Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

ASEGURAMIENTO Y SEGURIDAD EN SERVIDORES WEB CASO DE ESTUDIO: UNIVERSIDAD VIRTUAL DE LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Aura Luz Cifuentes Reyes
Asesorado por el Ing. Pedro Pablo Hernández Ramírez

Guatemala, noviembre de 2012

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ASEGURAMIENTO Y SEGURIDAD EN SERVIDORES WEB CASO DE ESTUDIO: UNIVERSIDAD VIRTUAL DE LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

AURA LUZ CIFUENTES REYES
ASESORADO POR EL ING. PEDRO PABLO HERNÁNDEZ RAMÍREZ

AL CONFERÍRSELE EL TÍTULO DE

INGENIERA EN CIENCIAS Y SISTEMAS

GUATEMALA, NOVIEMBRE DE 2012

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Murphy Olympo Paiz Recinos
VOCAL I	Ing. Alfredo Enrique Beber Aceituno
VOCAL II	Ing. Pedro Antonio Aguilar Polanco
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Juan Carlos Molina Jiménez
VOCAL V	Br. Mario Maldonado Muralles
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Murphy Olympo Paiz Recinos
EXAMINADOR	Ing. Marlon Antonio Pérez Turk
EXAMINADORA	Inga. Floriza Felipa Ávila Pesquera
EXAMINADORA	Inga. Sonia Yolanda Castañeda Ramírez
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

ASEGURAMIENTO Y SEGURIDAD EN SERVIDORES WEB CASO DE ESTUDIO: UNIVERSIDAD VIRTUAL DE LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas, con fecha octubre de 2011.

Aura Luz Cifuentes Reyes

Facultad de Ingeniería
Escuela de Ciencias y Sistemas

Guatemala, 17 de enero del 2012

Ingeniero

Carlos Alfredo Azurdia Morales

Coordinador de Privados y Trabajo de Graduación

Respetable Ingeniero Azurdia:

Por este medio le informo como asesor del trabajo de graduación de la estudiante universitario de la carrera de Ingeniería en Ciencias y Sistemas, AURA LUZ CIFUENTES REYES, carné 200614790, *que he revisado el trabajo de graduación titulado:* "ASEGURAMIENTO Y SEGURIDAD EN SERVIDORES WEB CASO DE ESTUDIO: UNIVERSIDAD VIRTUAL DE LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA", y a mi criterio el mismo está completo y cumple con los objetivos propuestos para su desarrollo según el protocolo.

Agradeciendo su atención a la presente,

Atentamente

Ing. Pedro Pablo Hernández Ramírez

Asesor de trabajo de graduación

Colegiado: 7240

Pedro Pablo Hernández Ramírez
Ingeniero en Ciencias y Sistemas
Colegiado 7240

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 08 de Febrero de 2012

Ingeniero
Marlon Antonio Pérez Turk
Director de la Escuela de Ingeniería
En Ciencias y Sistemas

Respetable Ingeniero Pérez:

Por este medio hago de su conocimiento que he revisado el trabajo de graduación de la estudiante **AURA LUZ CIFUENTES REYES** carné **2006-14790,** titulado: **"ASEGURAMIENTO Y SEGURIDAD EN SERVIDORES WEB, CASO DE ESTUDIO: UNIVERSIDAD VIRTUAL DE LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"**, y a mi criterio el mismo cumple con los objetivos propuestos para su desarrollo, según el protocolo.

Al agradecer su atención a la presente, aprovecho la oportunidad para suscribirme,

Atentamente,

Ing. Carlos Alfredo Azurdia
Coordinador de Privados
y Revisión de Trabajos de Graduación

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA - INGENIERIA EN CIENCIAS Y SISTEMAS - COORDINACION TESIS

ESCUELA DE CIENCIAS Y SISTEMAS

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERÌA
ESCUELA DE CIENCIAS Y SISTEMAS
TEL: 24767644

El Director de la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer el dictamen del asesor con el visto bueno del revisor y del Licenciado en Letras, del trabajo de graduación titulado ***"ASEGURAMIENTO Y SEGURIDAD EN SERVIDORES WEB CASO DE ESTUDIO: UNIVERSIDAD VIRTUAL DE LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA",*** *presentado por la estudiante AURA LUZ CIFUENTES REYES, aprueba el presente trabajo y solicita la autorización del mismo.*

"ID Y ENSEÑAD A TODOS"

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION DE INGENIERIA EN CIENCIAS Y SISTEMAS

Ing. Marlon Antonio Pérez Turk
Director, Escuela de Ingeniería en Ciencias y Sistemas

Guatemala, 13 de noviembre 2012

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

Ref.DTG.581.2012

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería en Ciencias y Sistemas, al trabajo de graduación titulado: **ASEGURAMIENTO Y SEGURIDAD EN SERVIDORES WEB CASO DE ESTUDIO: UNIVERSIDAD VIRTUAL DE LA ESCUELA DE CIENCIAS Y SISTEMAS DE LA FACULTAD DE INGENIERÍA DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**, presentado por la estudiante universitaria: **Aura Luz Cifuentes Reyes,** procede a la autorización para la impresión del mismo.

IMPRÍMASE.

Ing. Alfredo Enrique Beber Aceituno
Decano en funciones

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANO
FACULTAD DE INGENIERIA

Guatemala, noviembre de 2012

/cc

AGRADECIMIENTOS A:

Dios	Por darme la vida y por permitirme llegar hasta este logro fundamental en mi existencia.
Mis padres	Aura Reyes y Sergio Cifuentes, por el amor y buen ejemplo que han dado a mi vida, quiero compartir con ellos la alegría de nuestro triunfo.
Mis hermanos	Juan Leo Cifuentes y Karen Alejandra Sontay, por demostrarme la entrega que se debe dar para lograr nuestros sueños.
Dr. Héctor Sontay	Por ser un ejemplo a seguir y además ser un guía emocional, espiritual y profesional.
Ing. Pedro Pablo Hernández	Por enseñarme y guiarme sobre el presente trabajo y la vida, por su paciencia, por su entrega.
Ing. Darwin Hernández	Por su afecto incondicional y enseñarme a luchar a cada proyecto que me enfrento.
Mis amigos	Gracias por su apoyo y compresión.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

GLOSARIO

Ascii Es un código estándar de origen americano para el intercambio de información.

Captcha Es una prueba de desafío para identificar si el usuario que lo utiliza es humano o no, al lograr que un usuario escriba el contenido de una imagen distorsionada.

Compilador Es el proceso por medio del cual a partir de un programa o código fuente se ha traducido a lenguaje que es comprendido por una computadora y rápidamente ejecutable.

Dbms Es un software que se utiliza como interfaz entre la base de datos, el usuario y las aplicaciones que se utilizan.

Hosting Es un servicio que provee a los usuarios el poder almacenar información contenido de información web como imágenes, video o contenido que es accedido vía web.

Log Bitácora de eventos producidos durante un rango de tiempo, registrando datos de sobre quién, qué, cuándo, dónde y por qué en cualquier sistema.

Script	Es un programa de texto plano que se utiliza para realizar diversas tareas e interactuar con el sistema operativo o servicio, que facilita la implementación de manera automática de los procedimientos.
Tcp	Protocolo fundamentales de internet, que es utilizado para crear conexiones entre los usuarios de internet a través de un flujo de datos.
Trafico *web*	Son las cantidades de información que puede ser enviadas o recibidas por los usuarios de un sitio web.
Virus	Es un programa cuyo principal objeto es alterar el funcionamiento de una computadora, sin que el usuario de la misma dé su permiso.

RESUMEN

La Universidad Virtual de la Escuela de Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala; es una herramienta para la gestión y control de cursos para el pénsum de la carrera de Ingeniería en Ciencias y Sistemas; este servidor forma parte de la red de ingeniería.

Se realizó un análisis de las vulnerabilidades que pueden estar presentes en un servidor de aplicaciones web, así como también las medidas y políticas de seguridad que deben de llevarse a cabo para proteger al sistema, dado que un sitio web debe tomar medidas preventivas para evitar la pérdida o robo de información valiosa por parte de los estudiantes y profesores que utilizan el sistema.

La investigación se dirigió hacia los elementos que se tienen actualmente instalados en la Universidad Virtual, y las mejores prácticas de configuración de los servicios que están utilizando.

Se realizó un análisis de la situación actual de la Universidad Virtual a partir de una serie de recomendaciones que proporcionó el administrador de la misma; además, se encontró información publicada en la web por parte de estudiantes que utilizan la plataforma y se utilizó una herramienta de monitoreo de seguridad para encontrar problemas sobre el servidor; luego se propuso posibles soluciones a los problemas que se presentaron.

OBJETIVOS

General

Proponer tareas y normas de seguridad para asegurar que un sitio web responda a todos los servicios y planificar actividades para optimizar su rendimiento.

Específicos

1. Dar a conocer las vulnerabilidades más frecuentes en un sitio web, debido a que un atacante podría aprovechar para obtener el control total del sitio.

2. Enumerar las tareas y actividades que promueven el fortalecimiento de la seguridad en un sitio web.

3. Investigar las buenas prácticas de configuración para garantizar la seguridad del sistema y la continuidad de todos los servicios conforme transcurra el tiempo.

4. Evaluar y analizar la tecnología y tareas que actualmente están en la universidad virtual y detectar posibles mejoras en el sistema.

INTRODUCCIÓN

En la actualidad, en un sitio web es indispensable la seguridad, ya que de no existir, podrían cambiar la lógica de la ejecución de las aplicaciones logrando resultados no deseados; contiene información valiosa ya sea de la organización o de los usuarios que lo utilizan, que puede ser perjudicada por su mal uso; por tanto, las medidas de seguridad deben centrarse en la eliminación o reducción de vulnerabilidades.

Se describen inicialmente los elementos que forman parte de la seguridad o vulnerabilidades en un servidor de aplicaciones, entre estas el ambiente, los usuarios, atacantes, e historias de infiltraciones que han marcado historia.

Para disminuir el riesgo en un sitio web, la tecnología que se utilice deberá ser configurada para maximizar su rendimiento; se debe definir una serie de políticas de seguridad y programación de tareas, para prevenir detectar las medidas de corrección al cubrir una vulnerabilidad; por ello se definieron las vulnerabilidades más comunes que ha atacado a aplicaciones web.

Los sitios web deben mejorar sus políticas de seguridad continuamente, dado que nuevas vulnerabilidades son encontradas diariamente y se necesita el incremento de tareas que garanticen que el sistema siga respondiendo a todos los servicios; se incluyeron configuraciones que se realizaron en la Universidad Virtual y que facilitan detectar el estado de un servidor.

1. MARCO TEÓRICO

1.1. Seguridad informática

Esta área de la informática, es una guía a la protección de elementos computacionales, por ello existen estándares, protocolos, reglas, herramientas que protegen la información.

1.1.1. Cliente web

El cliente de aplicaciones web es el navegador web. Se comunica a través de *http* (entre otros protocolos) y bajo la combinación de *html* y *http;* estos presentan datos procesados por el servidor web.

1.1.2. Aplicaciones web

Estos son programas que se ejecutan a través de un navegador de internet, y su funcionamiento es de una forma dinámica; estos servidores son muy comúnmente representados por un *middleware* entre los servidores de bases de datos y el usuario que a menudo se conectan.

1.1.3. Servidor web

Un servidor dará acceso a los computadores para que se conecten a él, y puedan acceder a programas, documentos, archivos, u otra información del servidor.

Un servidor web es un programa que recibe peticiones que son realizados por los clientes (usuarios de internet); se encarga de contestar a estas peticiones en una página web o información según los parámetros solicitados. Este intercambio de información se realiza por medio de un navegador mediante protocolos como http *y https,* pero muchas veces estos servidores realizan más de una, funcionando al mismo tiempo como servidor de correo, o como bases de datos o de archivos, dado que estos pueden tener muchos programas funcionando simultáneamente.

1.1.4. Protocolo http

Su función es transferir información por medio de internet a través de páginas web, este es uno de los protocolos más usados; por lo general opera en puerto *tcp* 80, pero puede existir en cualquier puerto libre; *http* define un mecanismo para solicitar un recurso, y el servidor lo devuelve. Si el recurso tiene acceso, pueden ser desde páginas web hasta el contenido de un vídeo.

Este protocolo está basado en texto *ASCII*, accesible para que cualquiera lo pueda leer y no hay necesidad de decodificar, solo se necesita saber son las solicitudes y respuestas.

1.1.5. Protocolo ssl

Este es un protocolo de cifrado de información que nos provee comunicación segura por una red, por lo que el intermediario no podrá leer el texto plano como sería en el protocolo *http;* este no provee mayor seguridad general, además de poner más difícil de interceptar el tráfico entre el cliente y el servidor.

Otra característica de este protocolo es que el servidor de internet debe poseer un certificado digital de seguridad (estas son otorgadas por agencias independientes y autorizadas).

1.1.6. Seguridad en un sitio web

Un sitio web puede encontrase altamente vulnerable y todos los días se producen nuevos riesgos, para ello se deben tomar acciones preventivas para evitar que estos sean aprovechadas por otro usuario.

- Información de tarjetas de crédito, cuentas bancarias
- Información personal (*email*, direcciones, números de teléfonos)
- Listado de usuarios y contraseñas
- Información interna de la empresa

Para disminuir el riesgo en un sitio web se deben reducir al mínimo las aplicaciones que funcionarán en el servidor; la tecnología que se utilice deberá ser configurada para maximizar su rendimiento; se debe definir una serie de políticas de seguridad y programar tareas para prevenir, detectar y las medidas de corrección al cubrir una vulnerabilidad.

1.1.7. Usuarios de un sistema web

- Algunas operaciones básicas de los usuarios.

- Registro: el usuario puede crear su propia cuenta y tener acceso a la información personalizada del sistema, según permisos que le sean otorgados.

- Perfil público: la información de cada uno de los usuarios del sistema, debe modificarse o ampliarse según gustos del usuario.

- Publicar contenidos: los usuarios podrán crear, editar, gestionar y publicar contenido digital.

- Temas de foro: *posts* en las discusiones u opiniones en línea que permite al usuario expresar sus opiniones.

- Agenda: programar citas en unos calendarios o sucesos asociados a una fecha.

- Gestión de *blog*: los usuarios puedan crear contenidos o comentar y así darles respuestas a los autores del mismo.

- Publicar: los usuarios podrán subir cualquier tipo de contenido, en función de los objetivos del sitio web, para ser públicos a ciertos usuarios del sistema.

- Informes: podrá obtenerse información del comportamiento de los usuarios en un periodo de tiempo específico que tengan ciertas características comunes.

- Anuncios y enlaces: publicar y enviar información a los usuarios; estos pueden ser enviados a los datos que se encuentra en su perfil.

1.1.8. Roles del usuario

- Los usuarios son los que a través de un navegador realizan operaciones en un sitio web, ya sea teniendo una cuenta en el sitio o según los permisos que posea se le asignen recursos del sistema.

- Usuario registrado: este podrá acceder a través de un *login* (autentificación) podrá obtener información que requiera identificación; puede acceder a descargas o recursos que tenga el permiso de "registrado".

- Usuario anónimo: este tendrá acceso a información pública, él solo consulta la información en línea, sin tener ningún privilegio sobre el sistema.

- Administrador: de este tipo de usuario pueden existir diferentes clases y así variar según las necesidades del sitio; posee la mayoría de permisos para poder administrar el contenido del sitio pero sin tener acceso a todos los componentes y módulos como tampoco a la configuración global del sistema.

- Súper administrador: posee los permisos para poder acceder al sitio y a todas las zonas administrativas, no tiene ningún tipo de restricción y es el único capaz manipular usuarios, registrados y permisos, inclusive denegar la entrada al sistema.

1.1.9. Principales atacantes

- *Hacker*: esta es una persona que posee numerosos conocimientos de informática y tecnología asociada y que utilizará estos conocimientos para acceder a información privada del sitio web a través de una falla o deficiencias en el sistema; después de encontrar estas posibles fallas y soluciones, las publica por los diferentes medios, para así mejorar la seguridad en un sitio web.

- *Lammer*: este usuario posee menores conocimientos que los que posee el *hacker*, pero solo puede llegar a encontrar fallas menores, que divulga a través de la web. Este usuario está comenzando en el mundo del *hackeo* de aplicaciones y cualquier logro lo toma con satisfacción; esto sucede a base de tutoriales o manuales de *hacker* avanzados.

- *Crackers*: estos usuarios hacen uso incorrecto de la información de un sistema informático a partir de encontrar una debilidad o *bugs*, violando la seguridad y tomando control del mismo para obtener información o recursos.

1.1.9.1. Historias web de *hackers*

- Grace Hooper: es conocida como la primer *hacker*, produjo el primer compilador, logró el cargo de almirante y siempre tuvo la visión de que la computación se podría utilizar para más que fines militares, para ayudar al mundo, y participó en la creación del primer lenguaje de programación COBOL.

- Kristina Svechinskaya: joven rusa de 21 años que en el 2010 formó parte de una banda de *cracker* que intentó hacer un robo multimillonario a diversos bancos estadounidenses e ingleses utilizando un troyano llamado Zeus; ella fue acusada de desviar el dinero de cuentas bancarias y de haber abierto cinco cuentas y gracias a ello recibió 35 millones de dólares; fue condenada a cuarenta años de prisión.

- Jude Milhon: era defensora de los derechos informáticos como también de que las mujeres tengan una activa participación en la web; fue la impulsadora del grupo *ciberpunks*, una famosa frase de Jude Milhon: "Las piedras y los palos pueden romperme los huesos, pero las palabras en una pantalla pueden hacerme daño si y hasta que yo lo permita". El día de su muerte fue titulado en todos los grandes periódicos digitales que ha muerto la protectora de los *hackers*.

- Joanna Rutkowska: experta en el mundo del *malware*; ella inició desde la primaria donde aprendió ensamblador X86 con Ms-Dos; luego se pasó a Linux con desarrollos de exploids para Linux y Windows, el cual se centra en proteger el *kernels* de los sistemas operativos, sobre tecnologías de *rootkits*, *backdoors* etc. En el 2006 durante la presentación (*Black hat*) de Microsoft sobre la nueva versión de "Vista", en una sala siguiente, ella se encontraba demostrando cómo insertar código malicioso en el *core* de Windows vista, utilizando la píldora azul que fue programada por ella misma, que es 100% indetectable.

- Chen Ing-Hou: creó el virus CIH, que fue motivado por las insuficiencias en el desarrollo de virus; este lo desarrolló en mayo de 1998, exactamente, en el aniversario de la tragedia ocurrida en la planta nuclear rusa; actualmente trabaja como experto en Internet Data Security.

- Vladimir Levin: graduado de la Universidad Tecnológica de San Petesburgo, acusado por una serie de fraudes para substraer más de 10 millones de dólares de cuentas de Citibank; fue sentenciado a 3 años de prisión y a pagar una suma $ 240,015 a favor del banco, ya que las compañías de seguros habían cubierto el monto total de la pérdida; se tuvieron que mejorar los sistemas de seguridad, dado que actualmente Vladimir se encuentra en libertad.

- Mark Abene: a sus 17 años ya era conocido como un genio de la computación y de telefonía; experto en patrones de discado en recepción telefónica; en 1992 hizo colapsar a *WNET* un canal de la ciudad de New York al mandar un anuncio con un saludo por el Día de Acción de Gracias; fue catalogado como una de las 100 personas más inteligentes de la nación, accesó de manera desautorizada al sistema de compañías de teléfono; por ello fue condenado a 10 meses en prisión.

- Anonymous: entidad compartida por un grupo de usuarios sin nombre donde es imposible encontrar información real, pero trabajan bajo la misma dirección y se organizan por toda la red; uno de sus principales ataques es la denegación de servicios de forma coordinada y distribuida que el *hosting* no soporta la carga de peticiones; dejando suspendido el servicio. Han realizados diversos ataques a muchas compañías entre ella Sony, Telefónica, partidos políticos de España, las páginas del gobiernos de Guatemala, Egipto, Argelia, Libia, Irán, Chile, Colombia y Nueva Zelanda etc.

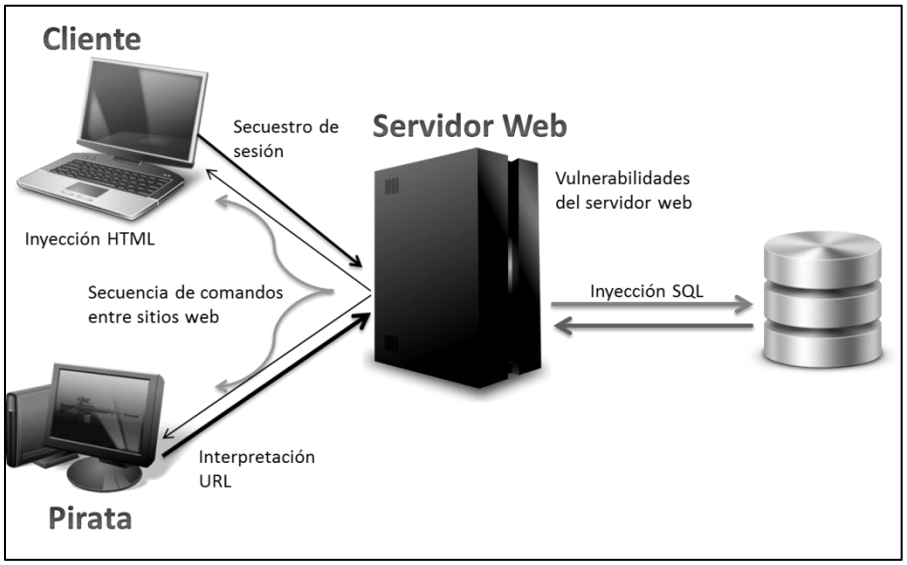
1.2. Vulnerabilidad en un sitio web

Esta es una falla, deficiencia, debilidad o *bugs* en un sistema; puede provocar que un usuario del sistema obtenga información interna del sitio.

Los primeros ataques a vulnerabilidades web fueron dirigidos hacia el conjunto de protocolos; con el paso del tiempo, estos son dirigidos hacía las capas de las aplicaciones web y a las redes externas, ya que las empresas empezaron abrir sus sistemas al tráfico del internet.

Los usuarios buscan fallas en los servidores web que cada vez deben estar más protegidos, porque a diario surgen nuevas amenazas y la seguridad se debe tener en cuenta desde el momento del diseño hasta el desarrollo del sitio.

Figura 1. **Ejemplo de principales amenazas para un sitio web**



Fuente: elaboración propia, con programa de Power Point.

1.2.1. Disponibilidad

De los elementos más importantes de un sitio web, es el de asegurar que el sistema siga funcionando; para ello se debe utilizar la protección en los canales de comunicación y los controles de seguridad; el no tener disponible el sistema se puede producir de diferentes formas:

- Tiempos de carga excesivos: la mayoría de usuarios de la web son impacientes, deben de tener tiempos de respuesta rápidos, de lo contrario abandonan el sitio.

- Detener el servidor: por tareas previstas por motivos de mantenimiento, actualizar información, copias de seguridad, actualizaciones de los programas instalados, etc. El servidor debe dejar de funcionar; sería ideal poseer un sistema para no tener este problema, pero por el costo que implica esto, muchas veces no es posible; por lo que es ideal mostrar un mensaje informando al usuario que el sistema no está disponible, avisando con tiempo anticipado el corte o parada del servidor.

- Imprevistos: muchas veces los servidores suspenden su funcionamiento por contratiempos que pueden ser algún recurso; deja de funcionar *hosting, dns, email* o cambios en un sistema operativo, por lo que es importante monitorear el sistema; existen sistemas que mandan mensajes de texto o correo electrónico cuando el sistema falla.

- Capacidad: puede suceder que se exceda la cantidad de tráfico y que estos dejen de funcionar lentamente o con errores.

1.2.1.1. Negación de servicio (DoS)

Este ataque produce que un servicio o recurso no pueda ser accesible por los usuarios del sistema; estas se originan por saturación de solicitudes que no son respondidas o aprovechan la vulnerabilidad de un sistema para volverlo inestable. Estos se pueden producir por: exceso de consumos de procesamiento de información, espacio en disco, etc. Generar tráfico en la red ocupando el ancho de banda disponible dejando cualquier recurso inoperable; alterar configuración de los sistemas por ejemplo modificar las rutas encaminamiento, produciendo que se manden a rutas que no son su destino.

Algunos de los ataques más comunes, que aparecen día a día son:

- Inundación de *SYN*: esta consiste en saturar el tráfico en la red haciendo uso del proceso de negociación de tres vías del protocolo *tcp*. Este consiste en que el cliente web realiza varias conexiones al servidor; este responde con un acuse de recibido, y el cliente web validará su conexión. En cambio suele suceder que se envían varias solicitudes con un ordenador, con información inexistente o no válida, y estos jamás reciben una respuesta de parte del servidor, provocando que este retenga conexiones abiertas en estructuras de memoria esperando respuesta, lo que puede provocar la caída del sistema si se producen muchas solicitudes.

- Ping de la muerte: este fue uno de los primeros ataques; consistió en enviar datagrama IP cuyo tamaño superaba el tamaño máximo que las almacena, produciendo la caída del sistema.

- Ataque *land:* consiste en la suplantación de direcciones IP; esta consiste en remplazar la dirección IP de un paquete existente, por la dirección de otro equipo; la idea es que el usuario logre enviar paquetes de red sin que sea interceptado por los sistemas de filtrado, esta parecerá provenir de un sistema de red interno y se transferirá al destino.

1.2.1.2. *Backups* incompletos o inexistentes

Puede ocurrir cualquier acontecimiento dentro de la organización que requiera la restauración de la información y muchas veces los respaldos de la información no han sido probados y no se verifican si están funcionando; o por ejemplo no se definen políticas o procedimientos para restaurar la información y estos errores comúnmente se detectan cuando se ha perdido o destruido la información.

1.2.1.3. *Log* incompleto o inexistente

Con el registro de eventos, se pueden ver los sucesos entrantes recientes, los salientes y los afectados con intrusiones. Mientras los usuarios se conecten a través de la red, esta puede ser atacada de una manera silenciosa y es necesario saber cómo atacaron el sistema con los registros de eventos; se puede conocer detalles de qué está ocurriendo.

Este proceso ayudará a saber qué problema sucedió y evitar que vuelva a suceder, dado que se puede realizar cambios en el sistema para que no ocurra.

1.2.2. Autenticidad

Es la forma de asegurarse o acreditar un usuario como auténtico; es decir que es quien dice ser y por ello se le puede asignar los recursos o servicios que tenga asignado.

En términos de seguridad de redes de datos, esto es conocido por (AAA) Autenticación, Autorización y Auditoría.

- Autenticación: es el proceso de verificar la identidad digital sobre una petición de conectarse a un sitio web; esta es una manera de confirmar que los usuarios sean quienes dicen ser; solo el usuario autenticado tiene que poder realizar funciones sobre un sistema.

- Autorización: sobre un sistema, es el proceso de autorizar a un usuario para acceder y así obtener determinados recursos que tiene asignados.

- Auditoria: este es el proceso de registrar todos los accesos a recursos autorizados o no por los usuarios sobre un sistema.

La diferencia entre la autenticación y autorización radica en que la primera es el proceso de verificar la identidad de un usuario; mientras que la segunda es el de asegurar que el usuario tenga la autoridad para realizar la operación.

1.2.2.1. Autenticación insuficiente

Esta ocurre cuando un sitio web permite que sus atacantes puedan acceder un contenido restringido o servicios, sin haberse autenticado correctamente.

La validación débil en la recuperación de contraseñas se produce cuando un sistema permite recuperar o modificar información de otro usuario al atacante, por ejemplo si un sitio web tiene una pregunta de seguridad que es fácilmente identificable por otro usuario, este puede obtener su acceso.

1.2.2.2. Autenticación y sesión interrumpida

Las sesiones creadas no tienen ninguna protección, y los agresores pueden obtener y enviar contraseñas, claves o cookies, logrando vencer las restricciones de autenticación, y así asumir la identidad de otro usuario.

Un ejemplo de caso es donde el servidor almacena las sesiones; por ejemplo estas algunas veces se almacenan en archivos temporales ubicados en la carpeta *"/tmp",* donde cualquier usuario puede acceder a los registros y ver el contenido de todas las sesiones; por ello se deben almacenar las sesiones ya sea en la base de datos o en lugares donde solo el administrador tenga acceso.

1.2.2.3. Autorización insuficiente

Este problema se produce cuando un sitio web permite acceso a recursos o servicios, a usuarios que no tienen los accesos necesarios.

- Predicción de credenciales: permite a un atacante asignar de una forma correlativa las contraseñas o claves de acceso al sitio web o implementar una contraseña que puede ser fácilmente predecible.

- Expiración de sesión insuficiente: se origina cuando se tiene mucho tiempo de expiración de las credenciales; es tanto tiempo, que esta puede ser reutilizada por un atacante u otros usuarios al autenticarse.

- Debe asegurarse que el sitio web esté diseñado para que después de un tiempo transcurrido en una sesión, se proceda a actualizar o destruirla según corresponda.

- Fijación de sesión: un sistema web siempre asigna las mismas credenciales o identificadores a los usuarios; estas pueden ser implementadas por otro usuario atacante para estar autenticado y así entrar al sistema.

1.2.2.4. Fuerza bruta

Es el proceso de automatizar la "prueba y error" o sea probar todas las combinaciones posibles hasta obtener el acceso y así adivinar usuarios, claves, número de tarjetas de crédito, etc.

Por ellos es que las contraseñas van a fijar una clave importante; por ejemplo, las claves de solo números son más fáciles de detectar, que aquellas que incluyen letras, símbolos y varios caracteres.

Casi todos los sitios deben poseer protecciones hacia intentos de acceso no autorizado, manteniendo vigiladas las direcciones IP desde su origen hasta su ingreso de usuario y contraseña; muy común es admitir tres intentos consecutivos, entonces bloquear la IP; pero muchos programas hoy en día pueden hacer combinaciones de IP para hacer creer al servidor que vienen de diferentes orígenes, logrando evitar este problema.

1.2.3. Integridad

Esta es la garantía de que la información no fue alterada, perdida o destruida, ya que esta puede ser alterada de forma accidental o por medio de un fraude. Es la constancia que se mantuvo una fiable comunicación de información entre el cliente y el servidor; este proceso puede ser certificado por un servicio de integridad que garantice que los datos son una fiel copia de los datos enviados.

La data digital es mucho más complicada de percibir si fue modificada y esta es casi imposible de detectar si fue modificada o mal intencionada durante su transmisión, por ello es primordial implementar un sistema de encriptación para evitar la amenaza.

1.2.3.1. Inyección de código SQL

Esta consiste en que los usuarios no autorizados logren modificar el comportamiento de las consultas, al introducir nuevos parámetros y así estos usuarios logren tener acceso a información que deben tener acceso en otros casos y logren modificar el comportamiento de las aplicaciones.

Este tipo de problema debe ser tomado en cuenta por parte del programador y así poder prevenirlo; un sistema creado con descuido, o ignorancia del problema puede quedar comprometido.

Al realizar una consulta en la base de datos, el usuario puede realizar un sin número de cosas como insertar registros, actualizar o eliminar datos, y autorizar acceso o agregar código malicioso en el sistema.

Por ejemplo en una aplicación web en la página de ingreso o autenticación, el usuario cambia los parámetros de la aplicación, cerrando la sentencia de SQL, eliminando la tabla y finalizando una consulta con otra a para lograr el acceso.

1.2.3.2. *Cross-site scripting*

Esta vulnerabilidad se puede crear en proceso de desarrollo o análisis de riesgos o en el proceso de diseño o desconocimiento de la vulnerabilidad; esta se concentra en falta de filtrado en los campos de entrada o ingreso de campos sin ninguna validación, creando secuencias de comandos maliciosos.

Los usuarios pueden activar *scripts* o comandos ejecutados desde alguna página web, estas podrían desarrollarse en el computador del usuario y si por ejemplo este usuario activa todos los privilegios en el navegador, podría obtener las cookies de otros usuarios, y activar servicios o componentes del sistema operativo.

Este código malicioso se encuentra oculto y el usuario sin saberlo puede activar el enlace y ejecutar alguna acción en su computador de manera indirecta; este ataque puede producirse ya sea por medio de correo electrónico, publicación de sitios web vulnerables, a través de un vínculo o imagen que sea atractiva para el usuario.

1.2.4. Antecedentes en Guatemala

Existen diferentes grupos de *hackers* que amenazan a Guatemala, para interferir los sistemas del gobierno, SAT y el Ministerio Público.

1.2.4.1. Anonymous Guatemala

El grupo de *hackers* amenaza al gobierno de Guatemala por la pobre actuación del gobierno por resolver casos de asesinatos y el 30 de agosto de 2011 se amenazó con tomar las páginas del gobierno y del Ministerio Público como protesta, por no resolver casos de homicidios a guatemaltecos.

1.2.4.2. Incursión en SAT

Un grupo de *hackers* logró ingresar a las páginas de SAT (Superintendencia de Administración Tributaria, Ministerio de Finanzas) modificando su página pública; a través de correos contactaron al periódico Prensa Libre; esto lo hicieron solo para demostrar que lo pueden hacer, dado que en Guatemala falta implementar mucha seguridad; además de mostrar las fallas que tienen los servidores de las empresas que están afectando, también han recibido remuneración por ello, ya que al demostrar las fallas en los sistemas, las empresas pueden pagar por saber por qué tienen esa falla.

Ellos ingresan en servidores que tengan sistema operativo de Microsoft Windows y tener el programa instalado *Frompage Server*, dado que el servidor al recibir una orden de *Frompage,* independientemente de donde provenga, expone toda la información.

1.3. Tareas de adaptación de la tecnología

Se han incluido los usuarios positivos y negativos que forman parte en un sistema web y los roles que estos pueden tomar durante la ejecución del mismo, así como las tareas más comunes que podrán ejecutar en un sitio web.

Se han definido las vulnerabilidades más relevantes ataques en sitios web, así como la historia de los más famosos atacantes, enfocando las vulnerabilidades y clasificándolas con base en la disponibilidad, autenticidad e integridad de un sitio, que son las principales características primordiales que debe cumplir cualquier sitio web.

2. SEGURIDAD EN UN SITIO WEB

2.1. Indicadores de seguridad en la web

Detectar si el sitio web es quien dice ser, es crucial pues en ellas se pide información confidencial; esta información que se esté mandando debe ser en forma segura y encriptada, cada transacción que se realice con el sitio web debe crear confianza y seguridad en el sitio web.

2.1.1. Cliente web

Es importante que los usuarios de la web se protejan y por ello existen una serie de medidas de prevención para evitar que los usuarios pierdan información mediante un sitio web.

2.1.2. Recomendaciones

- Utilizar las últimas versiones del sistema operativo, aplicaciones y navegadores.

- Implementar dispositivos que brinden una seguridad extra a sus equipos de computación como *firewall*s, antivirus, *antispyware* o cualquier programa que ayude a detectar código malicioso en sitios sensibles en la web.

- El sitio web debe tener implementado SSL para que la información enviada por el cliente sea segura.

- Poseer un certificado digital, esto garantiza que el sitio es real.

- El sitio debe brindar una autenticación para identificar a los usuarios; esto se realiza con una contraseña y usuario.

- Nunca acceder a páginas enviadas por correos electrónicos donde pidan información personal.

- Al finalizar de utilizar el sitio web, debe cerrarse la sesión y no solo finalizar el navegador.

- Utilizar contraseñas fuertes que contengan mayúsculas, minúsculas, números y signos especiales.

- Las preguntas de olvidar la contraseña o correos a donde enviar los datos no deben de ser públicos o fáciles de adivinar para elementos que nos rodean.

- No utilizar opciones de recordar contraseñas.

2.1.3. Secure Socket Layer (SSL)

Es un protocolo de seguridad desarrollado por la empresa *Netscape Communications* para lograr que la transmisión de datos entre un servidor y un usuario, o viceversa, a través de internet, sea completamente segura. El SSL es un protocolo abierto por lo que puede ser empleado por cualquier fabricante de aplicaciones para internet.

Proporcionan seguridad en la red, ofreciendo autenticación y privacidad para la información enviada entre extremos en la red mediante el uso de criptografía; el servidor demuestra su autenticación, los datos son codificados con algoritmos o descifrados, de modo que no pueden transformar la información en el camino que sigue el internet, garantizando que los datos no tienen ninguna alteración; este se desarrolla en las 3 fases que a continuación se describen:

- Realizar una negociación del algoritmo de encriptación que se usará para efectuar la transacción.

- Intercambiar claves públicas y validarlas basándose en certificados digitales.

- Cifrado simétrico del tráfico.

2.1.3.1. Fases

El cliente inicia enviando una lista de información de todos los algoritmos de encriptación que este soporta; el primero es el que el cliente prefiere, el servidor responde con la clave digital certificada e información sobre los sistemas encriptación que soporta.

El cliente selecciona un sistema de encriptación y desencripta la clave certificada; el servidor puede pedir al cliente un certificado para que la conexión sea segura mutuamente.

El cliente y servidor realizan el cifrado de una clave secreta, utilizándola a partir de una clave pública, que es descifrada por la clave privada de cada uno.

A continuación se describen ejemplos de software SSL gratuitos:

- *Open SSL*: realizado bajo código abierto, y compatible con muchos navegadores.

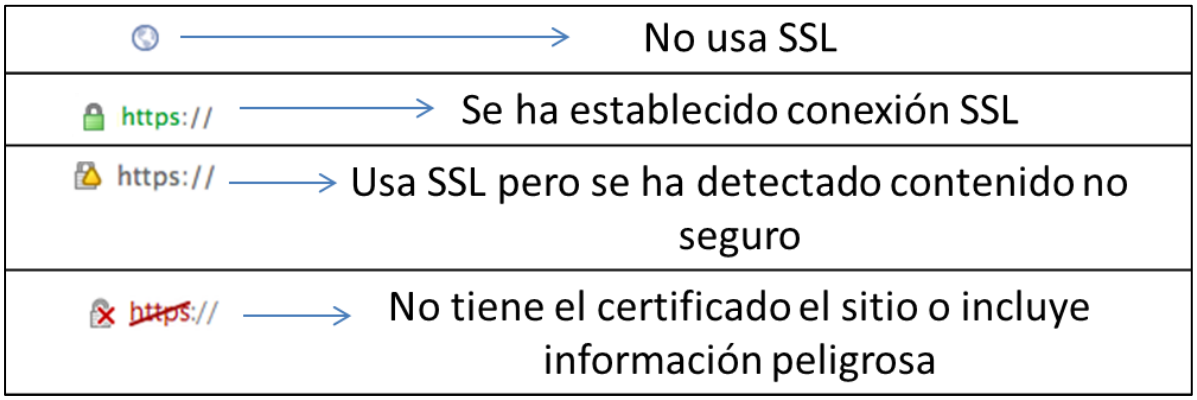
- *GnuTLS*: realizado con *software* libre con licencia GPL.

- *JSEE*: realizado en la máquina virtual de java e introducida en *Java Runtime Enviroment.*

2.1.3.2. Indicadores de seguridad en la web

Los navegadores que hoy proporcionan información se están implementando el protocolo SSL, presentando símbolo de un candado; si este presenta otro icono que no valide la seguridad sobre el navegador no se debe introducir información personal de ningún tipo; este puede informar si ya se venció el certificado o presenta una irregularidad.

Figura 2. **Indicadores de seguridad en un navegador**



Fuente: elaboración propia, con programa de Power Point.

2.1.3.3. *Hackeo* a empresa de certificados *SSL*

Un grupo de *hackers* logró engañar a las firmas digitales con certificados SSL falsos para *google, yahoo, Microsoft,* entre otras, logrando obtener información privada de los usuarios.

Estos obtuvieron certificados falsos de 9 sitios entre estos *Skype y Mozilla*; al momento de ser descubiertos fueron bloqueados y por ello permitían a un sitio web, hacerse pasar por el sitio original y obtener información encriptada como contraseñas de banco, correos, entre otros.

El hackeo fue rastreado en Irán; una empresa reconocida como *ComodoHack* se atribuyó el haberlo hecho, afirmando ser un *criptoanalista* que apoyan al régimen iraní.

Los actuales certificados que se están emitiendo, deberían brindar mayor seguridad durante las transacciones en el internet porque no se está implementando seguridad sobre una llave maestra, sobre revocar certificados, fraudes y prevenir certificados falsos a grandes empresas.

2.1.3.4. Cotizaciones de certificados SSL

- *VeriSign*

    - VeriSign protege los recursos de 500 compañías
    - 96% de los bancos utiliza VeriSign
    - 90% de los más famosos negocios de ecommerce usa VeriSign
    - 91% de los compradores online utiliza VeriSign
    - VeriSign realiza más de 175 millones transacciones por día

- VeriSign realiza un escaneado de malware en los sitios web
- Autenticación completa con una garantía de $1 000 000,00
- Cifrado de 40 bits como mínimo a 256 bits

- *GeoTrust*

  - Validación automática del nombre del dominio
  - Emisión rápida e instalación fácil con cifrado SSL de 256 bits
  - Compatibilidad con el 99% de navegadores
  - Compatibilidad con la mayoría de navegadores móviles

- Individuales

  - Estos son certificados pensados en garantizar seguridad 1 dominio con un certificado SSL

Tabla I. **Costos individuales de un certificado de seguridad**

Descripción	***Symantec***	***GeoTrust***
2 Certificaciones de seguridad en un sitio web por 2 años	$ 3 580,00	$ 3 580,00
2 Certificaciones de seguridad en un sitio web por 1 años	$ 1 990,00	$ 1 990,00
3 Certificados seguridad en un sitio web por 1 año	$ 2 985,00	$ 2 985,00
3 Certificados seguridad en un sitio web por 2 año	$ 5 370,00	$ 5 370,00

Fuente: datos proporcionados por Ing. Pedro Pablo Hernández.

- Multidominios

  - Estos son certificados diseñados para garantizar seguridad de 2 a más dominios con un certificado SSL

Tabla II. **Costos de multidominios de un certificado de seguridad**

Descripción	*GeoTrus*	*VeriSign Trusted*
1 Certificado SSL hasta de 5 dominios por 1 años	$359,00	$1 595,00
1 Certificado SSL hasta de 5 dominios por 2 años	$628,00	$2 891,00
1 Certificado SSL de 5 dominios por 3 años	$898,00	$4 391,00
1 Certificado SSL de 5 dominios por 4 años	$1 167,00	$5 760,00

Fuente: datos proporcionados por Ing. Pedro Pablo Hernández.

2.1.4. Firmas digitales

Una firma digital es la que aplica una función criptográfica asociada a un mensaje o documento de texto, y así que puede ser verificada por terceros, para poder identificar al firmante y la validez del documento firmado, y así garantizar un documento digital con un documento de papel que agilice procedimientos, y mejore la administración de la información.

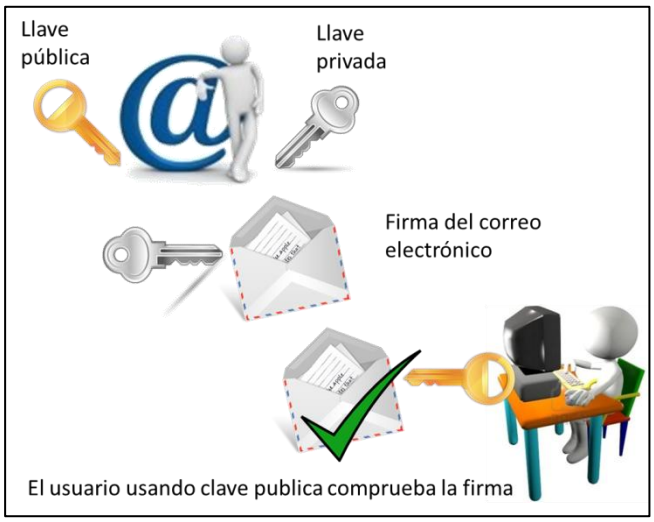
Las firmas digitales son implementadas para la distribución de *software*, transacciones financieras y así detectar la falsificación o modificación de un elemento ortográfico.

Ley de Firmas Digitales en Guatemala: esta ley obtuvo vigencia en octubre del 2008; permitió que empresas o personas individuales adquirieran su firma electrónica sin problemas.

2.1.5. Certificado digital

Son otorgados por autoridades de certificación; estas garantizan que una persona está asociada a una firma digital.

Figura 3. **Implementación de la firma digital**



Fuente: elaboración propia, con programa de Power Point.

Estos certificados son usados para comprobar que una clave pública pertenece a una persona o entidad, las que estas son aseguradas por una autoridad certificada.

Un certificado digital tiene validez legal y por ello tiene que garantizar integridad e identidad; emplea un método criptográfico usando una clave pública y otra privada, una autoridad correspondiente certifica que sean únicas y estén vinculadas.

El servidor firma utilizando su clave privada y lo envía a otros usuarios y estos con la clave pública del remitente.

Gracias al certificado digital se puede realizar cualquier proceso o gestión desde cualquier computadora evitando desplazarse, y así poder hacerlo en el momento más apropiado, dado que tiene validez jurídica; todos los formularios y documentos electrónicos son validados, ya que el usuario lo ha realizado.

Los elementos que tiene un certificado digital son:

- La identidad de la entidad
- La clave pública de la entidad
- Elementos del certificado (fecha que el certificado caduca)
- Nombre de la entidad que lo emite
- Firma autorizada

A través del navegador se puede asegurar si un certificado es válido o no, por ejemplo este puede caducar o no ser válido y el navegador nos puede ayudar a saber si un sitio es seguro.

2.1.6. Historial de visitas

Está presente si se ha visitado la página en una ocasión anterior; esto puede ser útil por si un sitio se está haciendo pasar por otro e incluso ha copiado todo el formato de la página que se está visitando; actualmente, la mayoría de navegadores pueden decir fácilmente si se ha visitado con anterioridad el sitio.

Esta información es guardada en memoria *cache* o en *cookies,* pero estos también se pueden eliminar.

Hoy en día *google* proporciona un historial web, únicamente se necesita una cuenta de *gmail* y esta se almacena de forma encriptada y finaliza al cerrar la sesión de la cuenta, protegiendo la privacidad.

2.2. Servidor de aplicaciones web

Un muestra de cómo se implementan estos servidores son los sitios de internet, donde las empresas popularizan información y es un punto de entrada para usuarios internos o externos, sobre una base de un servidor de aplicaciones; estos sitios también pueden almacenar accesos a información o servicios de manera segura, logrando que se pueda obtener a la información desde cualquier dispositivo.

2.2.1. Indicadores de rendimiento

Los indicadores de rendimiento proporcionan información de los recursos que están consumiendo en el servidor, y son los siguientes:

- Tiempos: estos representan el tiempo de respuesta, es decir, el tiempo en el que el servidor procesará la petición o transacción y devolverá la respuesta a la solicitud realizada.

- El *hardware*: este tema es de alta relevancia ya que se debe encontrar la forma de optimizar el *hardware* de los servidores, de tal forma que quede de una manera en la que se mantendrá disponible el servicio, a mayor cantidad de usuarios simultáneamente conectados. Con un *hardware* óptimo se podría aumentar el nivel de rendimiento general de los servidores, proporcionándole mayores recursos al servidor que tendrá la mayor carga de concurrencia.

- Los usuarios: la concurrencia de usuarios tiene un gran impacto en el rendimiento general de los servidores y por ende, repercuten directamente en los servicios proporcionados por el sistema de servidores. Deben existir formas o planes que definan claramente de qué forma se manejarán las concurrencias de los diferentes tipos de usuarios.

Se debe realizar la medición en función de las variables memoria, disco duro y procesador, para poder conocer el rendimiento y disponibilidad (garantizar que los usuarios puedan acceder a la información) del servidor aplicaciones Web.

Tabla III. **Indicadores de rendimientos del disco duro**

Disco duro	
• % de tiempo de disco • % de tiempo de escritura en disco • % de tiempo de lectura de disco • % de tiempo inactivo • Bytes de disco • Bytes de escritura en disco • Bytes e lectura en disco • Escritura en disco • Lectura en disco • Promedio de segundos de disco /transferencia	• Longitud promedio de la cola de escritura de disco • Longitud promedio de la cola de disco • Promedio de *bytes* de disco /escritura • Promedio de *bytes* de disco /lectura • Promedio de *bytes* de disco /transferencia • Promedio de segundos de disco /escritura • Transferencias de disco

Fuente: elaboración propia, con el uso de la herramienta *Performance Monitor.*

Tabla IV. **Indicadores de rendimientos de memoria**

Memoria	
• % de *bytes* confirmados en uso • Asignaciones de bloque no paginado • Asignaciones de bloque paginado • *Bytes* confirmados • *Bytes* de bloque no paginado • *Bytes* de bloque paginado • *Bytes de cache* • *Bytes* de lista de páginas libres y cero • *Bytes* de lista de páginas modificadas • *Bytes* disponibles • *Bytes* residentes de bloque paginado • *Bytes* residentes de cache del sistema • *Bytes* residentes de código del sistema • *Bytes* residentes de controladores del sistema	• Copias de escrituras • Entrada de páginas • Entradas libres de la tabla de páginas del sistema • Errores de *cache* • Errores de páginas • Errores de solicitud de cero • Errores de transición • Escrituras de páginas • *KB* disponibles • Lecturas de páginas • Límite de confirmación • *Mbyte* disponibles • Páginas de transición reasignadas • Páginas • Salida de páginas • Total de *bytes* de código del sistema • Total de *bytes* de controladores del sistema • Uso máximo de los *bytes* de cache

Fuente: elaboración propia, con el uso de la herramienta *Performance Monitor.*

Tabla V. **Indicadores de rendimientos de procesador**

Procesador	
• % de tiempo de *DPC* • % de tiempo de interrupción • % de tiempo de procesador • % de tiempo de usuario • % de tiempo en C1, C2 y C3 • % de tiempo inactivo • % de tiempo privilegiado	• Transiciones a C1/s, C2/s y C3/s • Velocidad de *DPC* • Interrupciones/s • *DPC* en colas/s (C1, C2, C3) potencia en estado inactivo del procesador

Fuente: elaboración propia, con el uso de la herramienta *Performance Monitor.*

2.2.1.1. Filtrado de variables

Todas las variables antes mencionadas están presentes en el procesador, memoria y disco duro, pero no todas ellas deben ser analizadas, porque hay gran cantidad de ellas que no varían, o no representan nada importante dentro del análisis de rendimiento.

Algunas variables frecuentes monitoreadas para medir el rendimiento de los servidores son:

- Memoria, *bytes* disponibles: este contador que indica cuánta memoria aún permanece funcionando cuando esta ya ha sido usada por varios procesos en cantidades grandes y memoria.

- Memoria, entrada de paginado/seg: este contador muestra la cantidad de páginas de memoria que se encuentran en memoria virtual.

- Memoria, falla en paginado/seg: esto sucede cuando un programa o proceso trata de utilizar memoria como parte de un grupo de trabajo y esta no se encuentra en dicha memoria.

- Memoria, lectura de paginado/seg: esta sucede cuando el contador cuenta el número de accesos y no el número de páginas. A partir de la cantidad de accesos que han tenido, estos se cargan a memoria virtual.

- Memoria, salida de paginado/seg: este contador muestra cuántas páginas han sido intercambiadas fuera del disco a memoria virtual.

- Memoria, paginado/seg: son las páginas requeridas durante el proceso que no se encuentre en *RAM* y tuvieron que ser leídas del disco duro.

- Disco duro, promedio de *bytes* escritos a disco: la cantidad de *bytes* que se escriben en una unidad secundaria de almacenamiento, desde la memoria principal u otro origen, es de suma importancia, ya que una anomalía en comportamiento habitual o normal del sistema operativo podría llevar a una caída drástica en el rendimiento general de una computadora.

- Disco duro, promedio de tiempo de ocioso: si una unidad de disco duro se encuentra ociosa, significará que no se está accediendo constantemente a la lectura o escritura de *bytes* sobre su superficie; una disminución en el porcentaje de tiempo ocioso que pasa el disco duro indicará un aumento en la tasa de lectura o escritura de *bytes*.

- Disco duro, *bytes* en la transferencia: la cantidad de *bytes* en la transferencia del disco duro especifica la información que era llevada desde esta unidad hasta la memoria principal (RAM).

- Disco duro, promedio de bytes leídos en disco: la cantidad de bytes que son leídos desde el disco duro para llevarlos a la memoria principal (RAM) o para utilizarlos para otro fin.

2.3. Indicadores web

- Tasa de rebote: es el porcentaje de personas que entran y salen de la página o sea abandonan el sitio inmediatamente al entrar y salen de un sitio web; cuando una página tiene una tasa de rebote muy alta, significa que su desempeño es muy bajo, porque no está atrayendo a nuevos usuarios.

- Tasa de conversión: es cuando los usuarios ha terminado de realizar una transacción, por ejemplo que han terminado de llenar un formulario, enviado un email, entre otros.

- Fuentes de tráfico: es de donde provienen las fuentes de tráfico; esto es de gran ayuda para saber si se está llegando al público objetivo; por ejemplo se podría saber la cantidad de tráfico que es redirigido desde *Facebook, twitter y blogs*, entre otros.

- Visitas: el número de visitas único, es medido a partir de las *ip* que visiten el sitio web.

- Palabras clave: son las razones por las que las personas están buscando un sitio web; ayudan a saber qué estaban buscando los usuarios cuando encontraron el sitio web; con este indicador se puede saber si han sido bien escogidas las palabras que describen el sitio web, o si por ejemplo una página está generando muy poco tráfico, tal vez sea mejor escoger otra palabra que lo describa.

2.3.1. Políticas de seguridad

Al hablar de un sitio web las medidas de seguridad deben ser tomadas en cuenta, dado que hoy en día ningún sistema está completamente seguro de ser hackeado por otro usuario.

Por ello al realizar una política de seguridad, se debe identificar su vulnerabilidad y fallas de la organización; así también deben de renovarse constantemente estas políticas de seguridad en función de un ambiente cambiante.

Las políticas de seguridad son el establecer las reglas y procedimientos que regulan a la organización para prevenir, proteger y manejar los riesgos que pueden venir de diferentes daños.

2.3.1.1. Evaluación de riesgos

Se debe evaluar la ponderación económica de que un desastre ocurra, así definir un costo de protección sobre esta falla o riesgo.

Se debe realizarse un análisis de la probabilidad, según el ambiente donde ocurra la contingencia, priorizar los problemas y el costo potencial.

Se debe conocer qué proteger, dónde y cómo, asegurando tener un costo que dé beneficios positivos.

Para identificar los problemas dentro del sistema se puede realizar una reunión con los principales miembros del equipo.

Planificar unas preguntas como por ejemplo:

- ¿Qué puede ir mal?
- ¿Qué tan frecuente puede ocurrir?
- ¿Qué consecuencias traería?
- ¿Cuánto tiempo puede permanecer el sistema fuera de línea y el costo por ello?
- ¿Qué tipos de ataques cubre el actual sistema y está adecuadamente preparado para nuevos ataques?
- ¿Qué sucederá si la seguridad es violada?
- ¿Quiénes pueden utilizar los recursos?
- ¿Qué protección se dará a la información sensitiva, para que esta permanezca aún en el sitio?
- ¿Quiénes son los responsables de restablecer el sistema?

- ¿Qué recursos se pueden proteger?

2.3.1.2. Niveles de riesgos

Los riesgos se pueden clasificar según un nivel de ocurrencia o importancia y el efecto en caso de ocurrir:

- Estimar el riesgo de pérdida del recursos
- Estimar la importancia de los recursos

2.3.1.3. Identificación de amenazas

Después de reconocer los riesgos, es necesario identificar cuáles son las vulnerabilidades que pueden dar de baja a algunos de los recursos más importantes.

- Seguridad física (problemas en el entorno)
- Seguridad lógica (amenazas en el sistema)
- Comunicación (amenaza en la red)
- Internas y externas (amenaza de las personas)

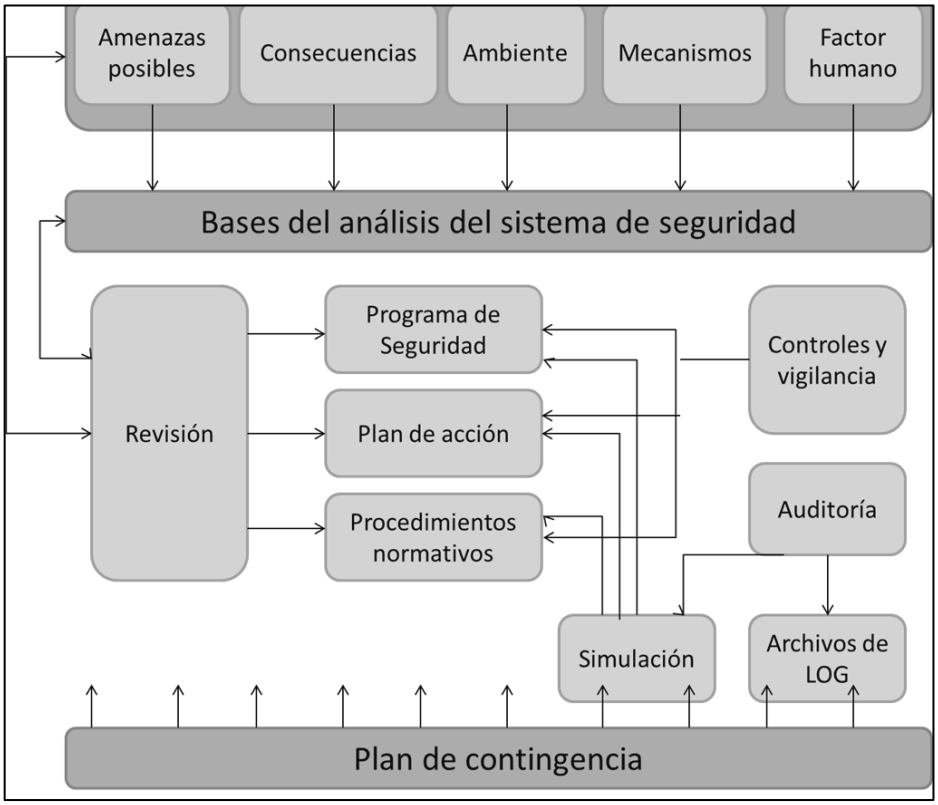
Al haber identificado las amenazas, puede ayudar a los administradores a tomar las medidas necesarias para identificar las herramientas y las técnicas para evitar estos ataques; por ello los administradores deben actualizar sus medidas continuamente.

Tabla VI. **Riesgo y factor de medida**

Riesgo	**Medida**
Robo de equipo informático	Alto
Robo de información de los usuarios	Alto
Fallas en los equipos	Medio
Accesos no autorizados	Medio
Fraude	Bajo

Fuente: elaboración propia.

Figura 4. **Procesos de seguridad de redes**



Fuente: elaboración propia, con programa de Power Point.

2.3.1.4. Evaluación de costos

Justificar los costos es uno de los elementos más importantes al planificar la respuesta ante los riesgos; por ello es necesario ponderar el riesgo en caso de ocurrir, pero es muy difícil de valorar debido a que no es un recurso intangible. Consiste en cuantificar los daños, y la posibilidad de la vulnerabilidad, además, en presentar un planeamiento de las acciones que se van a desarrollar a través de las siguientes preguntas:

- ¿Qué recursos proteger y cómo hacerlo?
- ¿Qué usuarios usan los recursos?
- ¿Qué tan importante es el recurso?
- ¿Cómo proteger los bienes de una forma económica y segura?
- ¿Qué importancia puede tener el recurso?
- ¿Qué tan reales son las amenazas?

Con este tipo de pregunta, se pueden justificar los costos para proteger los riesgos y entender cuál es más importante; las medidas que se proponen realizar son:

- Evaluación del factor humano y el medio donde se desempeña
- Mecanismos que llevan a cabo la tarea a realizar
- Las amenazas posibles y sus consecuencias
- Control periódico de las políticas de seguridad
- Asegurar el fiel cumplimiento de las políticas y procedimientos
- Realizar auditoría con los archivos *log*
- Simulación de los eventos que atentan contra la seguridad del sistema
- Revisión de las políticas de seguridad generadas en primera instancia
- Plan de contingencia en caso de que la política falle

2.3.1.5. Diseño

El diseño de seguridad debe ser realizado según el funcionamiento del servidor web; si este va a ser utilizado de manera local o en el mundo externo, (internet), sobre qué sistema operativo será instalado, qué servicios necesitará ya instalado, la estructura de red de la organización y costo de las licencias implementadas.

Los elementos que se van a configurar son:

- Instalación del sistema operativo.
- Análisis de la arquitectura de red que tiene la organización.
- Análisis de los servicios a los que los usuarios tendrán acceso y políticas de acceso a *firewall.*
- Analizar e instalar el *software* y servicios según los requerimientos de la entidad y los de red que esto implica.
- Configuración de reglas de salida del *firewall*, según necesidades de la organización.
- Creación de zonas desmilitarizadas para el manejo de servidores públicos.
- Configuración de entradas hacia servidores internos.

- Creación de *scripts* necesarios para que automáticamente cargue los servicios necesarios al iniciar el sistema.

2.3.1.6. Sistema operativo

Para mantener el sistema operativo actualizado con los últimos parches de seguridad y actualizaciones críticas implementadas, estas deben ser probadas antes en un ambiente de pruebas de producción.

Entre las acciones de control y manejo de cuentas de usuario, están: la definición de roles y restricciones, contraseñas eficientes y políticas de acceso; además se debe tomarse en cuenta lo siguiente:

- Otorgar acceso al sistema operativo solo al personal autorizado, con el mínimo de privilegios para realizar sus objetivos.

- Registrar la vigencia de todos los accesos concedidos a los usuarios del sistema operativo.

- Deshabilitar servicios del sistema operativo no necesario y evaluar la funcionalidad del mismo, mientras menos programas haya instalados, menos tendrán que actualizarse y habrá menos puertos abiertos.

Deberán aplicarse también las siguientes políticas de auditoría:

- Auditar inicios de sesión, los cambios de políticas y el acceso a objetos
- Controlar los errores
- Evitar la importación y exportación de datos
- Registrar las actividades de los usuarios en la red

- Encriptación de la información pertinente
- Manejar las contraseñas de acceso

Desinstalar los componentes no necesarios, entre ellos los productos de terceros, servicios innecesarios, *software* del sistema operativo por ejemplo un programa que necesita actualizarse; eso hace necesitar otro puerto abierto que provoca la necesidad de otra regla; es mejor tener lo mínimo instalado.

2.3.1.7. Seguridad en bases de datos

La seguridad en la base de datos estará relacionada con la integridad de ella; esta debe proteger la información, preservarlas de su destrucción y alteración, continuar su integridad logrando precisión y validez de los datos.

En términos de seguridad significa proteger la información de usuarios no autorizados que tengan acceso a ella; un cambio en la integridad significa proteger la información de personas que tienen autorización.

Deben tomarse en cuenta las siguientes consideraciones generales:

- Aspectos legales, sociales y éticos (la persona le pertenece la información que está solicitando o tiene derecho legal de conocer la información).

- Controles físicos (el computador está sobre protección, se encuentra bajo llave y no se le permite tener acceso).

- Cuestiones políticas (cómo la entidad u organización decide quién puede tener acceso a la información).

- Problemas operacionales (con la autenticación, cómo se mantienen las claves en secreto, con qué continuidad son cambiadas).

- Controles de *hardware* (las unidades de almacenamiento proporcionan alguna característica extra que brinde mayor seguridad y protección de la información).

Para acceder a la información de la base de datos esta debe poseer las siguientes medidas de seguridad para proteger la integridad y seguridad de la información.

- Seguridad en cuentas (logrando la validación de un usuario con una cuenta a la base de datos).

- Seguridad en acceso a objetos (con base en privilegios que permiten determinar a qué objetos puede tener acceso).

- Seguridad del sistema con gestión de privilegios globales (los roles pueden gestionar qué comandos del sistema tienen acceso o sea tener aquellos roles que sirvan para gestionar únicamente la parte del sistema en la cual tienen permiso).

- Para el manejo de *password* se debe realizar la encriptación de los campos para que no puedan ser descifrados.

2.3.2. Configuración recomendadas

Para cada elemento de riesgo que sea identificado, se debe prever que la vulnerabilidad no ocurra, en caso que suceda, no pueda producir muchos daños, es importante también monitorear parcialmente el estado y funcionamiento de cada uno de los elementos funcionando.

2.3.2.1. Negación de servicio (*DoS*)

La única forma de protegerse para estos ataques es mantenerse informado de los nuevos ataques y vulnerabilidades; además, mantenerse actualizado según los editores del *software* que se esté utilizando.

2.3.2.2. Respaldos de *backup*s incompleto

Planear y comprobar los *backup* debe ser una rutina que garantice saber cuándo existan fallos en el sistema SO, sobre cualquier acontecimiento que pueda ocurrir.

Las fallas en un sistema pueden ocurrir por distintas circunstancias por ejemplo: físicas (fallas en *hardware*, por ejemplo que falle el disco duro del CPU); diseño (problemas en el *software*, sistema operativo), funcionamiento (causadas por la intervención de un usuario, ya sea por configuraciones inapropiadas o mal empleo del *backup*) y del entorno (esto puede ser causado por desastres naturales, falla de corriente eléctrica, o una temperatura no apropiada).

Se pueden utilizar mecanismos de protección para garantizar una caída del sistema por pérdida de transacciones en la base de datos y se pueden utilizar mecanismos de protección de los datos, como:

- Utilizar UPS o fuentes para que no se interrumpa la corriente
- Raid o arreglo de discos espejo
- Poseer redundancia de componentes y de sistemas

También algunos *dbms* tienen lo que es *archivelog*, que permite realizar *backup* en caliente y estabilizar el sistema en caso de un fallo; pero en algún fallo del sistema este puede recuperar las últimas transacciones.

Reglas básicas al realizar un *backup*:

- Las copias de los ficheros deben estar en diferentes dispositivos que los originales.

- Mantener diferentes copias y control de versiones, y colocarlos en discos diferentes con diversos controladores.

- En el dbms, archivar los ficheros *redo log* (contiene todas las transacciones de la base de datos de aquellos cambios que no se han ejecutado) en disco y copiarlos a cinta.

2.3.2.3. Autenticación insuficiente

Al ser revelado datos internos de sesión, esta no debe contener la información interna, porque un atacante podría aprovechar esta vulnerabilidad.

Evitar la identificación de sesión predecible; el servidor no debe generar identificadores de sesiones predecibles ya que estos pueden ser suplantados por otros usuarios. Comprobar en todas las páginas, que el estado de sesión es correcto y que se poseen los suficientes permisos para acceder a él.

Al cerrar la sesión se debe asegurar que todas las variables han sido correctamente limpiadas, para que otro usuario no tenga permisos o acceso a datos.

2.3.2.4. Administración de autenticación

La autenticación no es segura por sí sola; es necesario utilizar la combinación de *https* en un servidor web; la autenticación nunca debe viajar en texto plano. Validar los campos de entrada, por ejemplo si este campo recibe un email, que este cumpla con las características; además, validar solo los caracteres permitidos.

Los formularios se pueden burlar de diferentes formas por ejemplo deshabilitando *javascript*, enviando formularios desde la web o implementando *firebug*. Las entradas de los usuarios puede contener caracteres no permitidos; por ello también se debe controlar convirtiendo la información que contenga estos problemas; se recomienda utilizar *prepared statements*.

2.3.2.5. Fuerza bruta

Implementar el uso de *captcha* es una alternativa de seguridad contra ataques de fuerza bruta; existe una variedad de proveedores de captcha de manera gratuita.

El uso de intentos fallidos sobre los de conexión del usuario, por ejemplo, cuando existan tres intentos fallidos este debe esperar 10 minutos.

2.3.2.6. Inyección de código SQL

Existen varias formas para evitar, por ejemplo se pueden filtrar las entradas de los usuarios sustituyendo la aparición de (comillas) y (dos comillas simples) incluso evitando que los usuarios puedan incluir caracteres al intercambiarlos por (diagonal seguida de comillas o dos comillas simples) o cualquier otro que pueda causar problemas. Estos filtros son sencillamente reemplazables utilizando sentencias de remplace, y usuarios distintos para sentencias *select, delete y update* o con los mínimos permisos en la base de datos, garantizando que cada sentencia solo ejecute lo permitido.

Otra manera de prevenir este ataque y agilizar también el proceso de análisis de las sentencias de SQL, es utilizar *prepared statements*, debido a que siempre se utilizan las mismas sentencias; muchas veces esto agiliza el tratamiento de transacciones.

2.3.2.7. *Cross-site scripting*

Existen casos para filtrar los caracteres de los *script* por ejemplo validar que cuando el usuario envíe datos se revisen los datos incrustados e informar al usuario con un *popup*.

Un modo de filtrar este problema es mirar el código de formulario y ver la dirección del *cgi* o *script* que procesa el formulario.

Otra forma de combatir este ataque es eliminar los caracteres peligrosos a su equivalente hexadecimal como < y >.

2.3.3. Mantenimiento

Es importante que cada elemento que se configure siga realizando sus funciones requeridas, por ello deben definirse acciones para reparar la unidad.

2.3.3.1. Mantenimiento preventivo

Realizando un mantenimiento preventivo puede aumentar la vida útil del equipo y así lograr disminuir los costos en reparaciones y la posibilidad de que el sistema no permanezca fuera de línea; asimismo, instalar *software* que ayude a monitorear el estado del equipo y prever una posible amenaza. Deben tomarse en cuenta las siguientes acciones:

- Aseguramiento de la calidad.

- Identificar fallas o posibles problemas que pueden surgir durante la vida útil del sistema y las consecuencias de no ser tratadas.

- Definir los procedimientos y planes de acción para el caso de una posible falla.

- Implementación de herramientas y medidas necesarias para la continuidad del proyecto.

Es necesario realizar un análisis cualitativo y cuantitativo de cada uno de los riesgos que pueden ocurrir al sistema y cómo en esta medida puede afectar al sistema así también medir la probabilidad de daño.

2.3.3.2. Plan de contingencias

El plan de contingencias ayudará a recuperarse después de que todas las medidas de seguridad tomadas para los equipos e información falle, y se hace necesario restaurarla.

Realizar un documento de plan de acción de riesgos e identificar todos los posibles riesgos y el alcance e impacto que estos tendrían en el sistema y desarrollar un plan para realizar todas las medidas y procesos que deben llevarse a cabo para implementar en caso de ocurra un riesgo.

2.3.3.3. Mantenimiento correctivo

Esto se define como la corrección de errores o fallas cuando estas ocurren; es muy importante proteger la información y llevar a cabo políticas y estándares; siempre existe la posibilidad de perder la información por factores externos que provoquen perder la información y no obtener respaldo; por ello siempre existe la posibilidad de que ocurra una falla.

Las fallas se pueden dar por muchas razones, se deben manejar los riesgos y las acciones necesarias para responder a ellos causando el menos daño posible, y la recuperación rápida del sistema.

El mantenimiento correctivo también se puede dar en el *software*; por ejemplo, cuando un *software* manda una versión de su sistema, dado que este contenía una vulnerabilidad y es necesario actualizar ya dado que cada día salen nuevas amenazas.

Dependiendo del tipo de impacto que tiene el problema, se debe de realizar un proceso para verificar cada uno de los posibles problemas sobre cada componente y si es posible, aislar el problema, para que solo una parte deje de funcionar.

2.3.3.4. Programado y no programado

El mantenimiento correctivo programado ocurre cuando se cuenta con el personal, la herramienta y el conocimiento necesario para llevar una reparación apta; este es afectado igualmente que el no programado; el fallo inmediato de la información hace detener el sistema, pero en diferencia con el programado presenta inconsistencia en un estado indeseable, y demora la recuperación del sistema.

2.4. Tareas de adaptación de la tecnología

En resumen, se ha integrado la seguridad que debe poseer tanto el cliente, como el servidor en servicio web, además de identificar y medir la seguridad del sitio para poder identificar si necesita llevar más medidas de prevención.

3. CONFIGURACIONES RECOMENDADAS

3.1. Gestor de bases de datos (MySQL)

Actualmente MySQL es una de las bases de datos más usadas debido a que es código abierto en compatibilidad e integración con las herramientas, utilidades disponibles, rendimiento, facilidad de uso, fiabilidad del sistema, pero la configuración de seguridad en este sistema siempre debe planificarse conforme a las mejores prácticas.

3.1.1. Políticas de seguridad

Define los elementos principales de los responsables directos o indirectos de un sistema, limita que se puede hacer y que no se puede realizar sobre el área de seguridad en la operación de un sistema.

3.1.1.1. Sistemas de seguridad

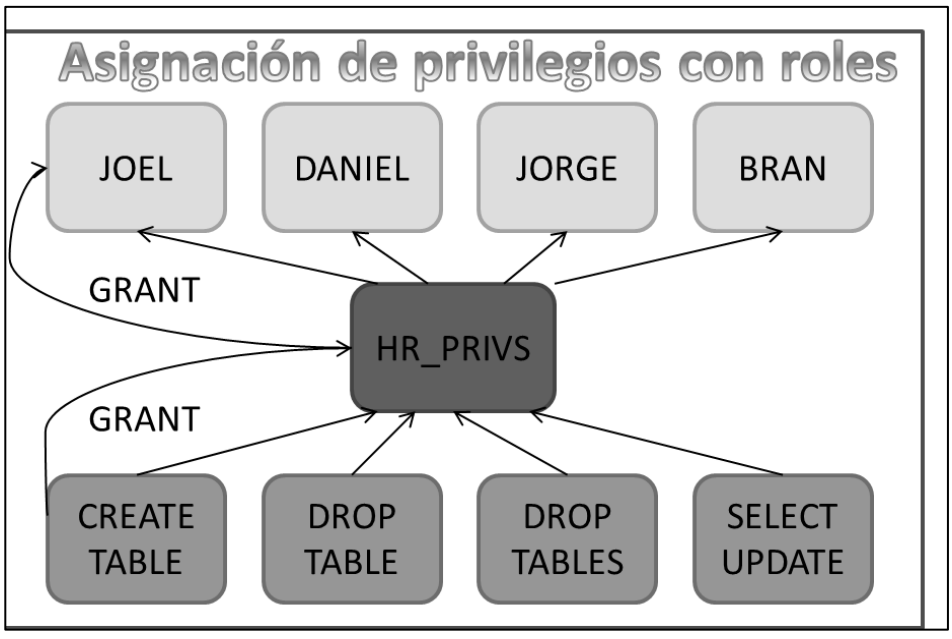
Entre los elementos de seguridad se definen los más importantes:

- Gestión de usuarios: dependiendo del tamaño del sistema y la cantidad de trabajo necesario para administrar la seguridad, el administrador del sistema debe ser el único usuario con los suficientes privilegios para crear, modificar o quitar usuarios en la base de datos, en caso de ser muy grande el sistema solo una persona de confianza pueden tener acceso para realizar esta acción.

- Políticas de seguridad en datos: incluye el acceso que puede tenerse a los datos y las acciones permitidas para cada usuario en un objeto. Por ejemplo, al usuario Daniel se le puede permitir *select, insert y update* pero no de *delete* en la tabla Empleados, estas también deben definir acciones que serán necesarias, por ejemplo, las acciones de auditoria. La seguridad general debe basarse en la sensibilidad de la información, en el caso de no serlo, se puede tener menor medida de seguridad; en caso contrario, deben tener un control estricto de acceso de los objetos.

- Políticas de seguridad del usuario: para todo usuario debe gestionarse una contraseña y privilegios sobre el sistema, por ejemplo debe observarse que los usuarios cambien sus contraseñas cada cierto tiempo, para evitar que las mismas sean reveladas.

Figura 5. **Asignación de privilegios**



Fuente: elaboración propia, con programa de Power Point.

- Desarrollo de aplicaciones por medio bases de datos prueba y producción: El desarrollo de aplicaciones no puede perjudicar a la base de datos de producción y de desarrollo de aplicaciones; estas no pueden ser perjudiciales en una base de datos de producción, después de realizar las pruebas correspondientes, se puede dar acceso a la base de datos, informando a los usuarios de producción.

- Roles y privilegios para los desarrolladores de aplicaciones: los administradores de seguridad pueden crear privilegios necesarios para desarrollo de aplicaciones típicas, por ejemplo, una función aplicación *developer* donde se pueda incluir la vista *create table, create view* y permisos de procedimientos del sistema.

- Gestión de políticas de contraseñas: la seguridad del sistema, se logrará si se mantienen las contraseñas en secreto; estas pueden ser vulnerables a robo, falsificación, entre otras. A continuación se describen las acciones que deben aplicarse

    - Bloqueo de cuentas al exceder un número de intentos fallidos en el sistema.

    - Especificar la duración máxima de contraseñas y la cantidad de tiempo para que estas expiren.

    - Historial de contraseñas para que los usuarios no puedan utilizar una misma contraseña durante algún periodo de tiempo.

    - Longitud mínima de contraseñas.

- Que las contraseñas no coincidan con un diccionario de datos o sean palabras sencillas.

- Políticas de auditoría: se deben crear políticas de auditoría de los elementos más importantes de la base de datos y nivel de detalle que se realizará sobre la misma.

- Administrar el espacio: las bases de datos siempre van en aumento y es importante asegurarse que se posee el suficiente espacio para el crecimiento del sistema; de considerarse, mantenerse al menos un 30% de su espacio libre disponible.

3.1.2. Configuraciones de seguridad

Las configuraciones en un servidor pueden variar según los elementos que utilice el sistema, y conforme el tiempo también vendrán nuevos riesgos, por lo que periódicamente deberán chequearse, para logar que el sistema siga protegido.

3.1.2.1. Asegurar el servidor web

Una de las mejores prácticas es separar el servidor de bases de datos del servidor de las aplicaciones, para reducir el número de puertos abiertos en el servidor y evitar accesos remotos; puede ocurrir que un usuario perjudique su base de datos y no tenga permisos para acceder a ella.

En el servidor debe configurarse elementos básicos de seguridad: antivirus, *antispam*, *firewall* y actualizaciones de paquetes del sistema operativo. La ubicación del servidor es de suma importancia, ya que este puede ser robado, inundado o afectado por una tercera entidad; se deben desactivar los servicios innecesarios, y asegurarse de que el servidor donde está instalado tenga estas medidas de seguridad al instalarse.

3.1.2.2. Desactivar el acceso remoto

Asegurarse que solo *host* definidos puedan acceder al ordenador; esto se puede realizar con *iptables* o *firewall* que posea el computador.

Para desactivar este parámetro desde MySQL se debe ir al archivo que carga la configuración de inicio; esta se encuentra en *"C:\Program Files (x86)\MySQL\MySQL Server 5.5\my.ini"* ó *"/etc/mysql/my.cnf",* dependiendo del servidor se añade el parámetro *"skip-networking"*, o se debe forzar para que solo escuche a *localhost,* añadiendo una dirección local únicamente.

Debe asegurarse que los usuarios que tengan acceso de forma remota tengan el menor número de permisos para conectarse al servidor como por ejemplo *"grant select, update, insert, delete on mydb.* to 'someuser'@'somehost';"*

3.1.2.3. Desactivar el uso de *LOCAL INFILE*

Se debe desactivar el comando *"LOAD DATA LOCAL INFILE"*, lo que ayudará a prevenir la lectura no autorizada de archivos locales, ya que esta es muy usada en vulnerabilidades de inyección de SQL.

Además, utilizando este comando, se puede acceder a otros archivos del sistema operativo como el comando *'SELECT load_file("/etc/passwd")'* para ello se debe agregar esta línea al archivo de configuración, *"set-variable=local-infile=0"*.

3.1.2.4. Cambiar el usuario de *root* y *password*

El nombre por defecto que trae MySQL es *root*, por lo, el que muchos usuarios mal intencionados pueden intentar acceder a él, por lo que es recomendable cambiar el nombre de usuario de *root*, como también su *password,* por una contraseña larga que contenga alfanuméricos y combinación de mayúsculas y minúsculas con este comando se puede cambiar el nombre de usuario *"RENAME USER root TO new_user;"* y para cambiar el password*" mysqladmin -u username -p passwordnewpass".*

3.1.2.5. Eliminar las bases de datos de prueba

MySQL por defecto trae una base de datos prueba, que puede ser punto de ataque de un usuario, por lo que se recomienda eliminar esta base de datos a través de este comando; se puede borrar *"drop database test;".*

3.1.2.6. Eliminar cuentas de usuarios obsoletas

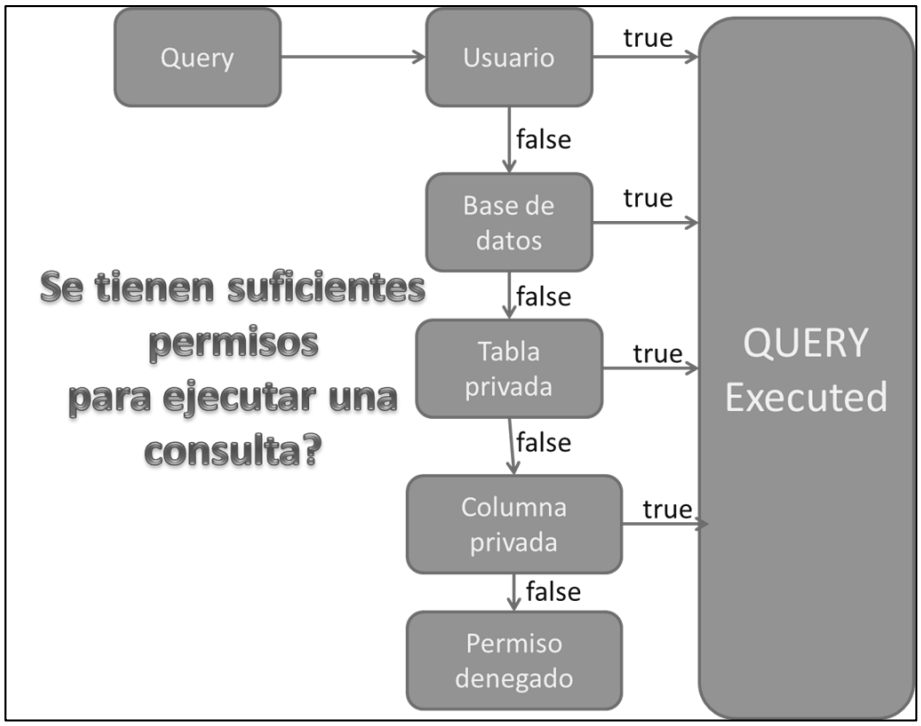
La base de datos de MySQL trae por defecto usuarios anónimos sin contraseñas logrando que cualquier persona pueda conectarse; para este caso se usa *"select * from mysql.user where user="";"* otra forma de realizarlo es *"SHOW GRANTS FOR "@'localhost' "* la forma de eliminar es únicamente ejecutando este comando *"DROP USER; ".*

3.1.2.7. Menor número de privilegios en el sistema

En el momento de desarrollo del proyecto, se realizan bajo usuarios con todos los permisos, pero esta es una mala práctica de ejecución de cualquier proyecto; se debe validar que se haya creado un usuario y grupo MySQL; las últimas versiones de MySQL 5.x solo se pueden instalar con estas medidas de seguridad.

Asegurarse que sobre el sistema solo los usuarios *root* y *MySQL* puedan acceder *"/var/lib/mysql"* y a *"/usr/bin/my*"*ya que otros usuarios pueden acceder a esta información.

Figura 6. **Permisos para ejecutar un *query***



Fuente: elaboración propia, con programa de Power Point.

3.1.2.8. Limitar los privilegios en la base de datos

Existen diferentes tipos de usuarios que utilizan la base de datos, por ejemplo el servidor web debe poseer un usuario que solo tenga acceso a las tablas; y únicamente poseer los permisos de *select, update, insert, delete*, funciones y procedimientos. Se debe realizar un análisis de los permisos y privilegios que realmente necesiten.

3.1.2.9. Habilitar el registro de *log*

Si el servidor no realiza muchas consultas se recomienda habilitar el registro de transacciones mediante el comando *"log =/var/log/mylogfile"* pero no en caso de realizar muchas transacciones, ya que puede sobrecargar al servidor además, se debe de comprobar quiénes tienen acceso a esta información; solo los usuarios de *root* y *MySQL* puedan tener acceso a ella, dado que esto contiene información confidencial.

3.1.2.10. Eliminar el historial de instalación

Durante el proceso de instalación existe mucha información sensible que puede ayudar a los *hacker* a obtener información sobre el sistema; esta se almacena en el historial del servidor y es útil si algo sale mal en la instalación donde los administradores pueden ver qué problemas ocurrieron, pero no son necesarios después de la instalación ejecutando este comando *"~/.mysql_history".*

3.2. *Test* de *Tunning MySQL*

Para mejorar el rendimiento de un servidor de *MySQL* existe un *script* escrito en *perl* que realiza pruebas a cualquier base de datos y muestra resultados además de dar información adicional sobre cómo resolver estos problemas.

Tabla VII. ***Test tunning* ejecutado**

Test de tunning de MySQL sobre el servidor de la Universidad Virtual
[root@uvchamilorackerhacker-MySQLTuner-perl]# perl mysqltuner.pl
>>MySQLTuner 1.2.0 - Major Hayden <major@mhtx.net>
>> Bug reports, feature requests, and downloads at http://mysqltuner.com/
>> Run with '--help' for additional options and output filtering
Please enter your MySQL administrative login: root
Please enter your MySQL administrative password:
-------- General Statistics --
[--] Skipped version check for MySQLTuner script
[OK] Currently running supported MySQL version 5.0.77
[OK] Operating on 32-bit architecture with less than 2GB RAM
-------- Storage Engine Statistics ---
[--] Status: -Archive +BDB -Federated +InnoDB -ISAM -NDBCluster
[--] Data in MyISAM tables: 114M (Tables: 10651)
[--] Data in InnoDB tables: 5M (Tables: 34)
[!!] BDB is enabled but isn't being used
[!!] Total fragmented tables: 18
-------- Security Recommendations ---
[!!] User '@ingenieria.usac.edu.gt' has no password set.
[!!] User '@localhost' has no password set.

Continuación de la tabla VII.

[!!] User 'root@127.0.0.1' has no password set.
[!!] User 'root@ingenieria.usac.edu.gt' has no password set.
-------- Recommendations ----------
General recommendations:
Add skip-bdb to MySQL configuration to disable BDB
Run OPTIMIZE TABLE to defragment tables for better performance
Enable the slow query log to troubleshoot bad queries
Set thread_cache_size to 4 as a starting value
Increase table_cache gradually to avoid file descriptor limits
Variables to adjust:
query_cache_size (>= 8M)
thread_cache_size (start at 4)
table_cache (> 64)

Fuente: datos proporcionados por Ing. Pedro Pablo Hernández.

Tabla VIII. **Análisis del *Test de Tunning MySQL***

Titulo	Recomendaciones
Estadísticas generales que proporcional el servidor.	La versión de MySQL es 5.0.77 La arquitectura del servidor es de 32 *bits*, por ello solo detecta 2 GB de RAM.
Estadísticas sobre el motor de almacenamiento	*MyISAM* es el sistema de administrador de base de datos de MySQL y en ella se tiene el mayor número de tablas, por lo que es recomendable cambiar las tablas de *InnoDB a MyISAM.*

Continuación de la tabla VIII.

Estadísticas sobre el motor de almacenamiento	*MyISAM* es el sistema de administrador de base de datos de MySQL y en ella se tiene el mayor número de tablas, por lo que es recomendable cambiar las tablas de *InnoDB a MyISAM.* Existen 18 tablas que necesitan ser optimizadas, esto se puede realizar con el comando *"myqlcheck -A –optimize"*
Recomendaciones de seguridad	El usuario *root*, con diferentes *Ip´s* o direcciones no tiene *password* definido. Al visualizar la tabla de usuarios con la siguiente consulta, podrá verse los usuarios y sus contraseñas *"SELECT User, Host, Password FROM mysql.user;"*, implementando *"SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');"*se puede actualizar su contraseña.
Recomendaciones generales	Se debe optimizar las tablas que están desfragmentadas; esto se puede realizar con el comando *"myqlcheck -A –optimize"*. Habilitar el registro de consultas; lentas para resolver las malas consultas; estas son provocadas por sentencias SQL que llevan mayor tiempo en ejecutarse, para habilitarlo al iniciar el servicio *mysqld"--log-slow-admin-statements".*

Continuación de la tabla VIII.

Recomendaciones generales	Es recomendable para cualquier servidor de MySQL utilizar conexiones permanentes de la base de datos, para evitar sobre carga; en caso que se realicen muchas conexiones se debe cambiar la variable *thread_cache_size*. Incrementar gradualmente las variables de *table_cache* en el archivo de límites de descripción.

Fuente: elaboración propia.

3.3. *Firewall*

Es un dispositivo *software* o *hardware* que funciona como cortafuegos entre redes, permitiendo o denegando los paquetes, o por ejemplo implementarlo como un dispositivo de seguridad para evitar que usuarios no autorizados puedan acceder a la información del sistema. Permite crear un filtro que acepta o deniega la comunicación que pasa por redes en función del servicio; el *firewall* permite si se realiza o no, además verifica si es que está entrando o saliendo y esto puede decidir enviarla o no.

3.3.1. Políticas de diseño de *firewall*

Se deben diseñar las políticas de acceso al *firewall* poniendo mucho énfasis en las amenazas y vulnerabilidades sobre una conexión insegura; para ello se deben responder las siguientes interrogantes:

- ¿Cuáles son los elementos que se van a usar para proteger a la red?

- ¿De quién se deben proteger sobre cualquier intento de acceso no autorizado sobre el exterior de la red hacia el interior?

3.3.1.1. DMZ

Las empresas crean diferentes redes con diversas políticas de seguridad y es necesario crear arquitecturas de *firewall* que aíslen las diferentes redes. Cuando una máquina de red interna es accesible por la red externa como los servidores web, servidores de correo electrónico y servidores FTP, es necesario crear una red separada que no comprometa la seguridad de la información, que hace referencia a una zona desmilitarizada que poseen las empresas, para aplicaciones que están disponibles al público.

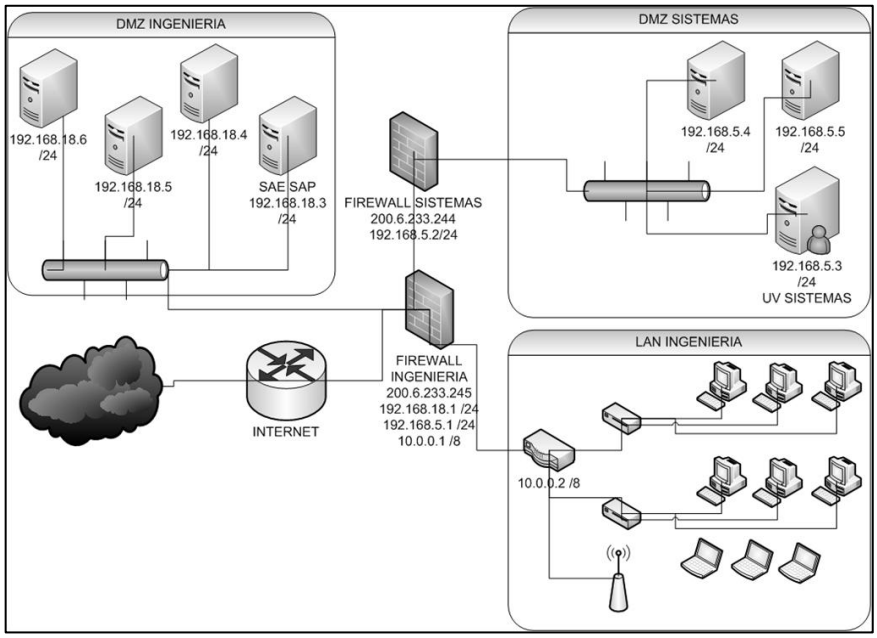
3.3.2. *Firewall* para la Escuela de Sistemas

La Escuela de Ciencias y Sistemas debería ser independiente, sus servidores tendrán que estar separados de los demás servidores y protegidos por un *firewall* dedicado. Al *firewall* de Ingeniería se le instalará una tarjeta adicional que se utilizará para conectar al *firewall* de sistemas en cascada.

Lo anterior implica una serie de cambios en la configuración de rutas y reglas. Universidad Virtual, actualmente tiene 2 direcciones IP una pública (200.6.233.245) esta dirección NO cambia porque es la dirección pública. Sin embargo tiene otra dirección IP que es la privada, imaginando que es (x1.y1.z1.w1),así es su actual funcionamiento. Toda petición que llega a la IP pública de sistemas (200.6.233.245) es redireccionada con NAT a x1.y1.z1.w1 que es la DMZ de Ingeniería.

Se necesita con el nuevo *firewall* que toda petición que llegue a la IP pública (200.6.233.245) sea redireccionada con NAT a x2.y2.z2.w2 donde esta nueva dirección pertenece a la nueva DMZ de sistemas. Es decir, que el *firewall* de Ingeniería tiene que redireccionar o *rutear* hacia el nuevo *firewall*.

Figura 7. **Esquema para la Escuela de Ciencias y Sistemas**



Fuente: elaboración propia, con de programa Visio.

3.3.3. Registro de *logs* en el firewall

El uso de las herramientas de seguridad necesarias (*firewall, switch, router*) en sí misma no asegura su red, pero los datos de seguridad de las herramientas necesitan ser analizadas y la seguridad de la información extraída debe ser reportada o se le avisará para asegurar que la red es asegurada. Por lo tanto, el análisis de los registros del *firewall* y otro dispositivo de seguridad y los registros es vital para la seguridad de la red.

El servidor de seguridad de los registros revela una gran cantidad de información sobre la amenaza en la periferia de la red y sobre la naturaleza del tráfico que entra y sale del *firewall*. El servidor de seguridad analizará los registros de información y proporciona información en tiempo real a los administradores, sobre la amenaza de los intentos de seguridad y para que con rapidez se pueda iniciar una acción de remediación.

Los servidores de seguridad de *log* son esenciales para el reconocimiento de los ataques, resolución de problemas y reglas de *firewall*, y darse cuenta de una actividad inusual en la red. Se deben incluir las normas de registro en el servidor de seguridad para que puedan ser generados; sin embargo, las reglas de registro debe venir antes de cualquier norma aplicable en terminación (una regla con el objetivo de que decida el destino del paquete, como ACCEPT, DROP, o REJECT).

3.3.4. Importancia de la bitácora

Una bitácora puede registrar información acerca de eventos relacionados con el sistema o servicios que la genera. La información que debe registrase en una bitácora es la siguiente:

- Fecha y hora
- Direcciones IP origen, destino y la que genera la bitácora
- Usuarios
- Errores

A continuación se presentan las ventajas del uso de bitácoras:

- Recuperación ante incidentes de seguridad
- Detección de comportamiento inusual
- Información para resolver problemas
- Evidencia legal
- Es de gran ayuda en las tareas de cómputo forense

3.3.4.1. Procedimiento para DMZ SISTEMAS

- Agregarle el módulo de la tarjeta de red al *firewall* de sistemas
- Definición de red y *subnetting* a la *DMZ* de Sistemas
- Asignación de las *Ip´s* a los diferentes equipos
- Ruteo a las otras redes de ingeniería (*DMZ sistemas, LAN Ingeniería*)
- Análisis y evaluación de políticas de *firewall* de sistemas
- Creación de *Nat* para ocultar la red interna de sistemas
- Las reglas de los *firewalls* deben de configurarse de la siguiente manera:
    - La regla de *Nat* se debe de aplicar al *firewall* interno de modo que la red interna de sistemas se traduzca a la dirección *ip* pública
    - Adición de la regla de acceso de *ip´s* a la red interna en el *firewall* de sistemas.

- En el *firewall* externo de Ingeniería se debe de definir reglas de acceso de la ip pública de sistemas, permitiendo el acceso hacia la red externa de internet, para permitir o denegar acceso por medio de diferentes puertos al sistema.

- Evaluación de puertos abiertos con un *sniffer*.

- Revisión del *log* del *firewall* semanalmente, por posibles infiltraciones.

3.4. *Windows Server* 2003 *vs CentOS*

El anterior sistema operativo utilizado por la UV fue Windows server, pero fue cambio dado por un fallo en el servidor y actualmente se encuentra instalado y configurado el sistema operativo *CentOS*.

3.4.1. Ventajas Windows Server 2003

Es fácil de implantar, administrar y usar gracias a que posee una interfaz familiar, y además asistentes que facilitan las configuraciones sobre servidores; también incluye herramientas para implementar mejoras como los servicios de instalación remota que ayudan a crear copias del sistema.

Proporciona herramientas que ayudan a ajustar el diseño e implementación de necesidades organizativas de red. Ayuda a administrar su red de forma proactiva con políticas de seguridad, tareas automatizadas y simplificaciones de actualizaciones. Garantiza la fiabilidad, disponibilidad, escalabilidad y rendimiento dado que tiene funciones como lo son memoria agregadas en caliente se permite agregar clústeres.

3.4.2. Desventajas de Windows Server 2003

Debido a que este es uno de los sistemas operativos más utilizado es muy conocida la cantidad de virus y de fallos que este puede ocasionar y comprometer la seguridad. Los precios de la licencia y elementos a configurar van a depender de ella, incrementando el costo del producto y las utilidades que se pueden lograr.

El sistema de archivos utilizado por Windows realiza mala utilización de los recursos causando fragmentación y recalentamiento de la computadora; la única opción para resolver el problema es desfragmentar la información que provoca pérdidas de recursos y tiempo.

3.5. Sistema Operativo *CentOS*

Es un sistema operativo libre que fue basado en *Red Hat Enterprise (RHE)* lo que significa que es compatible con todos los paquetes y elementos de RHE, *CentOS* incorpora por defecto muchas aplicaciones que son muy utilizadas para las empresas y facilita su utilización; está disponible para su descarga vía web, *ftp o torrent* de la página web de *CentOS* y existen muchos otros sitios que permiten descargar el *software*.

3.5.1. Elementos básicos de seguridad

Seguridad física, todo servidor debe ser encerrado en centros de datos con controles de seguridad y se debe validar y verificar la seguridad con *scripts* que validen que la misma ha sido ejecutada.

Muchas veces la información puede perderse y es primordial tener copias de seguridad sobre el sistema, además de validar que estas se realicen de una manera íntegra, realizando pruebas sobre las mismas.

Mantener particiones separadas del sistema entre ellas */boot, /usr, /var, /tmp, /home,* para evitar por ejemplo que un registro temporal como */var o /tmp,* dejen sin espacio al sistema.

IPtables es un dispositivo de *firewall* por *software* que está vinculado al *kernel* de Linux; esta aplicación puede ser detenida o iniciada según los requerimientos de la empresa; es necesario que los administradores de red creen un *script* que inicie o pare el *iptables* con las reglas implementadas. Se recomienda practicar las reglas de *firewall* en servidor de pruebas para hacer uso de la herramienta como *iptraf,* para comprobar que esté en funcionamiento y si las conexiones *tcp/ip* se establecen o no.

SELinux, es *software* incluido en la distribución de Linux que trae una serie de políticas de seguridad que fue desarrollada por la agencia de seguridad nacional (NSA), integrada en el *kernel 2.6.x* en módulos de seguridad de Linux.

SELinux proporciona un sistema con control de acceso obligatorio (MAC) bajo el estándar *DAC (discretionary Access control, DAC),* donde un proceso o aplicación se ejecuta con un usuario; con ello tiene permisos a objetos, archivos y otros.

SELinux define acceso a privilegios de transición a usuarios, aplicaciones, procesos y archivos del sistema; dando por ejemplo una política entre qué objetos puede configurar y utilizar un usuario.

Se debe conocer qué servicios y *software* se tienen instalados en el sistema, sino, no se podrá garantizar la seguridad es muy importante realizar un listado de todos los paquetes que no son necesarios en el sistema o que no cumplen con sus políticas de seguridad; por ejemplo, eliminar Samba del sistema si este no se está utilizando; además, es una buena práctica no tener instalados los paquetes de escritorio (X server) o para el desarrollo del sistema.

Se debe tener una política de seguridad para gestionar los procedimientos de actualización de seguridad de Linux; además, evaluar y probar las vulnerabilidades más críticas y ser tratadas en periodo corto de tiempo, además de llevar control sobre qué parches se han implementado y en qué periodo de tiempo se realizó.

Es importante detectar y cerrar los puertos que no son necesarios en la red; además, eliminar los procesos de arranque del sistema y estos llegarán a ser un montón, pero se deben desactivar los servicios de nivel de ejecución así como la detección y configuración de un nuevo *hardware* etc.

Se debe garantizar que todas las secuencias de arranque contenidas en los comandos de inicio o *inittab* sean legitimas de su computadora, por ejemplo, eliminar la opción *CTRL-ALT-SUPR* para evitar reinicios accidentales.

El sistema debe expirar las contraseñas de forma automática; además, habilitar seguridad en el uso de contraseñas para que gente que utiliza contraseñas sencillas no puedan violarle su seguridad, además de implementar el bloqueo de las cuentas a un número de intentos fallidos.

3.6. Servidor http/https Apache

La configuración de apache se encuentra en el archivo *"/etc/httpd/conf/httpd.conf",* es recomendable siempre hacer una copia de este directorio antes de hacer cualquier modificación, además registra los errores que se registran en el servidor sobre el archivo, *"/var/log/httpd/error_log";* las últimas entradas en el servidor sirven para saber qué ha sucedido.

3.6.1. Server Root

Especifica el directorio que contendrá el contenido web, que por defecto es configurado en *"/etc/httpd"*.

Apache inicia por defecto con el usuario *root* y es recomendable cambiarle el usuario por defecto, debido a las acciones que este puede realizar; los archivos y directorios es necesario protegerlos y para ello, se deben cambiar las del propietario y las autorizaciones; por ejemplo, para cambiarle los permisos a un propietario o grupo es con el comando *chown* cuando se realiza sobre consola bajo la combinación de tres dígitos que representan permisos del propietario, del grupo y el resto. Los números pueden producir los significados de no permisos, ejecución, escritura y lectura.

No se puede permitir que un usuario diferente de *root* pueda modificar cualquier archivo que ejecute o escribir en ellos; todo el sistema puede quedar desprotegido además debe protegerse el fichero *log*, ya que por ejemplo, puede ser remplazado por un enlace simbólico y podría sobreescribir sobre un archivo con datos arbitrarios y el fichero *log* permanecería con datos falsos.

3.6.2. Server Side Includes (SSI)

El *SSI* es un riesgo potencial para cualquier administrador debido a que por ejemplo aumenta la carga del servidor ya que tienen que ser analizados por el servidor apache, además plantean riesgos con archivos cgi, pues podrían ejecutarse bajo permisos del usuario y el grupo a que pertenece Apache.

Una solución es deshabilitar la opción de ejecutar *script* en programas y paginas SSI, remplazando los archivos "*Includes*" con "*Includes NOEXEC*".

3.6.3. CGIs

Ejecutar archivos *cgi* en general produce un agujero en la seguridad, solo en caso que se tenga plena seguridad en los usuarios, ya que estos *scripts* podrán ejecutarse y alterar el sistema.

Los archivos *cgi* solo deben utilizarse bajo las siguientes circunstancias:

- Si tiene total confianza sobre las personas que escribirán archivos cgi sobre el sistema.

- No tiene usuarios y visitas sobre el servidor.

3.6.4. Restringir a los usuarios

Uso de *.htaccess* permite definir directivas de configuración de cada directorio, evitando la posibilidad de editar el archivo de apache.

Las directivas de configuración cumplen con las siguientes acciones:

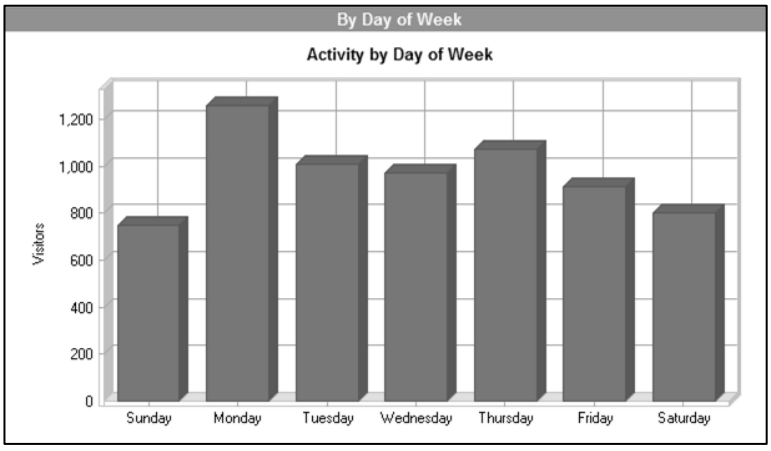
- Restringen el acceso a directorios
- Restringen *ip´s* o *ISP*
- Crean *URLs* amigables
- Formulan redirecciones estáticas

3.6.5. *Log* del servicio httpd visitas

Se realizó el análisis de *logs* que presenta el servidor de la Escuela de Ciencias y Sistemas, mostrando el número de visitas de parte del servidor y los navegadores utilizados por los usuarios.

Debe evitarse que los usuarios puedan modificar este archivo y anulen la seguridad del sistema, para ello Apache contiene la opción que evitará sobreescribir este archivo, y con él, las anulaciones, inclusiones y accesos.

Figura 8. **Actividades de los días de la semana en la Universidad Virtual**



Fuente: elaboración propia, con base a la herramienta *WebLog*.

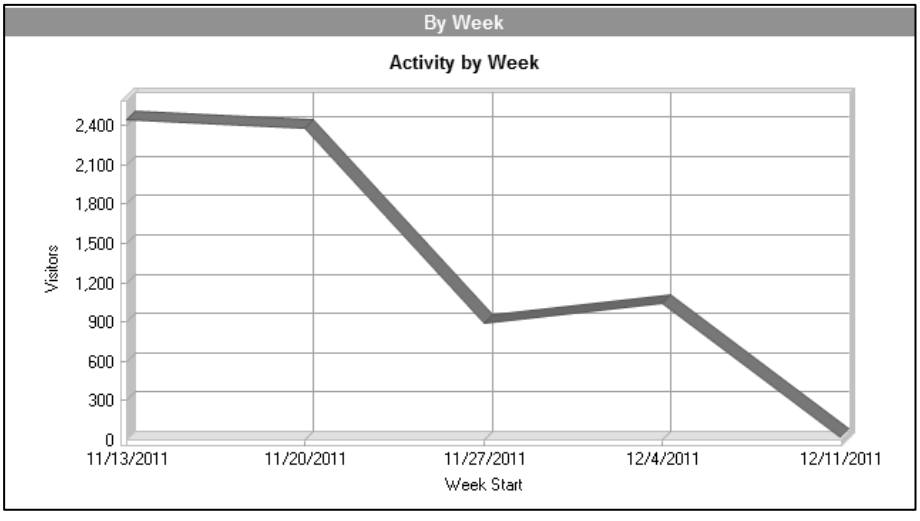
Los días que más se utiliza la Universidad Virtual, según estadisticas, son los días lunes con un total de vistantes de 1258.

Tabla IX. **Visitantes por día de la semana**

Día	Accesos	Visitas de Página	Visitantes	Ancho de banda
Dom.	1740	1549	749	11578
Lun.	3069	2857	1258	20722
Mar.	2424	2186	1010	15865
Mie.	2259	2082	972	15627
Jue.	4769	4507	1072	32974
Vie.	2314	2141	913	14218
Sab.	1981	1751	800	13291
Total	18556	17073	6774	124558

Fuente: elaboración propia.

Figura 9. **Actividades semanales en la Universidad Virtual**



Fuente: elaboración propia, con base en la herramienta *WebLog*.

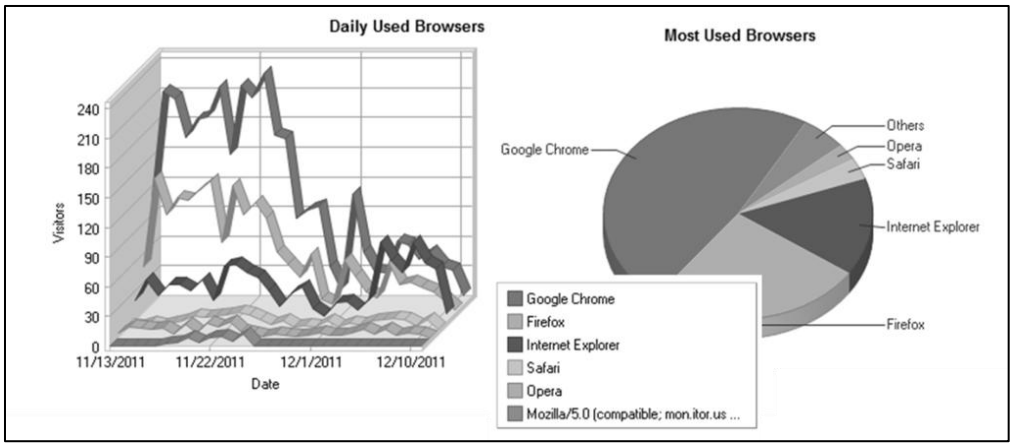
Tabla X. **Actividad semanal**

Semana	Accesos	Vistas de página	Visitantes	Ancho de banda
13/11/2011-19/11/2011	8202	7661	2449	58285
20/11/2011-26/11/2011	5610	5227	2376	39753
27/11/2011-3/11/2011	2313	2068	891	13948
4/12/2011-10/12/2011	2393	2089	1039	12396
12/11/2011-12/17/2011	38	28	19	174
Total	18556	17073	6774	124558

Fuente: elaboración propia.

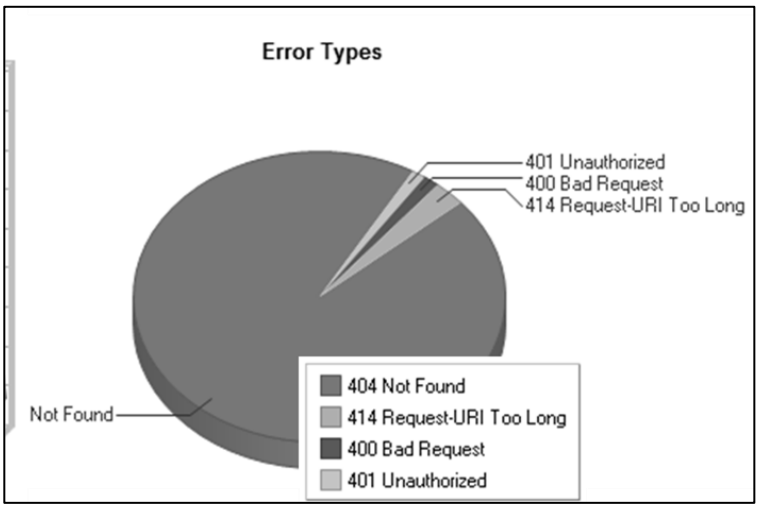
Según los archivos analizados, los navegadores más utilizados son google chrome, el cual tiene un total de 3,306 visitantes con 47.86% y Firefox con 3980 con 25.88%.

Figura 10. **Navegadores más utilizados en la Universidad Virtual**



Fuente: elaboración propia, con base en la herramienta *WebLog*.

Figura 11. **Errores en el *log* de visitas de la Universidad Virtual**



Fuente: elaboración propia, con base en la herramienta WebLog.

3.6.6. *Log* del servicio httpd errores

Se realizó el análisis de *log*s que presenta el servidor de la Escuela de Ciencias y Sistemas, mostrando los errores presentados por el servicio de httpd por parte del servidor.

Los archivos *log* fueron proporcionados por el administrador de la Universidad Virtual, con un total de 2268 eventos, que fueron realizados durante estos 2 días durante el mes de diciembre del 2011, donde fue analizada la cadena eventos y se incluye el análisis de lo encontrado.

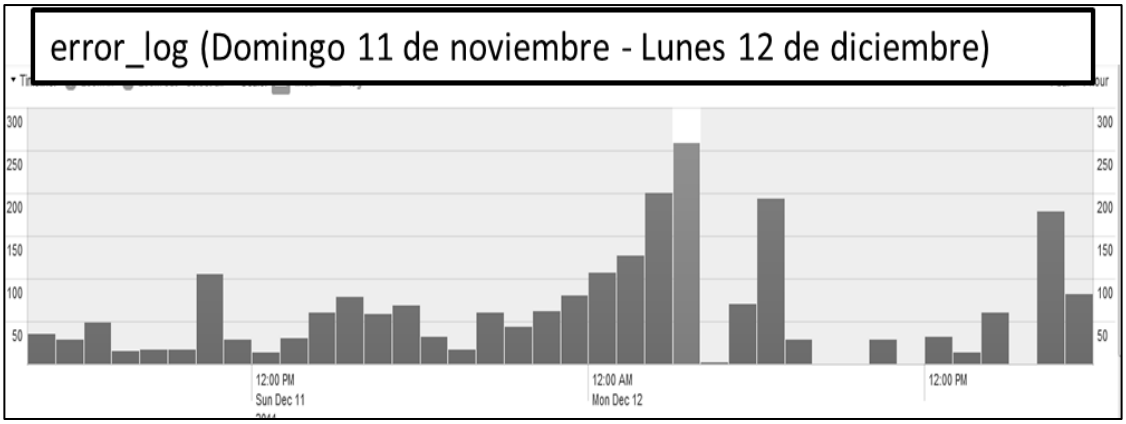
Los archivos *log* fueron proporcionados por el administrador de la Universidad Virtual, con un total de 2268 eventos que fueron realizados durante estos 2 días durante el mes de diciembre del 2011, donde fue analizada la cadena eventos y se incluye el análisis de lo encontrado.

Tabla XI. **Errores sobre el *log* de visitas de Apache**

No	Error	Accesos
404	*404 NotFound:* este es error en el navegador, que indica que no ha sido posible comunicarse al servidor.	1613
414	*414 Request-URI Too Long*: el servidor rechaza la petición porque la solicitud es mayor de la que el servidor está dispuesto a interpretar.	47
400	*BadRequest:* el servidor web, no respeta completamente el protocolo http; por lo que no puede entender la solicitud procesada.	24
401	*401 Unauthorized*: el error indica que se debe iniciar sesión con usuario y contraseña válida para acceder.	20
	Total	1704

Fuente: elaboración propia, con base en la herramienta *WebLog*.

Figura 12. **Registro de errores por hora en la Universidad Virtual**



Fuente: elaboración propia, con base en la herramienta *Splunk*.

Se llevó control, en qué horarios se realizó mayor número eventos para poder detectar en qué periodos de tiempo se produjo mayor cantidad de errores; gracias a ello se pudo detectar amenazas que están presentes en el servidor dado una vulnerabilidad conocida por los atacantes.

Además se buscó qué errores eran los más concurrentes en la plataforma y estos estuvieron relacionados con que las funciones que se están utilizando en la plataforma de Dokeos ya están en desuso en la versión PHP que se tiene instalada y esta informa que se debe actualizar esta parte del código.

Tabla XII. **Errores registrados sobre PHP en la Universidad Virtual**

Fecha 11 Diciembre	**Conteo**	**Fecha 12 Diciembre**	**Conteo**
04:00-5:59	60	0:00-01:00	232
06:00-7:59	63	02:00-2:59	200
08:00-9:59	32	03:00-4:59	260
10:00-11:59	233	05:00-05:59	70
12:00-13:59	44	06:00-06:59	194
14:00-15:59	90	07:00-9:59	29
16:00-17:59	137	10:00-11:59	28
18:00-19:59	47	12:00-13:59	46
20:00-21:59	103	14:00-15:59	239
22:00-23:59	141	18:00-18:59	81
Total	950	**Total**	1379

Fuente: elaboración propia.

Tabla XIII. **Errores producidos en el *log* sobre la Universidad Virtual**

No	Errores	Conteo
1	Function session_register() is deprecated in /var/www/dokeos/main/inc/lib/main_api.lib.php	600
2	Function session_register() is deprecated in /var/www/dokeos/main/inc/lib/main_api.lib.php, referer: http://sistemas.ingenieria-usac.edu.gt/	192
3	Function session_register() is deprecated in /var/www/dokeos/main/inc/lib/main_api.lib.php, referer: http://ecys.ingenieria-usac.edu.gt/	128
4	Assigning the return value of new by reference is deprecated in /var/www/dokeos/main/inc/lib/pear/HTML/Table.php	74
5	Assigning the return value of new by reference is deprecated in /var/www/dokeos/main/inc/lib/pear/HTML/Table.php	74
6	Assigning the return value of new by reference is deprecated in /var/www/dokeos/main/inc/lib/pear/HTML/Table.php	74
7	Assigning the return value of new by reference is deprecated in /var/www/dokeos/main/inc/lib/pear/HTML/Table.php	74
8	Assigning the return value of new by reference is deprecated in /var/www/dokeos/main/inc/lib/pear/HTML/Table.php	74
9	Assigning the return value of new by reference is deprecated in /var/www/dokeos/main/inc/lib/pear/Pager/Pager.php	74

Fuente: elaboración propia.

3.6.6.1. Amenaza detectada

El servidor fue afectado el 11 y 12 de diciembre del 2011; esta amenaza fue procedente de diferentes ip´s originarias de varias partes del mundo, sobre el registro de errores en diferentes horas de tiempo.

Los *hackers* buscan una vulnerabilidad sobre el sitio web, al encontrarlo lo usan para atacar servidores, y posteriormente pedir dinero para restablecer el sistema. Para ello averiguan cuál es el archivo (PHPMYADMIN) administrador de MySQL sobre *PHP,* dado que tiene un error que no todos conocen; probando todos los posibles nombres, se descarga un archivo *scripts/setup.php* y por medio de él descargan un archivo, lo ejecutan y se descarga una consola y el archivo con extensión "*.txt*" que automáticamente se cambia de directorio y se renombra a "*.php"*; y con ello obtiene acceso al servidor.

Tabla XIV. **Soluciones para evitar el ataque sobre *PHPMYADMIN***

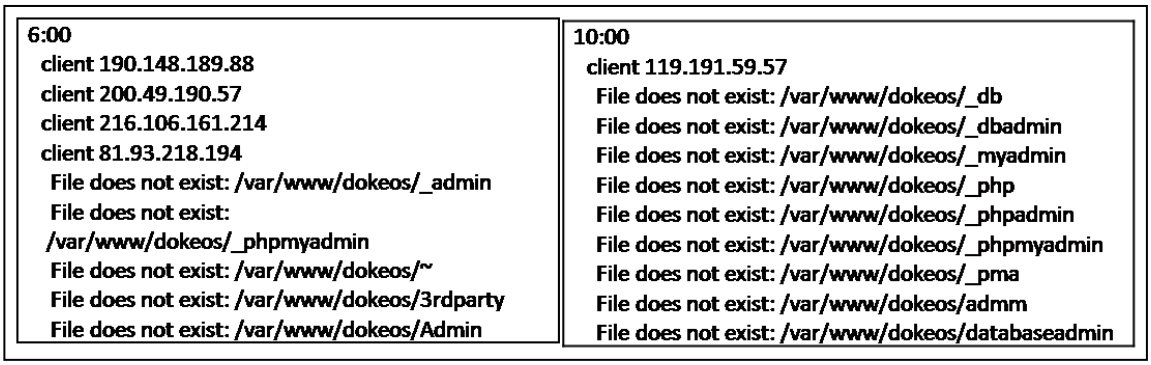
Restringir el acceso al administrador de *MySQL* sobre *PHP*
Configurar el *htaccess* para restringir el acceso a solo a una dirección ip. *<Directory /var/www/AdminFolder/>* *Options FollowSymLinks* *Order Deny, Allow* *Deny from all* *Allow from 128.98.2.4 #ip privado únicamente* *</Directory>*
Activar *ModSecurity*
Instalar un módulo de seguridad de Apache, que funciona como un *firewall* de aplicaciones web, haciendo un completo registro de tracciones, a partir de reglas configuradas para monitorear la seguridad.

Fuente: elaboración propia.

Se monitorearon los eventos registrados como actividad fuera de lo normal; estos monitoreos fueron realizados en diferentes horas y con diferentes ip; las actividades que demostraron mayor actividad sobre la vulnerabilidad, fueron realizadas sobre las horas 6:00 am del 11 de diciembre y 10:00 am del 12 de diciembre; los demás registros no muestran alta actividad sospechosa como describe esta vulnerabilidad.

Este evento se produjo durante varias horas del día, iniciando desde las 7:00 am del 11 de diciembre y el último evento registrado fue a las 17:00 pm del 12 de diciembre del 2011.

Figura 13. **Ip´s que realizaron el mayor número de transacciones**



Fuente: elaboración propia, con base en *log* de la Universidad Virtual.

Las *ip´s* que demuestran acción maliciosa provienen de diferentes partes del mundo, realizando diversas cantidades de transacciones en diferentes horas del día, donde destaca el IP 81.23.218.194 proveniente de España, con 89 transacciones de error.

Estos atacantes utilizan *proxies* para evitar dejar rastro y que las personas hallen el hecho malicioso, con ello pueden utilizar múltiples Ip´s y ocultar la IP real.

3.6.7. Ataques en los *log*s

Se pueden detectar ataques partir de análisis de registros de errores, cuando algún usuario quiera acceder a un recurso que no está disponible de forma pública, y puede sufrir la ejecución de comandos del servidor, inyección de virus, iframes ocultos con código, accesos no permitidos, obtener las contraseñas, visión de datos privados, borrado de datos, etc.

Figura 14. **Lugar de donde proviene IP atacante**

No	IP Vulnerable	Cantidad	Rango	Ubicación
1	client 81.93.218.194	89	81.93.218.192 - 81.93.218.255	Spain Madrid
2	client 119.191.59.57	33	119.0.0.0 - 119.255.255.255	China Shandong
3	client 157.55.16.230	6	157.54.0.0 - 157.60.255.255	United States Redmon
4	client 157.55.17.144	6	157.54.0.0 - 157.60.255.255	United States New York
5	client 61.38.186.50	4	61.32.0.0 - 61.43.255.255	Korea, Republic Of Seoul
6	client 95.108.150.235	4	95.0.0.0 - 95.255.255.255	Russian Federation Moscow
7	client 157.55.39.87	2	157.54.0.0 - 157.60.255.255	United States Redmond
8	client 193.47.80.46	2	193.47.80.0 - 193.47.80.255	France Paris
9	client 207.46.199.53	2	207.46.0.0 - 207.46.255.255	United States New York
10	client 65.52.109.146	2	65.52.0.0 - 65.55.255.255	United States New York
11	client 66.249.72.213	2	66.249.64.0 - 66.249.95.255	United States Mountain
12	client 119.63.196.57	1	119.0.0.0 - 119.255.255.255	Japan Naha Baidu
13	client 119.63.196.84	1	119.0.0.0 - 119.255.255.255	Japan Naha Baidu
14	client 157.55.16.178	1	157.54.0.0 - 157.60.255.255	United States Redmond
15	client 180.76.5.49	1	180.0.0.0 - 180.255.255.255	China Beijing Beijing
16	client 180.76.5.55	1	180.0.0.0 - 180.255.255.256	China Beijing Beijing
17	client 180.76.5.58	1	180.0.0.0 - 180.255.255.257	China Beijing Beijing

Fuente: elaboración propia, con base al *log* de la Universidad Virtual.

Por ello por cada recurso que presente un *log* que no demuestra una actividad inusual o pidiendo un recurso que no es de uso público se debe analizar:

- ¿Qué error genera?
- ¿Está intentando acceder a un recurso?
- ¿Cuándo se realizó la petición y con qué frecuencia se realizó?
- ¿Se puede detectar la ubicación? y ¿qué país lo realizó?

3.7. PHP

Php.ini, es el archivo de configuración que contiene aspectos sobre el funcionamiento del intérprete PHP; se encuentra en el directorio raíz, todo lo que inicia con punto y coma es ignorado, es inicialmente cargado cada vez que se reinicia el servidor; por ejemplo para configurar directivas, el mostrar errores y advertencias sobre que ejecuta el servidor e impedir la ejecución.

3.7.1. Autenticación y autorización

Se debe proyectar un test de *captcha* ante cualquier intento de sesión fallido, además de no almacenar contraseñas de los usuarios en cookies, es conveniente manejar *SSL* para evitar cambios en la información durante la llegada al destino.

En caso de que se utilicen preguntas de recuperación, asegurarse que los usuarios no utilicen preguntas con respuestas fáciles de adivinar por otra persona. Al finalizar las sesiones, destruir todos los datos y no solo borrar las *cookies*, dado que un usuario podría recuperarla y reutilizar la sesión.

3.7.2. Nomenclatura de los archivos

Un usuario puede querer obtener datos como la contraseña de la base de datos, o contraseñas para acceder al panel de actividades y muchas veces no utilizan la extensión correspondiente, lo que puede provocar que sean accesibles por los usuarios del sistema, y nunca se deben poner extensiones que no interprete el PHP.

3.7.3. Error *reporting*

Esta directiva, muestra los errores que se registran en tiempo de ejecución a través de niveles de seguridad, su valor es un entero y representan constantes predefinidas como errores o advertencias de los que no se logra recuperar el intérprete; aunque es bastante útil para reconocer errores en el servidor, puede provocar que este se convierta en vulnerable, dado que brindará información del sistema para un usuario mal intencionado, esto se puede evitar fijando "*error_reporting*" a 0.

3.7.4. *Register globals*

Esta directiva permite que las variables que proporciona el cliente se puedan registrar como variables globales en una aplicación web; por defecto, en la versión 4.1.0, ya se encuentra deshabilitada, en sí ella no es una vulnerabilidad sino que representa un riesgo en seguridad; si funcionan con *register_globals* activado.

Un ejemplo de esto sería: un usuario podría declarar una variable que no ha sido inicializada y pasarla a través de una URL con la intención de realizar una acción.

3.7.5. Validación de entradas

Nunca se debe confiar en las entradas de los usuarios, además de que todas las validaciones no solo se deben hacer del lado del cliente mediante *javascript,* estas pueden ser muy útiles pero no seguras, ya que la información tiene que pasar por múltiples servidores*, proxy* y *firewall,* de manera que la información puede variar durante el cambio, además de filtrar los caracteres especiales de una cadena y agregar una barra invertida a los caracteres como comillas y comillas dobles.

3.8. Configuración de seguridad de la Universidad Virtual

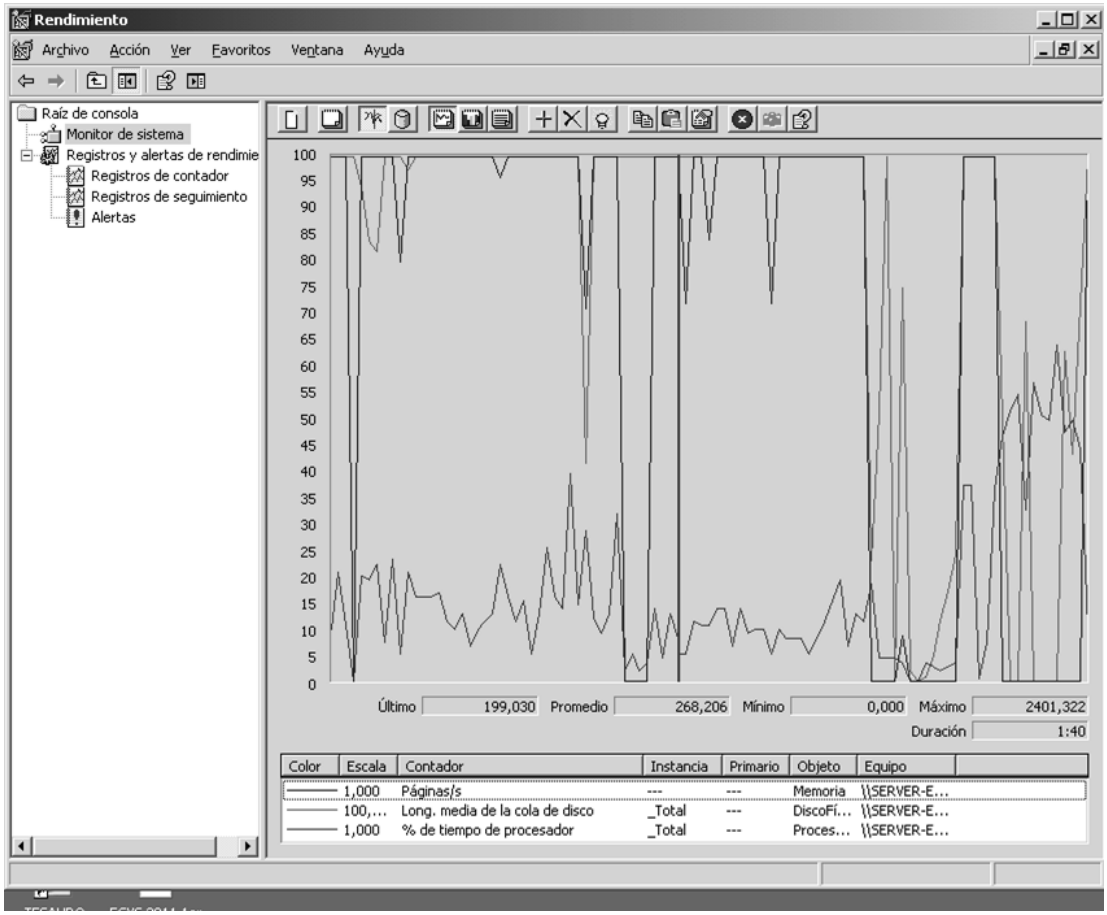
El administrador de la Universidad Virtual dio las credenciales para poder tomar información sobre el servidor, donde se pude comprobar el estado de la Universidad Virtual, además de las políticas que tenía configuradas.

3.8.1. Motivos de fallo

El servidor que contenía la Universidad Virtual, a las dos horas de estar funcionando tiraba fallo de memoria, pero debido a esto, no hubo ningún problema para sacar *backup* u obtener toda la información del sistema.

La figura siguiente demuestra el rendimiento de la Universidad Virtual donde la memoria se mantenía en alto índice de uso de la memoria a partir de la paginación, el servidor podía observarse que mantenía poco mantenimiento y es recomendable que se proceda a darle limpieza de ese modo, que pueda resolver los problemas de memoria que presenta el servidor.

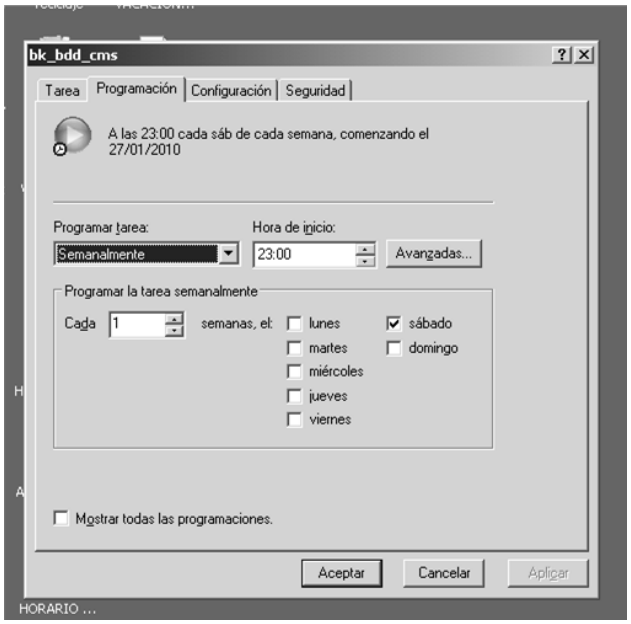
Figura 15. **Rendimiento de la Universidad Virtual**



Fuente: fotografía sobre el rendimiento en el servidor de la Universidad Virtual.

Se había realizado un *backup* semanal del sistema que se realizaba de forma automática para los sitios web de la Escuela de Ciencias y Sistemas, los días sábados a las 11:00 de la noche; estos se almacenaban en una partición diferente del sistema operativo.

Figura 16. ***Backup* semanal de la Universidad Virtual**



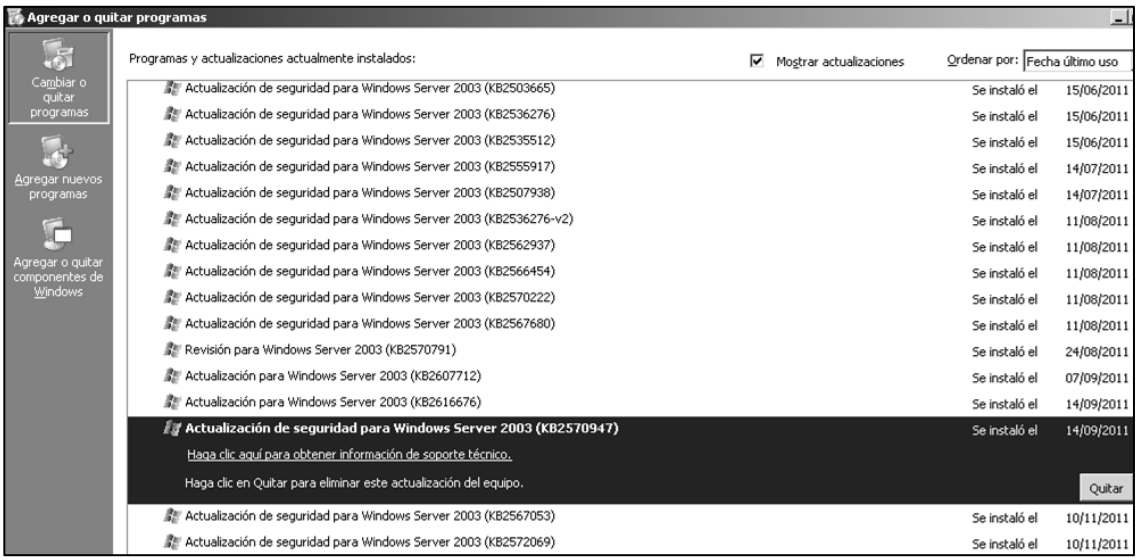
Fuente: fotografía del *backup* en el servidor de la Universidad Virtual.

La Universidad Virtual tenía todos los parches de seguridad antes de su fallo el 14 de septiembre del 2011, y este servidor no falló por un problema de seguridad, sino por falta de mantenimiento ya que, este no tenía asignado ninguna tarea de mantenimiento y el servidor se encontraba en funcionamiento continuo.

3.8.2. *Log* de errores del sistema

Los servicios y programas en el servidor que contenía el sitio web, tienen configurados los *logs* de errores ocurridos durante la compilación del sistema, estos pueden ayudar a conocer porque falló por la que falla el servidor y a partir del análisis del mismo, corregir los errores producidos.

Figura 17. **Actualización de seguridad de la Universidad Virtual**



Fuente: fotografía de las actualizaciones en el servidor de la Universidad Virtual.

3.8.3. MySQL logs

MySQL instalado en la Universidad Virtual, contenía muchos elementos de errores, el administrador de la Universidad Virtual mencionó que se debía a la cantidad de tablas que necesitaba el sistema de *e-learning* debido a la cantidad de cursos que manejaba la Universidad; por cada uno de ellos era necesario crear 20 tablas o más y esto ayudaba a que se produjeran más fallos en el servidor. Algunos de los errores producidos según los *logs* encontrados, se debían a la versión de MySQL instalada, y no a la configuración del mismo, resaltados por los usuarios de MySQL en la web como *bugs* de la versión.

La mayor de cantidad de errores se presenta sobre el puerto 3002, según el *log* de la herramienta Visnetic.

La mayor cantidad de errores fueron presentados durante el mes de junio, presentando un total de 2130 visitas, mostrando bastante variación con los meses anteriores, por lo que es recomendable revisar por qué falló.

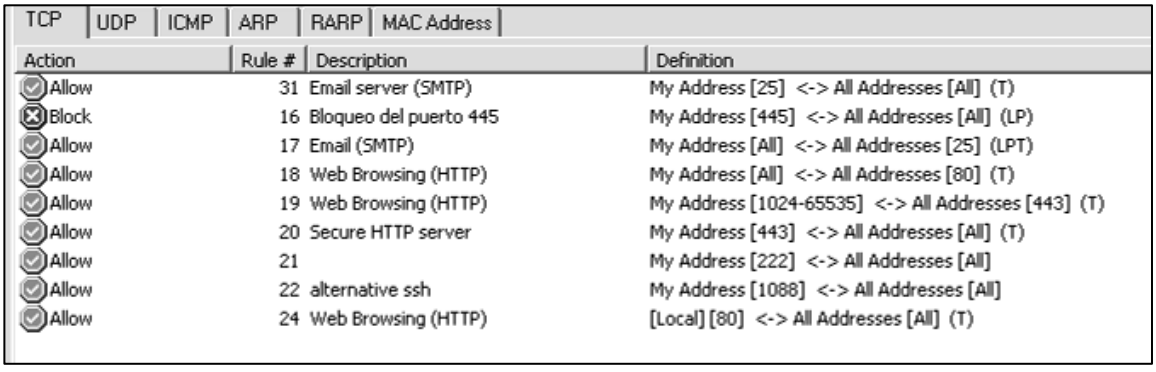
3.8.4. Firewall

La Universidad Virtual tiene instalado el *firewall Visnetic*, este solo tiene habilitados los servicios que implementa la Universidad Virtual, entre ellos: servicio smtp (protocolo de correo electrónico), http (protocolo de páginas Web), shh (protocolo para acceder a máquinas remotas).

3.8.5. PHP logs

Sobre el servicio de PHP se pudo observar que las fallas producidas recientemente sobre el mismo, no se debían a errores de seguridad del sistema, sino errores producidos por la concurrencia de accesos, o fallos sobre los servicios del sistema con *MySQL*.

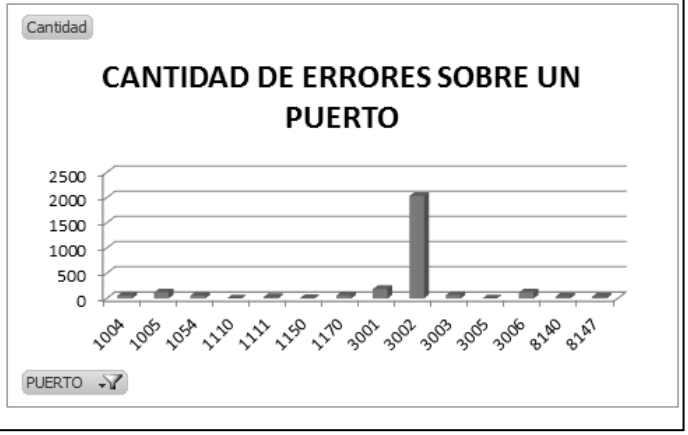
Figura 18. **Servicios habilitados en la Universidad Virtual**



Fuente: fotografía del *firewall* en el servidor de la Universidad Virtual.

Figura 19. **Errores de *log* sobre la herramienta Visnetic**

PUERTO	Cantidad
1004	59
1005	123
1054	59
1110	1
1111	29
1150	8
1170	64
3001	190
3002	2036
3003	70
3005	3
3006	123
8140	38
8147	33
Total general	2836

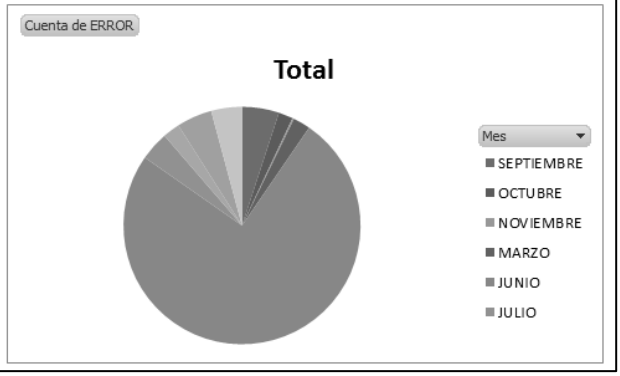


Fuente: elaboración propia.

La tabla muestra los errores más concurrentes en el archivo de *log* del *firewall*, la mayor cantidad de entradas de errores fueron sobre *"HTTP URL extensión not allowed"*; esto se debe a que un usuario intenta acceder a una dirección no permitida.

Figura 20. **Errores mensuales sobre el *log* de la herramienta Visnetic**

MES	Cuenta de ERROR
SEPTIEMBRE	143
OCTUBRE	56
NOVIEMBRE	6
MARZO	67
JUNIO	2130
JULIO	112
FEBRERO	66
ENERO	137
AGOSTO	119
Total general	2836



Fuente: elaboración propia.

Tabla XV. **Cantidad de errores en el *firewall***

No	*Error*	Conteo
1	*Device 1: Address=0.0.0.0*	59
2	*Device 1: Address=0.0.0.0 no longer in use*	59
3	*Starting firewall version 2.2.6.*	59
4	*Using configuration file "G:\herramientas\firewall\reglasECYS.rul"*	59
5	*Device 2: Address=192.168.18.20*	54
6	*66.249.71.105: HTTP URL extension not allowed: 'GET /function.getimagesize HTTP/1.1'*	36
7	*Firewall stopped at system shutdown.*	29
8	*66.249.68.123: HTTP URL extension not allowed: 'GET /function.getimagesize HTTP/1.1'*	28
9	*190.149.42.218: HTTP URL extension not allowed: 'GET /language/en-GB/en-GB.ini HTTP/1.1'*	13
10	*190.149.42.218: HTTP Method not allowed: 'OPTIONS / HTTP/1.1'*	12
11	*190.149.42.218: HTTP Method not allowed: 'TRACE / HTTP/1.1'*	12
12	*66.249.71.76: HTTP URL extension not allowed: 'GET /function.getimagesize HTTP/1.1'*	12

Fuente: elaboración propia.

3.9. Tareas de adaptación de la tecnología

Según las medidas de seguridad implementadas en el servidor, se analizaron los servicios instalados para verificar si este se encuentra en un estado correcto y prever posibles contingencias sobre estos servicios; durante la investigación falló el servidor por falla de *hardware* que procedió al cambio del servidor actual y nueva implementación del sistema operativo cambiando el servidor actual.

4. CONFIGURACIÓN ACTUAL DE LA UNIVERSIDAD VIRTUAL

4.1. Antecedentes de la Universidad Virtual

Los estudiantes de la Escuela de Ciencias y Sistemas han publicado diferentes problemas o inconvenientes que se han encontrado en la Universidad Virtual.

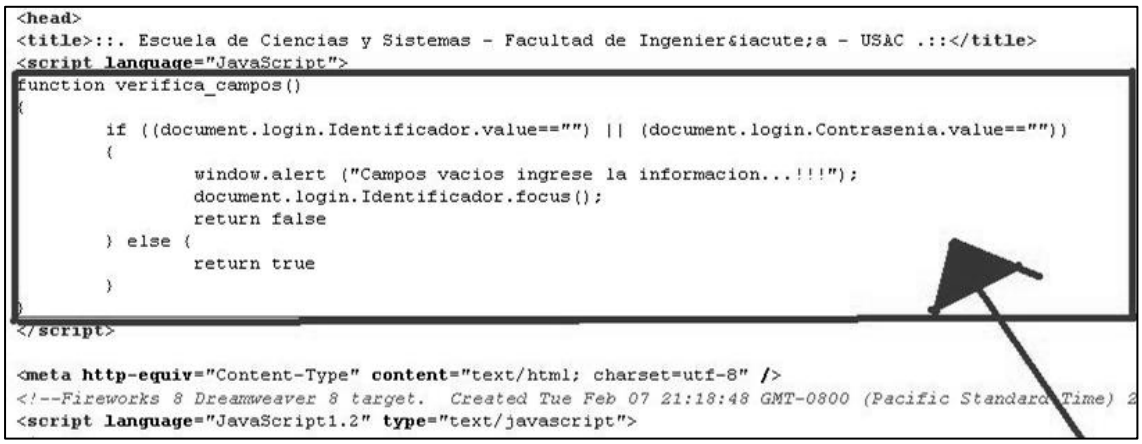
4.1.1. La antigua Universidad Virtual e Inyección SQL

El 11 de diciembre del 2006 un estudiante de la Facultad de Ingeniería de la Escuela de Ciencias y Sistemas publicó en su blog, que había logrado descubrir una vulnerabilidad de la Universidad Virtual de ese periodo.

El estudiante publicó un ejemplo sobre código de *javascript* donde indicaba que las entradas de usuario y contraseña están vacías; esto no es una vulnerabilidad a menos que no se realicen las validaciones de las entradas de parte del servidor, ya que estos se deben validar tanto del cliente como por parte del servidor.

No demostró nada únicamente una validación de parte del cliente que es código público; este código solo facilita la interacción entre un navegador y el usuario.

Figura 21. **Imagen de código de *javascript* de parte del cliente**



```
<head>
<title>::. Escuela de Ciencias y Sistemas - Facultad de Ingenier&iacute;a - USAC .::</title>
<script language="JavaScript">
function verifica_campos()
{
        if ((document.login.Identificador.value=="") || (document.login.Contrasenia.value==""))
        {
                window.alert ("Campos vacios ingrese la informacion...!!!");
                document.login.Identificador.focus();
                return false
        } else {
                return true
        }
}
</script>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<!--Fireworks 8 Dreamweaver 8 target.  Created Tue Feb 07 21:18:48 GMT-0800 (Pacific Standard Time) 2
<script language="JavaScript1.2" type="text/javascript">
```

Fuente: http://www.blogdelprofe.com/2006/12/11/la-uv-es-insegura/. Consulta: agosto de 2011.

En la figura 25, un usuario que no tomó las medidas de seguridad al escribir contraseña fuerte. Una aplicación puede ser segura. Pero si los usuarios no realizan las medidas de seguridad correspondientes, ninguna aplicación puede brindad seguridad; esto se describe en el capítulo 2, en indicadores de seguridad en un sitio web del lado del cliente.

En la figura 25, se muestra la prueba de un catedrático que realmente no tiene cursos asignados. El catedrático, al identificarlo, no era parte de la Escuela de Ciencias y Sistemas, y lo que sucedió es que antes, La Escuela de Ciencias y Sistemas se prestaba a otras escuelas porque no todas tenían este servicio. El resultado fue que no se exponía información confidencial pues el usuario está creado, pero nunca tuvo cursos asignados que pudieran ser visualizados.

Figura 22. **Ingreso al sistema de un usuario no autorizado**

Fuente:http://www.blogdelprofe.com/2006/12/11/la-uv-es-insegura/. Consulta: agosto de 2011.

Esta amenaza no influyó en ningún momento en la disponibilidad del servicio, la autenticación del curso fue cambiada por el Administrador de la Universidad Virtual, que respondió a la vulnerabilidad en 4:30 horas después de señalar la amenaza, resolviendo la vulnerabilidad.

En conclusión, esta vulnerabilidad no demostró riesgo alto, debido a que no tuvieron acceso a la administración de la Universidad Virtual donde se puede tener ingreso a información de estudiantes, profesores que no son información pública o manipular información que cambie resultados de algún curso; además no se mostró una serie de pasos sobre cómo provocar esta vulnerabilidad; el único curso que demostró tener acceso por no ser parte de la Escuela de Ciencias y Sistemas no tenía curso asignados y no pudo manipular ningún tipo de información, demostrando que la integridad del sistema no fue manipulada.

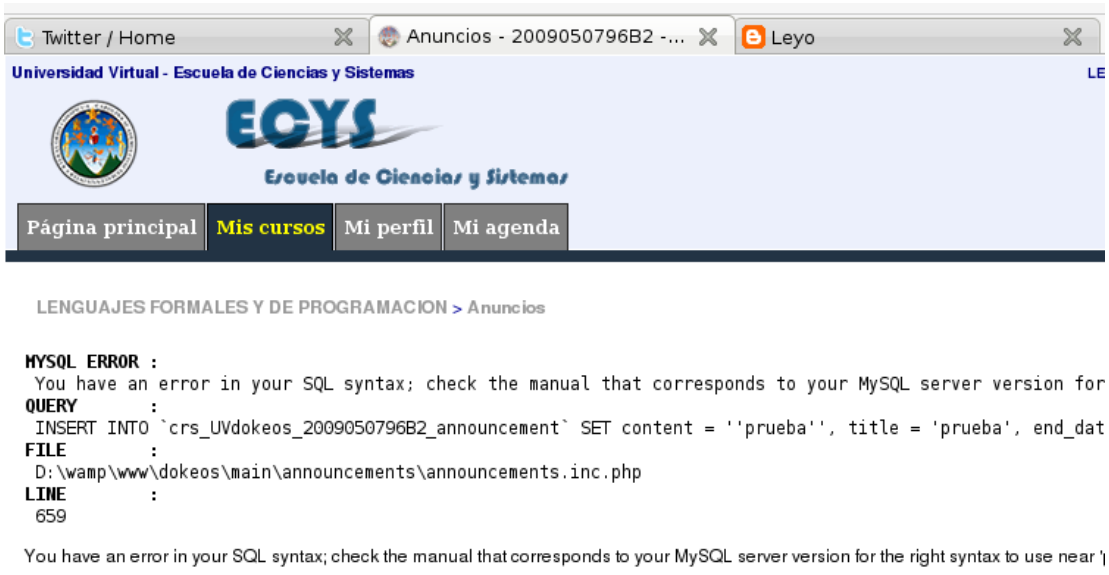
4.1.2. Caracteres especiales en las cadenas

La versión de *Dokeos* instalada en la Universidad Virtual, en algunas páginas no realizaba la validación de caracteres especiales como lo es comilla simple, hecho que fue detectado por un estudiante de la Escuela de Ciencias y Sistemas en julio del 2009, mostrando cómo al ingresar y colocar comillas simples sobre un mensaje, provoca un error en *SQL,* demostrando que no se validaron los caracteres especiales sobre esta página. Error que fue resuelto por el administrador de la Universidad Virtual, con referencia a la figura 27.

4.1.3. Rendimiento del servidor

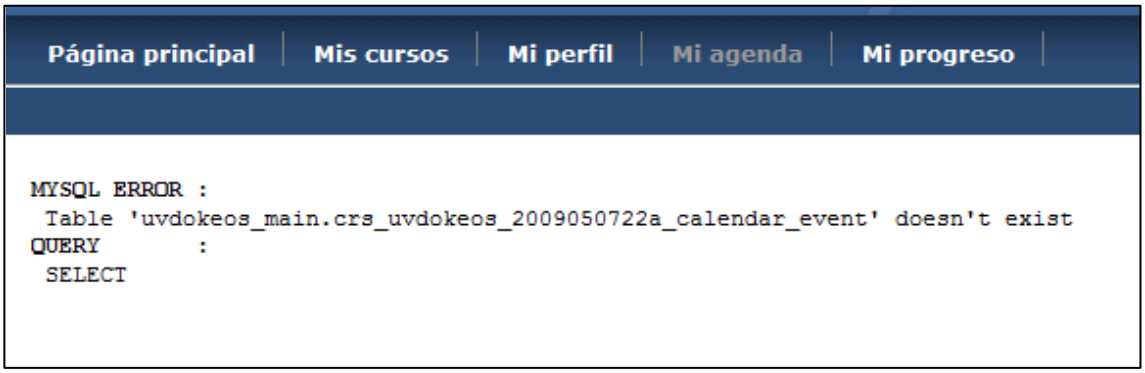
En un blog de una estudiante de la Escuela de Ciencias y Sistemas, se hace mención que el servidor que alojaba antiguamente la Universidad Virtual realizaba operaciones muy lentas provocando que no se puedan subir tareas, utilizar los foros, descargar archivos o no dejar entrar a la plataforma, lo que ocasionaba inconvenientes a los alumnos.

Figura 23. **Imagen del blog de LeoQuiroa**



Fuente: http://leoquiroa.blogspot.com/2009/07/la-nueva-uv-usac.html. Consulta: agosto de 2011.

Figura 24. **Errores de MySQL sobre la Universidad Virtual**



Fuente: fotografía tomada de un error en el servidor de la Universidad Virtual.

4.1.4. Errores de MySQL

En la Universidad Virtual publicada 2009-2011 sobre Windows server 2003, era poco frecuente encontrar un error de MySQL, la mayoría de veces, al actualizar la página se eliminaban los errores; estos errores se debían a la gran cantidad de tablas e información que contenía la base de datos de MySQL.

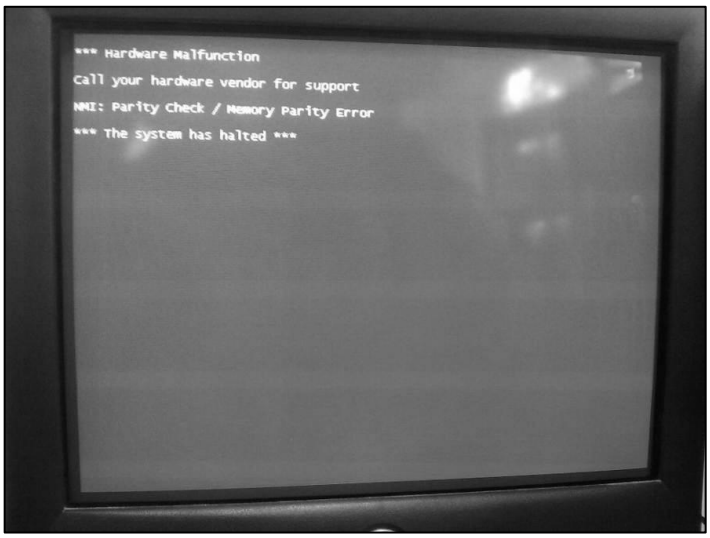
4.1.5. Fallo de memoria

El 14 de septiembre del 2011, el servidor de la Universidad Virtual tuvo inconveniente con el *hardware* de la computadora, donde al revisar el servidor, era provocado por falta de mantenimiento preventivo (limpieza interna del equipo); el sistema se encontraba al día en sus parches de seguridad, este acontecimiento provocó que estuviera fuera de línea hasta el mes de octubre, siendo restablecido por el administrador de la Universidad Virtual migrando hacia otro servidor en Linux y *Lamp*, el *backup* de la información se encontraba al día y no hubo ningún inconveniente para restablecer la información del año en curso, además el error estuvo relacionado con la memoria RAM y la información se encontraba intacta.

A partir de octubre del 2011, el administrador informó a los estudiantes sobre el arreglo del sistema, donde se restablecieron las funciones de enlaces, anuncios, exámenes online, agenda por curso, usuarios, foros, chat, y descarga de archivos, informes por curso sobre la nueva plataforma.

Durante el cambio de sistema operativo se resolvieron otros problemas de seguridad que poseía la Universidad Virtual como errores en certificado digital, que siempre los usuarios tenían que observar un anuncio proporcionado por los navegadores informando que no era seguro el sitio que se estaba visitando.

Figura 25. **Error de *hardware* sobre la Universidad Virtual**



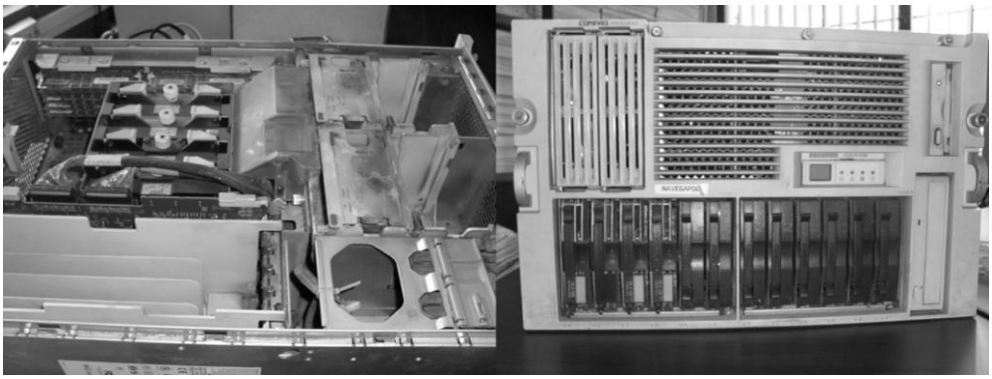
Fuente: fotografía en el servidor de la Universidad Virtual en la Escuela de Ciencias y Sistemas.

4.2. Problemas o inconvenientes en el servidor actual

El administrador de la Universidad Virtual, describió que los inconvenientes que actualmente le está dando la Universidad Virtual son debido a un fallo de memoria y falta de mantenimiento preventivo hacia el servidor de la Universidad Virtual.

Además menciona que el problema de seguridad más grande que posee está relacionado con el esquema de ingeniería donde se tienen vulnerabilidades mayores que no han sido corregidas y se escapan sobre el servidor y alcance de la Escuela de Ciencias y Sistemas.

Figura 26. **Foto del antiguo servidor de la Universidad Virtual**



Fuente: fotografía del servidor de la Universidad Virtual en la Escuela de Ciencias y Sistemas.

4.3. Propuestas a posibles soluciones en el servidor

De los problemas descritos, se propusieron una serie de recomendaciones, para el servidor de la universidad virtual.

4.3.1. Virtualización de la Universidad Virtual

La virtualización es la simulación de tener una o más computadoras sobre un ordenador real; a partir de esto surgen se sugiere la virtualización de *hardware* donde se utiliza el mismo CPU, memoria y dispositivos o la utilización de único *kernel* a nivel del sistema operativo.

Las ventajas que esto provee son:

- Simplificar la realización de copias de seguridad y de respaldo (*backup*), en caso de falla, debido a que todo el servidor virtual puede ser almacenado en único archivo.

- Migrar de una computadora a otra sin ningún inconveniente.

- Puede mejorarse la seguridad ya que un mismo servidor puede contener diferentes tipos de servicios en diferentes máquinas.

4.3.1.1. Virtualización por *hardware*

Este tipo de virtualización emula el *hardware* nativo de las computadoras, de manera que no se ejecuta a nivel de sistema operativo sino a nivel de *hardware*, pero solo es soportado por ciertos procesadores que ya contienen esta característica.

Este tipo de virtualización emula el *hardware* nativo de las computadoras, de manera que no se ejecuta a nivel de sistema operativo sino a nivel de *hardware*, pero solo es soportado por ciertos procesadores que ya contienen esta característica.

4.3.1.2. Virtualización de la Universidad Virtual

La virtualización de la Universidad Virtual proveerá una rápida recuperación de desastre, debido a que el *backup* se realizaría a nivel de sistema operativo; además, una fácil migración en caso de fallo o mejoras del sistema de información, se recomienda realizar la técnica de virtualización por *hardware*, que emulará todos los componentes de la máquina anfitrión.

4.3.2. Cloud computing

Es la tendencia de no tener almacenada la información de otros servidores en ordenadores propios, sino poder acceder a ella desde el internet; esto ayuda a responder a las necesidades de organizaciones y necesidades cambiantes de forma flexible, escalable y adaptable.

El acceso a este tipo de sistema puede realizarse desde cualquier PC que tenga acceso a internet incluso un celular; además, actualizaciones automáticas ya sea sobre servicios o *hardware*; esto proporciona a la vez, mayor capacidad de recuperación de desastres.

4.3.2.1. *Cloud computing* en la Universidad Virtual

Existe el modelo de nube llamado infraestructura como servicio, en lugar de tener un data center, el cual es el caso en el que se encuentra la Universidad Virtual, donde se tiene full control, más seguridad, capacidad limitada y servidores dedicados solo a estos servicios, que requieren mucho consumo de recursos. Tener una infraestructura como servicio (*Iass*) significa pagar por el uso de recursos como: espacio en disco, tiempos del CPU, espacio en la base de datos o transferencia de archivos al hacerlo así solo se pagará por los servicios consumidos logrando un ahorro de costos.

4.3.3. Firewall para la Escuela de Ciencias y Sistemas

Un *firewall* dedicado para la Escuela de Ciencias y Sistemas resolvería el problema de que la Facultad de Ingeniería aún tenga puntos críticos de seguridad por resolver; este se describe como parte del capítulo tres, al describir la seguridad en el *firewall*.

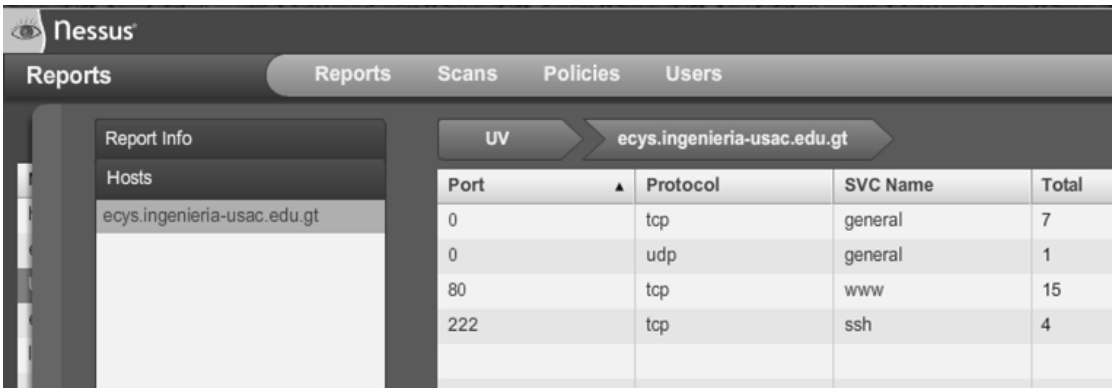
4.3.4. Mantenimiento preventivo

Al tener un mantenimiento preventivo se puede aumentar la vida útil de los equipos, disminuir costos que se utilizarían en reparaciones, y la posibilidad de que falle el sistema; pero el problema de la Universidad Virtual es que no posee recursos necesarios para llevar el mantenimiento preventivo, por lo que se propone que, como parte de un ejercicio profesional supervisado de tres meses, se realice el mantenimiento preventivo de los servidores de la Escuela de Ciencias y Sistemas.

4.4. Herramientas para monitorear seguridad Nessus

Se agrega el resultado del análisis con la herramienta *nessus* en el apéndice B.

Figura 27. **Monitoreo de seguridad con la herramienta Nessus**



Fuente: fotografía tomada al evaluar al servidor con la Herramienta *Nessus.*

Esta herramienta sirve para escanear la seguridad de un sitio web; es de gran alcance y fácil; esto permite realizar auditorías de una red de forma remota y determinar si es posible que acceda al sistema web o si existen puertas traseras realizando un análisis, para explorar las vulnerabilidades dentro del servidor y proporciona informes detallados sobre los elementos encontrados, además de trabajar en la mayoría de plataformas.

Tabla XVI. **Lista de vulnerabilidad y recomendaciones**

Parámetros potencialmente sensible para *CGI / index.php*. *Password*: posibilidad de encontrar una contraseña por ataque de diccionario.	
De acuerdo con los nombres de los parámetros CGI, se puede dar información de datos confidenciales como precios, datos de tarjetas de crédito y estados, los cuales pueden ser divulgados en el transcurso de la aplicación. Estos parámetros deben ser examinados para determinar qué tipo de datos determina un riesgo en seguridad.	Asegurarse que datos sensibles no sean revelados por parámetros CGI. Además no utilizar parámetros CGI para controlar los recursos o privilegios.
Alguna información sobre la configuración http se puede extraer	
El test dio información sobre el protocolo http remoto; la versión utilizada manteniendo conexiones http.	

Continuación de la tabla XVI.

El autocompletar no está desactivado en los campos de contraseñas	
El servidor contiene al menos un campo del formulario HTML que contiene el tipo *password* donde el autocomplete no se encuentra desactivado, esto puede representar un riesgo si tienen en su navegador credenciales guardadas.	Agregar el atributo autocomplete=off donde se previene que los navegadores usen credenciales guardadas.
El servidor implementa marcas de tiempo tcp	
El host remoto implementa marcas de tiempo tcp, como se define en RFC1323. Un efecto secundario de esta característica es que el tiempo de funcionamiento del servidor a veces puede ser calculado.	
Un servicio de SSH en el puerto 222 devuelve información	
Es posible obtener información sobre el servidor SSH remoto, mediante el envío de una solicitud de autenticación.	
Servidor http tipo y versión	
El servidor web es Apache/2.2.3 (*CentOS*), con el se puede determinar la versión del servidor web. La versión del sistema operativo fue encontrada en la bandera de Apache.	Se debe desactivar esta directiva en el archivo *httpd.conf* y poner "*ServerTokensProd*" y reiniciar el servidor.

Continuación de la tabla XVI.

Algunos *CGIs* son candidatos para las pruebas de inyección de datos.	
Nessus fue capaz de inyectar cadenas inocuas en los parámetros CGI y leer de nuevo en la respuesta http. Los parámetros afectados son candidatos a las pruebas de inyección extendidos como ataques *cross-site scripting*.	
Es posible adivinar el sistema operativo Linux *Kernel* 2.6 en *CentOS* 5	
Usando una combinación de sondeo remotas, (tcp/ip, smb, http, ntp, snmp, etc.) Es posible adivinar el nombre del sistema operativo remoto en uso, y a veces su versión.	
La configuración de PHP en el servidor permite la divulgación de información confidencial.	
La instalación de PHP en el servidor remoto se configura de manera que permite la divulgación de información sensible a un atacante a través de una URL especial.	En la configuración del archivo PHP, php.ini, se debe establecer el valor de '*expose_php*' a 'Off', para desactivar este comportamiento.

Fuente: elaboración propia, con base en la herramienta *Nessus.*

CONCLUSIONES

1. Todos los días se producen nuevas vulnerabilidades sobre las aplicaciones que se están utilizando y es importante mantenerse actualizado y analizar en qué aspectos puede afectar la seguridad del servidor web, ya que en él se tiene acceso a información valiosa para los usuarios, que no puede ser pública.

2. La seguridad en un sistema web se garantiza estableciendo políticas de seguridad en los diferentes elementos y servicios instalados en el servidor; por ello es importante optimizarlos y definir el mínimo de elementos que se necesita utilizar, desde el momento de la instalación, monitorearlos y actualizarlos, según las necesidades del sistema.

3. Se debe monitorear la seguridad y detectar posibles amenazas; ésta se puede realizar con herramientas de *scanner* de seguridad o de posibles errores que tiene el servidor, que pueden ser obtenidos por registro de eventos del sistema o de sus servicio; así como de historias de usuario que utiliza la aplicación; posteriormente, establecer qué políticas y medidas de seguridad pueden resolver estos problemas.

4. La Universidad Virtual de la Escuela de Ciencias y Sistemas, realizó un análisis y presentó vulnerabilidades de poco riesgo; además el administrador describió que las inseguridades más peligrosas se escapan del alcance de Escuela de Ciencias y Sistemas, debido a que son a nivel de la red; se están aplicando parches y actualizaciones de seguridad sobre el los servicios instalados. Aún así la Universidad Virtual no está a salvo, ya que continuamente siguen apareciendo nuevos *bugs* y vulnerabilidades, y se le debe dar seguimiento a los errores que esta pueda presentar.

RECOMENDACIONES

1. Mantenerse informado de nuevos ataques y vulnerabilidades sobre las aplicaciones que se estén utilizando en el servidor.

2. Conocer cuáles son las vulnerabilidades que pueden afectar al sistema web, para identificar las herramientas y técnicas y evitar los ataques.

3. Planificar los servicios que debe tener instalado un servidor para reducir al mínimo las vulnerabilidades que afecten al sistema y disminuir rendimiento.

4. Establecer los procedimientos a seguir en caso de pérdida de información o pérdidas del sistema, así como rutinas para probar que se podrá restaurar correctamente la información.

5. Cada elemento instalado debe ser personalizado y deben establecerse políticas de seguridad, para mejorar el rendimiento total del sistema.

6. Implementar un *scanner* de seguridad, para servidores web, que permite detectar vulnerabilidad, ya que estos son actualizados diariamente; además, se debe proporcionar la posible solución a la vulnerabilidad.

7. La Universidad Virtual necesita políticas de seguridad de mantenimiento preventivo, para evitar caídas del sistema y es recomendable modernizar el servidor que está en uso para la Universidad Virtual.

BIBLIOGRAFÍA

1. BALLAD, Tricia. *PHP Vulnerability. Security PHP Web Applications.* Boston: Pearson Education, 2007. 218 p.

2. BERENTZEN, Paul. *Seguridad informática* [en línea]. Kioskea, Comunidad Kioskea [ref. 19 de enero de 2008]. Disponible en Web: <http://es.kioskea.net/contents/secu/>.

3. CERULLO, Fabio. *Las 10 vulnerabilidades de seguridad más críticas en aplicaciones web.* [en línea]. OWASP TOP 10 [ref. 7 de agosto de 2007]. Disponible en Web: <http://www.lulu.com/items/volume_63/4114000/4114474/1/print/4114474.pdf>.

4. CREESON WOOD, Charles. *Practices of systems security planning*. *Information Security Policies Made Easy.* Houston: Magazine, 2002. 120 p.

5. DATE, CJ. *Introducción a los sistemas de bases de datos.* México: Pearson Education, 2009. 525 p.

6. GONZÁLEZ, Julia, et al. *Hacia la medición de calidad en uso web*. La Pampa, Argentina: Facultad de Ingeniería, Departmento de Informática, Universidad de Cáceres, 2007. 4 p.

7. HERNÁNDEZ, Jhon Jairo. *Seguridad en aplicaciones Web.* [en línea]. DragonJAR [ref. enero de 2008]. Disponible en Web: <http://search.4shared.com/postDownload/dVcwIe0h/Ezine_2_Comunidad_DragonJAR.html>.

8. *ORACLE Corporation. Establishing Security Policies: Oracle9i Database Administrator´s Guide* [en línea]. [ref. enero de 2002]. Disponible en Web: < http://docs.oracle.com/cd/B10501_01/server.920/a96521/secure.htm>.

9. PUSCHITZ, W. *Securing and hardening Red Hat Linux Production Systems.* [en línea]. Puschitz [ref. 10 de enero de 2007]. Disponible en Web: <http://www.puschitz.com/SecuringLinux.shtml>.

10. SCAMBRAY, Joel. “Hacking Web Server”. *Hacking Exposed Web Applications.* Osborne: McGraw-Hill, 2001. 333 p.

11. VENDOR, G.L. *MySQL Security Best Practices: Hardening MySQL Tips* [en línea]. GreenSQL [ref. marzo de 2009]. Disponible en Web: <http://www.greensql.com/articles/mysql-security-best-practices>.

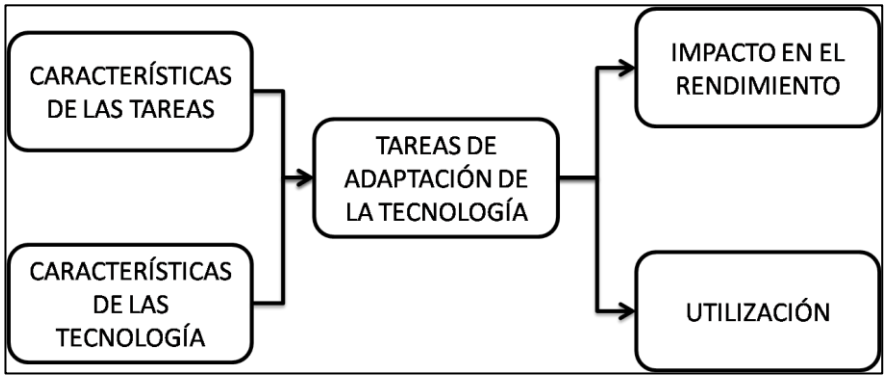
APÉNDICES

Apéndice 1. **Tareas de adaptación de la tecnología**

El modelo pretende analizar el sistema actualmente está en funcionamiento en la Universidad Virtual y realizar una investigación de las vulnerabilidades que pueden afectar y producir que la Universidad Virtual minimice su rendimiento, o que realicen un mal manejo de la información; así también definir nuevas tareas que ayuden al sistema para prevenir y detectar amenazas.

Plantear reglas y configuraciones de seguridad para que el sistema de web siga garantizando integridad y lograr impacto positivo en el rendimiento del sistema dando a los usuarios una mejor utilización del mismo.

Gráfico de tareas de adaptación de la tecnología



Fuente: http://www.fsc.yorku.ca/york/istheory/wiki/index.php/task-technology_fit. Consulta: marzo de 2011.

Las tareas que poseerá un sitio web pueden variar en función de la institución por ello mencionare las principales tareas del caso de estudio (Universidad Virtual) se mencionará:

- Administración de cursos
- Anuncios para los miembros de los cursos
- Exámenes en línea
- Agenda
- Foros
- Descarga de archivos
- Carga de archivos
- Informes de utilización del sistema
- Información de los miembros del curso
- Descarga de documentos
- Notificaciones del curso
- Información de los miembros del curso
- Foros
- Carga de archivos

En el área de tareas de adaptación de la tecnología se proponen las mejores prácticas en función de la tecnología que actualmente posee la Universidad Virtual. Se describen normas o estándares sobre las tareas que actualmente se encuentra en el servidor.

Para el caso en estudio, en la Universidad Virtual se pretende realizar un análisis de alternativas para mejorar el rendimiento sobre el estado actual del servidor, y así sumar vida útil al servidor actual.

La tecnología que se usará en un sitio web tiende a variar por los recursos que se disponen, por las licencias que se poseen, las características del sistema, costo de mantenimiento y seguridad; se menciona la tecnología, para el caso de estudio (Universidad Virtual):

La funcionalidad de la investigación se basará en:

- Que la Universidad Virtual siga garantizando como hasta hoy lo ha hecho, una fácil comunicación e interacción con el estudiante y profesores.

- Plantear las nuevas amenazas a que pueden, ser sometidas la Universidad Virtual, y plantear posibles soluciones a ellas.

- Describir las mejores prácticas que pueden realizarse para cualquier sitio web.

En la matriz de contenidos los elementos, incluidos en la investigación son organizados, en base de capítulos y variables.

- Capítulos

    - Marco teórico sobre seguridad informática
    - Aseguramiento de la seguridad en un sitio web
    - Configuraciones recomendadas en servidores web
    - Configuración actual de la Universidad Virtual

- Variables

    - Características de las tareas
    - Características de la tecnología
    - Tareas de adaptación de la tecnología
    - Impacto en el rendimiento
    - Utilización

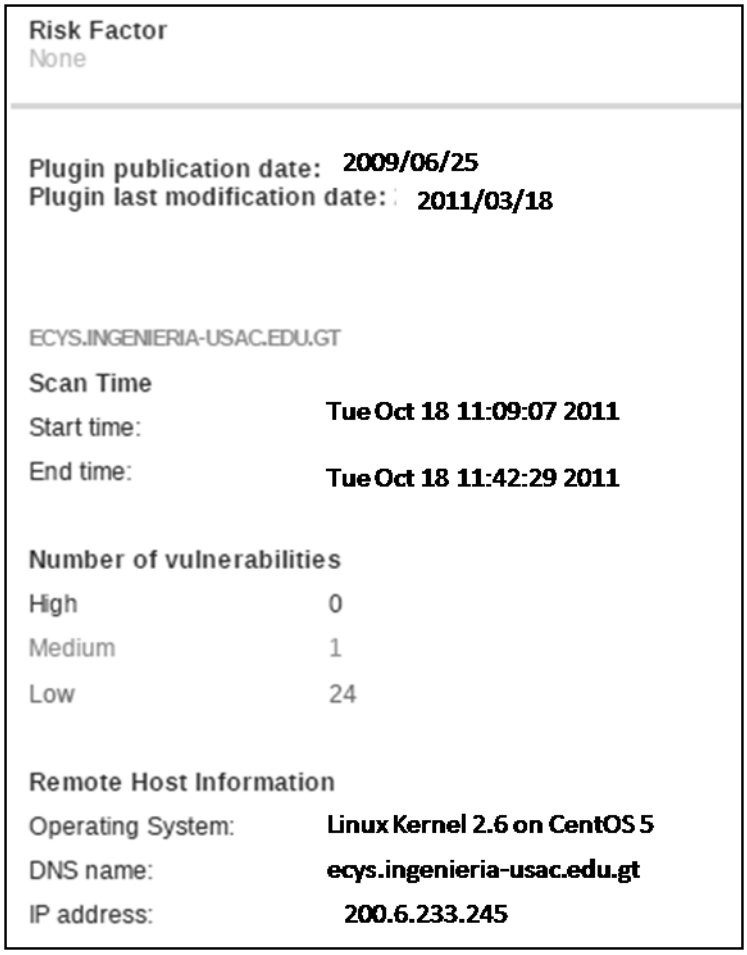
Apéndice 2. **Nessus y la Universidad Virtual**

Reporte de *Nessus* sobre la Universidad Virtual

PLUGIN NAME	SEVERITY
Service Detection	Low Severity problem(s) found
Web Server Uses Plain Text Authentication Forms	Low Severity problem(s) found
Web Server Allows Password Auto-Completion	Low Severity problem(s) found
Web mirroring	Low Severity problem(s) found
Web Application Potentially Sensitive CGI Parameter Detection	Low Severity problem(s) found
Traceroute Information	Low Severity problem(s) found
TCP/IP Timestamps Supported	Low Severity problem(s) found
SSH Server Type and Version Information	Low Severity problem(s) found
OS Identification	Low Severity problem(s) found
Nessus Scan Information	Low Severity problem(s) found
HyperText Transfer Protocol (HTTP) Information	Low Severity problem(s) found
HTTP Server Type and Version	Low Severity problem(s) found
HTTP Server Cookies Set	Low Severity problem(s) found
HTTP Methods Allowed (per directory)	Low Severity problem(s) found
Host Fully Qualified Domain Name (FQDN) Resolution	Low Severity problem(s) found
External URLs	Low Severity problem(s) found
Device Type	Low Severity problem(s) found
Common Platform Enumeration (CPE)	Low Severity problem(s) found
CGI Generic Tests Load Estimation (all tests)	Low Severity problem(s) found
CGI Generic Injectable Parameter	Low Severity problem(s) found
Backported Security Patch Detection (WWW)	Low Severity problem(s) found
Backported Security Patch Detection (SSH)	Low Severity problem(s) found
Apache Banner Linux Distribution Disclosure	Low Severity problem(s) found

Fuente: elaboración propia, con base en la herramienta *Nessus.*

Duración del análisis de Nessus



Risk Factor
None

Plugin publication date: 2009/06/25
Plugin last modification date: 2011/03/18

ECYS.INGENIERIA-USAC.EDU.GT

Scan Time

Start time:	Tue Oct 18 11:09:07 2011
End time:	Tue Oct 18 11:42:29 2011

Number of vulnerabilities

High	0
Medium	1
Low	24

Remote Host Information

Operating System:	Linux Kernel 2.6 on CentOS 5
DNS name:	ecys.ingenieria-usac.edu.gt
IP address:	200.6.233.245

Fuente: elaboración propia, con base en la herramienta *Nessus.*

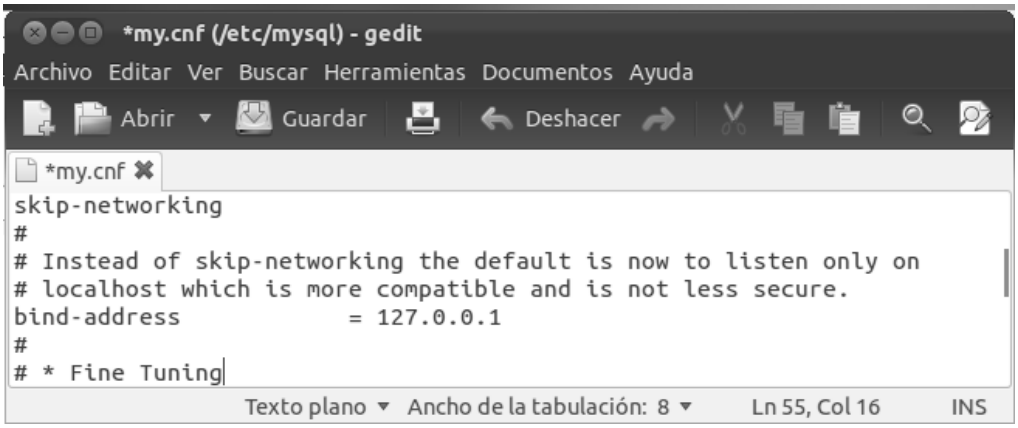
Apéndice 3. **Manual de configuración seguridad en Linux**

Entre las configuraciones de seguridad más confiables están:

- MySQL/Deshabilitar o restringir el acceso remoto

    - Se abre una terminal, y se escribe “sudo gedit /etc/mysql/my.cnf” y agregamos la línea, “skip-networking”, guardamos el archivo y reiniciamos el servidor de MySQL.

Archivo de configuración de MySQL

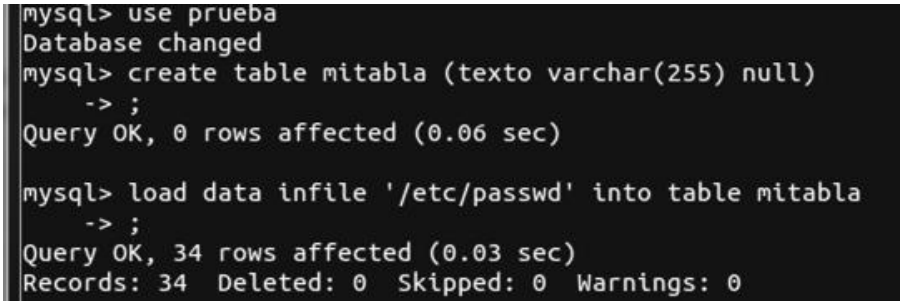


Fuente: elaboración propia, con el sistema operativo Ubuntu.

- MySQL/Deshabilitar LOCAL INFILE

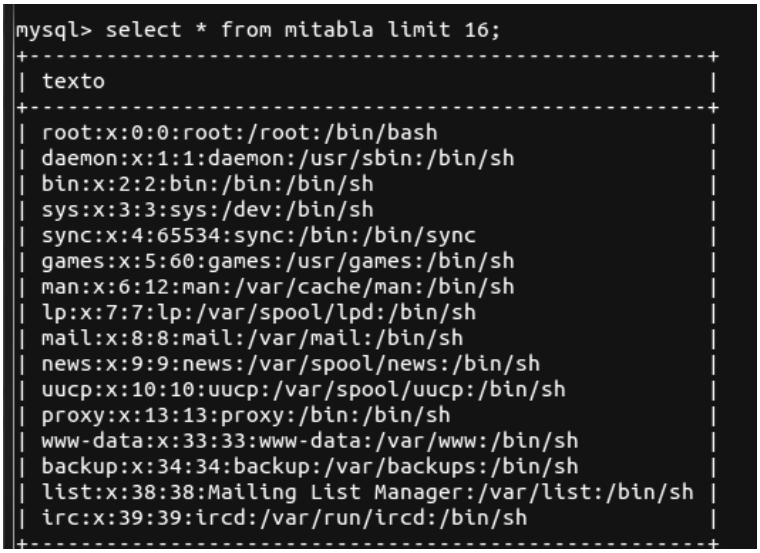
  - Para demostrar este problema se crea una base de datos de prueba, y una tabla para almacenar la información del archivo.

Consola de MySQL vulnerabilidad



Fuente: elaboración propia, con el sistema operativo Ubuntu.

Acceso no autorizado a archivo etc/passwd por MySQL



Fuente: elaboración propia, con el sistema operativo Ubuntu.

- Únicamente agregando en el archivo /etc/mysql/my.cnf el parámetro “set-variable=local-infile=0” y reiniciando el servidor.

- MySQL/Cambiar el nombre de usuario y contraseña de root

Cambio de usuario y contraseña de MySQL

```
mysql> rename user root@'localhost' to ecys_user;
Query OK, 0 rows affected (0.01 sec)
```

Fuente: elaboración propia, con el sistema operativo Ubuntu.

- MySQL/Eliminar la base de datos de pruebas.

Eliminación de bases de datos de prueba

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+
3 rows in set (0.00 sec)

mysql> drop database test;
Query OK, 0 rows affected (0.06 sec)
```

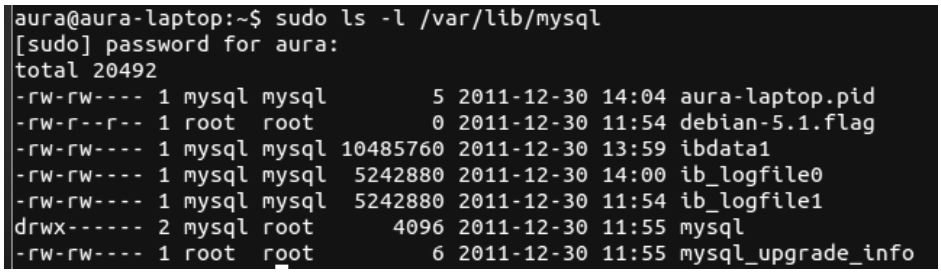
Fuente: elaboración propia, con el sistema operativo Ubuntu.

- MYSQL/Menor número de privilegios del sistema

  - Se debe asegurar que solo los archivos son propiedad de MySQL y del grupo de MySQL.

  - Además asegurar que solo el usuario MySQL y root tienen acceso de escritura a los archivos.

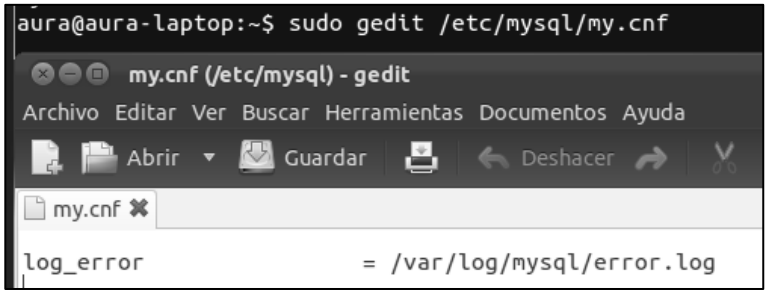
Verificación de usuarios sobre los archivos de MySQL



Fuente: elaboración propia, con el sistema operativo Ubuntu.

- MySQL/Habilitar el registro de *log* en el archivo de configuración

Habilitación del archivo *log* de MySQL



Fuente: elaboración propia, con el sistema operativo Ubuntu.

- PHP/Directivas de seguridad

Descripción de algunas directivas de seguridad en PHP

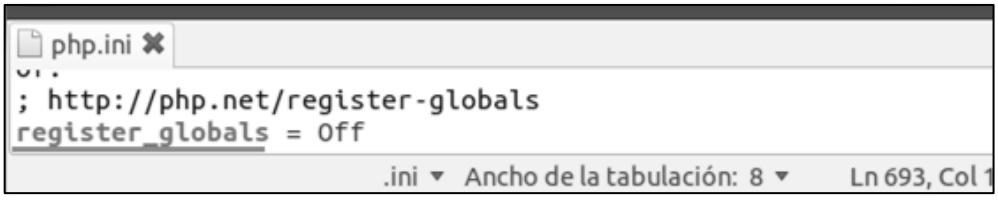
max_execution_time=30;	Tiempo máximo de ejecución
Max_input_time=60;	Tiempo máximo de espera para recibir datos de un formulario o cualquier input
Memory_limit=8M;	Máximo de memoria que puede ejecutar una aplicación
Upload_max_filesize=8M;	Tamaños de los archivos que se pueden subir al servidor
Upload_tmp_dir=\var\tmp\tmp2	Cambiar el directorio donde se guardan los archivos temporales
Open_basedir=/home/www	Denegar el acceso a PHP a directorios que no estén a su alcance

Fuente: http://www.centosni.net/instalar-php-en-gnulinux-centos-5/. Consulta: junio de 2011.

- PHP/Deshabilitar RegisterGlobals

  - Desde la versión de PHP 4.2 esta opción viene desactivada por defecto.

Deshabilitar la opción de *register globals* de PHP

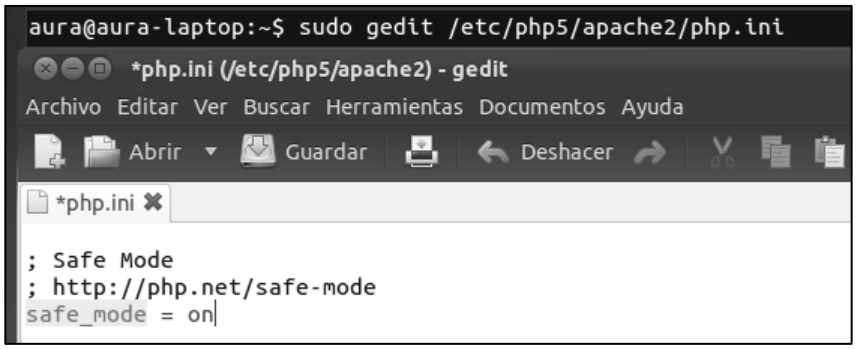


Fuente: elaboración propia, con el sistema operativo Ubuntu.

- PHP/Modo Seguro

  - Previene ataques externos utilizando scripts PHP para ejecutar comandos del sistema operativo.

Activación del modo de seguridad para PHP

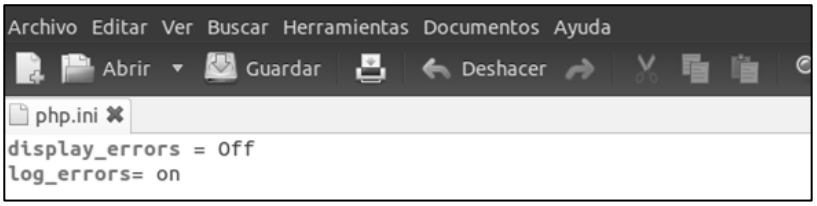


Fuente: elaboración propia, con el sistema operativo Ubuntu.

- PHP/Mensajes de Error y *Log*

  - Envía un mensaje de error al registro de errores de un servidor web o a un archivo.

Activación de mensajes de *log* por archivos

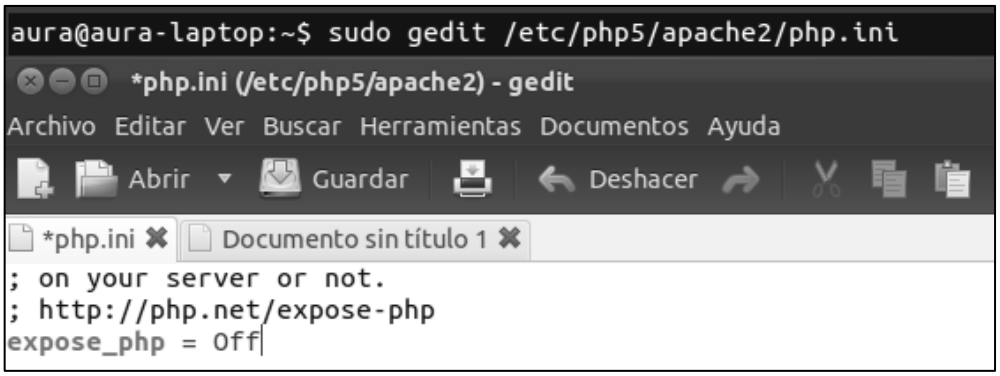


Fuente: elaboración propia, con el sistema operativo Ubuntu.

- PHP/Ocultar la presencia de PHP

    - Se utiliza para que no se pueda determinar si usted utiliza PHP en su servidor.

Ocultar la presencia de PHP

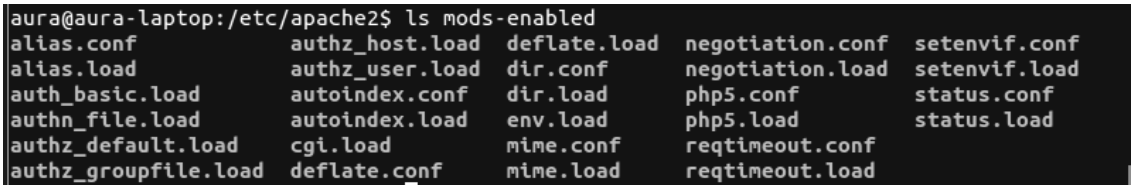


Fuente: elaboración propia, con el sistema operativo Ubuntu.

- Apache/Deshabilitar los módulos innecesarios

    - Con el comando “ls mods-enabled” ubicado en la carpeta “/etc/apache2” se puede ver los módulos activos en PHP.

Mostrar los módulos innecesarios



Fuente: elaboración propia, con el sistema operativo Ubuntu.

- APACHE/ Desahabilitando autoindex

    - Modificar la opción para que no muestre un listado de directorios cuando no hay ningún directorio index.html.

Desactivar la opción de auto índex

```
aura@aura-laptop:/etc/apache2$ sudo a2dismod autoindex
[sudo] password for aura:
Module autoindex disabled.
To activate the new configuration, you need to run:
  service apache2 restart
aura@aura-laptop:/etc/apache2$ sudo /etc/init.d/apache2 restart
 * Restarting web server apache2
```

Fuente: elaboración propia, con el sistema operativo Ubuntu.

- APACHE/ Ejecutar como usuario y un grupo

    - Crear un grupo apache y usuario

Crear un usuario en Linux

```
aura@aura-laptop:/etc/apache2$ sudo groupadd apache
```

Fuente: elaboración propia, con el sistema operativo Ubuntu.

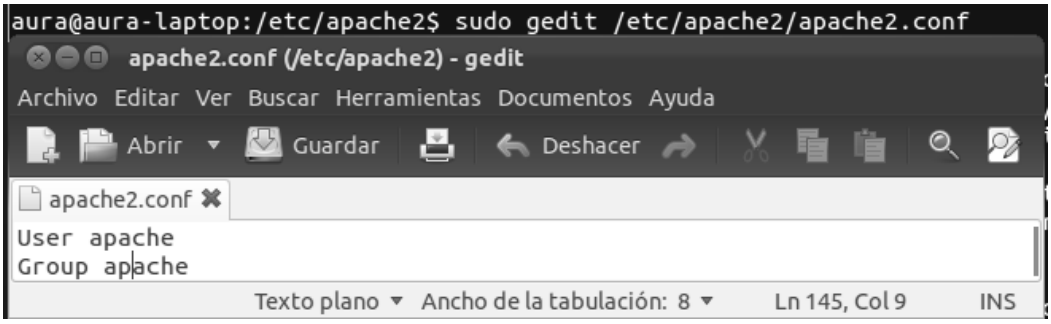
Crear un grupo en Linux

```
aura@aura-laptop:/etc/apache2$ sudo useradd -d /usr/local/apache2/htdocs -g
apache -s /bin/false apache
```

Fuente: elaboración propia, con el sistema operativo Ubuntu.

- Modificar el archivo de configuración con el usuario y grupo apropiado

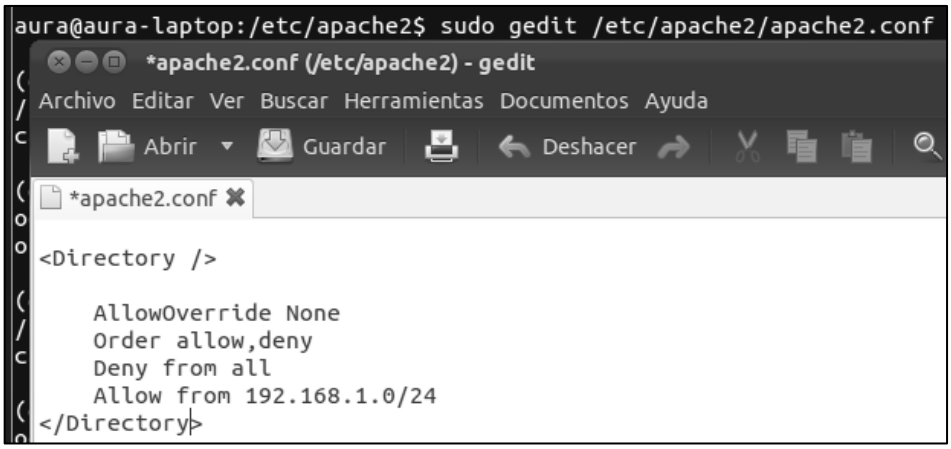
Agregando el usuario y grupo a apache



Fuente: elaboración propia, con el sistema operativo Ubuntu.

- APACHE/Establecer los permisos adecuado para una red
    - Establecer los permisos adecuados para conf y el directorio bin

Agregando una red específica a APACHE

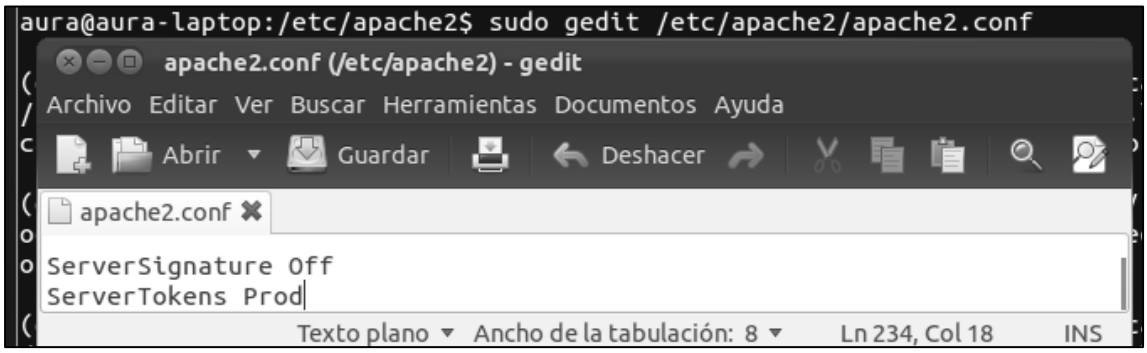


Fuente: elaboración propia, con el sistema operativo Ubuntu.

- APACHE/Ocultar la versión e información delicada

    - Las instalaciones de apache por defecto muestra el número de versión que está funcionando y el sistema operativo y módulos de apache que están funcionando, los usuarios pueden usar esta información para atacar el servidor, se necesitan corregir dos directivas.

Modificar Apache para no mostrar información del sistema



Fuente: elaboración propia, con el sistema operativo Ubuntu.