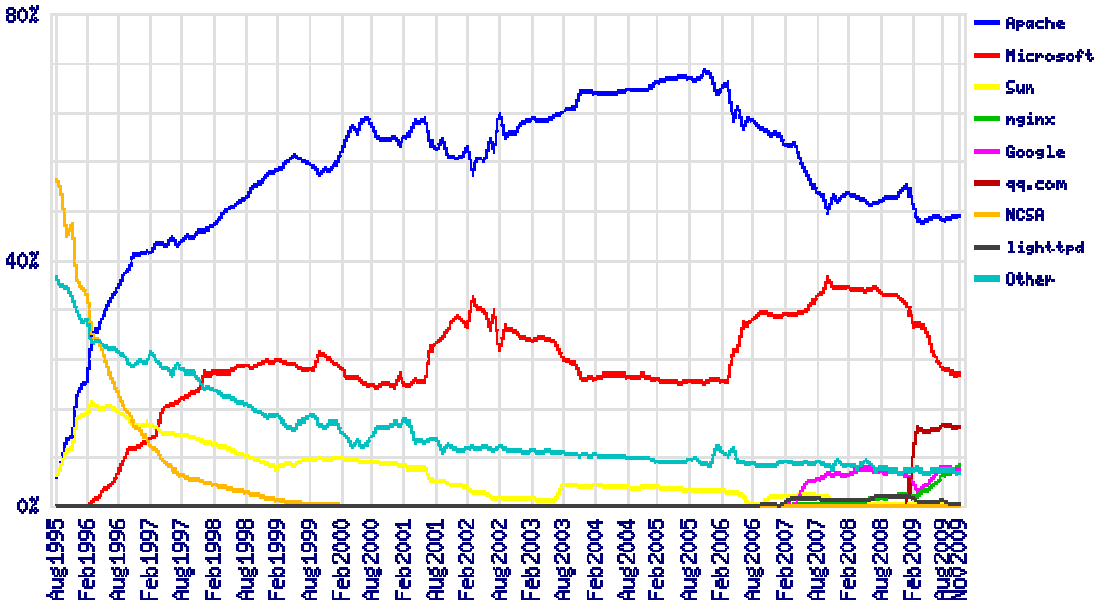

Figura 1.4.- Distribución mundial del uso de servidores web según Netcraft.

Apache es un servidor web creado por Apache Software Foundation y es el más utilizado en el mundo. Según Netcraft, a noviembre de 2009, Apache cuenta con cerca del 50% de usuarios en el mercado de servidores web.

El segundo servidor web más utilizado es IIS de Microsoft, el cual, además de proveer servicios web, tiene la capacidad de dar servicios de FTP y SMTP. Cifras de Netcraft indican que en el mercado IIS es utilizado por cerca del 35% de los usuarios de servidores web en el mundo (noviembre 2009).

1.7.1.-Comparativa entre Apache y IIS.

En la tabla 3 hacemos una comparativa entre Apache y IIS, en la cuál, pretendemos resaltar sus mejores características y ventajas para nuestras necesidades.

Característica.	Apache 1.X.	Apache 2.X.	IIS (HTTP).
Versión actual.	1.3	2.2	7.5
Versión estable.	1.3	2.0.3	7
Sistemas Operativos Compatibles.	Microsoft Windows, Linux, Novell NetWare, Solaris.	Microsoft Windows, Linux, Novell NetWare, Solaris.	Microsoft Windows.
Precio.	Gratuito.	Gratuito.	Gratuito.
Licencia.	GPL.	GPL.	Propietaria.
Soporte SSL-TLS.	Sí.	Sí.	Sí.
Principal ventaja.	Es compatible con múltiples plataformas.	Es compatible con múltiples plataformas.	Soporte nativo para ASP y ASP .NET.

Principal desventaja.	Con la aparición de la versión 2, se ha vuelto obsoleto.	Si no se tiene experiencia, resulta difícil de instalar y configurar.	Sólo es compatible con algunas versiones de Sistemas Operativos de Microsoft.

Tabla 3. Comparativa de los servidores web: Apache y IIS.

Con esta comparativa elegiremos el mejor servidor web de acuerdo a las necesidades del CDMIT.

1.8.-Manejadores de bases de datos.

En el mercado tenemos una gran cantidad de sistemas manejadores de bases de datos, algunos son de código abierto y otros son software propietario. Los de código abierto que son más conocidos son: MySQL, PostgreSQL y SQLite y los que son software propietario y que están más difundidos son: Microsoft SQL Server, Oracle, IBM DB2, IBM Informix y Sybase, entre otros.

En este punto sólo analizaremos 2 de los manejadores de bases de datos de código abierto, MySQL y PostgreSQL. Los manejadores con licencia propietaria son muy caros y llegan a alcanzar precios de entre $5,000 USD y $80,000 USD por licencia (por cada CPU).

1.8.1.-Comparativa entre MySQL y PostgreSQL.

A continuación mostramos la tabla comparativa de las características entre MySQL y PostgreSQL.

	MySQL.		**PostgreSQL.**
Característica.	MySQL Community Server.	MySQL Enterprise Server.	
Versión actual.	6.0	5.1	8.4.1
Versión estable.	5.1	5.1	8.4.1
Sistemas Operativos compatibles.	Windows, Linux, Solaris, FreeBSD, Mac OS X, HP-UX, IBM AIX, IBM i5/OS.	Windows, Linux, Solaris, FreeBSD, Mac OS X, HP-UX, IBM AIX, IBM i5/OS.	FreeBSD, Linux, Mac OS X, Solaris y Windows.
Precio.	Gratuito.	$599.00-$4,999.00 USD	Gratuito.
Licencia.	GPL.	Propietaria.	GPL.
Principal Ventaja.	Totalmente gratuito.	Soporte por parte de SUN.	Totalmente gratuito.

Principal desventaja.	No es posible realizar subconsultas.	Tiene un Costo muy alto.	Soportado por un menor número de plataformas que MySQL.

Tabla 4. Comparativa entre los manejadores de bases de datos: MySQL y PostgreSQL.

Se sabe que existen muchos manejadores de bases de datos pero con esta comparativa se podrá elegir el que más se adecue a las necesidades del CDMIT.

1.9.-Plataformas de Hardware.

Un aspecto muy importante a considerar para la seguridad en un servidor web y de bases de datos es también la plataforma de hardware en la cual será montado, ya que muchas veces si no se tiene el hardware con los recursos necesarios, resulta complicado ofrecer una seguridad robusta y garantizar la disponibilidad del servicio adecuadamente.

En esta sección hacemos una comparativa entre diferentes plataformas de hardware especializadas para el servicio web y de bases de datos. Analizamos un producto con características similares de tres de los mayores fabricantes de este tipo de hardware que ofrecen soluciones para servidores web en ambientes Linux y Windows, tal es el caso de IBM, Hp y Dell.

1.9.1.-Comparativa del hardware.

Para esta comparativa, se eligieron tres modelos de hardware para montar un servidor web y de base de datos con base en los 3 mayores distribuidores a nivel mundial para este tipo de productos en plataformas Linux y Windows: Dell Poweredge 1950 III, HP ProLiant serie BL260c G5 y IBM System x3200. Estos modelos, uno por fabricante, fueron elegidos de acuerdo a las recomendaciones hechas por los mismos fabricantes, ya que son productos apropiados para PyMES y que cumplen con las características necesarias para analizarlos en este trabajo (orientados a PyMES y a servicios web).

Característica.	Dell Poweredge 1950 III.	HP ProLiant serie BL260c G5.	IBM System x3200.
Procesadores compatibles.	Procesadores Intel Xeon.	Procesadores Intel Xeon.	Intel Xeon, Intel Pentium D.
Procesador.	Procesador Intel Xeon cuádruple; E5405, 2x6MB Cache, 2.0GHz, 1333MHz FSB.	Procesador Intel® Xeon® 445 Single-Core a 1,86 GHz.	Intel Xeon (doble núcleo) (4 MB/hasta 2,4 GHz/1066 MHz).
Número de núcleos soportados.	8	4	2
Sistemas operativos compatibles.	Microsoft Windows Server 2008, Microsoft Windows Server 2003, Red Hat Linux, Debian	Microsoft Windows Server 2003, Microsoft Windows Server 2008, Red Hat Enterprise	Microsoft Windows Server 2003, Microsoft Windows Server 2008, Red Hat Enterprise

	Linux, Ubuntu Linux, SuSE Linux, Novell Netware.	Linux, USE Linux Enterprise Server, Sun Solaris, Ubuntu Server Linux.	Linux, SUSE Linux Enterprise Server, Novell NetWare, IBM operating system 4690, Ubuntu Server.
Memoria RAM maxima.	64 GB.	48 GB.	8 GB
Memoria RAM.	Memorias DIMM 4GB, 667MHz, (4x1024MB).	1 GB (2 x 512 MB).	2 GB DDR II 667 or 800 MHz.
Capacidad máxima de almacenamiento en disco.	2 TB	2 TB.	2 TB.
Disco duro.	Disco duro de 250 GB, SATA, de 3.5 pulgadas, con velocidad de 7,200 RP.	Sin disco duro.	Sin disco duro.
Accesorios.	arjeta de interfaz de red Ethernet doble incorporada Broadcom® NetXtreme II 5708 Gigabit. 3 años de garantía.	1 adaptador de red adicional 10/100 dedicado a gestión iLO 2.	Integrated Gigabit Ethernet.
Temperatura idónea para el funcionamiento.	10 ºC-15ºC	Dato no disponible.	Dato no disponible.
Garantía.	3 años.	1 año.	3 años.
Precio aproximado.	$1,600.53 USD	$1,016.76 USD	$1,291.00 USD

Tabla 5. Comparativa de plataformas de Hardware.

Cabe señalar que el hardware anterior no cuenta con sistema operativo ni monitor.

Con todas las comparativas realizadas sobre las plataformas, servidores web, manejadores de bases de datos y del hardware nos será posible proponer la mejor solución web para cubrir las necesidades del CDMIT.

A continuación se explican las vulnerabilidades que tienen las aplicaciones web que ponen en riesgo a los servidores web y de bases de datos y también la forma de evitarlos para tener protegido al servidor, además de otros mecanismos de defensa.

Capítulo 2.- Vulnerabilidades en aplicaciones web.

En este capítulo se explican algunas vulnerabilidades en aplicaciones web que pueden ser explotadas por software o por personas malintencionadas y como consecuencia, provocar algún daño al servidor web y de bases de datos o a la información que se aloja en el mismo. También hacemos recomendaciones para poder evitarlas.

2.1.-Web.

Las aplicaciones web pueden presentar diversas vulnerabilidades que van de acuerdo a los servicios que prestan. De acuerdo con la OWASP (Open Web Application Security Project), las vulnerabilidades en aplicaciones web más explotadas a finales del año 2009, fueron las siguientes:

- Cross Site Scripting (XSS).
- Ataques de inyección de código.
- Ejecución de archivos maliciosos.
- Insecure Direct Object Reference.
- Ataques Cross Site Request Forgery (CSRF).
- Pérdidas de información y errores al procesar mensajes de error.
- Robo de identidad de autentificación.
- Almacenamiento criptográfico inseguro.
- Comunicaciones inseguras.
- Acceso a URLs ocultas no restringidas de manera adecuada.

El sitio www.opensecurity.es indica que los 5 principales tipos de vulnerabilidades en aplicaciones web son:

- Ejecución remota de código.
- SQL.
- Vulnerabilidades en formato de cadenas.
- Cross Site Scripting (XSS).
- Problemas atribuidos a los usuarios.

Otro sitio (www.vsantivirus.com) publica que las vulnerabilidades más comunes son:

- SQL Injection.
- Cross Site Scripting (XSS).

El sitio del Departamento de Seguridad en Cómputo de la UNAM (http://www.seguridad.unam.mx/vulnerabilidadesDB/) menciona en su lista de vulnerabilidades más comunes a las siguientes:

- Cross Site Scripting (XSS).
- SQL Injection.
- Buffer Overflow.

Podemos apreciar que en las anteriores listas, la vulnerabilidad en aplicación web en común es el ataque Cross Site Scripting (XSS), por lo que se debe poner atención en ella, así como las que se explicarán más adelante.

Para este trabajo, se analizaron las vulnerabilidades en aplicaciones web más comunes porque pueden representar un peligro para los servidores web y de bases de datos del CDMIT.

2.1.1.-XSS (Cross Site Scripting).

El XSS es un fallo de seguridad en sistemas de información basados en web, que más que comprometer la seguridad del servidor web compromete la seguridad del cliente.

El XSS es un ataque, que consiste en inyectar código, HTML y/o JavaScript en una aplicación web, con el objetivo de que el cliente ejecute el código inyectado al momento de ejecutar la aplicación.

El XSS se da cuando una aplicación web permite inyectar código en la página; esto se puede lograr por medio de campos de texto de formularios o por medio de la URL.

Por lo regular, el código inyectado se ejecuta de manera que el cliente no nota algún comportamiento fuera de lo normal, ya que la ejecución de este código se hace de manera simultánea con el código original de la aplicación. Dependiendo de otros factores, el XSS puede hacer que el navegador funcione de manera indebida y en algunos casos, llegar a provocar un fallo en el servidor. Aunque esto último es muy difícil porque el código HTML y JavaScript se ejecutan en el navegador y no en el servidor web.

Para poder evitar este tipo de ataque, en las aplicaciones web se debe:

- Evitar que llegue código HTML y/o JavaScript por medio del método Get o Post, es decir, hay que filtrar el código. Algunos lenguajes de programación del lado del servidor proveen funciones para evitar que el código HTML llegue como tal a nuestras aplicaciones. Un ejemplo de esto es PHP que cuenta con funciones que convierten el código HTML en texto o simplemente lo suprimen, tal es el caso de las funciones htmlspecialchars() y htmlentities().

- Procurar que los datos viajen por Post en lugar de Get para evitar ataques de XSS por URL.

- También es recomendable hacer uso de navegadores modernos, ya que algunos tienen la capacidad de detectar casos en donde se podría presentar la vulnerabilidad y lo invalidan. Como Mozilla Firefox 3 e Internet Explorer 8, que lo hacen.

Aunque algunos expertos en seguridad informática opinan que las vulnerabilidades a XSS son ya obsoletas, otros opinan lo contrario, pues el XSS es muy utilizado para realizar ataques como el *phishing*, robo de identidad y otro tipo de ataques, por lo que es importante conocer cómo funciona para poder evitarlo.

2.1.2. - CSRF (Cross Site Request Forgery).

El CSRF es prácticamente igual que XSS, salvo que este ataque se basa en explotar la confianza que un usuario tiene en un sitio web.

Se trata de una vulnerabilidad en aplicaciones web, en donde el usuario es forzado a ejecutar acciones no deseadas en una aplicación web en la cual se encuentra autenticado. Con un poco de ayuda de ingeniería social (como el envío de una liga vía correo electrónico o chat), una persona malintencionada podría forzar al usuario de la aplicación web a ejecutar acciones no deseadas por el mismo usuario, por ejemplo la ejecución de código de manera remota.

De ser exitoso, el CSRF puede comprometer la información del usuario y el comportamiento normal del usuario en la aplicación web. En caso de que el usuario sea el administrador de la aplicación web, este tipo de vulnerabilidad puede llegar a comprometer por completo al sistema en cuestión.

Existen numerosas formas en las que el usuario puede ser engañado cuando intercambia información con un sitio web. Con el fin de llevar a cabo un ataque de este tipo, primero se debe saber como generar una petición maliciosa para que nuestra víctima la ejecute.

Ejemplo:

El usuario B desea hacer una transferencia bancaria de $1,000 al usuario A, a través del portal de Internet de un banco (http://banco.com), la cabecera http de la petición al servidor generada por el usuario B podría ser de la siguiente manera:

```
GET http://banco.com/transfer.do HTTP/1.1
...
...
...
Content-Length: 19;

acct=USUARIOA&amount=1000
```

Pero el usuario C nota que la aplicación web realiza la transferencia de parámetros de la URL de la siguiente manera:

```
GET http://banco.com/transfer.do?acct=USUARIOA&amount=1000 HTTP/1.1
```

Y como consecuencia de lo anterior, el usuario C decide explotar esta vulnerabilidad en la aplicación web del banco tomando como su víctima al usuario B. De esta forma, el usuario C construye una URL que transferirá $100,000 de la cuenta del usuario B a su propia cuenta, como sigue:

```
http://banco.com/transfer.do?acct=USUARIOC&amount=100000
```

Ahora que el código de la petición maliciosa se ha generado, el usuario C debe engañar al usuario B para que este envíe la petición al servidor del banco, ya que el usuario B está autenticado en el sistema. El método más sencillo sería que el usuario C le envíe al usuario B el link con la petición y que este lo abra, ya sea por correo electrónico o vía chat. La etiqueta HTML sería la siguiente:

```
<a href="http://banco.com/transfer.do?acct=USUARIOC&amount=100000">Gane
$10,000.00 Ahora</a>
```

Asumiendo que el usuario B se encuentra autenticado (ha iniciado sesión) en la aplicación web del banco, cuando da clic al link que le envió al usuario C, él inconscientemente transfiere a la cuenta del usuario C $100,000. Para que el usuario B no lo note (ya que el banco le notificará de la transferencia), el usuario C decide esconder el ataque en una imagen de cero bytes, como sigue:

```
<img src="http://banco.com/transfer.do?acct=USUARIOC&amount=100000" width="1"
height="1" border="0">
```

Si esta etiqueta HTML de la imagen, fuese incluida en un correo electrónico, el usuario B sólo vería una pequeña caja indicando que el navegador no puede mostrar la imagen. Sin embargo, el navegador en cualquier caso, enviará la petición al sistema del banco sin ninguna indicación de que la transferencia se ha llevado a cabo.

Este tipo de ataque se puede prestar a fraudes como *phishing.*

Para evitar este tipo de vulnerabilidad en aplicaciones web, por el lado del programador, se debe:

- Hacer que las sesiones expiren en un tiempo corto. este tiempo tiene que ser el suficiente para que el usuario haga la transacción que requiere.
- Forzar a que el usuario termine su sesión para que de esta forma se evite que la sesión del usuario quede activa y se pueda hacer mal uso de ella.
- Hacer del conocimiento del usuario que los problemas mencionados anteriormente se pueden presentar y concientizarlo de cómo prevenirlos.
- Ocultar la URL en navegadores y códigos fuente de aplicaciones para evitar su mal uso.
- Hacer un filtrado de datos que llegan al servidor, es decir, verificar que los tipos de datos sean los que se esperan.
- Hacer que la composición de la URL sea compleja, es decir, cifrar los datos que viajan a través de esta.

Para evitar este tipo de vulnerabilidad en aplicaciones web, por parte del usuario, se debe:

- Hacer el intercambio de la información a través de internet en un lugar seguro y no sitios públicos como escuelas, lugares de trabajo o cibercafés.
- Proteger el equipo con software *antiphishing*, sobre todo para personas que no tienen muchos conocimientos de informática.
- Advertir al usuario de no abrir ligas o correos electrónicos sospechosos o de dudosa procedencia.

2.1.3.-Inyección de Código (Code Injection).

Code Injection es el nombre de un ataque, que consiste en insertar código que podría ser ejecutado por una aplicación. Un ejemplo de esto, es cuando se añade una cadena de caracteres en una cookie o los valores de un argumento en la URL. Este tipo de ataque hace uso de la falta de una validación correcta de los datos: tipos de caracteres permitidos, formato de datos, datos esperados, etcétera.

Code Injection y *Command Injection*, son ataques muy similares entre sí, por lo que no analizaremos a fondo este último, pero si diremos que la diferencia entre *Code Injection* y *Command Injection* son las medidas distintas que se toman para lograr objetivos similares. Mientras que *Code Injection* tiene como objetivo añadir código malicioso en una aplicación, que luego se ejecutará como parte de ésta, *Command Injection* no es precisamente código que pertenece a la aplicación y no necesariamente se ejecutará simultáneamente con ésta. Este tipo de vulnerabilidad en el código, puede ser muchas veces peor que cualquier otra vulnerabilidad, ya que la seguridad del sitio web y posiblemente del servidor, se ve comprometida.

Un ejemplo muy sencillo de cómo opera este tipo de ataque es el siguiente:

Hagamos la suposición de que el siguiente código en PHP se va a ejecutar:

```
<html>

<body>

<?php

$pagina=$_GET[‘page’];

Include(‘$pagina’);

?>

</body>

</html>
```

Si en un servidor externo tenemos un script, es posible realizar un ataque con Code Injection que el código anterior presenta. Digamos que la URL que corresponde al código de la aplicación de arriba es la siguiente: http://ejemplo.net/index.php y que el script con el código que se inyectará está en el servidor cuya URL es http://codigo.net/code.php, entonces basta con construir una URL como sigue para realizar el ataque y llamarla desde el navegador:

```
http://ejemplo.net/index.php?page=http://codigo.net/code.php
```

Con esto se ejecutará todo el código que el atacante haya escrito en el archivo "code.php".

Para evitar este tipo de ataques, basta con hacer una programación ordenada, con el filtrado del código y con la inicialización de todas las variables. El ejemplo aquí presentado es el más utilizado para realizar este tipo de ataques, por lo que se debe evitar a toda costa incluir archivos por medio de variables.

2.1.4.-Buffer Overflow.

Este tipo de vulnerabilidad, es quizá, la más conocida dentro de la seguridad en el software. La mayor parte de los desarrolladores de software saben lo que una vulnerabilidad del tipo Buffer Overflow es, sin embargo, este tipo de vulnerabilidad suele ser aún común hoy en día. Parte del problema se debe a la gran variedad de formas en las cuales el Buffer Overflow se puede dar.

Este tipo de vulnerabilidades no son tan fáciles de descubrir y cuando se llegan a descubrir, son por lo general, difíciles de explotar. A pesar de lo anterior, los atacantes han logrado identificar vulnerabilidades de este tipo en un sin fin de sistemas de todo tipo.

Es clásico que en esta vulnerabilidad, el hacker envíe datos a un programa, los cuales, son almacenados en una pila de tamaño inferior al de los datos. El resultado de esto es que la información en la pila es sobrescrita incluyendo las funciones de punto de retorno. Los datos establecen el valor del punto de retorno, así que si la función regresa, transfiere el control al código malicioso contenido en los datos del atacante.

Aunque este tipo de Buffer Overflow es aún común en algunas plataformas y en algunas comunidades de desarrollo, existen otras variedades de tipos de Buffer Overflow. Otra clase de vulnerabilidad muy similar y conocida es la llamada Format String Attack. Existe un gran número de información acerca de esta vulnerabilidad en Internet y en libros, que provee muchos detalles de cómo trabaja esta vulnerabilidad, por lo que sólo la mencionamos, por ser conocida al igual que las vulnerabilidades Heap Buffer Overflow y Off-By-One Error, que entran en esta categoría.

A nivel de código las vulnerabilidades de tipo Buffer Overflow envuelven comúnmente la violación a las sentencias del programador. Muchas funciones de manipulación de

memoria en lenguajes como C/C++ y sus derivados no ejecutan chequeos de límites y es posible sobrescribir fácilmente los límites asignados para su operación. La combinación de la manipulación de memoria y sentencias equivocadas referentes al tamaño son la principal causa de este tipo de vulnerabilidades.

A nivel de código, una vulnerabilidad de este tipo ocurre cuando una aplicación se basa en datos externos para controlar su comportamiento o depende de las propiedades de los datos que se ejecutan fuera del ámbito inmediato del código, o bien si es tan compleja que un programador no puede predecir con exactitud su comportamiento.

En las aplicaciones web, los atacantes explotan vulnerabilidades de este tipo para corromper la pila de ejecución de las aplicaciones web. Al enviar cuidadosamente datos de entrada a una aplicación web, el atacante puede causar que dicha aplicación ejecute código de una manera arbitraria y así hacerse de una manera efectiva del control del sistema.

Las vulnerabilidades Buffer Overflow pueden estar presentes tanto en el servidor web como en el servidor de aplicaciones, que da soporte de contenido estático o dinámico al sitio o a la aplicación web y pueden plantear un riesgo significativo para los usuarios del servidor web. Cuando las aplicaciones web hacen uso de librerías, se abre la posibilidad a ataques de Buffer Overflow.

La vulnerabilidad también puede encontrarse a nivel de código en las aplicaciones web y suele ser más probable dada la falta de control que se tiene en estas aplicaciones y lo difícil que resulta detectar dichas vulnerabilidades. Aunque son más probables las vulnerabilidades de este tipo en este caso, ya habíamos mencionado que son mucho más difíciles de encontrar y por lo tanto el número de atacantes que intentará encontrar y explotar estas vulnerabilidades será menor.

Casi todos los servidores web, servidores de aplicaciones y entornos de aplicaciones Web son susceptibles a vulnerabilidades de Buffer Overflow, sin embargo, se tiene una notable excepción en los entornos desarrollados con lenguajes como Java, que son inmunes a este tipo de ataques (a excepción de desbordamientos en el intérprete mismo), ya que Java tiene una máquina virtual que cuenta con ciertos niveles de seguridad que impiden que esta vulnerabilidad se presente.

Para terminar con esta vulnerabilidad, revisaremos un ejemplo de esta para comprender su funcionamiento.

El siguiente fragmento de código de ejemplo (en lenguaje C) demuestra que existe una vulnerabilidad de Buffer Overflow que se da a causa de que el código se basa en datos externos para controlar su comportamiento. El código hace uso de la función gets(), que quienes tienen algunos conocimientos del lenguaje de programación C/C++ sabrán que sirve para leer una cantidad arbitraria de datos y guardarlos en una pila. Como no hay forma de limitar la cantidad de datos que esta función lee, la seguridad del código depende de que el usuario siempre ingrese una cantidad menor a la capacidad en tamaño de la pila.

```
char buf[BUFSIZE];

cin >> (buf);
```

Para prevenir este tipo de vulnerabilidades es importante estar informado y al día con los últimos reportes de fallos para nuestro servidor web, para nuestras aplicaciones y para otros productos de nuestra infraestructura de Internet. Realizar una programación segura de nuestras aplicaciones es también muy importante. Hay que aplicar siempre los parches y las actualizaciones de seguridad de nuestros productos de software que por lo regular se encuentran disponibles en la web de nuestros proveedores, pero siempre hay que estar informado si son seguros o no. También es importante escanear y monitorear de manera periódica nuestros servidores y nuestras aplicaciones con las diversas herramientas existentes para la búsqueda de estos fallos de seguridad y en el caso de las aplicaciones, revisar todo el código que acepta datos de usuarios a través de peticiones HTTP y asegurarse que se dispone del tamaño adecuado de *buffer* para almacenarlos. Esto debe hacerse incluso en el caso de entornos que no son sensibles a este tipo de vulnerabilidades.

2.2.-Bases de datos.

Las bases de datos son vulnerables si no se tiene una buena configuración de seguridad y están propensas a experimentar el ataque llamado *SQL Injection* si no se tiene una buena validación en la aplicación con la que interactúa, por lo cual en este subcapítulo únicamente hablaremos acerca de este ataque que explota vulnerabilidades de una programación deficiente en los sistemas.

2.2.1.-SQL Injection.

Un ataque del tipo *SQL Injection* consiste en insertar o inyectar una consulta SQL a través del intercambio de datos entre el cliente y la aplicación. Un ataque de *SQL Injection*, es capaz de leer datos sensibles de la base de datos, modificar los datos de dicha base de datos (Insert, Delete, Update), ejecutar operaciones como administrador, recuperar el contenido de un archivo dado que se encuentra en el Sistema de directorios del Sistema Manejador de Bases de Datos (DBMS) y en algunos casos ejecutar comandos en el sistema operativo. Los ataques de *SQL Injection* son del tipo de ataques de inyección (como *Code Injection* y *Command Injection*).

Como ejemplo de *SQL Injection* tenemos el siguiente: supongamos que tenemos un formulario de autenticación de usuarios que requiere un nombre de usuario y una contraseña. Veamos la siguiente consulta, haciendo uso de PHP:

```
SELECT * FROM usuarios WHERE usuario=$_POST['usuario'] AND
contrasena=$_POST['contrasena'];
```

Ahora bien, si en el formulario de autenticación introducimos como usuario tom y como contraseña ‘’ OR 1 = 1, de tal forma que la consulta sería:

```
SELECT * FROM usuarios WHERE usuario= 'tom' && contrasena = ‘’ OR 1 = 1;
```

Con la consulta anterior, cualquier usuario quedaría autenticado, porque esta consulta siempre regresaría algo diferente de nulo.

Para evitar que nuestras aplicaciones sean vulnerables a un ataque de *SQL Injection*, nunca debemos confiar en la información que el usuario introduce en los formularios, toda esa información debe ser verificada y validada, se recomienda también no construir consultas de SQL de forma dinámica, ya que son susceptibles a ser cambiadas de forma externa, también hay que evitar el uso de cuentas con privilegios administrativos, al igual que se debe evitar proporcionar información innecesaria (mensajes como “contraseña incorrecta” o “usuario incorrecto” no deben ser utilizados, mejor utilizar “usuario y/o contraseña incorrectos”).

El conocer las vulnerabilidades en las aplicaciones web es de suma importancia por el riesgo que implica al servidor si no se encuentra protegido correctamente.

A continuación se dará el resultado del análisis realizado al servidor web y de bases de datos del CDMIT para determinar los activos, las amenazas, las vulnerabilidades y los mecanismos de control, así como las recomendaciones para minimizar los riesgos.

Capítulo 3.- Análisis de riesgos.

Un análisis de riesgos es el proceso de estimar la probabilidad de que ocurra un acontecimiento y la magnitud probable de sus efectos adversos (consecuencias).

La información que se recopiló ha permitido identificar que problemas han existido en el área de cómputo del CDMIT, y permite recomendar los posibles mecanismos que deben aplicarse para contar con una mejor seguridad informática en el CDMIT.

Parte de la información que se ha adquirido se encuentra evidenciada por encuestas realizadas al personal del área de cómputo del CDMIT y desarrolladores de aplicaciones que utilizan el servidor web y de bases de datos.

3.1.- Identificación de los activos.

Se realizó el análisis de riesgos para uno de los activos del Centro de Diseño Mecánico e Innovación Tecnológica (CDMIT): el servidor web y de bases de datos, el cual cuenta con las siguientes características:

Marca	Compaq
Procesador	Intel Pentium III
Arquitectura	i386
Memoria RAM	256 MB
Disco duro	160 GB
Sistema operativo	Fedora Core 4.
Versión del kernel	2.6.17-1.2142_FC4
Servidor web	Apache
Versión del servidor web	2.0.54
Motor de bases de datos	MySQL
Versión del motor de bases de datos	4.1.18
Administrador de MySQL	phpMyAdmin 2.8.0.1
Version de PHP	4.4.2

Tabla 6. Características del servidor web del CDMIT.

Es muy importante el servidor web y de bases de datos del CDMIT porque almacena la información de proyectos, cursos, trabajos, calificaciones y material de apoyo para clases de los profesores, entre otros.

En un principio, el servidor web y de bases de datos era un servidor de pruebas, que después con las necesidades del personal del CDMIT, se convirtió en un servidor web y de bases de datos fundamental. Este es el motivo por el que se requiere identificar que eventos podrían ocurrir, las posibles consecuencias, y el impacto en las actividades cotidianas del CDMIT.

3.2.-Identificación de las amenazas y vulnerabilidades.

Se realizaron una serie de encuestas a los usuarios del servidor web y de bases de datos y también con algunas otras evidencias se obtuvo la siguiente información:

- Existe falta de comunicación entre las áreas del CDMIT, lo cual puede provocar que en caso de que ocurra un incidente no se siga un solo procedimiento para resolverlo y se pueden duplicar actividades en lugar de trabajar conjuntamente para resolverlo.

- El servidor no se encuentra en un lugar restringido.

- No hay una señalización adecuada en el lugar de trabajo que permita a los miembros del CDMIT tomar precauciones al ingresar al área de servidores.

- No se cuenta con clima controlado especial para un área de servidores.

- No se cuenta con un extintor en el área donde está situado el servidor web y de bases de datos.

- No se le da mantenimiento al servidor web y bases de datos, es decir, no se revisan los registros, las bitácoras y aquellas aplicaciones o servicios que hacen vulnerable al mismo, así como la probabilidad de ser atacado.

- Los responsables de cómputo no hacen respaldos de los archivos de configuración de la información.

- El servidor no es seguro, ya que presenta diversas vulnerabilidades que podrían ser explotadas y así comprometer la información de los usuarios.

- No se hacen las actualizaciones de seguridad de los sistemas.

- Los desarrolladores no hacen aplicaciones seguras porque no tienen una metodología de desarrollo que considere la programación de aplicaciones de manera segura.

- Los usuarios no utilizan protocolos seguros para el intercambio de la información en los sistemas, que en algunos casos, puede provocar una pérdida de información importante.

- Los usuarios confían de más en los mecanismos de control de acceso en la entrada del CDMIT.

- No se cuenta con políticas de seguridad propias ni un plan de contingencia en el CDMIT.

- No se contaba con una adecuada infraestructura de red de datos, pero actualmente se está reestructurando.

A continuación se muestra la estructura funcional actual de la sección de cómputo en el CDMIT:

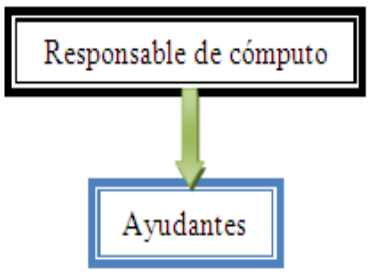

Figura 3.1.- Estructura actual de la sección de cómputo del CDMIT.

Con esta estructura, los ayudantes, quienes pueden tener algunas horas contratadas o son prestadores de servicio social, participan en todas las actividades de cómputo, pero

no hay nadie dedicado a una sola tarea o función, como por ejemplo, ser el administrador del servidor web y de bases de datos, y realizar todas las actividades que requiere el servidor para ser seguro.

3.3.-Determinación del impacto de la ocurrencia de una amenaza.

Si una amenaza se llegara a llevar a cabo, se debe evaluar lo que se puede suscitar y lo que implicaría para la organización tal suceso:

- Si no se tiene un extintor en el área de servidores y se llegara a presentar un incendio, la respuesta del personal del CDMIT sería tardía y esto pudiera provocar una pérdida mayor para la organización, dígase en cuanto a infraestructura e incluso en cuanto a personas.

- El no dar mantenimiento al servidor web y de bases de datos podría provocar que el servidor fuese vulnerable y por lo tanto que alguien explotara las vulnerabilidades y hubiera pérdida, robo o corrupción de información y/o afecte la disponibilidad de los servicios.

- El desconocer que personas ingresan al área de servidores es peligroso, pues si ocurriera la pérdida de algún equipo de cómputo o un daño físico provocado, no se sabría quien fue la persona responsable. Además, se tiene que hacer la limpieza para retirar el polvo que puede causar anomalías en el funcionamiento de los equipos de cómputo y demás dispositivos que se encuentran en el área. No es correcto que cualquier persona de la organización tenga acceso al área de servidores, ya que el acceso de cualquier persona podría provocar que personal ocioso dañara el servidor o hiciera mal uso de la información que en el se resguarda.

- La falta de comunicación entre las áreas puede provocar que en caso de que ocurra un incidente no se siga un solo procedimiento para resolverlo y pueden duplicar actividades en lugar de trabajar conjuntamente para resolverlo.

- Una sola persona no debería de hacer las cosas de varias personas porque nunca terminaría de cubrir todas las necesidades del CDMIT.

- Si se llegará a descomponer el no-break del servidor y hubiera una descarga eléctrica que provocara un daño irreparable al mismo, no sé podría adquirir prontamente otro servidor o estación de trabajo que lo supliera.

- Al no tener una organización de red adecuada, en caso de alguna falla, no sería posible detectarla de inmediato y esto causaría que el CDMIT dejara de realizar sus actividades normales.

- Al no revisar los registros, las bitácoras del servidor o no monitorearlo, si se presentan ataques, estos no serían detectados hasta que ocurriera pérdida de información o falta de disponibilidad en el servicio.

- El realizar aplicaciones con vulnerabilidades en código puede comprometer la información de los usuarios o incluso la operación del servidor web y de bases de datos.

- Se cuenta con un reglamento antiguo y obsoleto, que ya no se respeta, lo que puede provocar que ocurran incidentes de toda índole como mal uso del servidor, de la infraestructura de red o de las instalaciones y esto puede comprometer la información y la disponibilidad de los servicios del servidor.

- Al no contar con un dispositivo de control de temperatura adecuado se puede provocar que el hardware falle por causa de la temperatura ambiente y por lo tanto que los servicios no se encuentren disponibles.

- El no realizar respaldos de la información de los usuarios o de archivos importantes de los sistemas del servidor, existe el riesgo de que al suceder un desastre, esta información se pierda y no se pueda recuperar.

- La falta de señalización puede provocar que los miembros del CDMIT no tengan las precauciones debidas con activos importantes del área de cómputo del centro.

3.4.-Identificación de controles.

A continuación se muestran los controles que actualmente se llevan a cabo en el CDMIT:

- El personal del CDMIT cuenta con la identificación que los acredita como miembros del mismo.

- Los controles de seguridad con los que cuenta el CDMIT son: el acceso por huella digital o autorización en la entrada por parte de las secretarías en su horario de trabajo.

- El administrador asigna las contraseñas a los usuarios en el servidor web y de bases de datos.

- Los usuarios no pueden tener más de una cuenta en el servidor web y de bases de datos.

- Se dan de baja del servidor las cuentas de los usuarios que dejan de pertenecer al CDMIT.

- Los privilegios con los que cuentan los usuarios en el servidor web son: solo pueden leer, ejecutar y escribir en su directorio.

- El servidor web y de bases de datos cuenta con No-Break.

- Se sanciona a los usuarios que hacen uso indebido de sus cuentas en el servidor.

Lo que se puede decir de este análisis de riesgos, es que en el CDMIT existen vulnerabilidades que pueden ser explotadas en el servidor web y de bases de datos y en la propia organización.

Se propone realizar un análisis de riesgos de los demás activos que considere la organización como importantes, hacer las políticas de seguridad y hacer un plan de contingencia que consideré todos los aspectos de seguridad de la organización.

A continuación se presenta una propuesta de un servidor web y de bases de datos que cubra las necesidades del CDMIT con los recursos disponibles.

Capítulo 4.- Recomendaciones para un Servidor web y de bases de datos seguro.

Este capítulo explica las características que un servidor web y de bases de datos seguro debe tener. Esto es esencial para que nuestras aplicaciones web sean seguras, robustas y funcionen de una manera correcta, además de garantizar la amplia disponibilidad de los servicios.

4.1.-Recomendación del Hardware.

El CDMIT cuenta actualmente con un equipo dedicado a servidor web y de bases de datos con las características mencionadas en la tabla 6.

Para que el CDMIT cuente con un servidor web y de bases de datos seguro, se hace la siguiente propuesta en cuanto a hardware:

Marca	Dell
Modelo	Poweredge 1950 III
Procesador	Intel Xeon cuádruple
Memoria RAM	4 GB
Disco Duro	250 GB
Sistema Operativo	Ubuntu 9.04 Server
Servidor Web	Apache 2.0.3
Motor de bases de datos	MySQL 5.1 Community Server
Administrador de MySQL	phpMyAdmin 3.1.3
Lenguaje de scripting	PHP 5.2.9

Tabla 7. Características del hardware propuesto para el CDMIT.

Con base en la comparativa de hardware del capítulo 1, se considera que el producto que cubre mejor las necesidades del CDMIT es el servidor Dell Poweredge 1950 III, ya que a pesar de ser el de mayor precio, tiene mayores ventajas que los otros 2 (HP ProLiant BL260c G5 e IBM x3200).

En comparación con el producto de Hp, cuenta con 3 años de garantía, cuenta con disco duro, cuenta con una mejor capacidad de escalabilidad y mayor capacidad en memoria RAM, aunque es más caro.

Si lo comparamos con el producto de IBM, el precio es prácticamente el mismo, con la ventaja para el producto de Dell, que cuenta con un disco duro, soporta un mayor número de núcleos, tiene una mayor capacidad de escalabilidad, y mayor capacidad de memoria RAM.

4.2.-Selección de Plataformas.

Como resultado del análisis que se realizó de sistemas operativos para servidores, los que se consideran más adecuados para cumplir con los objetivos trazados en este trabajo son:

- Ubuntu Server 9.04.
- Microsoft Windows Server 2003 Web Edition.

Característica.	Microsoft Windows Server 2003 Web Edition.	Ubuntu Server 9.04.
Sistema de Archivos.	NTFS.	Ext3.
Versión Actual.	R2	9.10
Licencia	Propietaria.	GPL.
Precio Promedio.	$399 USD.	Gratuito.
Principales Servicios.	HTTP, FTP, SSH, DNS, DHCP, HTTPS.	HTTP, FTP, SSH, DNS, DHCP, HTTPS.
Versión Estable.	R2	9.04
Sistema de gestión de paquetes.	Windows Installer 3.1.	Aptitude.
Principal ventaja.	Soporte de Microsoft.	Es gratuito.
Principal desventaja.	No es gratuito.	Menor compatibilidad con hardware comparado con Windows.

Tabla 8. Selección de plataformas.

La plataforma que proponemos es Ubuntu Server, ya que Ubuntu es gratuito, de código abierto, robusto y brinda todo el soporte que se requiere. Posee un repositorio de software muy completo y es una versión especializada en servidores. Cuenta con una mejor integración con manejadores de bases de datos de código abierto (MySQL y PostgreSQL) y con el servidor web Apache. Se actualiza continuamente, consume muy pocos recursos, es rápido y muy sencillo de usar.

El servidor web que proponemos utilizar es Apache, ya que es el servidor web más completo, tiene compatibilidad con múltiples plataformas de sistemas operativos, es gratuito, es el servidor web más utilizado en el mundo, cuenta con un gran soporte y tiene muy buena integración con lenguajes de programación y manejadores de bases de datos muy usuales en el CDMIT.

El manejador de bases de datos que consideramos más adecuado para cubrir los objetivos del trabajo es MySQL, ya que hace un buen complemento con Linux, Apache y PHP (LAMP), es soportado por Sun Microsystems, además de ser el manejador de bases de datos que se utiliza en el CDMIT, también es posible ejecutarlo en máquinas con bajos recursos y la versión gratuita es ideal para bases de datos de tamaño mediano.

4.3.-Instalación y configuración de las plataformas.

Como lo hemos mencionado anteriormente, el sistema operativo que se propone utilizar es Ubuntu Server 9.04 con el servidor web Apache 2.0.3 y el manejador de bases de datos MySQL 5.1 Community Server.

Se recomienda que al realizar la instalación del sistema operativo, se instale sólo el software mínimo necesario para su inicio y que posteriormente se instalen los paquetes de software que se requieran para prestar los servicios que se desean proporcionar. Esta medida ayudará a que el funcionamiento del servidor sea eficiente, ya que no se

desperdiciarán recursos en servicios que no se utilizan y se reducirán vulnerabilidades que pudieran conllevar estos servicios no utilizados.

En la sección A del anexo de este trabajo se muestra la propuesta de cómo instalar el sistema operativo con el software mínimo necesario para su inicio.

Recomendamos la instalación de los siguientes paquetes de software:

- Cliente y servidor OpenSSH.
- Servidor web Apache 2.
- Cliente y servidor MySQL Community 5.1.
- Módulo MySQL para Apache.
- PHP 5.2.9.
- Módulo de PHP para Apache.
- OpenSSL.
- Módulo SSL para Apache.
- phpMyAdmin 3.1.3

Además recomendamos la habilitación del Firewall de Ubuntu, el cuál viene instalado por defecto con el sistema operativo, pero se encuentra desactivado.

Con la instalación de estos paquetes de software, pretendemos tener los servicios web y de bases de datos, además de contar con soporte para PHP y HTTPS. El servidor SSH nos permitirá administrar al servidor web de manera remota y el paquete phpMyAdmin nos permitirá administrar nuestras bases de datos remotamente por medio de una interfaz web.

El procedimiento para la instalación y configuración de estos paquetes de software se encuentra detallado en el anexo de este trabajo.

4.4.-Recomendaciones para la administración del servidor web y de bases de datos.

Es recomendable para un buen funcionamiento del servidor web y de bases de datos realizar las siguientes tareas:

- Realizar respaldos continuos de los archivos importantes en el servidor y almacenarlos en un lugar seguro.

- Actualizar de manera periódica los programas críticos y el sistema operativo con los parches de seguridad más actuales disponibles.

- Crear cuentas de usuario con privilegios mínimos necesarios.

- Llevar un control de accesos y uno de sucesos en el servidor web: un *Access Log* (Control de accesos) y un *Activity Log* (Control de sucesos).

- Llevar un control de las acciones de los usuarios, para saber si realizan tareas acordes a los fines para los que funciona el servidor o no.

- Revisar continuamente los *logs* del sistema para detectar problemas y darles una solución.

- Utilizar software especializado para el monitoreo del servidor y la detección de *exploits* o programas que comprometan la seguridad del servidor web y de bases de datos.

- Revisar el estado del hardware, para detectar anomalías y evitar una posible suspensión del servicio.

- Si se desean añadir servicios al servidor, hay que verificar que se instalen sólo los paquetes necesarios.

- Se debe procurar el uso de servidores dedicados en actividades críticas, para evitar un daño de grado mayor en caso de un incidente.

- Utilizar el protocolo seguro (HTTPS) cuando se maneje el intercambio de información confidencial a través de la red.

4.5.- Recomendaciones de seguridad para aplicaciones que utilizan bases de datos.

Aquí presentamos algunos puntos a tomar en cuenta para que las aplicaciones hospedadas en el servidor web del CDMIT se encuentren seguras:

- Si se hace uso de contraseñas, éstas deberán almacenarse cifradas en la base de datos.

- Al realizar consultas para obtener datos, se deberán traer solamente los datos que se necesitan y evitar en lo posible el uso del comodín (*).

- La longitud de valores que se le asigna a cada campo de una base de datos debe ser la necesaria y se debe validar desde el código de la aplicación que ésta no se exceda.

- Se deberá llevar un registro de accesos a las aplicaciones, con los datos de quien accede, como accede, IP, hora y fecha.

- Todo el código que se comunique con la base de datos deberá ser filtrado.

- No se deberán referenciar archivos con una ruta absoluta, en especial si se comunican con la base de datos.

- La información que se obtiene de la base de datos o que se inserta en la base de datos deberá ser enviada por el método POST o bien si se hace uso del método GET, la información deberá ir cifrada.

- No se recomienda el uso del método REQUEST, ya que es un método genérico que puede recibir valores por POST, GET o desde las COOKIES.

4.6.- Recomendaciones generales.

En esta parte se presentan una serie de recomendaciones importantes de carácter general que sería provechoso llevar a cabo para tener un servidor web y de bases de datos seguro:

A continuación se muestra una propuesta para la sección de cómputo en el CDMIT:

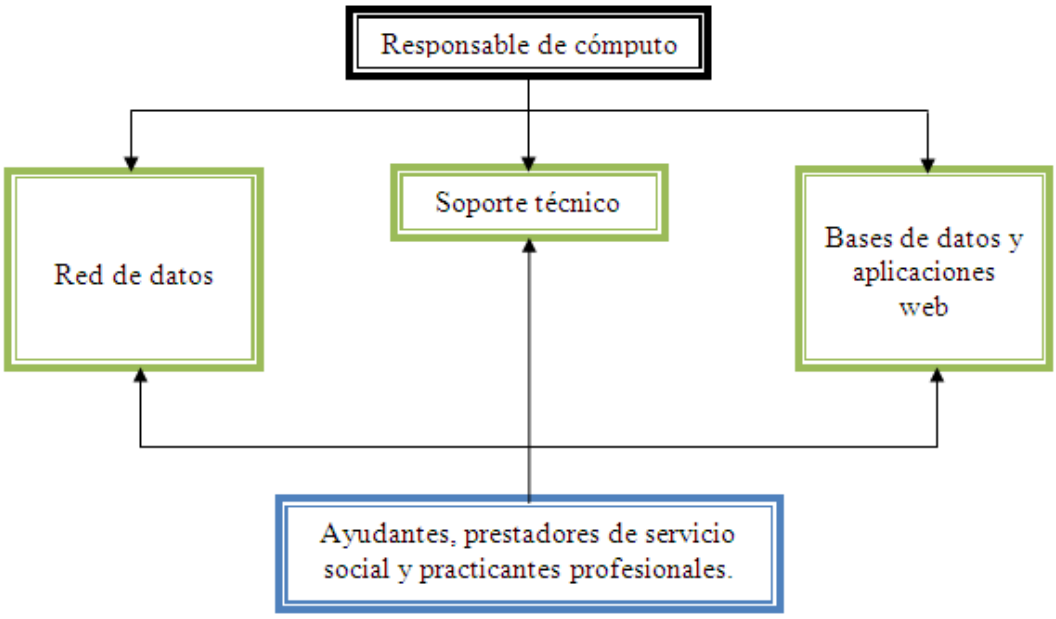

Figura 4. 1.-Estructura propuesta para la sección de cómputo del CDMIT.

Se propone que dentro de la sección de cómputo del CDMIT existan tres grupos:

Red de datos: Grupo donde deberá existir un responsable o administrador de red, quien supervisará el buen funcionamiento de la misma, las configuraciones de los equipos que la integran, así como resolver los incidentes de seguridad que se llegaran a presentar.

Soporte técnico: Grupo que se encargará de dar mantenimiento preventivo y correctivo menor a los equipos de cómputo, así como apoyar en instalaciones y configuraciones de software y hardware.

Bases de datos y aplicaciones web: Grupo donde deberá existir un responsable o administrador del servidor web y de bases de datos, responsable de las configuraciones, adecuaciones y respaldos.

Con esta estructura, el responsable de la sección de cómputo del CDMIT podrá delegar responsabilidades y responderá a las necesidades del CDMIT de una manera más adecuada.

Otras recomendaciones que se hacen son:

- Hacer la limpieza de la sala de cómputo frecuentemente.
- Asignar a una persona para realizar el mantenimiento del servidor web y de bases de datos.
- Elaborar una guía para los usuarios sobre la buena programación de aplicaciones.
- Delegar la responsabilidad a varias personas para administrar la seguridad en el CDMIT.
- Asignar a alguien que se encargue de difundir las vulnerabilidades y los riesgos que encontramos en Internet a través de artículos, cursos, etcétera sobre la seguridad de la información.
- Crear una cultura en seguridad en los miembros de la organización.
- Se necesita más personal en el área de cómputo para el apoyo de problemas en los equipos o en las redes de datos, que pueden ser becarios, prestadores de servicio social o prácticas profesionales.
- Delegar las responsabilidades a las personas que ayudan al área de cómputo dependiendo de la dificultad de estas, así como de su capacidad para realizarlas.
- Elaborar Políticas de Seguridad propias del CDMIT cumpliendo con las de la Facultad de Ingeniería, así como asignar a alguien que verifique su cumplimiento.
- Elaborar un Plan de Contingencia.
- Informar a los miembros de la organización sobre las personas que prestarán servicios y/o apoyaran al área de cómputo y avisar cuando la persona ya no preste servicios.
- Dar capacitación a las personas que prestan su servicio en el área de cómputo para que el área siga con su mejor funcionamiento.
- Hacer documentación de las configuraciones que se le hacen a los servidores.
- Quitar los dispositivos de la sala de cómputo que no se estén utilizando.

- Reportar cualquier anomalía en la red.

- Verificar que los dispositivos funcionen correctamente, es decir, no-break, router, switch, etcétera.

- Hacer un registro de las personas que visitan el CDMIT, para evitar algún incidente.

- Resguardar muy bien las claves de administrador, para evitar que alguna persona ajena al Centro haga mal uso de estas.

- Asignar una persona que verifique el cumplimiento del reglamento interno del CDMIT.

- Toda la información obsoleta debe ser destruida, de lo contrario, debe ser archivada en un lugar seguro.

- Conocer donde se encuentran los discos de respaldos del software que se utilice en el CDMIT.

- Dejar a alguien encargado en el CDMIT en la hora de la comida para mantener el control de acceso.

Se recomienda la existencia de un administrador de redes de datos, un administrador de bases de datos y aplicaciones web, una persona del soporte técnico, así como ayudantes, prestadores de servicio social y los que realizan prácticas profesionales.

Una vez hecho el análisis de riesgos la información obtenida servirá para realizar las políticas de seguridad para el servidor web y de bases de datos que se tuvo como objeto de estudio, así como, las recomendaciones para garantizar la protección del mismo.

Capítulo 5.- Propuesta de Políticas de Seguridad.

En este capítulo se presentan una serie de políticas de seguridad, propuestas para el uso y la operación del servidor web y de bases de datos del CDMIT. Dichas políticas han sido diseñadas tomando en cuenta las políticas de seguridad en cómputo de la Facultad de Ingeniería y con base en un análisis de riesgos realizado en el CDMIT acerca de la situación actual del servidor web y de bases de datos perteneciente al Centro. Dicho análisis ha considerado aspectos como la infraestructura, la configuración del servidor, los hábitos de los usuarios y algunos otros aspectos importantes que nos permitieron hacer una propuesta adecuada al CDMIT.

5.1.-Políticas de seguridad física.

a) Restringir el acceso a la sala de servidores a personal no autorizado.

b) La sala de servidores deberá contar con puertas cerradas con chapas y con ventanas selladas.

c) Queda prohibido introducir y consumir alimentos y bebidas a la sala de servidores.

d) Queda prohibido fumar en la sala de servidores.

e) Hacer uso de Reguladores y/o No-Breaks para proteger a los servidores de fallas eléctricas que puedan causar un daño físico.

f) En la sala de servidores deberá haber un extintor visible y listo para ser utilizado en caso de incendio.

g) Deberá mantenerse una temperatura adecuada en la sala de servidores para la correcta operación de estos.

h) Cuando se realice limpieza en la sala de servidores, se deberá hacer bajo la supervisión de una persona autorizada y capacitada.

5.2.-Políticas de cuentas.

i) Las cuentas de usuario en los servidores serán únicamente otorgadas a miembros de la DIMEI o del CDMIT o en su defecto a personas que las requieran para realizar una labor acorde dichas organizaciones.

j) Las cuentas de usuario en servidores serán únicamente asignadas por un administrador autorizado.

k) Las cuentas de usuario contarán con los privilegios suficientes para realizar la actividad para la cual fueron creadas.

l) Las cuentas de usuario que se encuentren inactivas deberán ser dadas de baja por el administrador.

m) Las cuentas de usuario son personales e intransferibles.

n) Las cuentas de usuario serán utilizadas únicamente para fines académicos y de investigación y no con otros fines diferentes al giro del CDMIT.

5.3.-Políticas de contraseñas.

o) Las contraseñas de las cuentas son asignadas únicamente por el administrador de los servidores.

p) El usuario puede solicitar elegir su contraseña, siempre y cuando esta cumpla con los criterios para contraseñas robustas y esta solicitud tenga el visto bueno del administrador.

q) Las contraseñas deben ser robustas, es decir: contarán con al menos 8 caracteres, que contengan combinaciones de caracteres numéricos, alfanuméricos y símbolos y se deberá evitar que sean palabras de diccionario.

r) Las contraseñas deben ser cambiadas al menos cada semestre, excepto las de administrador que deberán ser cambiadas cada 3 meses.

5.4.-Políticas de control de acceso (lógico y físico).

s) El acceso remoto a los servidores se hace utilizando el protocolo SSH exclusivamente.

t) Queda restringido el acceso remoto por medio de protocolos como RSH, FTP, Telnet y aquellos protocolos que se consideren inseguros o que no manejen conexiones cifradas y de los cuales existan vulnerabilidades conocidas.

u) Cualquier usuario debe autenticarse con su cuenta y queda prohibido hacer uso de sesiones activas de otros usuarios.

v) El acceso físico al área de servidores sólo se encuentra permitido para personal autorizado del área de cómputo del CDMIT.

5.5.-Políticas de uso adecuado.

w) La información que un usuario suba al servidor web debe ser acorde a los propósitos con los que fue creada la cuenta de usuario y por ningún motivo deberá ser utilizada con otros fines.

x) La información que un usuario suba al servidor web debe encontrarse libre de virus, por lo que el usuario deberá asegurarse de que su información se encuentra limpia.

y) Queda prohibido a los usuarios la instalación de paquetes en equipo en el cual reside el servidor web, por lo que de requerir algún programa en especial, deben

solicitarlo al administrador del sistema, quien evaluará si es procedente la petición del usuario.

5.6.-Políticas de mantenimiento.

z) El administrador debe actualizar el sistema operativo y el software necesario con las últimas actualizaciones de seguridad para reducir vulnerabilidades.

aa) Se debe hacer uso de herramientas de seguridad, para encontrar posibles amenazas dentro del sistema (*exploits*, virus, códigos malignos, etc.).

bb) Se debe hacer un monitoreo permanente del servidor web, en donde se detallen los puertos que se encuentran abiertos, los servicios que se están ejecutando y la información de los hosts remotos que están accediendo a los servicios.

cc) Se deben construir bitácoras con toda la información de los cambios de configuración que se han hecho en el sistema a fin de tener un control que permita identificar el origen de posibles contingencias.

dd) Se debe hacer una revisión periódica del estado del hardware, para evitar que una falla de este tipo pueda provocar la no disponibilidad del servicio.

5.7.-Políticas de respaldos.

ee) Cada usuario es responsable de la integridad de su información, por lo que es el único al que le concierne hacer respaldos de tal información.

ff) El administrador debe hacer respaldos, por lo menos semestralmente, de los servidores web del CDMIT (información, configuración y archivos que se consideren importantes).

gg) Los respaldos hechos por parte del administrador deben almacenarse en medios resistentes y confiables, tales como cintas, servidores de respaldos u otros que se consideren convenientes.

5.8.-Sanciones.

Las sanciones que se proponen para cuando se hace uso indebido son las siguientes:

I. Uso de cuentas de usuario ajenas: Primera vez, amonestación. Reincidencia, cancelación de las cuentas de usuario de quien resulte responsable.

II. Cambio de configuración del servidor: Cancelación de la cuenta del usuario.

III. Instalación no autorizada de paquetes: Cancelación de la cuenta de usuario.

IV. Escalada de privilegios: Cancelación de la cuenta de usuario.

V. Uso de servicios con fines no acordes a las actividades del CDMIT: Cancelación de la cuenta de usuario y consignación al tribunal universitario si la falta se considera grave.

VI. Subir archivos infectados por virus: Reducción de privilegios.

VII. Violación de las políticas o reglamentos de cómputo del CDMIT: Amonestación, cancelación de la cuenta de usuario o consignación al tribunal universitario según la gravedad de la falta.

VIII. Ejecución de programas que intenten adivinar contraseñas de otros usuarios o del sistema: Primera vez, amonestación. Reincidencia, cancelación de la cuenta de usuario.

IX. Ejecución de programas que intenten explotar o encontrar vulnerabilidades en el sistema: Cancelación de la cuenta de usuario.

X. Daño de cualquier índole al sistema: Cancelación de la cuenta de usuario y consignación al tribunal universitario según la gravedad de la falta, así como el pago por los daños causados.

Con estas políticas de seguridad se mantendrá seguro el servidor reduciendo los riesgos y las amenazas que hoy en día se hacen más frecuentes y más fuertes por la falta de medidas y controles dentro del CDMIT.

Conclusiones.

El presente trabajo ha llevado a cabo una investigación con base en una serie de requerimientos propuestos por el CDMIT, y cubriendo sus necesidades más importantes en cuanto a la seguridad de su servidor web y de bases de datos.

Es de suma importancia para la organización el contar con un servidor web y de bases de datos robusto y seguro, con una amplia disponibilidad por el uso que se le da, ya que tal servidor hospeda trabajos de investigación y sistemas de estudiantes de licenciatura, posgrado, profesores, prestadores de servicio social y practicantes.

Con los mecanismos de seguridad implementados actualmente en el servidor web y de bases de datos del CDMIT, resulta complicado tener una robustez en cuanto a seguridad, ya que el servidor se encuentra expuesto a amenazas que podrían comprometer la información y la disponibilidad de los servicios.

Al día de hoy se han cubierto todos los objetivos propuestos en este trabajo, ya que se han elaborado propuestas de las medidas de seguridad a llevar a cabo por parte de los administradores del servidor web y de bases de datos del CDMIT, con la finalidad de salvaguardar la información que se encuentra en el servidor y garantizar la disponibilidad de los servicios.

Además se ha elaborado una guía básica de recomendaciones para que las aplicaciones web y de bases de datos que utilizan los usuarios cumplan con una serie de requisitos mínimos.

También se ha hecho una propuesta tecnológica completa que considera aspectos de Hardware y sistema operativo apropiados para servidores, además del software necesario para prestar los servicios de web y bases de datos.

Por último se elaboró una propuesta de políticas de seguridad para el servidor web y de bases de datos, la cual contempla aspectos de seguridad física, de cuentas de usuario, de contraseñas, de control de acceso físico y lógico, de uso adecuado, de mantenimiento, de respaldos y sanciones.

Anexo.

A.-Instalación de Ubuntu 9.04 Server.

El primer paso que se debe dar para tener un servidor web seguro es la instalación del sistema operativo, en nuestro caso será Ubuntu 9.04 Server. Este sistema operativo se puede descargar de la página oficial de Ubuntu o bien se puede pedir por correo postal en la misma página.

Una vez que se cuenta con un CD de instalación de Ubuntu 9.04 Server, para instalarlo hay que poner en la unidad de CD ROM el disco de instalación y reiniciar el equipo con el CD dentro. Si la BIOS del equipo se encuentra configurada para que el sistema arranque desde el CD, se accederá a la pantalla principal de la instalación, si no, se deberá configurar la BIOS para ello.

La pantalla principal que aparece al iniciar la instalación, es la siguiente:

Figura A1.-Pantalla de instalación de Ubuntu 9.04 Server.

Aquí se seleccionará la opción *Install Ubuntu Server*, luego de esto aparecerán pantallas de configuración en donde se podrán elegir el idioma de instalación, el idioma del sistema operativo, el nombre de la máquina y la hora. Después de esto aparecerá un asistente que nos permitirá particionar el disco duro para la instalación del sistema operativo.

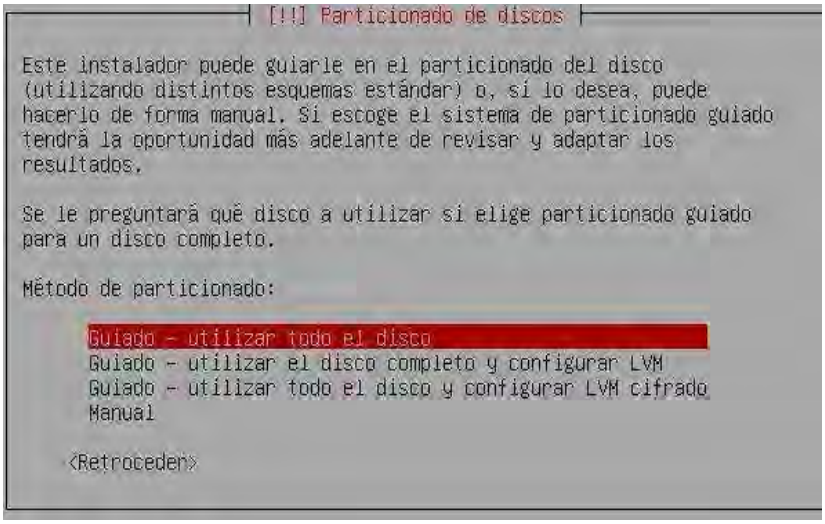

Figura A2.-Asistente de particionado de Ubuntu 9.04 Server.

En nuestro caso utilizaremos el particionado guiado. Al seleccionar esta opción, el asistente de instalación nos pedirá seleccionar el disco duro en el que deseamos instalar Ubuntu 9.04 Server, se seleccionará el disco duro y el sistema propondrá una estructura de particionado, la cual nosotros aceptaremos.

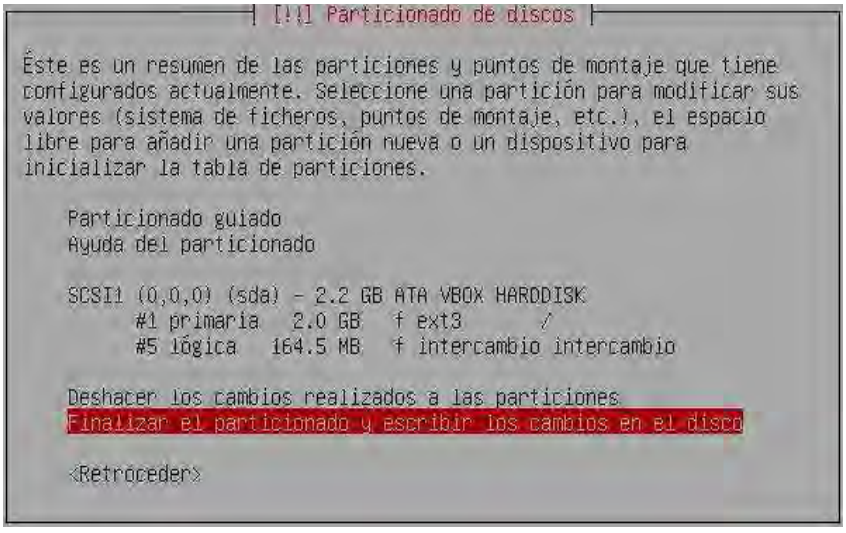

Figura A3.-Creación de particiones para Ubuntu 9.04 Server.

Después de seleccionar la opción de finalizar el particionado, aparecerá una ventana de confirmación, en la que se seleccionará aceptar. Posterior a esto, el disco será formateado y se iniciará la instalación de Ubuntu 9.04 Server.

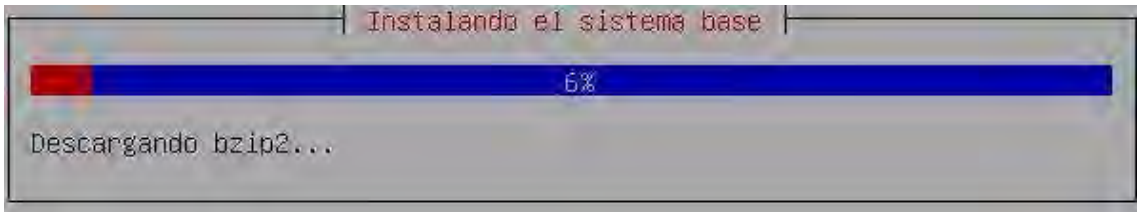

Figura A4.-Proceso de instalación de Ubuntu 9.04 Server.

Al terminar el proceso de instalación, se nos pedirá crear una cuenta de usuario, en donde se requerirá escribir el nombre completo del usuario, el nombre de usuario para la cuenta y la contraseña de la cuenta en dos ocasiones. Finalmente aparecerá una ventana en la que se nos permitirá seleccionar algunos paquetes de software a instalar. En dicha ventana no se seleccionará paquete alguno y se elegirá la opción continuar. El asistente de instalación dará aviso de que la instalación ha terminado y el equipo será reiniciado. Con esto habrá terminado la instalación de Ubuntu 9.04 Server.

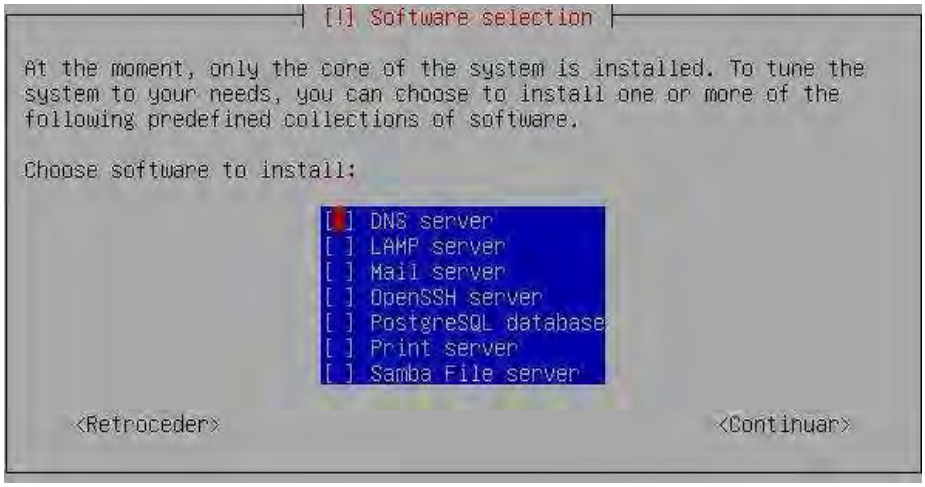

Figura A5.-Selección de software de Ubuntu 9.04 Server.

B.-Instalación de OpenSSH utilizando aptitude.

Para poder administrar de manera remota nuestro servidor web o nuestras cuentas, es necesario hacer uso de algún protocolo que nos permita hacerlo (Telnet, RSH, FTP), en nuestro caso hemos optado por utilizar el protocolo SSH que es una buena alternativa.

Como nuestro sistema operativo no cuenta con este servicio, se deberá instalar, para ello utilizaremos un paquete de código abierto llamado OpenSSH, que cuenta con un cliente y con un servidor SSH. Esto lo haremos con la ayuda del gestor de paquetes de Ubuntu *aptitude*.

Primero se deberá iniciar sesión y luego de ello ejecutar una serie de comandos como superusuario.

Para instalar el cliente SSH, deberemos teclear el siguiente comando:

```
sudo aptitude install openssh-client
```

Con esto el paquete será descargado e instalado en el sistema.

Lo siguiente será instalar el servidor, como sigue:

```
sudo aptitude install openssh-server
```

Y por último se reiniciará el servicio:

```
sudo /etc/init.d/ssh restart
```

Para verificar que el servicio SSH funciona correctamente, nos conectaremos de manera remota al servidor.

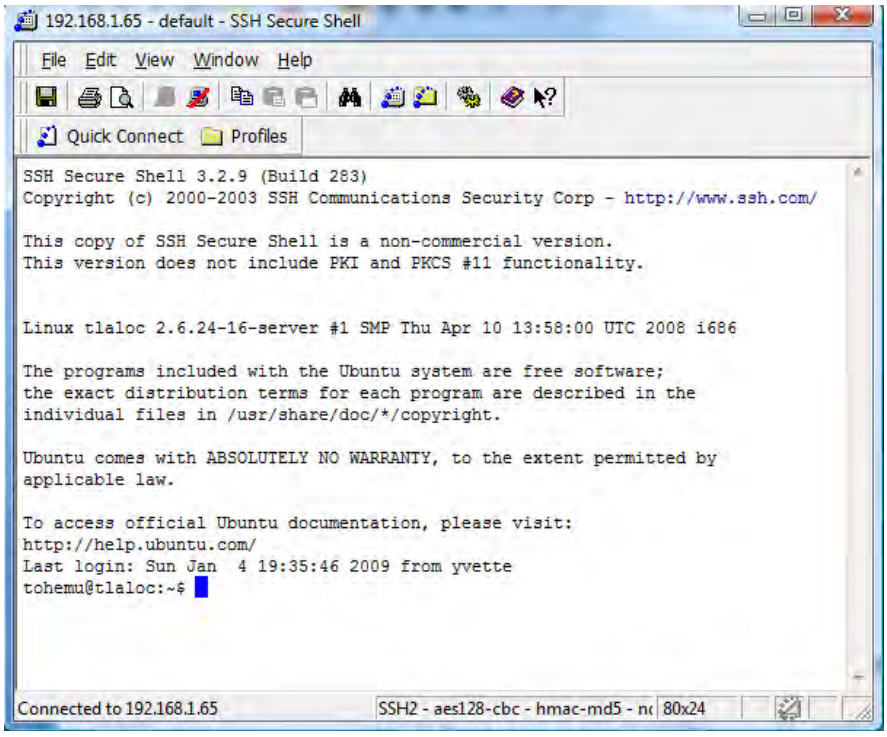

Figura A6.-Conexión remota al servidor por SSH.

C.-Instalación de Apache utilizando aptitude.

Para instalar Apache, deberemos descargarlo por medio de la herramienta de gestión de paquetes con la que cuenta nuestra distribución de Linux (Ubuntu 9.04). Basta con abrir una terminal y teclear el comando que se muestra (como superusuario, utilizando el comando sudo):

```
sudo aptitude install apache2
```

Cuando haya finalizado la descarga e instalación de Apache, se deberá reiniciar el servicio para que nuestro servidor web entre en funcionamiento:

```
sudo /etc/init.d/apache2 restart
```

Una vez terminada la instalación, el sistema nos dará aviso y para comprobar que nuestro servidor web funciona correctamente hay que abrir un navegador y teclear la IP de nuestro servidor. Si todo se hizo correctamente aparecerá en el navegador una pantalla como la que sigue:

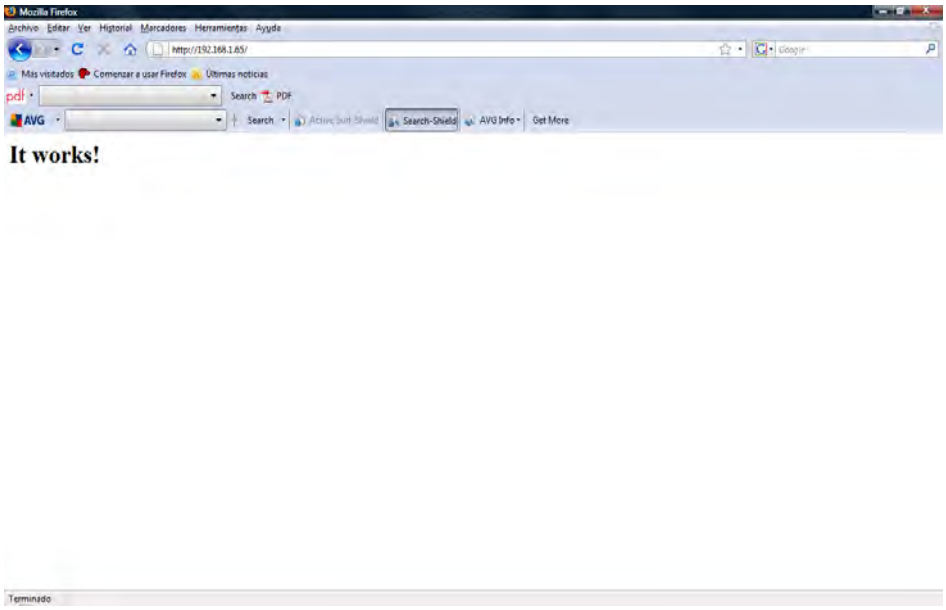

Figura A7.-Comprobación del Servidor web Apache.

D.-Instalación de MySQL utilizando aptitude.

En esta sección instalaremos el sistema manejador de bases de datos MySQL en nuestro servidor, para ello haremos uso del gestor de paquetes de Ubuntu 9.04 Server. Como en

el caso de OpenSSH, aquí también requerimos de un cliente y un servidor. El procedimiento es como sigue:

Primero descargaremos e instalaremos el cliente:

```
sudo aptitude install mysql-client
```

Seguido de ello instalaremos el servidor:

```
sudo aptitude install mysql-server
```

Cuando la instalación haya terminado, el sistema solicitará que se ingrese la contraseña de *root* en dos ocasiones.

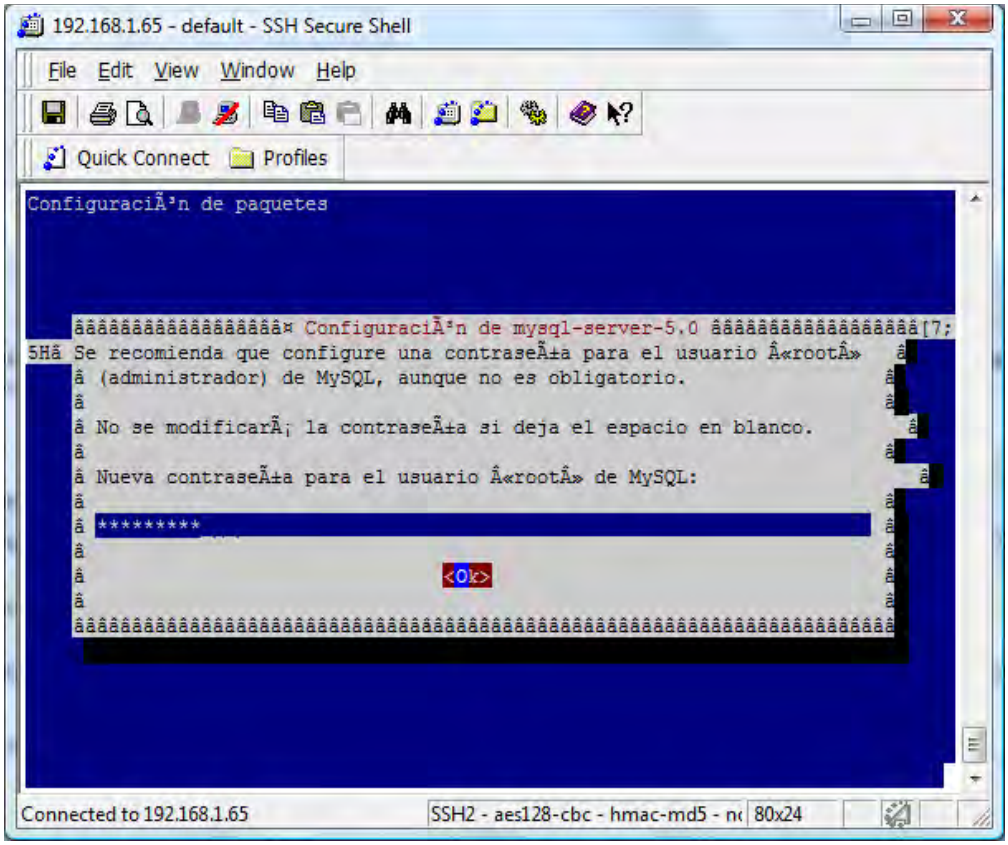

Figura A8.-Elección de contraseña de administrador para MySQL.

Posteriormente reiniciaremos el servicio:

```
sudo /etc/init.d/mysql restart
```

Para comprobar que MySQL se ha instalado correctamente en nuestro servidor, iniciaremos sesión remotamente en el servidor MySQL y nos será mostrada la consola del servidor.

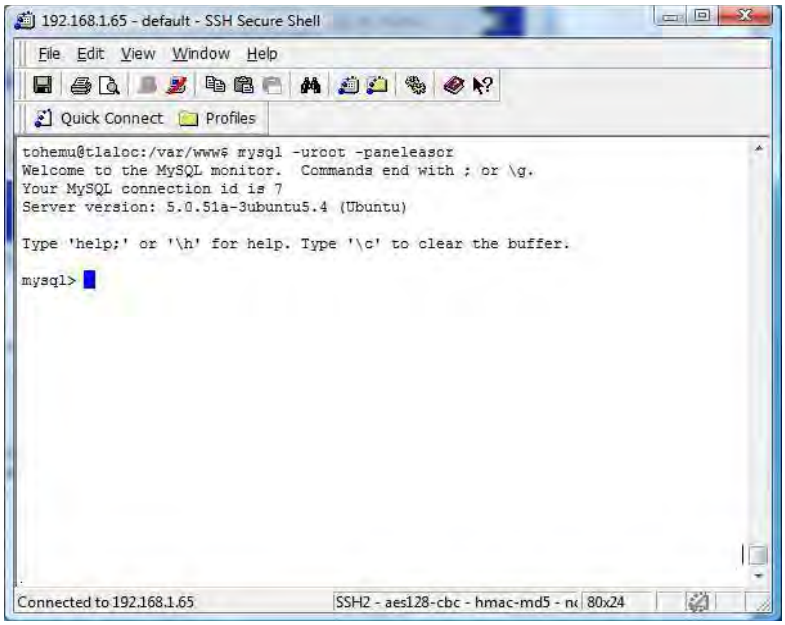

Figura A9.-Inicio de sesión en el servidor MySQL.

E.-Instalación de PHP utilizando aptitude.

Lo siguiente que haremos es instalar el soporte para PHP como módulo de Apache, para ello haremos uso del gestor de paquetes de Ubuntu 9.04 Server de nueva cuenta.

Para descargar e instalar este paquete, se tecleará el siguiente comando en consola:

```
sudo aptitude install php5
```

Seguido de esto debemos instalar otros dos paquetes que nos permitirán ligar a PHP con MySQL y con Apache. Esto se hace con los siguientes comandos:

```
sudo aptitude install libapache2-mod-auth-mysql
sudo aptitude install php5-mysql
```

Lo siguiente es reiniciar Apache:

```
sudo /etc/init.d/apache2 restart
```

Y por último verificaremos que PHP funciona, escribiendo un *script* sencillo en PHP y corriéndolo desde un navegador web.

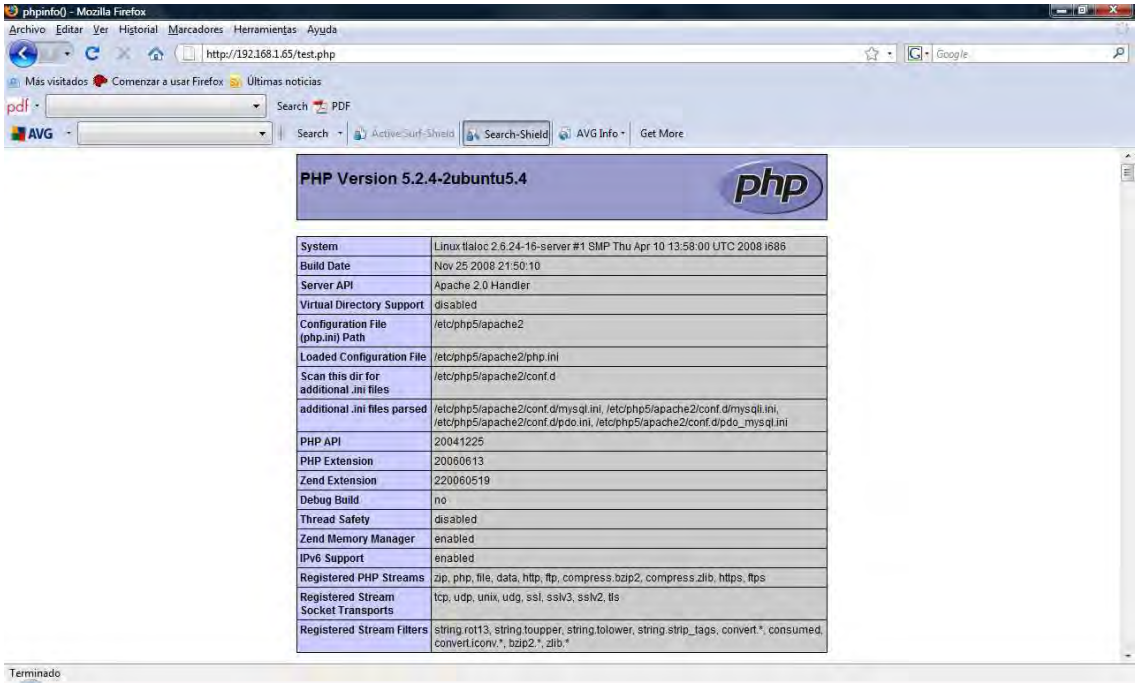

Figura A10.-Prueba de funcionamiento de PHP.

F.-Instalación de phpMyAdmin utilizando aptitude.

En esta parte instalaremos una interfaz web escrita en PHP que nos permitirá administrar nuestras bases de datos de manera remota, se trata de phpMyAdmin. Para utilizarlo se deben tener PHP y MySQL funcionando.

Primero se debe descargar este paquete:

```
sudo aptitude install phpmyadmin
```

Cuando se ejecute este comando, se nos pedirá la autorización para descargar e instalar el paquete, seleccionamos aceptar. Cuando termine de instalarse, un asistente de configuración aparecerá, en el cual se nos preguntará el servidor web que utilizamos, seleccionamos apache 2 y pulsamos Enter.

Lo que sigue es crear una liga simbólica al directorio web:

```
sudo ln -s /usr/share/phpmyadmin /var/www/
```

Posteriormente a esto reiniciamos Apache:

```
sudo /etc/init.d/apache2 restart
```

Posteriormente abrimos un navegador en cualquier cliente web y tecleamos la dirección URL: http://IP_SERVIDOR/phpmyadmin/ y verificamos que phpmyadmin se encuentra funcionando.

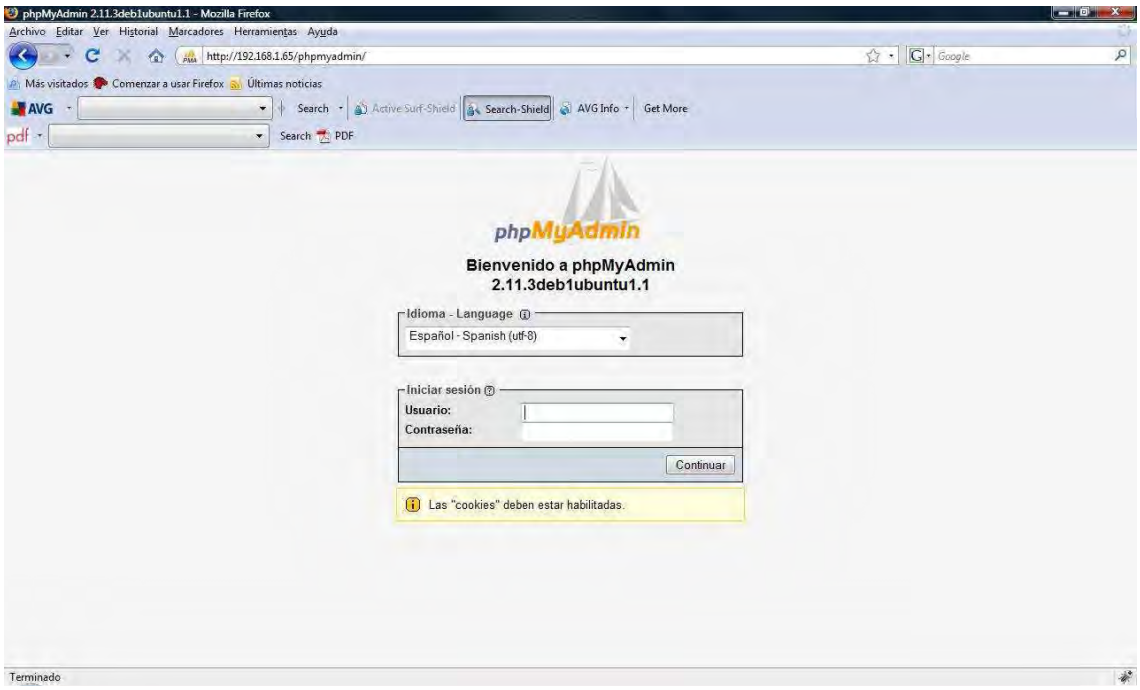

Figura A11.-Cliente MySQL phpmyadmin.

G.-Configuración de UFW.

Lo que haremos en esta parte será configurar una utilería con la que cuenta el sistema operativo Ubuntu 9.04 Server, esta utilería es UFW (*Uncomplicated Firewall*), que es un firewall que viene por defecto con este sistema operativo y es bastante sencillo de utilizar. UFW se basa en *iptables* y se utiliza principalmente como firewall basado en un *host*.

Como UFW ya viene instalado por defecto en nuestro sistema operativo, sólo hay que activarlo, ya que inicialmente viene desactivado. Para activar este firewall se debe utilizar el siguiente comando:

```
sudo ufw enable
```

Una vez que el firewall ha sido activado, aparecerá el mensaje que nos dará aviso de ello.

Ahora que nuestro firewall ha sido activado, nos daremos a la tarea de restringir los puertos de los servicios que no utilizaremos y sólo dejar abiertos los puertos para los servicios que nos interesan. Estos puertos son: 80-http, 22-ssh, 3306-MySQL y el puerto 443-https.

Primero que nada, denegaremos todas las conexiones:

```
sudo ufw default deny
```

Ahora deberemos abrir los puertos de los servicios que son de nuestro interés, en este caso: http, ssh, MySQL y https:

```
sudo ufw allow 80/tcp
sudo ufw allow 80/udp
sudo ufw allow 22/tcp
sudo ufw allow 22/udp
sudo ufw allow 3306/tcp
sudo ufw allow 3306/udp
sudo ufw allow 443/tcp
sudo ufw allow 443/udp
```

Luego de esto reiniciamos el firewall para que los cambios surtan efecto:

```
sudo ufw disable
sudo ufw enable
```

Para verificar el status del firewall, tecleamos el siguiente comando:

```
sudo ufw status
```

Con el comando anterior, se mostrará en pantalla información de los puertos que se encuentran abiertos en nuestro servidor.

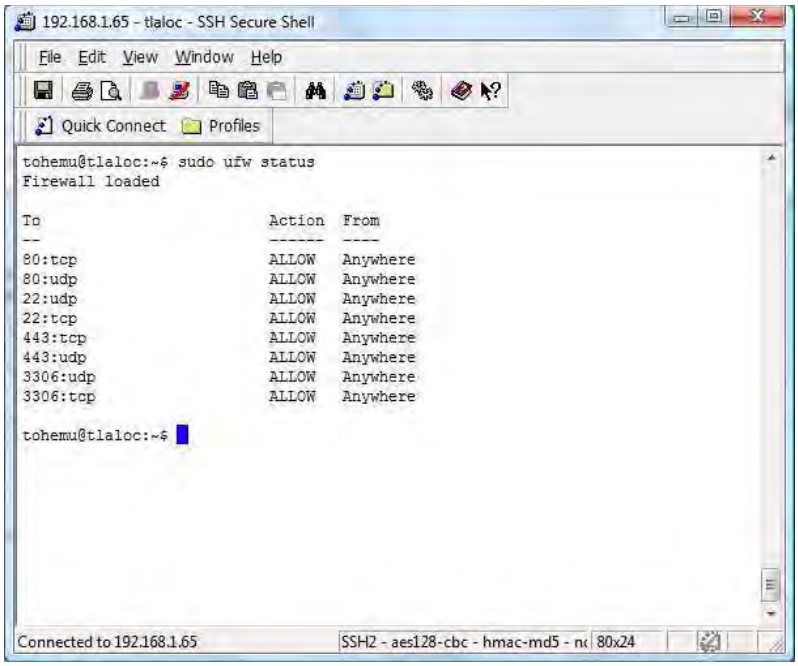

Figura A12.-Puertos permitidos por el Firewall.

Se han realizado una serie de pruebas que nos permitieron verificar el correcto funcionamiento de nuestro firewall, estas pruebas consistieron en cerrar y abrir puertos, verificando que los servicios se encontraban disponibles o negados.

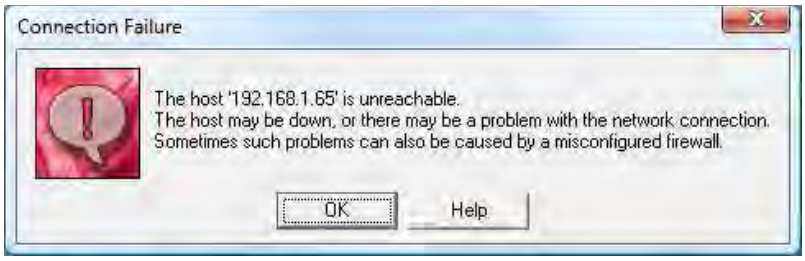

Figura A13.-Puerto 22 cerrado por el Firewall.

H.-Configuración segura de Apache por medio de HTTPS.

En esta parte haremos la configuración de nuestro servidor web para que se encuentre configurado de una manera más segura, instalaremos y configuraremos el módulo SSL/TLS para Apache (mod_ssl), que nos brindará soporte para HTTPS. Dicho módulo es de gran importancia, ya que le añade a nuestro servidor web la capacidad de cifrar los datos que intercambia y procesa, haciendo uso de algoritmos de cifrado como 3DES, RSA y otros.

Para instalar este módulo en nuestra distribución Linux (Ubuntu 9.04 Server), hay que teclear los siguientes comandos:

```
sudo aptitude install apache2 libapache-mod-ssl
sudo aptitude install openssl
```

Una vez instalado este módulo, se debe habilitar para su uso:

```
sudo a2enmod ssl
```

Después de esto, es posible generar un CRS (*Certificate Signing Request*), para lo cuál debemos crear nuestra propia clave, dicha clave se crea con el siguiente comando:

```
sudo openssl genrsa -des3 -out server.key 1024
```

Con el comando anterior estamos generando una clave privada RSA de 1024 bits para el servidor, dicha clave se guardará en el archivo server.key. Después de teclear el comando anterior, se nos pedirá una contraseña (debe ser robusta, por lo menos 8 caracteres, aunque cuando se especifica el argumento –des3, permite elegir de 4 caracteres en adelante), se la proporcionamos y con esto hemos terminado de crear la clave privada.

Lo siguiente es generar el CRS y para ello utilizaremos el siguiente comando:

```
sudo openssl req -new -key server.key -out server.crs
```

Después de teclear este comando se nos pedirá la contraseña que elegimos cuando creamos la clave privada y además se nos pedirán una serie de datos como el nombre de la compañía, la ciudad, el estado, el país, nuestro nombre, correo electrónico, etc. Luego de esto, el CRS se habrá creado y se habrá almacenado en el archivo server.crs.

Una vez que creamos nuestro CRS, deberemos crear un certificado autofirmado utilizando el CRS, para lo cual el siguiente comando es el apropiado:

```
sudo openssl x509 -req -days 3650 -in server.crs -signkey server.key -out server.crt
```

Aquí estamos especificando que el certificado autofirmado es un certificado del tipo x509, con una vigencia de 10 años (3650 días). Este certificado autofirmado se almacenará en el archivo server.crt.

Ya que hemos creado el certificado, es hora de instalarlo. Para ello ejecutaremos los siguientes comandos, con los que copiamos el certificado autofirmado y la clave privada del servidor a los directorios /etc/ssl/certs/ y /etc/ssl/private respectivamente, como se muestra:

```
sudo cp server.crt /etc/ssl/certs/
sudo cp server.key /etc/ssl/private/
```

Ahora deberemos crear el archivo /etc/apache2/sites-available/ssl. Este archivo será una copia del archivo /etc/apache2/sites-available/default, la copia la haremos de la siguiente manera:

```
sudo cp default ssl
```

Las primeras líneas de este archivo serán modificadas de manera que queden como sigue:

```
NameVirtualHost *:443
<VirtualHost *:443>
ServerAdmin webmaster@localhost
SSLEngine On
SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile
/etc/ssl/private/server.key
DocumentRoot /var/www-ssl/
<Directory />
```

Con esto, los archivos que se encuentren en el directorio /var/www-ssl/ serán los que utilizarán HTTPS (este directorio deberá ser creado).

Lo que deberemos hacer luego de esto, es editar el archivo /etc/apache2/ports.conf. Este archivo es muy pequeño, por lo que a continuación mostramos el archivo completo y como debe quedar una vez que se ha modificado:

```
Listen 80
<IfModule mod_ssl.c>
Listen 443
</IfModule>
```

Después de esto hay que crear una liga simbólica del archivo /etc/apache2/sites-available/ssl en /etc/apache2/sites-enabled/ssl:

```
ln -s /etc/apache2/sites-available/ssl /etc/apache2/sites-enabled/ssl
```

Para finalizar, sólo resta reiniciar nuestro servidor web Apache y todo ha quedado listo.

```
sudo /etc/init.d/apache2 restart
```

Con esto hemos configurado nuestro sitio web con el soporte para HTTPS.

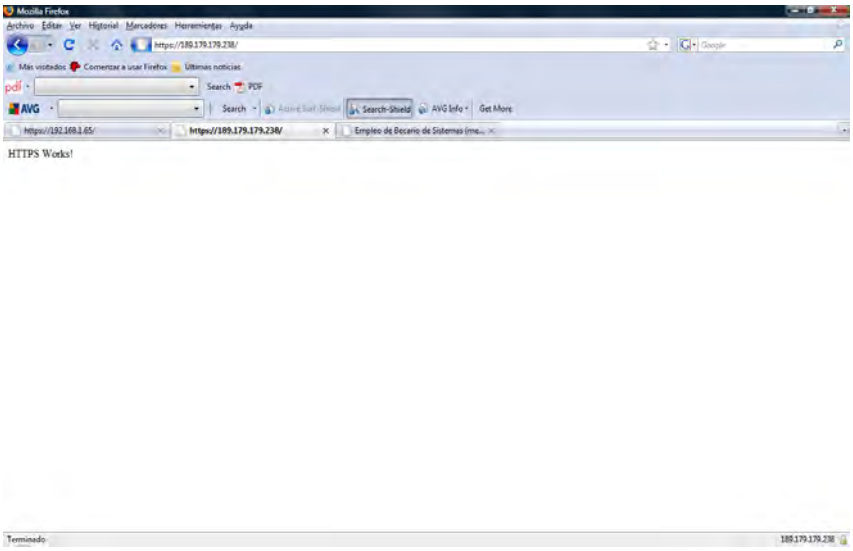

Figura A14.-HTTPS funcionando.

I.-Restricción de acceso de un usuario al home de otros usuarios.

A nosotros nos interesa que el directorio web de un usuario se encuentre en su directorio dentro de /home y que el usuario no tenga la posibilidad de ingresar al directorio de otros usuarios dentro de /home.

Para comenzar debemos crear un usuario, en nuestro caso lo hemos llamado “jaula”:

```
sudo adduser jaula
```

Después de esto se nos pedirá cierta información como la contraseña para el usuario y otros datos, tales como su nombre, su teléfono y otros.

Después de esto teclearemos el siguiente comando:

```
sudo dpkg-reconfigure -plow adduser
```

Al teclear este comando, aparecerá un ventana que nos explicará para qué sirve dicho comando, responderemos que No.

El comando anterior sirve para permitir o denegar el acceso al directorio de cada usuario dentro del directorio /home. Al teclear dicho comando y seleccionar la opción no, hemos hecho que cada usuario pueda solamente modificar el contenido de su directorio en /home y no pueda visualizar ni modificar el contenido de los directorios de otros usuarios en /home.

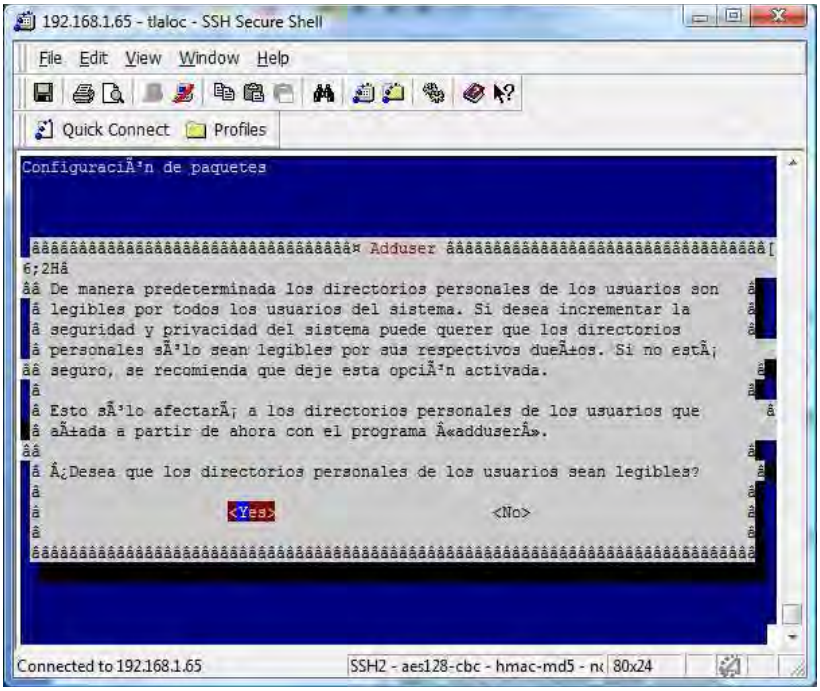

Figura A15.-Denegar el acceso de usuarios a directorios de otros usuarios.

Por último asignaremos permisos a los directorios de los usuarios dentro de /home:

```
sudo chmod o-r /home/*
```

Con este último comando estamos haciendo lo anterior para todos los archivos y subdirectorios del directorio de los usuarios dentro de /home.

Con lo anterior hemos logrado nuestro cometido de denegar el acceso de un usuario a un directorio web que no le pertenece.

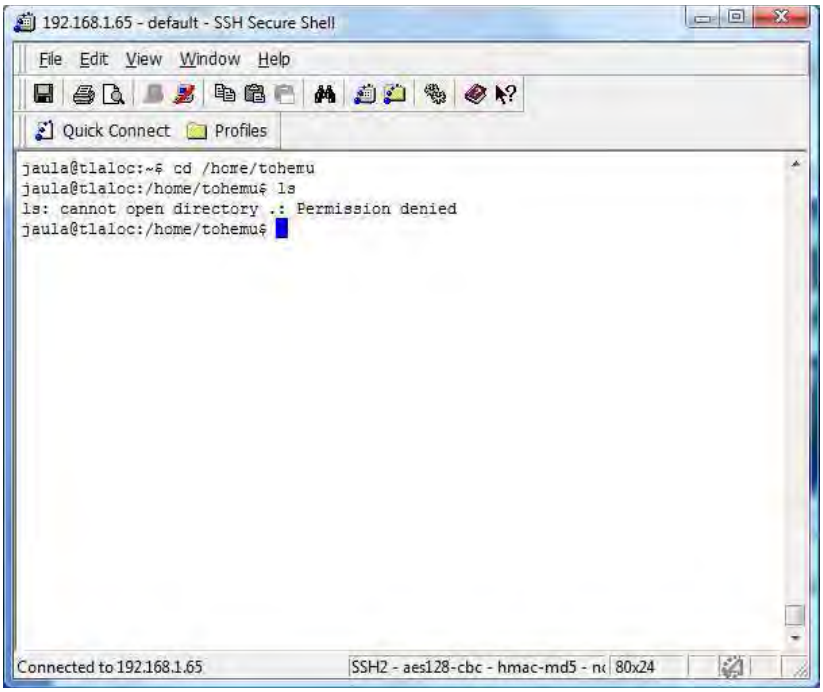

Figura A16.-Prueba de denegación de acceso al directorio de otro usuario.

Para hacer que el directorio de un usuario dentro de /home sea su directorio web, es necesario crear una liga simbólica hacia el directorio web:

```
sudo ln -s /var/www/tohemu /home/tohemu
```

Con el comando anterior, se logra que los archivos que el usuario ponga en su directorio, puedan ser leídos desde Internet. En este caso el directorio /home/tohemu hace referencia al directorio /var/www/tohemu.

J.-Respaldo de archivos del sistema.

Una buena práctica de seguridad es realizar respaldos de los archivos de los usuarios del servidor web. Muchos administradores realizan tareas programadas, que ejecutan un programa que hace esta tarea cada determinado tiempo. Nosotros realizaremos esta tarea con la ayuda de un demonio llamado cron, que nos permitirá hacer la ejecución de programas en ciertos periodos de tiempo definidos por nosotros.

Primero hemos escrito un script en bash que nos permite hacer un respaldo de los directorios web (/var/www y /var/www-ssl), este script copia estos directorios en un directorio llamado backupaammdd (aa-año,mm-mes,dd-día), luego de esto comprime dicho directorio y lo envía a un ordenador remoto utilizando un programa llamado scp.

El código de dicho script es el que se muestra abajo:

```
#
#Script para respaldar los directorios web de este servidor
#

cd /root

DATE=$(date +%y%m%d)

#Creamos el directorio de respaldos
sudo mkdir /root/backup

#copiamos el directorio /var/www a /root/backup
#y el directorio /var/www-ssl a /root/backup
sudo cp -r  /var/www /root/backup/www
sudo cp -r /var/www-ssl /root/backup/www-ssl

#Comprimimos el directorio backup
sudo tar -cvvf backup$DATE.tar backup/

#Copiamos el archivo de respaldo en un host remoto con ssh
sudo scp /root/backup$DATE.tar tohemu@192.168.1.64:/home/tohemu/www/backups


#borramos el directorio /root/backup y el archivo backup$DATE
sudo rm -rf backup
sudo rm  backup$DATE.tar
```

Para automatizar esta tarea se hará uso de cron, pero antes de esto se debe hacer una configuración en el ordenador remoto para que no solicite contraseña al iniciar sesión en él y poder enviar el archivo de respaldo. Esto sólo sirve mientras el servidor web se encuentra encendido.

Lo primero es crear un par de llaves pública y privada en el servidor web, como sigue:

```
sudo ssh-keygen -t dsa
```

Se nos solicitará el nombre del archivo donde se almacenarán las llaves, en nuestro caso, lo hemos denominado llaves, luego de esto se nos pedirá una contraseña y su confirmación, en donde no ingresaremos nada.

Una vez que se ha creado el archivo llaves, hay que copiarlo al ordenador remoto, en el que se almacenarán nuestros respaldos:

```
sudo scp llaves.pub tohemu@192.168.1.64:/home/tohemu/.ssh/authorized_keys
```

Con esto será posible transmitir archivos desde el servidor web a un ordenador remoto que almacenará nuestros respaldos.

Ahora si podremos automatizar esta tarea con cron, para lo cual ejecutaremos el comando:

```
sudo crontab -e
```

Este comando nos permitirá editar el archivo crontab, que es el archivo que nos permitirá programar tareas cada cierto tiempo.

En este archivo haremos la configuración para que nuestro script de respaldos se ejecute diario a las 6 a.m.

Luego de editar el archivo crontab, este debe quedar como sigue para que los datos del servidor web se respalden automáticamente en el ordenador remoto:

```
# m h dom mon dow command
  0 6  *    *    *    /root/backup-script
```

K.-Configuración segura de PHP.

El archivo de configuración de PHP se llama php.ini y en nuestro servidor web se encuentra localizado dentro del directorio de apache (/etc/apache2/php.ini*). En este archivo se encuentran todas las configuraciones de PHP. A nosotros nos interesa editar este archivo para darle una mayor seguridad a nuestro servidor web.

Básicamente en este archivo existen varias opciones de configuración que vienen establecidas y que nosotros modificaremos, ya que pueden representar un hoyo de seguridad en algún momento.

A continuación mencionamos las opciones de configuración que consideramos importantes para tener una seguridad mayor a la hora de servir páginas web dinámicas con PHP.

- Es recomendable deshabilitar el acceso remoto a archivos, ya que con funciones como fopen, file_get_contents, include, entre otras, permiten el acceso a archivos que se encuentran en otros hosts, además se les permiten a los

programadores considerar a las URLs como archivos. Para lograr deshabilitar el acceso remoto a archivos editamos la siguiente bandera del archivo de configuración de PHP como sigue:

```
allow_url_fopen = Off
```

De ser necesario el acceso a archivos remotos, se recomienda utilizar funciones de CURL (Client URL Library) o funciones como fsockopen.

- La transformación de parámetros en las peticiones HTTP que utiliza PHP es insegura, por lo que se debe estar deshabilitado el uso de variables globales. En la versión 4 de PHP y en versiones posteriores, esta opción viene deshabilitada por defecto, pero no está por demás verificar que la bandera se encuentre apagada:

```
register_globals = Off
```

- Es recomendable restringir los archivos a los que puede acceder PHP. La directiva open_basedir nos permite hacer lo anterior limitando el árbol de directorios a un directorio al que se puede acceder. Quizás esta sea la bandera más importante en relación a la seguridad de PHP. Su funcionamiento es como sigue:

```
open_basedir = /var/www/
```

El valor de esta bandera en el ejemplo nos indica que PHP únicamente podrá acceder a los archivos del directorio /var/www/.

- PHP también cuenta con un modo seguro en su archivo de configuración, sin embargo, el activarlo puede traer inconvenientes, ya que en el modo seguro permite a Apache acceder sólo a archivos de los cuales este sea dueño. Es recomendable activar esta bandera si no se configuró a PHP como módulo de Apache. La mejor forma de utilizarlo es configurando las banderas como sigue:

```
safe_mode = Off
safe_mode_gid = On
```

- Si se está trabajando con PHP en modo seguro, la ejecución de archivos binarios no está permitida, pero existe una bandera que nos permite especificar un

directorio en el cual se permite que los archivos binarios en el directorio especificado se puedan ejecutar, la bandera es la siguiente:

```
safe_mode_exec_dir = /www/ejecutables
```

- Al trabajar en el modo seguro de PHP, el acceso a variables de entorno tampoco se encuentra permitido, si se requiere el uso de algunas banderas, estas se pueden utilizar especificándolas en una lista (separada por comas) de prefijos que se permiten para las variables necesarias. Con la siguiente bandera podemos especificar las variables de entorno a las cuales se puede tener acceso:

```
safe_mode_allowed_env_vars = PHP_VARIABLE
```

En el ejemplo de arriba, la variable de entorno dada por VARIABLE, será la que contará con permiso para ser utilizada.

- Es importante controlar los archivos que se van a subir a un servidor. Si no es necesario subir archivos al servidor, se recomienda que esto no se permita o bien que se utilice un filtro para permitir subir solamente algunos tipos de archivos (no se recomiendan archivos binarios). Para impedir que se suban archivos al servidor basta con que la siguiente sentencia se encuentre como sigue:

```
file_uploads = Off
```

- Es importante limitar el tamaño de los archivos si se requiere permitir subirlos, esto lo podemos controlar con la siguiente bandera:

```
upload_max_filesize = 2M
```

- El límite en el tiempo de ejecución de un script es otro de los factores a tomar en cuenta, así como el tamaño de memoria máximo que tiene un script para ejecutarse, esto se controla con las siguientes banderas respectivamente:

```
max_execution_time = 30
memory_limit = 16M
```

Glosario.

Amenaza: Circunstancia, suceso o persona con el potencial para dañar un sistema mediante la destrucción, la divulgación, la modificación de datos o la negación de servicios.

Apache: Programa que opera en la capa de aplicación del modelo OSI, el cual emplea el protocolo HTTP para servir documentos HTML. Es desarrollado por la organización Apache Software Foundation.

DBMS (DataBase Management System): Son sistemas gestores de bases de datos, que permiten el manejo adecuado de datos, con un esquema de almacenamiento ordenado que permite realizar consultas ordenadas y controladas. Utilizan el lenguaje SQL.

DoS (Denial of Service): Se trata de un ataque que tiene como consecuencia la no disponibilidad de un servicio a usuarios autorizados de este. Por lo general es provocado por el envío de peticiones masivas al servidor que provee el servicio, lo que propicia que el servidor sea incapaz de atender las peticiones de los usuarios.

Exploit: Proceso que se encarga de encontrar y explotar vulnerabilidades en un sistema.

Firewall: Es un sistema diseñado para impedir el acceso no autorizado o el acceso desde una red privada. Pueden implementarse firewalls en hardware, software o en ambos. Los firewalls se utilizan con frecuencia para impedir que los usuarios de Internet no autorizados tengan acceso a redes privadas conectadas a Internet. El firewall personal protege al equipo frente a ataques de Internet, contenidos Web peligrosos, análisis de puertos y otros comportamientos de naturaleza sospechosa.

FTP (File Transfer Protocol): Se trata de un protocolo que tiene como objetivo promover el intercambio de ficheros entre computadoras de manera remota de manera rápida y eficiente. Las especificaciones completas de este protocolo se encuentran descritas en el RFC 959.

HTML (HyperText Markup Languaje): Es un lenguaje de marcado de hipertexto, que se basa en el uso de etiquetas y que es interpretado por clientes HTTP (navegadores web). Estándar para las páginas de internet.

HTTP (HyperText Transfer Protocol): Es un protocolo de la capa de aplicación con la ligereza y la velocidad necesaria para sistemas de información hipermedia colaborativos y distributivos. HTTP ha estado en uso desde 1990 por el *World-Wide Web*. Las especificaciones del protocolo HTTP se encuentran descritas en el RFC 1945, dicha especificación describe las características que se encuentran implementadas en la mayor parte de clientes y servidores HTTP.

HTTPS(HyperText Transfer Protocol Secure): Es una versión segura del protocolo HTTP, la cual permite transacciones seguras a través de la red, como por ejemplo en operaciones bancarias.

IIS (Internet Infirmation Services): Es una aplicación desarrollada por Microsoft Coorporation, que es útil para proveer servicios de Correo y Web por medio de los protocolos SMTP, HTTP y FTP.

MySQL: Es un manejador de base de datos (véase DBMS) soportado desde 2008 por Sun Microsystems, es ampliamente conocido por su gran adaptación con los sistemas operativos de la familia Linux, con el servidor Web Apache y el lenguaje PHP, a lo que se conoce como LAMP (Linux, apache, MySQL, PHP), aunque también tiene una gran adaptación con Java.

Netcraft: Compañía que tiene el objetivo de proveer servicios de seguridad en Internet, incluyendo servicios anti-fraude y anti-phishing, pruebas de seguridad en aplicaciones, revisión de códigos pruebas automáticas de penetración en sistemas.

OWASP (Open Web Application Security Project): Es una comunidad libre y abierta centrada en la mejora de la seguridad del software de aplicación. Su misión es hacer visible la seguridad de las aplicaciones, de modo que las personas y organizaciones puedan tomar decisiones informadas sobre los verdaderos riesgos de seguridad en las aplicaciones. La OWASP Foundation es una organización sin fines de lucro.

Phishing: Es un ataque que suele comenzar con un mensaje de correo electrónico falseado, que en apariencia procede de una compañía reconocida y fiable. En tal mensaje se le engaña a un usuario para que revele información confidencial como contraseñas de tarjetas de crédito y de cuentas bancarias. En dichos mensajes de correo, se incluyen ligas en las que se guía al usuario a una réplica de un sitio de internet reconocido en donde el usuario engañado revela su información confidencial, la cual es utilizada para cometer fraudes.

PHP: Es un lenguaje de *scripting* de propósito general, que es especialmente orientado a la construcción de sitios web dinámicos.

SSH (Secure Shell): Es un protocolo para el inicio de sesión remoto y el uso de otros servicios de red de manera segura. Sus especificaciones completas se encuentran descritas en el RFC 4254.

SSL (Secure Socket Layer): Se trata de un protocolo que nos permite hacer uso de comunicaciones cifradas para la transmisión segura de datos. SSL fue desarrollado conjuntamente por Netscape Communications y RSA Data Security.

TLS (Transport Layer Security): El protocolo TLS permite a aplicaciones cliente/servidor comunicarse a través de un canal seguro, diseñado para prevenir la escucha, manipulación o falsificación de mensajes.

Vulnerabilidad: Una vulnerabilidad es un estado en un sistema informático que: permite a un atacante ejecutar comandos como otro usuario, permite a un atacante acceder a los datos en contra de las restricciones de acceso especificadas para esos datos, permite a un atacante hacerse pasar por otra entidad o permite a un atacante realizar una negación de servicio.
XSS (Cross-Site Scripting): Es una vulnerabilidad que permite a un atacante inyectar código (por lo regular de JavaScript o HTML) en una aplicación web, modificando el comportamiento habitual de la aplicación.

Mesografía y Bibliografía.

Capítulo 1.

[1]- ROJAS Nava, Leticia. Arquitectura de seguridad perimetral y en sitio para sistemas Unix Caso: Unidad de Servicios de Cómputo Académico de la Facultad de Ingeniería. México, D.F., Facultad de Ingeniería UNAM [2007].

[2]-LÒPEZ Jaquelina y QUEZADA Cintia. Fundamentos de Seguridad Informática México, D.F., Facultad de Ingeniería UNAM [2006].

[3]- Guía para elaboración de políticas de seguridad. Universidad Nacional de Colombia [2003]. [Fecha de Consulta: 10 de Enero de 2010] Disponible en: http://www.dnic.unal.edu.co/docs/guia_para_elaborar_politicas_v1_0.pdf

[4]- Políticas de Seguridad de la Información para el Sector Público. Oficina Nacional de Tecnologías de Información, Coordinación de Emergencias en Redes Teleinformáticas [Septiembre 2005]. [Fecha de Consulta: 10 de Enero de 2010] Disponible en: http://www.arcert.gov.ar/ncursos/material/Presentacion_Curso_Politica_de_Seguridad_09-2005.pdf

[5]- Concepto de Análisis de Riesgos. Academia Latinoamericana de Seguridad Informática [2003?]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://download.microsoft.com/download/C/C/0/CC0E4675-59CF-477D-BB8A-CEC0A937C288/Modulo_2.pdf

[6]- Manual de la Metodología Abierta de Testeo de Seguridad. Institute For Security And Open Methodologies [2000-2003]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://isecom.securenetltd.com/OSSTMM.es.2.1.pdf

[7]- Windows 2000. Microsoft Corporation [11 de Marzo de 2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/spain/windows2000/default.mspx

[8]- Windows 2000 Server Features and Functionality. Microsoft Corporation [2000?]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://technet.microsoft.com/en-us/windowsserver/2000/bb735343.aspx

[9]- Lanzamiento de Windows 2000. Microsoft Corporation [28 de Enero de 2000]. [Fecha de Consulta 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/venezuela/windows2000/launch.htm

[10]- Las 10 ventajas principales para las organizaciones que se actualicen desde Windows 2000 Server. Microsoft Corporation [18 de Noviembre de 2002]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/latam/windowsserver2003/evaluation/whyupgrade/top10w2k.mspx

[11]- Windows Server 2003 R2. Microsoft Corporation [11 de Marzo de 2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/spain/windowsserver2003/default.mspx

[12]- Windows Server 2003 R2 Pricing. Microsoft Corporation [4 de Abril de 2007]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/windowsserver2003/howtobuy/licensing/pricing.mspx

[13]- Información General técnica de la familia de productos Windows Server 2003. Microsoft Corporation [Enero 2003]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/latam/windowsserver2003/techinfo/resumen/default.mspx

[14]-Windows Server 2008 R2 Pricing. Microsoft Corporation [2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/windowsserver2008/en/us/pricing.aspx

[15]- Windows Server 2008, una nueva plataforma productiva. Microsoft Corporation [27 de Julio de 2007]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/latam/technet/articulos/tn/2007/jul-01.mspx

[16]- Windows Server 2008 R2. Microsoft Corporation [2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/windowsserver2008/en/us/default.aspx

[17]- What's New in Windows Server 2008 R2. Microsoft Corporation [2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/windowsserver2008/en/us/whats-new.aspx

[18]- Fedora. Red Hat Inc [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://fedoraproject.org/

[19]- Fedora 10 Accepted Features. Red Hat Inc [2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://fedoraproject.org/wiki/Releases/10/FeatureList

[20]- Security: Ubuntu Server Edition. Canonical Ltd [2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.ubuntu.com/products/whatisubuntu/serveredition/features/security

[21]- Apache http Server Project. The Apache Software Foundation [2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://httpd.apache.org/download.cgi

[22]- MySQL. Sun Microsystems Inc [2008]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.mysql.com/

[23]- MySQL How to Buy. Sun Microsystems Inc [2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: https://shop.mysql.com/

[24]- PostgreSQL. PostgreSQL Global Development Group [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.postgresql.org/

[25]- Servidores en Rack Power Edge. Dell Inc [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www1.la.dell.com/content/products/category.aspx/rack_optimized?c=mx&cs=mxbsdt1&l=es&s=bsd&~tab=2

[26]- IBM System X. IBM [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www-03.ibm.com/systems/x/?cm_re=masthead-_-products-_-sys-xseries

[27]- Blades de servidor BladeSystem ProLiant. Hewlett-Packard Development Company [2010]. [Fecha de Consulta: 10 de enero de 2010]. Disponible en: http://h10010.www1.hp.com/wwpc/es/es/sm/WF05a/3709945-3709945-3328410-3722790-3722790-3682822.html

Capitulo 2.

[28]- TERCERO Armendáriz, Guillermo. Seguridad Informática: Código Malicioso y Virus Informáticos. San Juan de Aragón, Estado de México. Facultad de Estudios profesionales Aragón, UNAM [2008].

[29]- Revista: @rroba, Nº123, Editorial @rroba, Málaga España [2008].

[30]- Open Security. Open Security [2009]. [Fecha de Consulta: 10 de enero de 2010]. Disponible en: http://www.opensecurity.es/

[31]- Las 10 vulnerabilidades Web más explotadas. Open Security [18 de Octubre de 2007]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.opensecurity.es/las-10-vulnerabilidades-web-mas-explotadas/

[32]- Las principales 5 vulnerabilidades Web. Desarrollo Web [2009?]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.desarrolloweb.com/articulos/principales-vulnerabilidades-web.html

[33]- Cross Site Scripting. Sl4xUz [2008?]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.milw0rm.com/papers/207

[34]- XSS: Cross Site Scripting – ¿Cómo evitarlo?. Stephen Reinhardt [29 de Octubre de 2007]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://10typesofpeople.wordpress.com/2007/10/29/xss-cross-site-scripting-%C2%BFcomo-evitarlo/

[35]- Code Injection Vulnerabilities Explained. The Server Pages [30 de Julio de 2004]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.theserverpages.com/articles/webmasters/php/security/Code_Injection_Vulnerabilities_Explained.html

[36]- Cross-Site Request Forgery. OWASP [23 de Octubre de 2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.owasp.org/index.php/Cross-Site_Request_Forgery

[37]- Buffer Overflow. OWASP [21 de Febrero de 2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.owasp.org/index.php/Buffer_Overflow

[38]- SQL Injection Attacks by Example. Steve Friedl [10 de Octubre de 2007]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.unixwiz.net/techtips/sql-injection.html

[39]- Ataques mediante SQL Injection. Issel Guberna [2007?]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.programacionphp.net/recursos-articulos/articulos-de-Vulnerabilidades/SQL-injection_0001601.html

Capitulo 4.

[40]- Introducción a Windows Server 2003, Web Edition. Microsoft Corporation [18 de Noviembre de 2002]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.microsoft.com/latam/windowsserver2003/evaluation/overview/web.mspx

[41]- Features: Ubuntu Server Edition, Canonical Ltd [2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.ubuntu.com/products/whatisubuntu/serveredition/features

Capitulo 5.

[42].- HERNÁNDEZ Martínez, Miguel Ángel, Desarrollo del plan de seguridad informática para el departamento de sistemas de la empresa El Hilo Megro S.A. de C.V. Coatzacolacos, Veracruz. Universidad de Sotavento, A.C. [2007].

[43]- Políticas de Seguridad en Cómputo de la Facultad de Ingeniería. Facultad de Ingeniería [2008]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.ingenieria.unam.mx/~cacfi/politicas_seguridad2.html

[44]- Mecanismos básicos de Seguridad para Redes de Cómputo. DGSCA-UNAM [8 de Diciembre de 2005]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.seguridad.unam.mx/eventos/admin-unam/politicas_seguridad.pdf

Anexo.

[45].- TENORIO Ruiz, Guillermo, Informe de un sitio web basado en el diplomado de desarrollo e implementación de software libre Linux. San Juan de Aragón, Estado de México. Facultad de Estudios profesionales Aragón, UNAM [2006].

[46]- Ubuntu Server Edition. Canonical Ltd [2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.ubuntu.com/products/whatisubuntu/serveredition

[47]- Ubuntu Server Guide. Canonical Ltd [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: https://help.ubuntu.com/8.04/serverguide/C/index.html

[48]- OpenSSH Server. Canonical Ltd [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: https://help.ubuntu.com/7.04/server/C/openssh-server.html

[49]- ApacheMySQLPHP. Canonical Ltd [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: https://help.ubuntu.com/community/ApacheMySQLPHP

[50]- PhpMyAdmin. Guía Ubuntu [22 de Febrero de 2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.guia-ubuntu.org/index.php?title=PhpMyAdmin

[51]- Uncomplicated Firewall. Canonical Ltd [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://doc.ubuntu-es.org/UFW

[52]- PHP – Configuración segura de php.ini. Hacktimes [18 de Julio de 2005]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.hacktimes.com/files/PHP-HackTimes.com.V1.0.pdf

Glosario.

[53]- HTML. W3C [29 de Enero de 2009]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.w3.org/MarkUp/

[54]- Request for Comments. IETF [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.ietf.org/rfc.html

[55]- SSL. VeriSign Inc [2010]. [Fecha de Consulta: 10 de Enero de 2010]. Disponible en: http://www.verisign.com/ssl/ssl-information-center/how-ssl-security-works/