UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGÓN

"IMPLEMENTACIÓN DE SEGURIDAD EN EL SISTEMA DE INSCRIPCIONES POR INTERNET DE ESPECIALIZACIONES DE MEDICINA DE LA UNIDAD DE ADMINISTRACIÓN DEL POSGRADO"

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN

P R E S E N T A:

FERNANDO ZARAGOZA HERNÁNDEZ

ASESOR DE TESIS:
M. EN C. MARCELO PÉREZ MEDEL

MÉXICO, 2004.

Agradecimientos

Primero agradezco a Dios por todas sus bendiciones y por permitirme cumplir esta meta después de mucho sufrimiento y sacrificios.

A mi familia por todo el apoyo, cariño y amor que hemos compartido durante mi vida.

A mi mami Paulis por apoyarme siempre. Gracias por que la sencillez y humildad que me has enseñado ha marcado mi vida.

A mi papá One, por que el estudio es la mejor herencia que puedo recibir.

A mis hermanos Alberto, Carlos, Jaime, Daniel y Mónica que amo y respeto profundamente. Comparto esta meta cumplida con ustedes.

A mis sobrinos Carla Fernanda, Carlos Alberto, Jaina y los que vienen, por que representan para mi nuevas esperanzas y parte de la alegría de mi vida.

A Mary, gracias por apoyarme en las buenas y en las malas, por todo tu cariño y enseñanzas durante muchos años.

A mi mejor amigo Pepe, que aunque estas muy lejos sabes que siempre estas en mis recuerdos. Gracias cachanilla por enseñarme a reir y tomar las cosas con alegría.

Agradecimientos

A mi Universidad por ampliar mi visión profesional y humana. Por darme la oportunidad de prepararme en sus aulas donde conocí a grandes personas.

A mi director de Tesis el M. en C. Marcelo Pérez Medel por guiarme en este trabajo.

A mis asesores la Lic. Carmen Ma. Herrera Reyes, Ing. José Manuel Quintero Cervantes, Ing. Cesar F. Germán Rosas e Ing. Alejandro R. González Ponce por sus comentarios y aportaciones a este trabajo.

En la Unidad de Administración del Posgrado agradezco al Maestro Francisco Incera Ugalde, Jefe de la Unidad y al Fis. Efraín López San Salvador, Jefe de la Unidad de Cómputo, por el apoyo y facilidades para llevar a cabo este proyecto.

Y a todos mis compañeros y amigos en la UAP, por hacer más placentera mi estancia.

A la Dirección General de Servicios de Cómputo Académico por permitirme ser parte de la gran familia de becarios donde he conocido a personas excelentes y de gran talento, además de toda la capacitación técnica recibida.

Especialmente quiero agradecer a la Lic. Rosario Salinas Cuellar por la confianza que siempre me ha tenido.

Al Lic. Juan Carlos Guel López por todas las oportunidades brindadas en el Departamento de Seguridad en Cómputo de la UNAM.

A la Lic. Mabel Entzana Martínez por demostrarme que la mente no tiene barreras, sólo las que nosotros mismos nos procuramos, gracias mabeliux :D

A todos las personas que han pasado por mi vida y han dejado una marca en mi corazón, muchas gracias.

Índice general

Introducción

Esta tesis tiene como objetivo definir e implementar los mecanismos de seguridad en el sistema de administración escolar de la Unidad de Administración de Posgrado (UAP). Este sistema de administración escolar está orientado a las especialidades de medicina por lo que para fines de este trabajo lo llamaré el Sistema de Inscripciones por Internet de Especializaciones de Medicina de la Unidad de Administración del Posgrado, SIIEM/UAP para abreviar.

En este trabajo se plantean, en base a la experiencia y entrenamiento del autor obtenido como becario en la Dirección General de Servicios de Cómputo Académico (DGSCA) - además del uso de métodos probados a nivel internacional-, los elementos de seguridad que deben ser cubiertos en un sistema que tenga como misión proporcionar el acceso a través de Internet a una base de datos mediante un servidor de páginas web. En el SIIEM/UAP se proporciona el acceso a los estudiantes a sus registros de inscripción.

Para lograr el objetivo planteado, se ha divido esta tesis en cuatro capítulos para cubrir cada uno de los elementos que deben ser considerados en el SIIEM/UAP que permitan alcanzar un estado aceptable de seguridad.

El primer capítulo titulado *Seguridad en Cómputo*, aborda los conceptos básicos de seguridad y los elementos que la conforman como una introducción al tema. En este capítulo se realiza un análisis de riesgos del SIIEM/UAP como primer paso en el proceso de seguridad. Y se definen las políticas necesarias, inexistentes al momento de iniciar este trabajo, que sirvan como marco de referencia para la implementación de la seguridad.

El segundo capítulo titulado *Seguridad en Web*, se encarga de definir los problemas de seguridad en el canal de comunicación que se forma entre el cliente y el servidor a través de Internet. Los usuarios del SIIEM/UAP acceden al sistema mediante Internet, sin embargo, en la llamada super carretera de la información, existen millones de usuarios que pueden tener acceso sin autorización al sistema. Mediante la seguridad en web se plantean e implementan los medios necesarios para asegurarse que sólo los usuarios con autorización tengan acceso al sistema y a su información.

El tercer capítulo titulado *Seguridad en la Base de Datos*, toma el otro eslabón que debe ser asegurado: la base de datos. Primero se proporciona una introducción a la seguridad en base de datos y las posibles amenazas que existen. Finalmente se explican de manera

puntual los mecanismos de seguridad que proporciona el manejador de la base de datos utilizado en el SIIEM/UAP y cómo fueron puestos en operación dichos mecanismos.

El último capítulo titulado *Herramientas de software libre y la seguridad en el SIIEM/UAP*, tiene como finalidad describir las ventajas que proporciona el uso de software libre en un sistema con características similares al SIIEM/UAP. Este capítulo también tiene como propósito mostrar que el software libre cuenta con soluciones de seguridad. En este cuarto capítulo pretendo dar una guía de las herramientas utilizadas para asegurar el sistema mediante el bloqueo del acceso al sistema, monitoreo y detección de intrusos.

La metodología que se emplea en este trabajo es la recuperación de la experiencia profesional, la investigación documental y el análisis de los elementos para definir un esquema de seguridad.

Como es ampliamente conocido desde el inicio de los llamados sistemas en línea o sistemas *punto com*, una gran cantidad de organizaciones de todo tipo, sean académicas, comerciales, financieras y gubernamentales se han visto en la "necesidad" de proporcionar servicios a través de Internet gracias a la imagen de competitividad e innovación que proporcionan creando sus páginas web (ver figura 1).

Figura 1: Sitio en línea de la Secretaría de Gobernación.

En el ambiente académico, Internet se ha utilizado para proporcionar servicios a la comunidad como sistemas de inscripciones para sus alumnos, difusión de las actividades

de investigación y docencia, convocatorias para ingreso a nivel bachillerato, licenciatura y posgrado, etc.

Muchas dependencias de la Universidad Nacional Autónoma de México (UNAM), a partir de la huelga estudiantil vivida a lo largo de nueve meses entre los años de 1999 y 2000, y debido a la situación física del cierre de las instalaciones y uso de sedes alternas de las mismas, se vieron en la necesidad de crear sistemas que no dependieran del lugar físico y fuera posible que muchas actividades se realizarán desde diferentes sitios.

Uno de los mayores problemas fue la inscripción y reinscripción de los alumnos, de tal manera, que los sistemas en Internet resultaron ser una solución muy importante para que los alumnos pudieran inscribirse al siguiente semestre, sin tener que hacerlo en el edificio donde se encuentra el Departamento de Control Escolar, sino que fuera posible inscribirse desde cualquier lugar con acceso a Internet: casa, oficina o *ciber-café*.

A nivel posgrado, la Unidad de Administración de Posgrado (**UAP**) es la encargada de realizar las inscripciones de todas las facultades, escuelas e institutos a este nivel de estudios en la UNAM. En la actualidad se cuenta con dos sistemas, a saber, el sistema de la UAP (ver figura 2) en el cual se inscriben todos los posgrados de la UNAM y el Sistema de Inscripciones por Internet de las Especializaciones de Medicina (SIIEM, ver figura 3), el cual quedó para uso exclusivo de la Facultad de Medicina.

Figura 2: Ventana de inicio del Sistema de Inscripciones de la UAP.

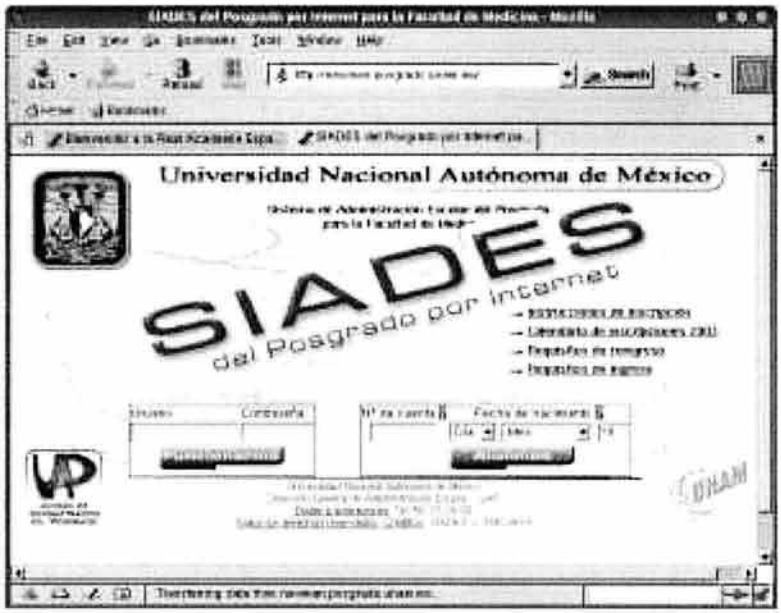

Figura 3: Ventana de inicio del SIIEM de la UAP.

Antes del sistema por Internet las inscipciones se realizaban localmente en cada dependencia mediante el siguiente procedimiento:

1. Se instalaba en cada instituto por parte de la UAP un programa creado en Visual Basic que accedía localmente a una base de datos en Access.

2. Cada dependencia realizaba la inscripción de acuerdo al calendario propuesto por ella.

3. Se regresaba la base de datos a la UAP para su análisis e incorporación a la base de datos general.

Esta forma de operar tenía como ventajas que cada escuela o facultad controlaba sus datos, recursos humanos y técnicos asignados para el proceso de inscripción y reinscripción. Sin embargo, tenía la desventaja que los alumnos debían de trasladarse hasta la facultad para realizar el trámite de inscripción, y cuando se vivió la huelga tuvieron que transladarse a las sedes alternas. Se debe mencionar que existen sedes de especializaciones de medicina ubicadas en todo el país, y los estudiantes de estas sedes tenían que trasladarse al Distrito Federal para realizar algún trámite. Esta forma de operar también hacía que se generaran filas para poder inscribirse lo que repercutía en tiempo que los alumnos debían invertir.

En la actualidad, este proceso se ha simplificado de la siguiente manera:

1. Se definen los horarios de inscripción por la UAP y las distintas escuelas y facultades.

2. Se habilita el servidor de web y base de datos en la UAP para que las dependencias puedan realizar el registro de inscripción.
3. Los alumnos se inscriben a través de Internet de acuerdo al calendario y horario definidos.

Ahora bien, esto tiene como ventajas que los coordinadores de posgrado de cada una de las facultades asignan los recursos necesarios para que los alumnos que tengan o no acceso a Internet lo puedan hacer desde sus instalaciones. Se reduce la cantidad de alumnos que deben ser atendidos en la dependencia en el período de inscripción, ya que los alumnos no tienen que trasladarse hasta la facultad o escuela para poder realizar su trámite en caso de que cuenten con acceso a Internet desde otro lugar, agilizando con esto el proceso.

Sin embargo, las desventajas de este esquema se presentan debido a que existe sólo un servidor que centraliza la información. Debido a la demanda, el servicio puede verse afectado y la información está expuesta al público, no sólo a los interesados en la inscripción.

Pare realizar el procedimiento anterior el SIIEM/UAP, al igual que cualquier sistema que proporciona una servicio en Internet, está compuesto de varios elementos que deben ser considerados al definir un esquema de seguridad:

1. Infraestructura de red. Esto incluye dispositivos de conexión.
2. Servidor de Web. Programa encargado de contestar las peticiones para páginas html.
3. Servidor de Bases de Datos. Programa encargado de almancenar la información de manera estructurada y relacional.
4. Computadora. Uno o más equipos encargados de dar el servicio de web y base de datos.

Si bien es cierto que los sistemas en línea han permitido el uso de servicios a través de Internet, existen problemas inherentes a esta tecnología como la exposición de la información en un medio inseguro; entendida como una tecnología que utiliza un medio público para acceder a la información.

Desafortunadamente, la seguridad es un elemento que no se considera en el diseño de los sistemas, siendo algo tan importante como el sistema mismo. Dadas las características de estos sistemas se debe considerar la seguridad en sus componentes:

- Seguridad en el sistema. Esto tiene que ver con la definición de políticas, estándares y procedimientos que determinen las reglas de operación y uso del sistema. Además de la prevención como elemento proactivo.
- Seguridad en el web. Debido a que éste es el medio de entrada al sistema, es conveniente tomar las medidas pertinentes para su buen funcionamiento y el viaje seguro de la información que se da entre la máquina cliente y el servidor.

- Seguridad en la base de datos. Debido a que en esta parte del sistema se mantiene la información, se deben determinar los mecanismos necesarios para la protección de los datos.

Este trabajo de tesis pretende cubrir estos tres elementos en la seguridad de un sistema a través de Internet, además de las herramientas utilizadas para tal propósito. Debido a la falta de recursos económicos se considerará en esta tesis la seguridad física de forma básica, ya que existen mecanismos de seguridad de este tipo que pueden resultar muy costosos, sin embargo, existen medidas preventivas que pueden ser implementadas fácilmente.

Capítulo 1

Seguridad en Cómputo

En este capítulo se hará una introducción a los conceptos básicos de la seguridad en cómputo y se identificarán los principales problemas que la afectan.

Se abordarán los siguientes puntos:

- *Introducción a la seguridad en cómputo*
- *Evaluación de riesgos*
- *Definición de Políticas*

1.1. Introducción a la seguridad en cómputo

En la actualidad, cualquier computadora conectada a la Internet [1] es potencialmente blanco de un ataque. Aunque se ha dicho que algunos *hackers* [2] son personas que ingresan a los sistemas de cómputo por entretenimiento, es cierto que en la actualidad estos accesos son más severos y persiguen otros fines (lucro, espionaje industrial, terrorismo) con efectos que puede dejar a las computadoras inservibles; la información es robada y/o modificada sin autorización; el software es reemplazado por programas con *back doors*; y son capturadas millones de contraseñas sin que los usuarios se percaten de ello.

De acuerdo a una encuesta realizada por el Instituto de Seguridad en Cómputo (CSI por sus siglas en inglés) a mas de 500 organizaciones en Estados Unidos, se dio a conocer que en contraparte a la tendencia que se venía dando en años anteriores, el 2003 vió una reducción en los costos económicos propiciados por ataques relacionados a la infraestructura de cómputo. Aunque el total de pérdidas en el 2002 fueron de casi 456 millones de dólares y en el 2003 de 201 millones de dólares entre la misma cantidad de empresas encuestadas, no deja de ser considerable la pérdida económica que tiene para las organizaciones en el rubro de la seguridad en cómputo. De este sondeo se desprenden algunos datos interesantes que se muestran en la tabla 1.1 [3].

Tipo de crimen	Costo 2002 (mdd)	Costo 2003 (mdd)
Robo de información propietaria	170	70
Negación de Servicios (DoS)	18	65
Virus	50	27
Abuso de la red por parte de usuarios internos	50	11
Penetración por intrusos ajenos	13	2.7

Cuadro 1.1: Encuesta 2003 del Instituto de Seguridad en Cómputo.

Esta encuesta, aunque reducida, no deja de ser representativa y muestra una clara tendencia del problema de la seguridad en cómputo. Debido a las pérdidas que se pueden generar de distintas amenazas es importante que la seguridad en cómputo sea considerada muy en serio y por tal motivo se debe asumir el riesgo si no se plantean los elementos adecuados que propicien un ambiente de seguridad.

Sin embargo, definir la seguridad en un sistema es difícil, de tal manera que no se refiere a un sistema como "seguro" o "inseguro". No obstante, la *confianza* es empleada para describir el grado de seguridad a través del comportamiento del equipo, ésto es, un sistema es seguro en la medida de que sus elementos de hardware y software se comporten como se espera que lo hagan. En seguida se definen los tipos de seguridad que existen.

[1] Entiéndase por Internet a la red mundial de computadoras.
[2] Persona con muchos conocimientos en el área del cómputo, ver glosario para más detalles.
[3] Fuente: CSI http://www.gocsi.com

1.1.1. Tipos de Seguridad

Existen distintos tipos de seguridad que deben ser considerados:

- *Confidencialidad.* Consiste en proteger la información de ser accedida por cualquiera que no esté autorizado para ello.
- *Integridad.* La integridad consiste en la protección de la información (incluyendo programas) de ser borrados o alterados sin el permiso del dueño de la información.
- *Disponibilidad.* La información debe ser utilizable cuando es requerida por el usuario.
- *Consistencia.* Asegurarse que el sistema se comporta como se espera que lo haga. Esto es, si el sistema opera como y para lo que fue diseñado.
- *Control.* Consiste en regular el acceso al sistema. Se debe cuidar que no se encuentren en el sistema individuos o programas no autorizados.
- *Auditoría.* La auditoría reside en tener un registro de las actividades dentro del sistema que identifique actores y acciones. En aplicaciones críticas, la auditoría será suficiente para poder revertir los posibles cambios a un sistema y reestablecerlo por completo.

Aunque todos estos aspectos de seguridad son importantes, cada organización determina el orden de prioridad para cada una. Esta variación se debe a que cada organización tiene diferentes preocupaciones de seguridad.

Por ejemplo, una institución financiera tiene diferentes prioridades de seguridad a las que puede tener una universidad. En el caso de un banco como institución financiera, es más importante la *integridad* y la *confidencialidad*, de tal manera que los clientes del banco tengan conocimiento exacto de sus activos. Por el contrario, en una universidad la prioridad puede estar en la *disponibilidad*, así la comunidad puede tener acceso a la información siempre que lo requiera.

Para iniciar el proceso de seguridad en un sistema, primero se debe definir el marco de acuerdo a una evaluación de riesgos que permitan identificar las amenazas al mismo. Posteriormente se deben definir las políticas que definan los lineamientos básicos para el funcionamiento del sistema.

Teniendo como fundamento las políticas, se definen los estándares y procedimientos necesarios que cumplan con las mismas y sirvan como guía en el uso del sistema. A continuación se verán con mayor detalle estos elementos básicos en la seguridad iniciando con la evaluación de los riesgos en el SIIEM/UAP.

1.2. Evaluación de riesgos

El primer paso antes de implementar la seguridad en un sistema es responder a las siguientes preguntas básicas[4]:

- ¿Qué es lo que se desea proteger?
- ¿Qué se debe hacer para protegerse?
- ¿Cuánto tiempo, dinero y esfuerzo se está dispuesto a invertir para obtener la protección adecuada?

La evaluación de riesgos es muy importante debido a que no es posible proteger algo si no se sabe de qué o de quién hay que defenderse. Para llevar a cabo una evaluación de riesgos podemos seguir una serie de pasos claves que nos ayuden a determinar dichos riesgos: identificar los bienes a asegurar, identificar las amenazas y realizar un análisis costo beneficio. En seguida veremos estos puntos importantes en esta evaluación.

1.2.1. Identificación de bienes

La identificación de bienes consiste en realizar un listado de lo que es necesario proteger. Los elementos a proteger incluyen las partes tangibles como dispositivos de almacenamiento electrónico, monitores, cables de red, manuales, etc.; y las partes intangibles tales como la capacidad de continuar un proceso, imagen pública, reputación, acceso a la computadora, contraseña del administrador del sistema, etcétera. La lista debe incluir todo lo que se considere valioso dentro de la organización. Para determinar si algo es valioso, se debe tomar en cuenta el costo en tiempo, dinero y esfuerzo para recuperar el elemento perdido o dañado.

Algunos de los elementos que pueden ser considerados dentro de la lista de bienes valiosos para la organización pueden ser los siguientes:

Tangibles

- Computadoras.
- Datos propietarios.
- Respaldos.
- Manuales, guías, libros.
- Impresoras.
- Medios de distribución de Software Comercial [5].

[4]Garfinkel Simson, Practical Unix & Internet Security.

[5]Por ejemplo CD de software propietario con licencia como Ms Windows 9x/2000/XP, Ms Office 97/2000/XP.

- Equipo de comunicaciones (routers, gateways,etc.).
- Registros del personal.
- Registros de auditoría, etc.

Intangibles

- Privacidad de los usuarios.
- Imagen pública y reputación.
- Contraseñas personales, etc.

El análisis de todos los bienes de la Unidad de Administración del Posgrado va más allá de este trabajo, así que sólo me limitaré a definir los bienes que tienen que ver con el SIIEM/UAP.

1. **Computadora**
   - Procesador: Intel Pentium III 750 MHz
   - Memoria RAM: 256MB
   - Disco Duro:18 GB
   - CDROM: 52x
   - Monitor: 15"
   - Marca: ENSAMBLADA
   - Costo Estimado: $3000.00 pesos
   - Uso: En esta computadora reside información de los alumnos de medicina inscritos en alguna especialización.
2. **Base de Datos**
   - Manejador: PostgreSQL 7.4.1
   - Tamaño: 40 MB
   - Registros: 61144

   Esta base de datos incluye información sobre:
   - Alumnos de Especialización de Medicina (Aspirantes e Inscritos).
   - Sedes de la Facultad de Medicina donde se imparten las especialidades.
   - Planes de estudio.
   - Profesores.
   - Jefes de enseñanza.

- Calendario de inscripción.
- Catálogos.
- Documentación de alumnos.

3. **Código Fuente**
   - Páginas HTML
     - Lenguaje de programación: HTML 1.0
     - Total de archivos: 11
     - Tamaño total: 60 KB
   - Programas CGI
     - Lenguaje de programación: Perl v5.6.0
     - Total de archivos: 98
     - Tamaño total: 2.8 MB

1.2.2. Identificación de Amenazas

El siguiente paso es determinar las amenazas que acechan los bienes. Algunas de estas amenazas pueden ser ambientales (fuego, terremotos, explosiones, inundaciones, etcétera). También se pueden incluir eventos muy raros pero que pueden suceder (por ejemplo el colapso de un avión con un edificio o una huelga estudiantil). Se deben tomar en cuenta también los actos adversos que pueden generarse por los miembros o personas ajenas a la organización. Algunos ejemplos de estas amenazas son:

- Enfermedad del personal importante dentro de la organización.
- Interrupción en el suministro de la energía eléctrica.
- Introducción de un virus.
- Empleados subversivos.
- Errores de software.
- Terrorismo político.
- Robo de medios de almacenamiento de datos tales como discos duros, cintas de respaldo, etcétera.
- Robo de computadoras personales de los empleados.
- Pérdida de utilidades.
- *Script kiddies*.

Debido a que la UAP pertenece a la UNAM, se tienen las siguientes amenazas latentes:

- Huelga estudiantil.
- Huelga de trabajadores del STUNAM[6].
- Robo de equipo.
- Terremotos[7].

El SIIEM, de acuerdo a sus características y a los bienes de los cuales se compone es amenazado por los siguientes factores:

- Accesos no autorizados al sistema operativo.
- Scaneo de puertos abiertos.
- Scaneo de vulnerabilidades.
- Negación de Servicios.
- Pérdida de información.
- Falta de políticas básicas de seguridad[8].

La base de datos del SIIEM tiene las siguiente manezas:

- Modificación de la base de datos sin autorización.
- Ingreso remoto no autorizado.
- Eliminación de la base de datos en sí.

1.2.3. Análisis Costo-Beneficio

Una vez realizada la evaluación de riesgos, es necesario asignar un costo a cada riesgo, y determinar el costo a defenderse contra él. Para determinar el costo-beneficio de la implementación de un esquema de seguridad se consideran dos criterios básicos: el costo de la pérdida y el costo de la prevención. A continuación se aborda cada uno de estos costos.

Calcular el costo de la pérdida puede ser algo muy difícil. Un cálculo simple puede incluir solamente el costo de reparar o reemplazar un elemento particular. Un análisis más sofisticado del cálculo del costo puede considerar:

- Costo de tener el equipo fuera de servicio.
- Costo de entrenamiento del personal dentro de la organización.

[6] Sindicato de Trabajadores de la Universidad Nacional Autónoma de México.
[7] La ciudad de México pertenece a una zona sísmica de gran actividad.
[8] Este tema se abordará más adelante.

- Costo de procedimientos adicionales resultado de una pérdida.
- Costo de la reputación de la organización.

Para cuestiones prácticas no es necesario asignar un valor exacto a cada riesgo posible. Regularmente, se asigna un promedio del costo para cada elemento. Por ejemplo: la pérdida de una base de datos puede ser "al rededor de $30,000 pesos", mientras que el robo de un servidor se puede estimar "al rededor de $60,000 pesos". Este cálculo aproximado puede estar en función de distintos factores como por ejemplo la devaluación del bien.

Se pueden asignar los costos basados en una escala de pérdida tan simple como "pérdida o no pérdida". Por ejemplo, se puede asignar un costo separado para cada una de las siguientes categorías:

- No disponible por un período corto (entre 7 y 10 días).
- No disponible por un período medio (entre 1 y 2 semanas).
- No disponible por un largo período (más de 2 semanas).
- Pérdida permanente o destrucción.
- Pérdida o daño parcial accidental.
- Pérdida o daño parcial inducidos.
- Acceso no autorizado dentro de la organización.
- Acceso no autorizado fuera de la organización.
- Acceso total fuera de la organización y competidores.
- Costo de reemplazo o recuperación.

En el SIIEM/UAP, al pertenecer a una institución de educación pública sin fines de lucro, el costo económico estará en función del reemplazo del equipo de cómputo. Sin embargo, la imagen de la Unidad de Posgrado también se vería afectada aunque no tendría repercursiones monetarias considerables.

Por otro lado, el costo de la prevención se obtiene desde el anásis costo-beneficio al calcular el costo de prevenir cada tipo de pérdida. Por ejemplo, el costo de recuperarse de una falla momentánea en el suministro de energía eléctrica es posiblemente el tiempo necesario para reiniciar el equipo o el tiempo que esté fuera de servicio. Sin embargo, el costo de prevención puede ser que se compre e instale una fuente de energía ininterrumpida conocida como *UPS*[9].

[9]Uninterruptable Power Suply.

En el SIIEM/UAP el costo de prevención depende de los aditamentos físicos que se añadan a las instalaciones de la UAP como: instalación de energía regulada, compra de no-brakes, instalación de dispositivos de circuito cerrado, etc. La prevensión lógica es de bajo costo debido a que el uso de software libre en detectores de instrusos, firewalls, herramientas de monitoreo, etcétera hace que la inversión en seguridad se reduzca considerablemente. Sin embargo, el posible costo económico se puede ver reflejado en la contratación de recursos humanos que instale, configure y administre estos elementos de software.

1.2.4. Reducción de Riesgos

Es posible identificar y reducir riesgos, pero no pueden ser eliminados por completo. Por ejemplo, se puede comprar una UPS para reducir el riesgo de una falla de energía eléctrica y consecuentemente evitar un daño en los datos. Pero la UPS puede fallar cuando se le necesite. Una evaluación cuidadosa identificará estos *riesgos secundarios* y ayudará a planear soluciones para ellos.

Sin embargo, existe una verdad en seguridad en cómputo: *"No importa que tan segura sea una computadora, ésta puede sufrir en cualquier momento una intrusión si el atacante cuenta con los suficientes* ***recursos, tiempo, motivación y dinero***"[10].

Al haber determinado el análisis de riesgos, el siguiente paso es definir los lineamientos necesarios mediante políticas de seguridad, para evitar o reducir el daño en caso de que se concreten las amenazas al sistema.

1.3. Políticas

Las Políticas ayudan a definir lo que es valioso, y éstas especifican que pasos se deberían seguir para asegurar dichos bienes.

Las políticas pueden ser formuladas de diferentes maneras. Se pueden escribir políticas muy generales de unas cuantas páginas que cubran muchas posibilidades. También se pueden escribir políticas para diferentes conjuntos de bienes o servicios: Políticas de control de acceso físico y lógico; Políticas de uso de equipos y Políticas de respaldo de información. Las políticas juegan tres papeles principales:

1. Definen qué es lo que debe ser protegido y cómo debe ser protegido.

2. Aclaran las responsabilidades para la protección.

3. Proveen una base sobre la cual se interpreta y resuelve cualquier conflicto posterior que pueda surgir.

[10] ibdem

Sin embargo, las políticas NO deben listar amenazas específicas, equipos e individuos, por el contrario, deben ser generales y cambiar poco con el tiempo.

En las siguientes secciones propongo las políticas básicas de seguridad que deben ser implementadas en el Sistema de Inscripciones.

1.3.1. Política de uso y modificación de la información en el SIIEM

La información manejada en el **SIIEM** de la **UAP** es responsabilidad del administrador del sistema y de los desarrolladores del mismo y se rige de acuerdo a los siguientes lineamientos:

1. Todos los desarrolladores involucrados en el SIIEM comparten la responsabilidad de mantener la operación correcta del sistema y la protección de toda la información que sea parte del mismo.

2. La información debe ser protegida de ser descubierta por personas u organizaciones ajenos a la UAP y a la Unidad de Posgrado de la Facultad de Medicina.

3. La información personal de los alumnos se considera confidencial y debe ser protegida.

4. Las actualizaciones a la información por personal autorizado fuera de la UAP será a través de la interfaz proporcionada por el equipo de desarrollo de la UAP.

5. Las modificaciones realizadas a través del acceso directo a la base de datos será realizada por el administrador del sistema a través de un usuario y contraseña definidos.

6. La información almacenada en cualquier medio (electrónico o impreso) es responsabilidad del jefe del área de cómputo y del administrador del sistema siempre y cuando el primero lo apruebe.

7. El administador del sistema es el responsable de formular estándares y procedimientos apropiados, de acuerdo a las prácticas de la UAP y el SIIEM, para asegurar la protección y correcta operación del proceso de información.

1.3.2. Políticas de control de acceso al SIIEM

El acceso físico y lógico a cualquier parte de la información del SIIEM por parte de los usuarios será a través de algún mecanismo que permita acceder sólo a aquellos involucrados en la modificación de la información.

Es responsabilidad del administrador del sistema definir los mecanismos adecuados para el acceso correcto a la información y al sistema en general.

1.3.3. Políticas de respaldos para el SIIEM

La información contenida en la base de datos del SIIEM y los programas que permiten el acceso a dicha base de datos deberán ser respaldados en medios de almacenamiento alternos (CD, cintas, papel) por el administrador del sistema.

El administrador del sistema será el encargado de definir, implementar y verificar la periodicidad y el procedimiento para realizar los respaldos que permitan mantener segura y recuperable la información.

1.4. Estándares

Los estándares son el código de prácticas exitosas de seguridad en una organización. Estos estándares están definidos en términos de compartir. Son generalmente plataformas independientes, e implican una métrica para determinar si han sido alcanzados los objetivos planteados. Los estándares son desarrollados como soporte de las políticas y cambian léntamente con el tiempo.

En seguida propongo los estándares para el SIIEM en base de las políticas planteadas anteriormente. Considero que la Política de uso y modificación de la información en el SIIEM es lo suficientemente clara y detallada que no incluyo un estándar para ella.

1.4.1. Estándar para el acceso lógico al SIIEM

El acceso de los usuarios al SIIEM estará determinado de acuerdo al papel que juega el usuario en el sistema. Estos usuarios los defino de acuerdo a las necesidades del mismo sistema. Los usuarios principales son:

- El Administrador del Sistema Operativo. El Administrador del sistema operativo tiene acceso total a la información. Este acceso será realizado a través de un nombre de usuario y contraseña. El Administrador es el encargado de registrar, modificar y/o eliminar a otros usuarios, además de mantener el equipo trabajando en condiciones óptimas y con las herramientas de seguridad mínima instaladas.

- El Administrador de la Base de Datos. El Administrador de la base de datos es el encargado de mantener la base de datos del SIIEM, generar los usuarios de la base de datos y respaldar lo datos contenidos en la misma. El Administrador de la base de datos tendrá acceso a través de un nombre de usuario y contraseña que será definida en primera instancia por el Administrador del Sistema Operativo.

- El Desarrollador. El desarrollador/programador del sistema tendrá acceso a la base de datos y a programas que tengan que ver con la interfaz de acceso para otros usuarios. El acceso se llevará a cabo por medio de un nombre de usuario y una contraseña definidos en el sistema operativo.

- Los usuarios de la Unidad de Posgrado de la Facultad de Medicina. Estos usuarios tendrán acceso a la información almancenada en la base de datos del SIIEM y sólo podrán realizar las modificaciones pertinentes de acuerdo a los permisos definidos en la base de datos. El acceso de estos usuarios será a través de un nombre de usuario y una contraseña definidos en la base de datos del SIIEM y no tendrán acceso a un shell en el sistema operativo.

- Los alumnos. Los alumnos (aspirantes, de primer ingreso o reingreso) de las Especializaciones de Medicina sólo podrán acceder a información personal y ésto se llevará a cabo mediante su número de cuenta y fecha de nacimiento.

- Los Jefes de Enseñanza. Los jefes de enseñanza sólo podrán acceder a la información referente a su sede e institución por medio de un nombre de usuario y una contraseña definidos en la base de datos del SIIEM.

1.4.2. Estándar para el acceso físico al SIIEM

El acceso físico al centro de cómputo donde reside el servidor del SIIEM/UAP debe ser controlado mediante la identificación de la persona que ingresa y con autorización del personal que labora en el lugar.

En horas de comida, o en cualquier momento que no se encuentre un colaborador de la unidad de cómputo, el sitio de servidores debe mantenerse cerrado para que personas ajenas al mismo no puedan ingresar sin autorización.

1.4.3. Estándar para los respaldos de la base de datos del SIIEM

Dada la importancia de la información contenida en la base de datos del SIIEM se debe definir una serie de elementos básicos para mantener copias de respaldo. El principal factor es la periodicidad, la cual está determinada por el período de inscripciones del sistema que es el tiempo en el cual se efectuan las inscripciones de los alumnos al ciclo escolar. En este período la información que se agrega y modifica en la base de datos es muy grande. El otro período es el de mayor duración en el cual no se realizan las incripciones y es el tiempo que pasa el sistema sin inscribir alumnos de primer ingreso o reingreso. En los casi 9 meses que dura este período, la información que es agregada o modificada en el sistema varía poco.

En base a estos períodos de uso de la base de datos, se define la frecuencia de respaldos de acuerdo a este criterio.

- Respaldo en período de inscripciones. El respaldo en período de inscripciones debe realizarse tomando en cuenta el volumen de la información y las modificaciones que se hacen en las tablas, de tal manera, que se realizarán respaldos totales e incrementales de la siguiente forma:

- Respaldo inicial. Se realizará un respaldo total un día hábil antes del primer día de inscripciones.
- Respaldos semanales. Los respaldos semanales incluyen toda la base de datos y se deben realizar los días viernes al final del horario de inscripciones.
- Respaldos diarios. Se deben hacer respaldos diarios de las tablas que tengan mayor demanda durante el período de inscripciones.

- Respaldo en período de no inscripciones. Los respaldos en períodos en los cuales no existen registros de alumnos se deberán hacer mensual o quincenalmente tomando en cuenta toda la información de la base de datos.

Las copias de respaldo, sin importar la periodicidad, deben cumplir con una serie de elementos que permitan salvaguardar la información. A continuación se listan los elementos mínimos que deben tener dichos respaldos:

1. Utilizar el programa que proporciona el manejador de la base de datos (DBMS [11]) para realizar los respaldos.
2. El nombre de los archivos debe ser definido de tal manera que permita identificar la fecha en la cual fue realizado y el nombre de la base de datos o de la tabla en particular. Se pueden utilizar las siguiente características en el nombre del archivo:
   - Dos dígitos para indicar el día.
   - Tres caracteres para indicar el mes.
   - Cuatro dígitos para indicar el año.
   - Nombre de la base de datos o de la tabla.
   - Extensión .sql

   Por ejemplo: **13Ene2004medicina.sql**
3. Generar la firma digital del respaldo.
4. Cifrar el archivo.

CONSIDERACIONES SOBRE EL ALMACENAMIENTO DE LAS COPIAS DE SEGURIDAD

El respaldo de la información del SIIEM debe ser almacenada en diferentes medios que permitan tener una mayor disponibilidad de la información. Por tal motivo se debe guardar una copia de seguridad en los siguientes medios de almacenamiento.

- En el disco duro del servidor donde se encuentra la base de datos.
- Medio de almancenamiento removible (cinta, CD, disquete,etc).
- En el disco duro de otro servidor (de preferencia fuera del edificio donde se encuentra la máquina que funciona como servidor de la base de datos) que cumpla con los mecanismos mínimos de seguridad.

[11] Data Base Manager System

1.5. Procedimientos

El objetivo de los procedimientos es interpretar los estándares para ambientes particulares, ya sea un ambiente físico o de software. A diferencia de los estándares, los procedimientos pueden ser modificados si es necesario. Como el nombre lo indica, los procedimientos no son usados como estándares de desempeño, si no más bien como una guía.

En seguida propongo el procedimiento para el respaldo de la información de acuerdo al estándar propuesto. En el caso de los estándares de control de acceso físico y lógico considero que son suficientes por lo que no se definen procedimientos para ellos.

1.5.1. Procedimiento de respaldo de información del SIIEM

Dados los estándares de respaldo para el SIIEM, a continuación se muestra una guía que permita generar los respaldos de información de la base de datos.

1. Generación de la copia. Para generar el respaldo de seguridad, se debe utilizar el programa proporcionado por el manejador de la base de datos diseñado para dicho propósito. El DBMS utilizado en el SIIEM es PostgreSQL[12], el cual proporciona el comando **pg_dump** para generar un vaciado de la información contenida en la base de datos.

   La forma de generar un respaldo de toda la base de datos del SIIEM es:

   pg_dump -D nom_bd >ddmmmaaaanom_bd.sql

   Donde:

   - pg_dump. Nombre del programa para realizar el vaciado de la información.
   - -D. Opción que permite respaldar la estructura y los datos en un sólo archivo. *En el caso de que se respalde una tabla, es necesario incluir la opción -t y el nombre de la tabla.*
   - nom_bd. Nombre de la base de datos.
   - ddmmmaaaanom_bd.sql. Nombre del archivo resultante:
     - dd. Día.
     - mmm. Nombre abreviado del mes.
     - aaaa. Año.
     - nom_bd. Nombre de la base de datos. *En el caso de que sea el respaldo de una sola tabla, debe sustituirse el nombre de ésta por el de la base de datos.*
     - .sql. Extensión que indica que se trata de un archivo con instrucciones SQL.

[12] En el capítulo Seguridad en Bases de Datos se abundará más sobre este DBMS.

2. Firma digital. Para realizar ésto se utiliza el programa **md5sum** que genera una firma utilizando el algoritmo MD5 (Message Digest):

   md5sum ddmmmaaaanom_bd.sql >ddmmmaaaanom_bd.sql.md5

   El archivo ddmmmaaaanom_bd.sql.md5 contiene la firma digital.

3. Cifrado. Para realizar el cifrado se puede utilizar el comando **gpg** [13], el cual implementa el estándar de Internet OpenPGP descrito en el *RFC* 2440. Esta herramienta es utilizada para cifrar y firmar datos. A continuación se muestra un ejemplo:

   gpg -r id_usuaio -e ddmmmaaaanom_bd.sql

   La opción -e permite cifrar el archivo con la llave pública. La opción -r indica el usuario del cual se utilizará la llave pública para cifrar. Con ésto se crea un archivo **ddmmmaaaanom_bd.sql.gpg** que sólo puede ser descifrado por la llave secreta de quién cifró el archivo.

En seguida se muestra un ejemplo completo del procedimiento (el símbolo $ indican el prompt del sistema):

- Genera el vaciado de la base de datos

```
$ pg_dump -D medicina > 06Ene2004medicina.sql
```

- Crea la firma digital

```
$ md5sum 06Ene2004medicina.sql > 06Ene2004medicina.sql.md5
```

- Muestra el contenido de la firma

```
$ cat 06Ene2004medicina.sql.md5
dc95c99b8f256f5106c40aed574afc35  06Ene2004medicina.sql
```

- Cifra el archivo

```
$ gpg -r postgres -e 06Ene2004medicina.sql
```

- Transfiere el archivo y la firma digital a otro servidor.

```
$ scp 06Ene2004medicina.* noether:.Posgrado/respaldos/newman
fer@noether's password:
06Ene2004medicina.sql.md5   100% |**********|  56      00:00
06Ene2004medicina.sql.gpg   100% |**********| 3586 KB  00:00
```

[13] GNU Privacy Guard. El capítulo cuarto abunda más en esta herramienta.

Es importante que se lleve a cabo este procedimiento para respaldar adecuadamente la información de la base de datos del SIIEM y cumpla con la integridad y confidencialidad necesarias.

Se puede apreciar que el procedimiento de respaldo está compuesto de distintas partes que lo pueden hacer engorroso o que se omita por error alguno de los pasos, por lo que se diseñó un programa para automatizar este proceso.

1.5.2. Automatización del procedimiento de respaldo

Debido a que los respaldos se realizarán periódicamente es conveniente generar un programa que permita realizar estas tareas de manera automática y que corresponda con las características pianteadas en el procedimiento del respaldo.

La automatización de los respaldos está compuesta de tres elementos:

1. Programa. Esta parte será la encargada de realizar todo el procedimiento del respaldo.
2. Transferencia segura de la copia de respaldo. Como parte del procedimiento de respaldo, se envía una copia de seguridad a un equipo remoto. Para realizar esta tarea es necesario crear una llave pública para que los equipos se puedan comunicar de forma segura.
3. Calendarización de las tareas. En sistemas Unix/Linux existen mecanismos para programar tareas que pueden ser realizadas en un tiempo determinado. En este caso se hará uso de la herramienta *cron* para calendarizar el proceso de respaldo.

Programa de respaldo

El programa que realiza los respaldos está escrito en lenguaje perl con ayuda de instrucciones del sistema operativo. Este programa se muestra a continuación:

```
#!/usr/bin/perl
use Date::Calc qw/Today_and_Now/;
#############################################################
#Descripcion:   Programa que hace  el respaldo  total o     #
#               parcial de la base de datos medicina.       #
#               Este programa esta disenado para ser        #
#               utilizado por el cron                       #
#Autor:         Fernando Zaragoza Hernandez                 #
#Fecha:         13 de Marzo del 2003                        #
#Modificaciones:28 de Junio del 2003                        #
#               25 de Enero del 2004                        #
#############################################################
```

```
#Define variables
$DIR='/home/postgres/respaldos';
$log="/home/postgres/respaldos/respaldo.log";
$bd='medicina';

#Usuario con el cual se cifra el respaldo.
$ugpg="postgres";

#Direccion de correo electronico para enviar un mensaje de error
$umail="fer\@newman.posgrado.unam.mx";

#Ruta del comando de copia segura
$scp="/usr/local/bin/scp";

#En este arreglo se tienen los meses abreviados
%meses=('1'=>'Ene',
        '2'=>'Feb',
        '3'=>'Mzo',
        '4'=>'Abr',
        '5'=>'May',
        '6'=>'Jun',
        '7'=>'Jul',
        '8'=>'Ago',
        '9'=>'Sep',
        '10'=>'Oct',
        '11'=>'Nov',
        '12'=>'Dic');

#Verifica si esta incluida la opcion para respaldar una tabla
if ( $ARGV[0] =~ '-t' )
{
  parcial();
}
else
{
  total();
}
exit;

sub parcial
{
  my ($anio,$mes,$dia,$hora,$min,$sec) = Today_and_Now();

  #Inicia la bitacora.
  open FH,">> $log" or die "No fue posible crear el archivo de
```

```
            bitacora $log para el respaldo de la base de datos";
print FH "Inicio del Respaldo: $dia/$meses{$mes}/$anio
          $hora:$min:$sec \n";
shift @ARGV;
foreach $tabla (@ARGV)
{
  ($anio,$mes,$dia,$hora,$min,$sec) = Today_and_Now();
  $fecha="$dia$meses{$mes}$anio";
  print FH "$dia/$meses{$mes}/$anio $hora:$min:$sec Respaldando
            $tabla en $bd\n";

  ##### Proceso del respaldo #####
  #Genera el vaciado de la base de datos

  system("/home/postgres/bin/pg_dump -Dt $tabla $bd >
          $DIR/${fecha}${tabla}.sql");
  if ( $? != 0 )
  {
    print FH "Error al vaciar la tabla $tabla de la base de
              datos $bd\n";
    system("echo No fue posible completar el respaldo de la base
            de datos | mail -s 'Error en el respaldo (pg_dump)'
            $umail");
    close(FH);
    exit 1;
  }

  #Crea la firma digital del archivo
  system("md5sum $DIR/${fecha}${tabla}.sql >
          $DIR/${fecha}${tabla}.sql.md5");

  #Cifra el archivo
  system("gpg -r $ugpg -e $DIR/${fecha}${tabla}.sql");

  if ( $? != 0 )
  {
    print FH "Error al cifrar $DIR/${fecha}${tabla}.sql\n";
    system("echo No fue posible completar el respaldo de la base
    de datos | mail -s 'Error en el respaldo (gpg)' $umail");
    close(FH);
    exit 1;
  }

  #Borra el archivo plano
  system("rm $DIR/${fecha}${tabla}.sql");
```

```
    #Envia el respaldo al equipo remoto
    system("$scp $DIR/${fecha}${tabla}.sql* fer\@noether:.Posgrado
            /respaldos/newman");
    }

  #Fin del proceso
  ($anaio,$mes,$dia,$hora,$min,$sec)=Today_and_Now();
  print FH "Fin del Respaldo: $dia/$meses{$mes}/$anio $hora:
            $min:$sec \n";
  close(FH);
}

sub total
{
  #Obtiene la fecha actual
  my ($anio,$mes,$dia,$hora,$min,$sec) = Today_and_Now();

  #Inicia la bitacora.
  open FH,">> $log" or die "\n No fue posible crear el archivo
                            de bitacora $log";
  print FH "Inicio del Respaldo: $dia/$meses{$mes}/$anio $hora:
            $min:$sec \n";
  print FH "Respaldando la BD $bd\n";
  $fecha="$dia$meses{$mes}$anio";

  ##### Proceso del respaldo #####
  #Genera el vaciado de la base de datos

  system("/home/postgres/bin/pg_dump -D $bd >
          $DIR/${fecha}$bd.sql");
  if ( $? != 0 )
  {
    print FH "Error al vaciar la base de datos $bd\n";
    system("echo No fue posible completar el respaldo de la base
            de datos | mail -s 'Error en el respaldo' $umail");
    close(FH);
    exit 1;
  }

  #Crea la firma digital del archivo
  system("md5sum $DIR/${fecha}$bd.sql >
          $DIR/${fecha}$bd.sql.md5");

  #Cifra el archivo
```

```
  system("gpg -r postgres -e $DIR/${fecha}$bd.sql");
  if ( $? != 0 )
  {
    print FH "Error al cifrar la base de datos $bd\n";
    system("echo No fue posible completar el respaldo de la
            base de datos | mail -s 'Error en el respaldo
            (gpg)' $umail");
    close(FH);
    exit 1;
  }

  #Borra el archivo plano
  system("rm $DIR/${fecha}$bd.sql");

  #Envia el respaldo al equipo remoto
  system("$scp $DIR/${fecha}$bd.sql* fer\@noether:.Posgrado
          /respaldos/newman");

  #Fin del proceso
  ($anio,$mes,$dia,$hora,$min,$sec)=Today_and_Now();
  print FH "Fin del Respaldo: $dia/$meses{$mes}/$anio
            $hora:$min:$sec \n";
  close(FH);
}
```

Con este programa se realiza la automatización del proceso de respaldo. El siguiente paso es generar las llaves para que se pueda llevar a cabo la copia del respaldo a un sistema remoto.

Transferencia segura de la copia del respaldo

La transferencia de los archivos de respaldos se realiza en horarios en los cuales ningún operador está presente para que pueda ingresar una contraseña que permita copiar los archivos en el equipo remoto. Para poder realizar una copia de respaldo en el servidor remoto es necesario autorizar mediante una llave. La forma de hacer esto se muestra a continuación:

- Generar la llave. Se debe generar una llave en la máquina local para que sea aceptada en el equipo remoto. Con la siguiente instrucción se genera la llave.

  ```
  $ ssh-keygen -d
  ```

  Con este comando se genera la llave y el comando solicita la ruta donde se guardará la misma:

```
Generating public/private dsa key pair.
Enter file in which to save the key (/home/fer/.ssh/id_dsa):
```

Posteriormente se solicita una frase de paso para la llave. En este punto se debe dejar en blanco la contraseña.

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

Finalmente, el comando muestra un mensaje cuando la llave ha sido creada.

```
Your public key has been saved in
/home/postgres/.ssh/id_dsa.pub.
The key fingerprint is:
f8:20:5b:61:e0:aa:e0:7e:3a:92:ff:42:92:d4:c5:ad
postgres@newman.posgrado.unam.mx
```

- Copiar el archivo id_dsa.pub en el sistema remoto.
- En el sistema remoto agregar el archivo id_dsa.pub al archivo de llaves autorizadas authorized_keys2 (el cual se encuentra ubicado en el directorio .ssh en el directorio de trabajo del usuario remoto) con el siguiente comando:

```
cat id_dsa.pub >> .ssh/authorized_keys2
```

Con el procedimiento anterior ya se ha garantizado la comunicación entre los equipos para realizar una copia remota, ahora se procede a calendarizar el respaldo.

Calendarización

El administrador de la base de datos utiliza el comando *cron* para definir la hora en la cual se ejecutará el programa de respaldo. Las instrucciones para el comando *cron* se muestran a continuación:

```
#Realiza el respaldo semanal de toda la base de datos
55 23 * * 5 /home/postgres/progs/respalda/respaldo.pl

#Realiza el respaldo diario de solo tres tablas
55 23 * * 1,2,3,4 /home/postgres/progs/respalda/respaldo.pl
                  -t almed inscrito aspirantes
#Realiza el respaldo cada 15 dias de tres tablas (para respaldos
 en periodos que no sean de inscripcion)
#55 23 1,15 * *  /home/postgres/progs/respalda/respaldo.pl
                  -t almed inscrito aspirantes
```

Donde las líneas que inician con # indican comentario. La segunda línea, de acuerdo a la sintáxis de *cron* significa lo siguiente: se ejecutará el comando *respaldo.pl* los días viernes a las 23 horas con 55 minutos. En este caso se realiza el respaldo total.

La cuarta línea indica que se ejecutará el comando *respaldo.pl -t almed inscrito aspirantes* los días lunes,martes,miércoles, y jueves a las 23 horas con 55 minutos.

La última línea define el respaldo de tres tablas los días 1 y 15 de cada mes a las 23 horas con 55 minutos.

De esta forma se estará realizando el procedimiento del respaldo de manera automática.

Como se ha visto a lo largo de este capítulo, la seguridad en cómputo no tiene una definición totalmente aceptada, sin embargo, se considerará a un equipo seguro en la medida de que funcione como se espera que lo haga. Existen diferentes tipos de seguridad que tendrán un peso distinto de acuerdo a las necesidades de seguridad de cada organización:

1. Confidencialidad.
2. Integridad.
3. Disponibilidad.
4. Consistencia.
5. Control.
6. Auditoría.

Sin embargo, para poder realizar un buen esquema de seguridad, es importante llevar a cabo un procedimiento cuidadoso que permita definir adecuadamente lo siguiente:

- Identificación de bienes.
- Identificación de amenazas.
- Análisis costo-beneficio.

Como parte final de este capítulo se determinaron las políticas, estándares y guías que permitan definir el control de acceso y metodología de respaldos en el SIIEM.

Por lo anterior concluyo que la Seguridad en Cómputo es un elemento importantísimo en un sistema en el cual la información que se mantiene es accedida por diferentes personas con diversas características, además de que un sistema en Internet, el cual es un ambiente áspero donde se goza del anonimato, está expuesto a una gran cantidad de personas de las cuales algunas pueden ser posibles intrusos que desearán ingresar a nuestro sistema sin autorización.

El siguiente componente del SIIEM que debe ser asegurado es el servidor web, debido a que mediante este servicio acceden los usuarios del sistema a través de Internet. Este tema se analiza en el siguiente capítulo.

Capítulo 2

Seguridad en Web

Internet es la plataforma de uso del Sistema de Inscripciones por Internet de Especializaciones de Medicina de la Unidad de Administración de Posgrado (SIIEM/UAP), de tal manera que existe un servidor de web el cual responde a las peticiones realizadas por los usuarios. Sin embargo, como se verá a lo largo de este capítulo, existen amenazas que afectan este servicio tan popular actualmente, por lo que es necesario definir los mecanismos para asegurarlo.

En este capítulo se abordarán los problemas concernientes a la Seguridad en Web y las implicaciones que esto tiene en el SIIEM/UAP. En esta parte del trabajo se abordarán los siguientes temas:

- *Breve Historia del Web*
- *El problema de la seguridad en Web*
- *Criptografía*
- *Secure Socket Layer*
- *Certificados digitales*
- *Instalación y configuración de SSL en el SIIEM/UAP*
- *Creación del Ceritificado Digital para el SIIEM/UAP*

2.1. Breve Historia del Web

A finales de los 60 surge la Agencia de Proyectos de Investigación Avanzados (ARPA[1]) del Departamento de Defensa de los Estados Unidos con el fin de desarrollar una tecnología de comunicaciones que fuera capaz de subsistir, aún si alguna línea de comunicación quedara fuera. A esta tecnología se le conoce como *switcheo* de paquetes.

El *switcheo* de paquetes empezó con el único objetivo de hacer las redes de teléfono y teletipos más confiables. El *switcheo* de paquetes resuelve el problema de conectar diferentes partes de una red, aún cuando las líneas de comunicación directa fueran interrumpidas. En una red con *switcheo* de paquetes, cada pieza de la información que viaja a través de la red es dividida en trozos llamados paquetes. Además de los datos que lleva el paquete, éste contiene la dirección de la fuente y la dirección de destino. Estas direcciones son la clave de que cada red con *switcheo* de paquetes funcione adecuadamente.

Con el paso del tiempo y debido al desarrollo de nuevos protocolos, la ARPANET evolucionó a lo que conocemos hoy en día como Internet. Dentro de Internet han existido una gran variedad de servicios como:

- Gopher
- Archie
- Correo electrónico
- FTP (File Transfer Protocolo)
- WWW (Worl Wide Web)
- Mensajería instantánea

Ghopher y Archie han desaparecido, actuamente el WWW, el correo electrónico y la mensajería instantánea son los servicios más populares, esto ha ocasionado que Internet tenga un gran éxito.

Pero, ¿por qué Internet se ha vuelto en una herramienta tan importante como lo es ahora?, a tal grado que es capaz de tener una gran cantidad de información asi como proporcionar una gran cantidad de servicios en línea como banca y comercio electrónicos. Es fácil de atribuir el éxito de Internet y el web a la combinación de tres elementos:

- Consumismo.
- Necesidades de mercado.
- Determinación de grandes empresas e individuos.

O, también se puede asumir que el uso masivo de computadoras y el acceso a la red de redes a través de líneas telefónicas han influido en este auge.

También se puede argumentar que el web fue empujado al mundo por compañías como Cisco, IBM, Dell y Compaq, empresas que se empeñaron en convencer a los líderes de negocios que ellos no tendrían éxito si no hacían negocios *en línea*. Por supuesto, el aparente éxito de compañías como Amazon.com, Yahoo y Verisign ayudaron a crear un clima de miedo hacia la "vieja economía" al final del siglo XX; el rápido crecimiento de

[1] Advanced Research Projects Agency

firmas que basaban sus negocios en Internet y sus repuntes en Wall Street, hicieron que muchas firmas sintieran que la única elección de continuar con vida era sólo por medio de los servicios en línea.

Desgraciadamente, estos argumentos tienen ciertas fallas. Es un error atribuir el éxito de Internet y el web a la combinación de tiempo y fuerzas de mercado. Después de todo, Internet fue sólo una de muchas redes de computadoras a gran escala que fueron desarrolladas en las décadas de los 70, 80 y 90. En cambio, por muchos años los visionarios de la industria apostaron sobre una red llamada la Interconexión de Sistemas Abiertos (OSI[2]). Por ejemplo, IBM y HP gastaron cientos de millones de dólares desarrollando productos basados en el modelo OSI, este estándar fue asignado por mandato del gobierno de los Estados Unidos, el cual en 1990 vió a Internet y al TCP/IP como un paso de transición.

Sucedió algo similar con el World Wide Web, con su Protocolo de Transferencia de HiperTexto (HTTP[3]) y el Lenguaje de Marcas de HiperTexto (HTML[4]), se pensó que podría llegar a ser la biblioteca de información universal del mundo a principios del siglo XXI. En los últimos treinta años se han visto docenas de sistemas de recuperación de información e hipertexto, desde Xanadu de Ted Nelson (1960), hasta las redes Gopher y Archie a principios de los 90.

La razón por la cual Internet y el web han sido ampliamente exitosos no se debe al resultado del mercado o tiempo, sino más bien es el resultado de su diseño. Un diseño que fue técnicamente superior a sus competidores, abierto a los desarrolladores, fácil de utilizar para la gente y libre de ser adquirido.

2.2. El problema de la Seguridad en Web

Como se trató anteriormente, el Web es uno de los servicios más utilizados en Internet y uno de los medios más socorridos por muchas organizaciones para reafirmar su presencia globalizadora. A nuevas tecnologías también surgen nuevas amenazas y el web no es la excepción. ¿Por qué la seguridad en Web es importante?. Porque las organizaciones deben anticipar las posibilidades de ataque hacia el servidor web y estar preparadas para lidiar con este problema. Si un ataque es exitoso y no se está preparado, la organización puede sufrir la modificación del sitio o que sea innaccesible. La figura 2.1 muestra la página principal de la Facultad de Medicina de la UNAM alterada el 22 de marzo del 2003 en favor de la guerra de Estados Unidos contra Irak.[5].

Debido a la alteración de esta página surgieron notas en medios de información como

[2]Open System Interconnection.
[3]HyperText Transfer Protocol.
[4]HyperText Markup Language.
[5]Fuente: http://www.zone-h.org

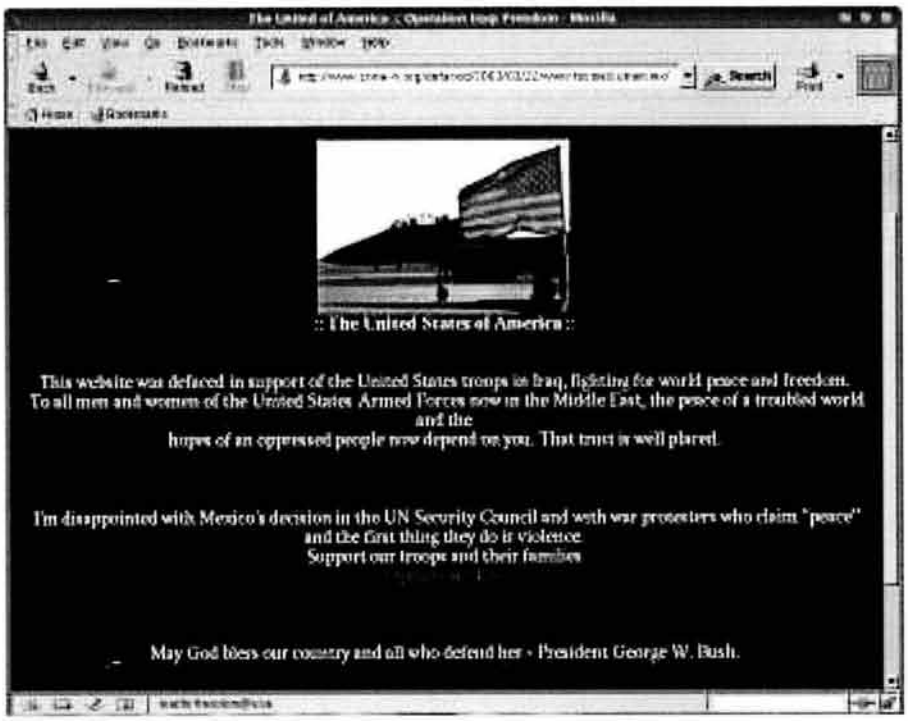

Figura 2.1: Página alterada en favor de la guerra de Estados Unido vs Irak

el periódico la Jornada [6] y CNN [7] acerca de un intruso que ingresó al equipo donde se aloja la página de la Facultad de Medicina.

Como se mencionó en el capítulo anterior, el objetivo principal de la seguridad en cómputo es minimizar los riesgos y propiciar que las computadoras se comporten como esperamos que lo hagan. Siendo específicos, la Seguridad en Web es un conjunto de procedimientos, prácticas y tecnologías para asegurar la confianza, operación predecible y todo aquello que rodea la infraestructura en Internet. Desafortunadamente, la escala total y la complejidad de Internet hacen que este problema sea más difícil que el problema de la seguridad del web específicamente.

En la actualidad, el problema de la seguridad en web tiene cuatro facetas principales:

- *Asegurar el equipo de cómputo.* Utilizar medidas adecuadas de seguridad física y lógica para que la computadora en la que residen las aplicaciones sea lo suficientemente segura.

- *Asegurar el servidor de web y los datos que en él residen.* Es necesario estar seguro que el servidor pueda continuar su operación, que la información en el servidor no pueda

[6] http://www.jornada.unam.mx/2003/mar03/030324/020n3pol.php?origen=politica.html
[7] http://www.cnn.com.mx/2003/tec/03/24/unam_hacker.ap/index.html

ser modificada sin autorización y que la información sea distribuida sólo a aquellos individuos autorizados.

- *Asegurar la información que viaja a través del servidor web y el usuario.* Asegurar que la información que proporciona el usuario al servidor web (nombres de usuario, contraseñas, información financiera, nombres de las páginas web visitadas, etc.) no sea leída, modificada o destruida por terceros. También se debe proteger la información que va del servidor web al usuario. Es importante asegurar que el enlace entre el usuario y el servidor no sea interrumpido fácilmente.

- *Asegurar la computadora del usuario final y cualquier otro dispositivo que utilice para acceder a Internet.* Por último, la seguridad en web requiere que la computadora del usuario final esté asegurada en la medida de lo posible. Los usuarios necesitan ejecutar sus *navegadores* de web y cualquier otro software en un ambiente de cómputo seguro, libre de virus y otros componentes hostiles de software. Los usuarios también necesitan protección de su privacidad e información personal, estar seguros que ésta no será comprometida ya sea por su computadora o por su servicio en línea.

Cada una de estas tareas puede ser dividida en muchas otras, esto dependerá de las necesidades que se tengan. Por ejemplo, en el caso de un editor, el objetivo de asegurar el servidor web usado en banca electrónica puede incluir las siguientes actividades:

- Determinar e implementar un sistema para verificar la identidad de los usuarios que se conectan al servidor web para visualizar sus estados de cuenta, este proceso es conocido como *autenticación*. Un sistema de autenticación incluye la implementación de un sistema de usuarios y contraseñas, determinar una técnica para la distribución del nombre de usuario y contraseña iniciales, y crear un mecanismo para que los usuarios puedan cambiar de manera segura sus contraseñas u obtener nuevas contraseñas cuando ésta sea demasiado vieja o sea olvidada.

- Analizar los programas y *scripts* que operan en el sitio web por posibles fallas o vulnerabilidades.

- Proveer respaldos de la información del usuario, fuera del sitio donde se aloja originalmente dicha información.

- Crear un mecanismo de ingreso seguro y auditoría al sistema que pueda ser utilizado para la facturación, resolución de conflictos, también llamado *no repudiación*, y la investigación de mal uso del sistema.

- Balancear la carga entre múltiples servidores para proteger contra el tráfico excesivo de información, fallas de hardware y proveer un servicio siempre disponible.

- Crear un centro de datos alterno, así que en el caso de un desastre (por ejemplo un terremoto, inundación o explosión) que afecte el centro de datos primario, el servicio pueda mantenerse sin ser afectado.

- Proveer conexiones redundantes a Internet, utilizando múltiples proveedores de servicio, para minimizar las oportunidades de que el servicio sea interrumpido sobre Internet y prevenir que los usuarios siempre encuentren el sitio web.
- Asegurar el servicio del Sistema de Nombres de Dominio (DNS[8]), así que un atacante no pueda cambiar el nombre de dominio para que haga referencia a otro servidor externo o de la misma organización.
- Proteger los registros de facturación, así los clientes pagarán la cantidad exacta por los servicios proporcionados por el banco.
- Crear un Centro de Operaciones de la Red (NOC[9]) que proporcione servicio las 24 horas, o emplear los servicios de una organización de monitoreo externo, de tal manera que si existe un incidente de seguridad el banco será capaz de responder rápidamente.
- Proporcionar la seguridad física del sitio y los servidores.
- Proveer entrenamiento adecuado para el personal, para que este grupo de trabajo conozca lo que se debe hacer en caso de una emergencia y puedan resistir un ataque de *ingeniería social*.

Como se puede ver, los elementos en la lista anterior incluyen tecnología que necesita ser creada y desarrollada, procedimientos que deben ser seguidos y políticas que deben ser desarrolladas. La seguridad no es una característica adicional que debe ser tomada en cuenta antes de que suceda un problema y simplemente adaptarlo a un sistema existente. La seguridad no es solamente un conjunto de políticas que puedan ser implementadas dentro de una organización por una persona quien tiene el nombramiento de Jefe de Seguridad.

Construir un ambiente de cómputo seguro es una tarea que requiere planeación cuidadosa y vigilancia constante. La recompensa es una infraestructura de cómputo continua y que se enfrenta a la adversidad, ya sea que esta adversidad sea generada por una persona o por algún desastre natural. En las siguientes secciones analizaré cada componente asegurado en el SIIEM/UAP.

2.2.1. Asegurar el equipo de cómputo

Asegurar el servidor web es un proceso que consta de tres partes. Primero, la computadora debe ser asegurada utilizando técnicas tradicionales de seguridad en cómputo. Segundo, los programas especiales que proveen el servicio de web deben ser asegurados. Por último, se debe examinar el sistema operativo y el servicio de web para ver si hay cualquier interacción inesperada que pueda comprometer la seguridad del sistema.

[8]Domain Name System.
[9]Network Operations Center.

La seguridad del servidor es complicada debido a que muchos servidores web son ejecutados en sistemas operativos de propósito general, tales como Unix o Windows NT/2000/XP. El servidor web puede ser utilizado para explotar algún *bug* en la seguridad del equipo, de igual forma, problemas de seguridad en el equipo pueden ser utilizados para obtener acceso al servidor web. Considérense las dos situaciones típicas:

- Un script o aplicación escritos sin consideraciones mínimas de seguridad, pueden permitir cambiar el archivo de configuración del servidor web, el cual puede ser entonces modificado de tal manera que el servidor web se ejecute con exceso de privilegios. Para explotar un error de seguridad del equipo, un atacante podría crear un script con privilegios sobre el servidor web que le permitiría obtener acceso completo al sistema de cómputo.

- Por otro lado, un servidor web puede tener scripts bien escritos y puede estar ejecutándose sobre un sistema operativo seguro, pero posiblemente, el servidor de base de datos al cual tiene acceso, puede contener una cuenta preconfigurada que permita el acceso completo a cualquier persona en Internet. Por conectarse al servidor de la base de datos y ejecutar algunos comandos, un atacante puede obtener acceso a los nombres, direcciones de correo eléctronico y números de tarjeta de crédito de muchos clientes que hayan enviado esta información al sitio web.

Asegurar el sistema de cómputo tiene que ver con una revisión completa del hardware de la computadora, sistema operativo y programas agregados. El objetivo de este proceso es asegurar que los usuarios autorizados del sistema tienen suficientes capacidades o los privilegios necesarios para desarrollar su trabajo y nada más. Técnicas tradicionales de seguridad en cómputo también están diseñadas para asegurar el sistema, de tal manera que personas ajenas en Internet no rompan las barreras de seguridad y obtengan control.

El SIIEM/UAP está soportado por el sistema operativo Linux, por tal motivo el aseguramiento práctico del servidor es significativamente diferente al sistema operativo Windows, sin embargo en escencia es muy similar. Para asegurar el equipo de cómputo se proponen las siguientes tareas que se detallan más adelante:

1. Mantener el sistema operativo y las aplicaciones actualizadas.
2. Ofrecer sólo los servicios necesarios.
3. Mantener mecanismos de autenticación de usuarios.
4. Configuración adecuada de permisos de objetos.
5. Identificar y habilitar los mecanismos de registro del sistema y red.
6. Configurar la computadora para la realización de respaldos.
7. Implementar medidas de seguridad física.

Mantener el sistema operativo y las aplicaciones actualizadas

Debido a que el diseño e implementación de software es sumamente complejo, es frecuente que aún después de haber sido liberado se encuentren problemas en la programación que afecten la seguridad del software. Después de que ha sido descubierta una vulnerabilidad en el software[10] los fabricantes requieren de un tiempo para realizar las modificaciones necesarias antes de liberar código libre de la vulnerabilidad descubierta, algunos fabricantes lo llaman parche, actualización, hot fix o service pack.

La importancia en la actualización del software utilizado en un sistema radica en la trascendencia que puede tener un sistema vulnerable mientras no sea actualizado. Esto es, en el período que transcurre entre el descubrimiento de la vulnerabilidad y la liberación del parche, el sistema se encuentra vulnerable a la existencia de un programa que explote dicha vulnerabilidad. La actualización del software es más importante cuando se trata de elementos críticos en el sistema, por ejemplo el sistema operativo o los servicios proporcionados por el equipo, dado a que éstos son la puerta de entrada al sistema.

Para mantener el sistema operativo y las aplicaciones actualizados, se definen a continuación los elementos necesarios para ello:

- *Mantener fuentes de información acerca de problemas de seguridad y actualización de software.* Definir las fuentes de las cuales se puede obtener información concerniente a la seguridad y actualización del sistema. Para obtener información actualizada sobre seguridad se puede recurrir a sitios en línea acerca del sistema operativo y de seguridad en general como:
  - Red Hat. Sitio de la distribución de Linux utilizada por el SIIEM/UAP. *http://www.redhat.com.*
  - Computer Emergency Response Team (Cert). *http://www.cert.org*
  - Forum of Incident Response and Security Teams (First). *http://www.first.org*
  - Departamento de Seguridad en Cómputo de la UNAM. *http://www.seguridad.unam.mx*
  - Equipo de Respuesta a Incidentes de Cómputo de la UNAM, el UNAM-CERT. *http://www.unam-cert.unam.mx*
  - Security Focus. *http://www.securityfocus.org*
  - SysAdmin, Audit, Network, Security (Sans). *http://www.sans.org*
- *Definir procedimientos de monitoreo de las fuentes de información.* Determinar los medios necesarios para obtener la información de seguridad y actualización de software. Las listas de correo son muy útiles debido a que en el momento de una noticia relevante, ésta llegará a los suscriptores de la lista. Algunas de las listas más importantes de seguridad se muestran a continuación:

[10] Algunas veces las vulnerabilidades son descubiertas por el fabricante del software, pero la mayoría de las veces son descubiertas por usuarios o investigadores externos.

- Bugtraq. Lista abierta de seguridad manejada por Security Focus la cual se basa en el concepto de *full disclosure*.
- cert-advisory. Lista del CERT que maneja los boletines de seguridad de aplicaciones que puedan tener un amplia repercusión a nivel internacional.
- El Departamento de Seguridad en Cómputo de la UNAM mantiene listas de correo como son:
  - GASU. Lista del Grupo de Administración y Seguridad en Unix.
  - CERT-Avisos. Lista donde se emiten los boletines de seguridad del CERT en español.

- *Evaluación de las actualizaciones.* No todas las actualizaciones aplicarán a las herramientas de software instaladas o a las políticas de seguridad definidas en la organización. Por lo anterior, es necesario determinar si las nuevas modificaciones al software deben ser instaladas en el servidor.

- *Planeación de las actualizaciones aplicables.* Cuando se instalan actualizaciones de aplicaciones es importante planear cuando y como debe ser realizadas dichas actualizaciones. En redes de pocas computadoras tal vez la actualización no sea problema, sin embargo, en organizaciones con cientos de computadoras que requieran la actualización de algún elemento de software serán necesarias herramientas que permitan realizar esta tarea de forma sencilla.

  No importa como se realice la actualización, se deben definir las formas de cómo llevar a cabo tal actividad. Sin embargo, existen ocasiones en las cuales no se pueden planear las actualizaciones debido a que al recibir un reporte a través de los canales definidos anteriormente para la obtención de información sobre seguridad, la recomendación es realizar las actualizaciones lo más pronto posible, ya que la vulnerabilidad reportada representa un problema grave de seguridad para el sistema.

- *Instalación de las actualizaciones de acuerdo a un plan documentado.* Después que se ha definido el plan para la actualización del sistema, se deben realizar dichas actualizaciones de acuerdo al sistema operativo que se esté utilizando y a las herramientras proporcionadas por el mismo.

  La distribución de Linux Red Hat utiliza el sistema llamado *Red Hat Network* con el cual mantiene a sus usuarios enterados sobre la actualización de software. En el panel de tareas se encuentra un ícono indicando el estado de actualización de aplicaciones del sistema. Cuando el sistema requiere una actualización el ícono cambia de una flecha a un signo de admiración, como se muestra en la figura 2.2.

  Una vez que el administrador recibe el aviso (ver figura 2.3) de la existencia de una actualización utiliza la herramienta gráfica *up2date* encargada de conectarse a un

Figura 2.2: Indicador del estado de actualización del sistema.

servidor maestro para obtener las actualizaciones necesarias e instalarlas. La figura 2.4 muestra la herramienta *up2date*.

Figura 2.3: Herramienta de notificación de software disponible.

- *Después de realizar cualquier cambio a la configuración del sistema y la información del mismo, actualizar el sistema de monitoreo de archivos.* Es posible que después de la actualización de alguna aplicación se agreguen o modifiquen archivos de programas y de configuración, de tal manera que se deben actualizar las bases de datos para que las herramientas que monitorean la integridad del sistema de archivos también tomen en cuenta los archivos agregados y/o modificados.

  Existen herramientas como *Tripwire* que permiten identificar los cambios realizados en el sistema de archivos. Esta herramienta se abordará en el último capítulo.

OFRECER SÓLO LOS SERVICIOS NECESARIOS

Un servidor en red debe proporcionar sólo los servicios para los cuales está planeado. Tener mas servicios de los que debe proporcionar el equipo puede afectar la seguridad del sistema de diferentes maneras:

- Cada servicio proporciona una entrada potencial que puede ser explotada por un intruso.
- La administración se vuelve más compleja debido a que más personas deben administrar los servicios en el equipo.

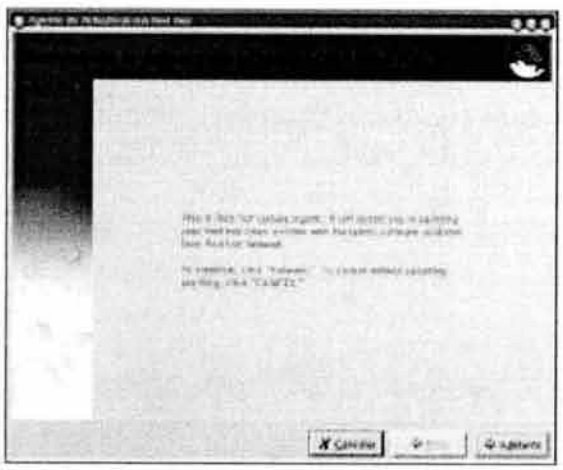

Figura 2.4: Herramienta *up2date* para la actualización de aplicaciones en Red Hat.

- Las bitácoras se vuelven más grandes debido a que cada servicio registra los eventos particulares y se vuelve más complejo el análisis de esta información.

Dado que la reducción de servicios a los mínimos necesarios ayuda en gran medida a mejorar la seguridad del servidor, a continuación se muestran algunos elementos a considerar en este tema.

- *Determinar las funciones del servidor.* Los servicios que debe dar el equipo dependen en gran medida de los objetivos y necesidades de la organización. Los servicios que proporcione un host pueden variar entre:
  - Servicios de impresión.
  - Servicios de almacenamiento remoto.
  - Servicios en Internet (correo electrónico, transferencia de archivos, web, etc.)

  En el equipo que hospeda el SIIEM/UAP se ha definido que los servicios necesarios para proporcionar el desarrollo, implementación y mantenimiento del sistema son:

  1. Web.
  2. Correo electrónico.
  3. Conexión remota.
  4. Base de datos.

- *Utilizar la forma más segura de proveer un servicio.* Una vez que se han determinado los servicios que debe proporcionar el equipo, es importante considerar las implementaciones existentes para proporcionar el servicio y determinar la manera más segura de proveerlo.

En el caso del servidor web se eligió **Apache** como implementación de este servicio debido a la popularidad que tiene actualmente. Cerca del 62 % de servidores web en el dominio **.com** a nivel mundial utilizan Apache, según estadísticas de SecuritySpace[11](ver figura 2.5). La aplicación como tal ha tenido problemas de seguridad que han sido corregidas durante el tiempo. El servicio, además del servidor Apache, debe ser asegurado utilizando otras medidas, como las que se están analizando durante este trabajo de Tesis.

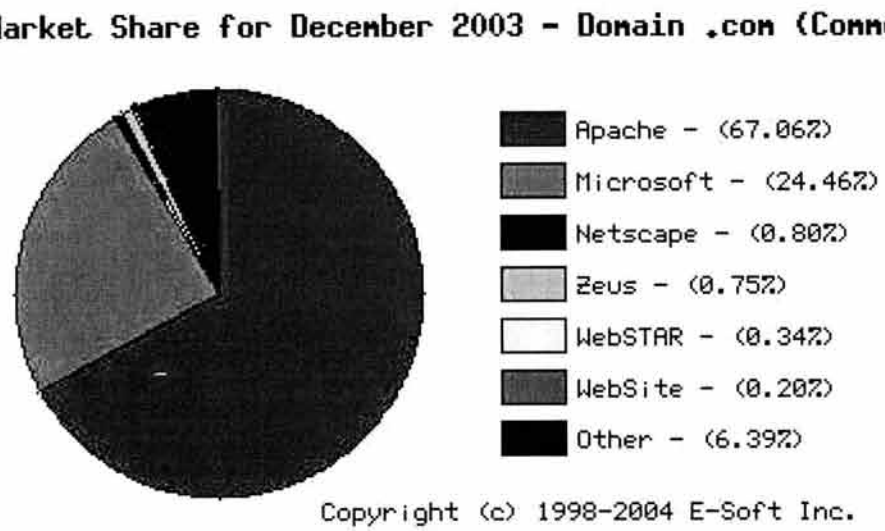

Figura 2.5: Estadística de uso de servidores web.

Como otro ejemplo, el servidor de correo electrónico que se estaba utilizando en el servidor que hospeda el SIIEM/UAP era **Sendmail**, sin embargo esta implementación se ha visto envuelta en serios problemas de seguridad y como muestra de ésto, en el mes de marzo del 2003 se publicaron 2 boletines de seguridad que comprometían el sistema a través de Sendmail. Estos boletines son el número 07 y el 12 que pueden ser consultados en: *http://www.unam-cert.unam.mx*. Debido a lo anteriormente expuesto, se optó por cambiar la implementación de correo electrónico por **Postfix**, un programa que está pensado en la seguridad del servicio de correo electrónico desde un inicio.

Por último, la conexión remota es fundamental en equipos que dan un servicio, ya sea para realizar mantenimiento del mismo o para el desarrollo de aplicaciones por parte del grupo encargado de ésto. La conexión remota se puede hacer utilizando diferentes mecanismos como **telnet** o **remote shell**, sin embargo estos mecanismo transfieren la información entre el cliente y el servidor en claro, de manera que

[11]Fuente: http://www.securityspace.com

cualquiera que intercepte la información en la red puede ver su contenido. Para resolver este problema se sustituyen estos servicios por **secure shell**, mediante el cual la información que es transferida entre el cliente y el servidor viaja cifrada a través de la red.

- *Una vez que se han determinado lo servicios del sistema, se debe asegurar que sólo sean esos servicios los que están instalados.* Para determinar ésto se utilizan herramientas para verificar los servicios en el equipo. Dos de estas herramientas son **netstat** y **nmap**.

  netstat. Es un programa del sistema operativo que determina el estado de la red del equipo. En seguida se muestra la salida de este comando para obtener sólo aquellos programas que dan un servicio.

```
# netstat -a|grep -w LISTEN
tcp        0      0 *:printer               *:*         LISTEN
tcp        0      0 *:x11                   *:*         LISTEN
tcp        0      0 *:http                  *:*         LISTEN
tcp        0      0 *:ssh                   *:*         LISTEN
tcp        0      0 *:postgres              *:*         LISTEN
tcp        0      0 *:smtp                  *:*         LISTEN
tcp        0      0 localhost:6010          *:*         LISTEN
```

  La salida anterior muestra los procesos en ejecución que proporcionan un servicio.

  nmap. Nmap es una herramienta libre que permite *escanear* los puertos abiertos que tiene un equipo. A continuación se muestra la salida de este comando aplicado al servidor donde está alojado el SIIEM/UAP.

```
# nmap localhost
Starting nmap V. 2.54BETA32 ( www.insecure.org/nmap/ )
Interesting ports on localhost (127.0.0.1):

Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
80/tcp     open        http
5432/tcp   open        postgres
6000/tcp   open        X11
Nmap run completed -- 1 IP address (1 host up)
```

  nmap reporta los puertos abiertos con su respectivo servicio.

- *Eliminar cualquier puerto abierto que sea innecesario.* Quitar del sistema puertos TCP o UDP no necesarios minimizan el riesgo de que estos puertos puedan ser utilizados como una puerta de entrada para un intruso.

  La forma de eliminar los puertos o servicios en el sistema dependerán de la forma en cómo son iniciados. En sistemas Linux, los servicios son iniciados de dos maneras: mediante los scripts de inicio o a través del super demonio de internet mejorado xinetd. Los servicios que son iniciados en el primer caso son manejados a través de una serie de programas que se encuentran en el directorio /etc/init.d. Por ejemplo, para detener el servicio de correo electrónico de postfix basta con ejecutar el siguiente comando: `/etc/init.d/postfix stop`. Para el caso de los servicios que son levantados mediante xinetd, se deben verificar los archivos que se encuentran ubicados en el directorio /etc/xinetd.d. Mediante la bandera *disable* se indica si debe o no ser iniciado el servicio cada vez que se hace una petición al mismo. En el SIIEM/UAP todos los servicios son controlados mediante los scripts de inicio y no por xinetd.

MANTENER MECANISMOS DE AUTENTICACIÓN DE USUARIOS

Otra tarea importante dentro del aseguramiento del equipo es definir e implementar los medios necesarios para validar la autenticidad de los usuarios del sistema. Además de que en el primer capítulo ya se definieron las políticas para el control de acceso físico y lógico en el SIIEM, a continuación se explican más detalladamente las consideraciones que se deben tener en cuenta al respecto.

- *Configurar el sistema para utilizar mecanismos de control de acceso mediante software o hardware.* En la actualidad, las computadoras permiten definir una contraseña en el BIOS, ésto con el fin de que un usuario no autorizado no obtenga acceso en caso de reiniciar el equipo. Sin embargo esta forma de asegurar el inicio de una computadora presenta un inconveniente cuando el equipo da servicio las 24 horas y se presenta una falla de energía eléctrica. Si se da esta situación y no hay nadie quien pueda proporcionar la contraseña del BIOS, el equipo estará fuera de servicio durante ese tiempo. Este mecanismo puede ser sustituido si se define correctamente el control de acceso físico, como se verá mas adelante.

- *Eliminar cuentas predeterminadas de usuarios y grupos innecesarios.* Existen sistemas operativos (y también otro tipo de software como bases de datos) que proporcionan cuentas predefinidas, sin contraseña o con contraseña predeterminadas. Los administradores de sistemas deben deshabilitar estas cuentas o modificar la contraseña para que no sea un punto vulnerable dentro del sistema.

- *Deshabilitar cuentas no interactivas.* Existen organizaciones que proporcionan a sus integrantes cuentas de correo, sin embargo la mayoría de ellos no requieren de un shell o programa interactivo en el sistema. Para los usuarios que no requieren interacción directa con el sistema debe ser eliminado el shell, en caso de Unix se puede

definir /bin/false para que no les permita acceder al sistema y ejecutar comando alguno. De esta manera, si por alguna razón es robada la contraseña de algún usuario, se reduce el riesgo de que un intruso ingrese al sistema y pueda ejecutar algún programa no autorizado.

- *Crear grupos y cuentas de usuarios particulares.* En sistemas multiusuario existen conjuntos de personas que realizan actividades similares de acuerdo al departamento en que laboran. En esta situación es recomendable que se definan grupos en el sistema de acuerdo a la similitud de funciones que tienen los usuarios, para ello se utilizan grupos como el de *contabilidad, administración, desarrollo,* etc. En el SIIEM/UAP se creó un grupo de trabajo llamado **desarrollo** al cual pertenecen las personas que pueden modificar archivos del sitio web.

- *Definir políticas de contraseñas.* La contraseña es uno de los eslabones más importantes de seguridad en un sistema debido a que es el mecanismo mediante el cual se le proporciona el acceso a un usuario. En un sistema que utiliza este mecanismo de autenticación deben estar definidas las políticas para las características que debe tener una contraseña fuerte.

  Una de las técnicas más utilizadas por los intrusos es el *ataque de fuerza bruta* que consiste en encontrar una coincidencia para una contraseña débil. De tal manera que la definición de contraseñas es un elemento importante. Las políticas para contraseñas deben incluir:

  - Longitud. Entre mayor sea el tamaño de una contraseña será mas difícil obtenerla.
  - Complejidad. Una contraseña debe ser lo más difícil posible, para ello se pueden utilizar caracteres alfanuméricos, caracteres especiales, etc.
  - Tiempo de duración. Se debe indicar con que regularidad debe cambiar una contraseña, se recomienda que por lo menos sea cada mes.
  - Autoridad. También se debe definir quién es el autorizado para cambiar la contraseña de un usuario.
  - Responsabilidad. Indicar a los usuarios la responsabilidad que tiene el uso de contraseñas adecuadas para el ingreso al sistema.

CONFIGURACIÓN ADECUADA DE PERMISOS DE OBJETOS

Los permisos en los objetos como archivos, directorios y dispositivos son un mecanismo muy utilizado para restringir el acceso de los usuarios y/o procesos. En seguida se explican algunas consideraciones al respecto.

- *Identificar la protección necesaria para archivos y directorios en la computadora.* Esto consiste en definir los permisos adecuados para los diferentes objetos dentro del sistema. La matriz que propongo en el SIEM/UAP es útil para definir la correlación

Objeto/Usuario	Normal	Desarrollador	Admin del sistema	DBA	Admin de Web
Archivos de usuarios	rwx	-	rwx	-	-
Aplicaciones	rx	rx	rwx	rx	rx
Servicios	-	-	rwx	-	-
Bitácoras	-	-	rwx	-	-
Servidor Web	-	-	rwx	-	-
Páginas Web	-	rw	-	-	rwx
Servidor BD	-	-	-	rwx	-
Servidor SSH	-	-	rwx	-	-
Servidor de Impresión	-	-	rwx	-	-

Cuadro 2.1: Matriz de definición de permisos de objetos y usuarios.

entre permisos (lectura **r**, escritura **w** y ejecución **x**), usuarios y objetos, como se muestra en la tabla 2.1.

- *Configurar control de acceso.* La configuración del control de acceso debe realizarse en base a la matriz de acceso definida previamente. Los programas binarios del sistema no deben tener el permiso de escritura a excepción del administrador para prevenir que cualquier usuario pueda modificarlos como se muestra en la tabla 2.2.

Dueño	Programa	Permisos
root	sleep	-rwxr-xr-x
root	sort	-rwxr-xr-x
root	stty	-rwxr-xr-x
root	sync	-rwxr-xr-x
root	tar	-rwxr-xr-x
root	tcsh	-rwxr-xr-x
root	touch	-rwxr-xr-x
root	true	-rwxr-xr-x
root	uname	-rwxr-xr-x
root	unicode_start	-rwxr-xr-x
root	unicode_stop	-rwxr-xr-x
root	unlink	-rwxr-xr-x
root	usleep	-rwxr-xr-x
root	vi	-rwxr-xr-x

Cuadro 2.2: Definición de permisos de archivos binarios en el sistema

IDENTIFICAR Y HABILITAR LOS MECANISMOS DE REGISTRO DEL SISTEMA Y RED

En el sistema se llevan registros de usuarios, aplicaciones y una gran variedad de eventos que pueden ayudar a mejorar la seguridad del sistema y permiten identificar una posible intrusión. Por tal motivo se deben identificar los mecanismos que utiliza cada sistema específico. Para llevar a cabo ésto es necesario tener en cuenta lo siguiente :

- *Identificar la información que debe ser registrada.* Identificar:
  - Tipo de información que se puede registrar.
  - Mecanismos utilizados para registrar.
  - Lugar donde los archivos de registro son almacenados.

  La tabla 2.3[12] muestra las categorías y tipos de registros de información más comunes. Se debe considerar que no todas estas categorías son tomadas en cuenta por todos los sistemas.

Categoría de registro	Tipo de información registrada
Usuarios	- Información de entrada y salida de usuarios: fecha y hora, intentos fallidos.
Procesos	- Ejecución de procesos de usuarios. - Tiempo de inicio de procesos. - Estado al finalizar el proceso, tiempo y recursos consumidos.
Sistemas	- Acciones solicitando privilegios especiales. - Estado y errores reportados por subsistemas de hardware o software. - Cambios en el estado del sistema, incluyendo reinicio y baja.
Red	- Solicitud de inicio de servicios. - Nombre del usuario y/o equipo que solicita el servicio. - Tráfico de mensajes de red. - Nuevas conexiones. - Duración de las conexiones. - Flujo de la conexión.
Sistema de Archivos	- Cambios en la lista de control de acceso y protección de archivos. - Acceso a archivos (apertura, creación, borrado y ejecución).
Aplicaciones	- Información específica de aplicaciones y servicios (por ejemplo: registros de correo, registros del servidor web, registro de firewall, etc.)

Cuadro 2.3: Categorías de registros de información en un servidor

[12] Allen Julia, Securing network servers, CERT

- *Determinar si los mecanismos de registro del sistema y red son suficientes.* Analizar si la información registrada por el sistema es suficiente, si no es así se deberán utilizar otras herramientas que permitan identificar la información que sea necesaria. Por ejemplo, un problema común de seguridad es el escaneo de puertos abiertos, eventos que el sistema operativo generalmente no registra. Para detectar escaneos de puertos hacia el equipo se agrega al sistema la herramienta Portsentry[13].

- *Habilitar el registro.* Utilizar el sistema de registro que provee el fabricante del sistema y cualquier otra herramienta para complementar esta tarea.

  En sistemas Linux existe el registro de sistema conocido como **syslog**. El archivo de configuración de syslog es /etc/syslog.conf en el cual se definen los eventos que debe registrar y el archivo donde debe ser almacenada dicha información. A continuación se ejemplifican unas líneas de este archivo de configuración:

```
# Log anything (except mail) of level info or higher.
# Don't log private authentication messages!
*.info;mail.none;authpriv.none;cron.none    /var/log/messages

# The authpriv file has restricted access.
authpriv.*                                          /var/log/secure

# Log all the mail messages in one place.
mail.*                                              /var/log/maillog
```

  Los eventos que se relacionan con el correo electrónico son almacenados en el archivo */var/log/maillog*, los accesos restringidos se almacenan en */var/log/secure* y los eventos comunes serán almacenados en el archivo */var/log/messages*.

- *Proteger los registros para asegurarse que son confiables.* Debido a la importancia de la información que guardan las bitácoras, se debe asegurar que no deben ser leídas, accedidas y mucho menos modificadas por cualquier usuario, excepto por usuarios autorizados. Se pueden utilizar los siguientes métodos para proteger estos archivos:

  - Dedicar un equipo dentro de la red para almacenar los registros.
  - Mantener esta información en medios de almacenamiento como CD-ROM o CD-RW.
  - Si es posible, habilitar el sistema para que los registros sólo puedan ser agregados, así los registros existentes no podrán ser modificados.
  - Cifrar los archivos de registros, especialmente los que contienen información sensitiva.

[13] Más adelante se abundará en esta herramienta.

Debido a que los archivos de registro pueden llegar a contener gran cantidad de información es necesario considerar el uso de herramientas que permitan analizar esta información de manera más rápida y eficiente. En el SIIEM, como se verá en el cuarto capítulo, se utiliza la herramienta LogWatch para analizar las bitácoras del sistema.

CONFIGURAR LA COMPUTADORA PARA LA REALIZACIÓN DE RESPALDOS

Los respaldos de información permiten la *integridad* y *disponibilidad* de la información. Sin un respaldo, es imposible restaurar la información en caso de que exista un problema en el sistema que haya alterado los datos almacenados en el equipo. Para ésto, es necesario cubrir una serie de elementos que se muestran a continuación.

- *Desarrollar un plan de respaldo y recuperación de la información.* Determinar los lineamientos necesarios para el respaldo de información después de realizar un análisis costo-beneficio entre la velocidad de respaldo, la cantidad de información que debe ser almacenada y los recursos técnicos y humanos requeridos para llevar a cabo esta actividad. El respaldo de información puede hacerse de dos maneras:

  - Realizar el respaldo en el equipo que contiene la información original. Esto tiene como ventaja que la información no necesita ser "trasladada" hacia otro lugar y no se pone en riesgo durante el trayecto. La principal desventaja de este método es que el equipo necesita más recursos para poder almacenar los respaldos.
  - Otro método es centralizar los respaldos en un equipo destinado para ello. La desventaja de ésto es que si no se utiliza un canal seguro para la transferencia del respaldo entre la máquina cliente y el servidor, la información respaldada puede ser interceptada en el trayecto.

  En el SIIEM/UAP se desarrollaron políticas de respaldos para la base de datos en el Capítulo 1 *Seguridad en Cómputo*, en el cuál se consideran los elementos necesarios que debe cubrir el respaldo de esta información.

- *Instalar herramientas de respaldo.* Una vez determinado el plan de respaldo, se deben instalar las herramientas para poder efectuarlos. Los sistemas Linux proporcionan herramientas como **dump**, **cpio**, **tar**, siendo la última la más utilizada.

- *Calendarizar los respaldos.* Las herramientas deben estar configuradas para que reflejen el plan de respaldos y recuperación de información. En los sistemas Linux, existen programas como **cron** y **at** que permiten calendarizar actividades en determinadas fechas.

- *Verificar que los respaldos se están llevando a cabo en los tiempos definidos.* Una vez configuradas las herramientas para la realización de respaldos, se les debe dar seguimiento a los mismos y verificar que se están realizando en tiempo y forma de acuerdo al plan.

- *Probar la recuperación de la información.* Es frecuente que se dé por hecho que la planeación y realización de respaldos son efectivos, sin embargo, también se debe considerar un prueba de recuperación de información, simulando un escenario de pérdida de datos y verificar que la información obtenida a partir del respaldo es la esperada.

IMPLEMENTAR MEDIDAS DE SEGURIDAD FÍSICA

Hasta el momento se han analizado prácticas de seguridad que tienen que ver con el sistema operativo y las aplicaciones, sin embargo, también se debe considerar la seguridad física en el equipo, ya que finalmente las computadoras son aparatos que están expuestos al desgaste y factores ambientales como el polvo, humedad, calor, etcétera.

En esta parte se consideran dos cosas importantes: definir los medios necesarios para que la computadora funcione en instalaciones físicas adecuadas y restringir el control de acceso físico.

En el primer caso se propone que se mantengan las condiciones ambientales adecuadas de temperatura y humedad para que los equipos de cómputo funcionen correctamente. También se determinan los medios necesarios para que no se vea interrumpido el suministro de energía eléctrica. Dado que la UAP no cuenta con una planta auxiliar de energía eléctrica, cada equipo de misión crítica, como es el caso del SIIEM, cuenta con no-brakes en caso de que se sufra una falla de este tipo. Además se cuenta con cableado estructurado que evita que un cable de red pueda ser dañado y afecte la disponibilidad del servicio.

En el segundo caso, si una persona no autorizada tiene acceso físico a un equipo de cómputo, éste puede estar en grave riesgo. Para asegurar la integridad y confidencialidad de la información, el acceso físico es un elemento importante, las herramientas de software no ayudan a prevenir el robo físico de una computadora, o de un disco duro, y mucho menos de la instalación de hardware no autorizado. Considérense los siguientes elementos:

- *Prevenir la instalación de hardware no autorizado y la modificación de hardware autorizado.* La instalación de nuevo hardware puede poner en riesgo un sistema en diferentes formas:

  - La instalación de un módem, por ejemplo, permite la conexión directa a la computadora desde la red telefónica, haciendo posible que alguien se conecte desde cualquier lugar evadiendo las medidas de seguridad en el perímetro.
  - La instalación de un medio de almacenamiento removible o impresora permitirá la copia de información de manera sencilla.

  En la medida de lo posible es recomendable asegurar el equipo con llave y tener una de repuesto en un lugar fuera del área de trabajo.

- *Ubicar la computadora en un ambiente seguro.* Ubicar la computadora en un lugar seguro previene el acceso no autorizado, el robo o destrucción de la información. Es recomendable ubicar el equipo en un lugar vigilado con cámaras de video y con mecanismos de registro para los usuarios. No es conveniente que una servidor se encuentre en la oficina de un empleado, sino en un lugar destinado para ese propósito.

No es posible acceder en el lugar donde está ubicado el equipo que aloja al SIIEM/UAP, y otros servidores más, a menos que sea una persona que labora en el área de informática de la Unidad de Posgrado, sin embargo, falta mucho por hacer en este rubro debido a que existen tiempos "muertos", generalmente en la hora de la comida, cuando no hay nadie en el lugar de servidores y es factible que alguien pueda acceder, no obstante, las políticas que he propuesto con anterioridad ayudarán a reducir este riesgo.

2.2.2. Asegurar el servidor Web

En los últimos años, un "servidor web seguro" ha significado cosas diferentes para personas distintas:

- Para los vendedores de software, un servidor web seguro es un programa que implementa ciertos protocolos criptográficos, de tal manera que la información transferida entre un servidor web y un navegador de web no pueda ser legible para nadie más.

- Para los usuarios, un servidor web seguro es aquel que protegerá cualquier información personal que reciba o recopile. Es un servidor que garantiza la privacidad de los usuarios y no provocará que el navegador descargue algún virus o cualquier otro programa malicioso en la computadora del usuario.

- Para la organización que proporciona el servicio de web, un servidor seguro es aquel que resiste a determinados ataques desde fuera (Internet) o desde dentro de la organización (empleados mal intencionados).

De cierta manera, un servidor web seguro es todo lo anterior y más. Un servidor web se puede considerar seguro cuando cuenta con las siguientes características:

- Confiabilidad. Se puede fiar en que el servidor se comportará como se espera que lo haga.

- Respaldado. Cuenta con los mecanismos necesarios para que pueda ser restituido rápidamente en el caso de fallas de hardware o software.

- Expandible. Es posible adaptar los servicios a grandes cantidades de tráfico.

Sin embargo, cuando los vendedores usan la frase "servidor web seguro", siempre hacen referencia a un servidor web que implementa el protocolo criptográfico SSL. Estos protocolos permiten que los navegadores y los servidores web intercambien información

sin el riego de ser descubierta por otras entidades con acceso a los mensajes que se encuentran transitando a través de la red. El cifrado es recomendado ampliamente como un prerrequisito para el comercio en Internet.

Si bien es cierto que los protocolos criptográficos son útiles para proteger la información que es enviada a través de Internet de ser vista por otros entes, estos protocolos no son estrictamente necesarios para la seguridad en web o no son suficientes para asegurarla. Muchos de los problemas más dramáticos de seguridad en cómputo de los últimos años han tenido que ver con servidores web que implementan protocolos criptográficos dado que los intrusos simplemente robaron los números de tarjetas de crédito después de que han sido descifrados por el servidor web y almacenados en la base de datos relacional.

Una vez que se han determinado los elementos necesarios para asegurar el equipo que hospeda un servicio como el de páginas web, lo siguiente es asegurar el servidor web como tal. Esto puede ser realizado a través de diferentes elementos que juntos harán de un servidor web común, un servidor web seguro. Para lograr esto se debe tomar en cuenta la configuración del servidor web y la forma de mantener la integridad del servidor. Este proceso se compone de distintas tareas como son:

1. Aislar, en la medida de lo posible, el servidor de redes públicas.
2. Configurar los objetos del servidor web a través del control de acceso.
3. Identificar y habilitar sistemas de registro en el servidor web específico.
4. Considerar implicaciones de seguridad antes de elegir scripts y plug-ins.
5. Configurar el servidor web para utilizar mecanismos de autenticación y cifrado.
6. Mantener una copia de páginas, scripts y objetos del servidor web en un host seguro.
7. Proteger el servidor web contra ataques comúnes.

En seguida se describen detalladamente cada una de estas actividades.

Aislar el servidor de redes públicas

Es recomendable ubicar el servidor Web en una subred que esté apartada de la red pública. De esta manera un intruso no podrá monitorear el tráfico que llega al mismo y es más fácil controlar la información que llega al servidor Web, ésto puede ser logrado a través de las siguientes medidas:

- *Ubicar el servidor web en una red aislada de la red pública.* Mediante un *firewall* o *ruteador* para protegei el acceso a la red y detectar intentos de intrusión.
- *Uso de firewall para restringir el tráfico entre la red pública y la red local.* El uso de un firewall efectivamente restringe el tráfico de información entre la red pública como

Internet y la red local. En la Unidad de Administración de Posgrado está instalado un firewall, implementado en el sistema operativo *OpenBSD*, que tiene como fin aceptar conexiones de servicios públicos como el correo electrónico y web, además define una red interna con direcciones IP no homologadas[14].

El firewall puede consistir de una computadora o conjuntos de computadoras para filtrar el tráfico al servidor web, la red local y otros servidores de los cuales dependa el servicio web como puede ser un servidor de base de datos.

Como se vió en la sección anterior, en el equipo que hospeda el SIIEM/UAP se encuentran otros servicios como correo electrónico, en este caso no es necesario un firewall para los servicios pero si para dividir la red local de la pública.

CONFIGURAR LOS OBJETOS DEL SERVIDOR WEB A TRAVÉS DEL CONTROL DE ACCESO

La configuración apropiada de objetos puede ayudar a prevenir la revelación de información sensitiva o restringida que no debe ser pública. Además, los controles de acceso pueden ser usados para limitar la cantidad de recursos utilizados para prevenir un ataque de negación de servicio. Para llevar a cabo la configuración de objetos considérense los siguientes elementos.

- *Configurar el servidor web para ser ejecutado por un usuario y grupo.* Crear un nuevo usuario y grupo para ejecutar el software del servidor web es una buena práctica. Aunque en Sistemas Linux, es necesario que primero sea ejecutado por el usuario *root*, quien es el administrador del equipo, las peticiones que se hacen hacia el servidor web se ralizan mediante un usuario sin privilegios. El servidor web en el SIIEM/UAP está configurado para que el usuario *nobody* realice las peticiones al servidor web como se muestra en el siguiente fragmento del archivo de configuración:

```
# If you wish httpd to run as a different user or group,
you must run httpd as root initially and it will switch.

User nobody
Group nobody
```

- *Identificar la protección necesaria para archivos, directorios y objetos del servidor Web.* La matriz de accesos definida en el aseguramiento del equipo ayuda a identificar los permisos que los usuarios tienen sobre los objetos del servidor web. La parte de la matriz de acceso que propongo para el SIIEM/UAP relacionada exclusivamente a los objetos en el servidor web se muestra en la tabla 2.4 donde r es igual al permiso de lectura, w es equivalente a permiso de escritura y x al permiso de ejecución.

[14]Direcciones del protocolo IP utilizadas en redes locales que no son ruteables a Internet.

Objeto/Usuario	Desarrollador	Admin. del sistema	Admin. de Web	nobody
Servidor web	-	rwx	-	-
Páginas web	rwx	-	rwx	r
Programas CGI	rwx	-	rwx	r-x

Cuadro 2.4: Matriz de control de acceso en el servidor web.

La tabla 2.5 muestra un ejemplo de objetos en el servidor web en el SIIEM/UAP con el usuario y permisos correspondientes.

Objeto	Dueño del objeto	Grupo	Permisos	Descripción del objeto
httpd	root	root	rwxr-xr-x	Programa del servidor web
/home/httpd	httpd	desarrollo	rwxr-xr-x	Directorio donde se alojan las páginas web y CGI's.
index.html	httpd	desarrollo	rwxrwxr-x	Página de inicio del SIIEM/UAP
logo_uap.gif	httpd	desarrollo	rwxrwxr-x	Imagen
Entrada_jefes.pl	httpd	desarrollo	rwxrwxr-x	Programa CGI
Altas.pm	httpd	desarrollo	rwxrwxr-x	Módulo CGI

Cuadro 2.5: Características de objetos en el servidor web.

- *Limitar el uso de recursos del servidor Web.* Limitar recursos como memoria, procesos y espacio en disco duro ayudarán a mitigar el efecto de un posible ataque de negación de servicios, los cuales intentan llenar el disco duro o la memoria disponible para que el equipo no sea capaz de continuar con sus actividades normales. En el SIIEM/UAP, los recursos que puede utilizar el servidor web están limitados de la siguiente manera:
  - Procesos. Se limitan 150 conexiones remotas. Este límite está definido en el archivo de configuración del servidor web.
  - Espacio en disco. Propongo una partición de disco de 150 MB en donde se ubican las páginas web y programas CGI. Debido a que estos archivos son muy pequeños, no es necesario asignar mas espacio que el propuesto ya que estaría desperdiciado.

IDENTIFICAR Y HABILITAR SISTEMAS DE REGISTRO EN EL SERVIDOR WEB ESPECÍFICO

La recolección de información generada por el servidor web puede ser de gran ayuda para detectar signos de intrusión y analizar problemas de seguridad.

Aunque el sistema operativo proporciona sistemas de registro, como se vió en la sección anterior, la información recolectada puede no ser suficiente para el desempeño del

servidor web como pueden ser las páginas solicitadas o los intentos de ataques al servidor en búsqueda de vulnerabilidades conocidas.

Frecuentemente, los archivos de registros son la única fuente para detectar comportamientos sospechosos. Errores al habilitar estos registros pueden ser una debilidad para identificar posibles problemas de intrusión. Se pueden necesitar los registros del servidor web para:

- Alertar sobre actividad sospechosa que requiera investigación posterior.
- Determinar la actividad de un intruso.
- Ayudar a recupear el sistema.
- Ayudar a conducir una investigación.

Considérense los siguiente elementos para identificar y habilitar el sistema de registro en el servidor web.

- *Identificar la información registrada por el servidor web.* Pueden existir diferentes registros:
  - Registros de transferencia. Cada transferencia es registrada incluyendo información como: equipo remoto, usuario remoto, fecha, *URL* solicitado, estado de la solicitud y bytes transferidos. En el SIIEM/UAP estos registros son almacenados en el archivo *access_log*.
  - Registros de error. Cada error es representado en un registro. En el SIIEM/UAP se almacena esta información en el archivo *error_log*.

  Debido a que estos archivos de registro pueden llegar a ser muy grandes, propongo que en el SIIEM/UAP se utilicen herramientas para facilitar el análisis de esta información. En el último capítulo expondré la herramienta Webalizer para realizar esta tarea de análisis de forma más sencilla con la ventaja que esta aplicación proporciona al presentar los resultados de forma gráfica.

CONSIDERAR IMPLICACIONES DE SEGURIDAD ANTES DE ELEGIR SCRIPTS Y PLUG-INS.

En su forma básica el servidor web espera por solicitudes y responde al transmitir el archivo solicitado. El servidor web, además puede ejecutar programas o procesos para producir información personalizada en respuesta a una solicitud. Ejemplos de estos mecanismos incluyen *CGI* (Common Gatway Interfaz), *scripts* y *plug-ins*.

CGI es un estándar aceptado que agrega una extensión al servidor web mediante la ejecución de programas que son invocados por solicitar un URL específico. Los programas CGI o scripts, como se verá abajo, se ejecutan como una tarea adicional del servidor web. La salida de un CGI es entregada de vuelta al navegador web que hizo la solicitud. Los CGI pueden ser utilizados como una interfaz para motores de búsqueda y bases de

datos, crear páginas web dinámicamente y responder a entradas de datos proporcionados por el usuario.

Un script es un programa ejecutado por un intérprete escrito en lenguajes como Perl, JavaScript, Shell, Phyton y VBScript.

Un plug-in funciona como la extensión de otro programa. Los plug-ins proporcionan funcionalidad adicional.

Considerar la seguridad en los elementos anteriormente descritos es importante debido a que vulnerabilidades de seguridad pueden ser introducidas en la adquisición, instalación o ejecución de un script o plug-in. Las vulnerabilidades pueden incluir programas escritos póbremente, por ejemplo, muchos ataques exitosos sobre servidores web han explotado vulnerabilidades conocidas en scripts CGI disponibles comúnmente.

Existen diversos puntos a considerar en la elección de programas externos como se verá en seguida, aunque es importante considerar que los scripts que son ejecutados en el SIIEM/UAP fueron programados por el grupo de desarrollo de la unidad de cómputo y no existen programas obtenidos de fuentes externas.

- *Realizar un análisis costo beneficio.* Considerar si la funcionalidad obtenida por el programa externo es escencial para el funcionamiento del servidor web o si puede ser omitido.
- *Seleccionar programas, scripts y plug-ins de fuentes confiables.* Evitar programas externos producidos por autores desconocidos o descargados de Internet de un sitio poco confiable.
- *Entender completamente el funcionamiento del programa.* Determinar que tan complejo es el programa, si tiene interacción con otros programas, si escribe o lee información en alguna parte del sistema, si es necesario ejecutarlo con privilegios de administración. Esto es de gran utilidad para identificar si el programa no está abriendo una puerta en el sistema que pueda ser utilizada por un intruso.
- *Revisar información pública acerca de vulnerabilidades conocidas en los programas externos.* Muchos organismos dedicados a la investigación de vulnerabilidades en productos de software proporcionan información pública sobre problemas de seguridad. Cuando se elija un programa externo, es conveniente que se revisen estas fuentes de información para indagar si existe o ha existido un problema de seguridad en el programa que se pretende instalar.

CONFIGURAR EL SERVIDOR WEB PARA UTILIZAR MECANISMOS DE AUTENTICACIÓN Y CIFRADO

El servidor web posiblemente necesite soportar un rango de tecnologías para identificar y autentificar usuarios con diferentes privilegios de acceso a la información. Algunas

de estas tecnologías son basadas en funciones criptográficas que proporcionan un canal cifrado entre el servidor web que soporta cifrado y el cliente. Tecnologías que realizan el cifrado incluyen SSL (Secure Socket Layer), S/HTTP (Secure Hypertext Transport Protocol) y SET (Secure Electronic Transaction).

Sin autenticación fuerte no es posible restringir el acceso a información específica para usuarios autorizados. Además, sin autenticación fuerte, los usuarios del servidor web no son capaces de determinar si están contectados al servidor web auténtico o a una versión falsa ejecutada por un intruso.

El cifrado puede ser usado para proteger la información entre el cliente y el servidor web. Sin cifrado, cualquier persona con acceso al tráfico de la red puede determinar y posiblemente alterar, el contenido de información sensitiva o restringida. Ésto puede violar la confidencialiad y la integridad de información crítica.

La autentificación es importante tanto para el cliente como para el servidor de maneras diferentes. El usuario externo (cliente) debe asegurarse de que:

- Se está comunicando con el servidor correcto.
- Lo que el usuario envía es entregado sin modificación al servidor.
- Que es posible comprobar que el usuario envió el mensaje.
- El mensaje sólo puede ser leído por el servidor adecuado.
- Que la entrega está garantizada.

Por otra parte, el servidor web debe asegurarse de que:

- Se está comunicando con el cliente correcto.
- El contenido del mensaje es adecuado.
- La identidad del autor del mensaje no es errónea.

Debido a la gran importancia que tiene el uso de mecanismos de autenticación y cifrado en un servidor web, este tema se abordará con mayor detalle adelante en este capítulo de forma que quede claro el uso de estas técnicas y la forma en cómo fueron utilizadas para proporcionar parte de la confidencialidad e integridad en el SIIEM/UAP.

MANTENER UNA COPIA DE PÁGINAS, SCRIPTS Y OBJETOS DEL SERVIDOR WEB EN UN HOST SEGURO

Si la seguridad del servidor web es comprometida, con una copia de respaldo será posible recobrar el contenido del servidor web. Generalmente un intruso modifica la información contenida en el servidor web y es muy poco probable que altere el respaldo ubicado en un lugar seguro. ¿Cómo se puede hacer ésto?, a continuación se explican algunos elementos a considerar y las medidas tomadas en el SIIEM/UAP.

- *Definir un plan de respaldo.* Como se vió anteriormente, es importante definir un plan de respaldo del contenido del servidor web. Pueden darse dos situaciones distintas en el servidor web que pueden influir en el plan de respaldo:

  - Sistema en desarrollo. Cuando el sistema se encuentra en constante cambio, es conveniente hacer los respaldos frecuentemente, así, cuando sea necesario la recuperación de información, ésta sea lo más actualizada posible.
  - Sistema en producción. Aunque un sistema siempre está en desarrollo, existe un punto en el cual el sistema es liberado y el trabajo consiste en darle mantenimiento de tal manera que los cambios son menores a las páginas o programas que dan soporte al sitio web. En esta situación es recomendable realizar el respaldo cada vez que se haga una modificación considerable al sistema.

  El SIIEM/UAP se encuentra en la segunda situación de tal manera que se realiza un respaldo cada vez que existe un cambio importante en alguna página o programa. El último cambio que se realizó en el sistema fue a finales de febrero del año 2004 por lo que el archivo de respaldo está fechado con el 27 de febrero del 2004.

- *Utilizar mecanismos de cifrado para transferir la copia de seguridad.* Se debe configurar el servidor web que contiene la información del sitio, así como el servidor que aloja la copia de seguridad para transferir la información de manera segura entre ellos. En el SIIEM/UAP propongo que se utilice *secure copy* para la transferencia de información del respaldo de tal forma que la información se envía por un canal cifrado.

PROTEGER EL SERVIDOR WEB CONTRA ATAQUES COMÚNES

Un servidor web público generalmente es utilizado por las organizaciones para proporcionar servicios en línea o simplemente proporcionar información de cualquier índole. Sin embargo, generalmente este servicio es objeto de ataques debido a que un intruso puede tener como objetivo obtener acceso al servidor, cambiar el contenido del sitio, negar el acceso a los usuarios del servidor o poner en rídiculo a la organización propietaria del servidor web.

Para proteger el servidor contra ataques comúnes en el SIIEM/UAP se llevaron a cabo la siguientes actividades.

- *Revisión regular de información pública acerca de nuevas vulnerabilidades.* Como se mencionó antes, es muy importante la actualización de información de seguridad, debido a que cada día surgen nuevas vulnerabilidades que pueden ser utilizadas por los intrusos para afectar un equipo.

- *Trabajar conjuntamente con el proveedor de Internet y su equipo de seguridad para evitar ataques de negación de servicios distribuidos.* En muchas ocasiones, los intrusos utilizan un servidor web comprometido como medio de ataque hacia equipos de otras organizaciones. El equipo desarrollador del SIIEM/UAP está en constante comunicación con el equipo de respuesta a incidentes UNAM-CERT.

Para verificar las vulnerabilidades a las cuales puede estar expuesto el servidor web, propongo utilizar la herramienta nessus, la cual es útil en la búsqueda de vulnerabilidades conocidas en servicios de red comúnes. Esta herramienta se explica de forma detallada en el último capítulo de esta tesis.

2.2.3. Asegurar la información en tránsito

Gran parte del énfasis inicial en el campo de la seguridad en web tiene que ver con el problema de proteger la información y cómo ésta viaja sobre Internet desde un servidor web a la computadora del usuario final. Es una preocupación que alguien pueda "escuchar" sobre la red (a través de nodos intermediarios), copiar información sensitiva o alterar información en el tránsito de la misma.

Existen muchas formas de proteger la información de ser "escuchada" mientras ésta viaja a través de la red:

- Aseguramiento físico de la red. De manera que no pueda ser interceptada la información.

- Ocultamiento de la información. De tal forma que la información vista sea irrelevante.

- Cifrado de la información. De esta forma la información no podrá ser descifrada por nadie más que no tenga en su posesión la llave adecuada.

Debido a que el aseguramiento físico de Internet es imposible, de las técnicas anteriores, el cifrado es la única técnica práctica en redes públicas a gran escala. Además el cifrado puede prevenir la alteración de la información, o por lo menos hacer esta situación obvia cuando la información haya sido modificada.

Uno de los eventos más importantes en el lanzamiento del World Wide Web fue el desarrollo, por parte de Netscape Communicator, de un sistema fácil de usar para enviar información cifrada en Internet. Llamado el Secure Socket Layer (SSL[15]), este sistema hace posible a los usuarios poco sofisticados emplear seguridad criptográfica similar a la que

[15]Mas adelante se abordará SSL en este capítulo.

ha sido reservada para organizaciones bancarias y del gobierno. El cifrado que provee SSL hace posible que la gente que transmite números de tarjeta de crédito lo haga de manera segura sobre Internet utilizando el Web, lo cual se consideraba un requisito previo para el comercio electrónico.

SSL no es la panacea, es sólo un componente de la seguridad en web. SSL hace posible enviar nombres de usuarios, contraseñas, números de tarjeta de crédito de manera segura, pero SSL no proporciona protección para la información en los dos puntos de la conexión.

Un efecto adverso en el tránsito de la información es el ataque de negación de servicios (DoS - *Denial of Service*), el cual resulta en una interrupción de la red. Un ataque de negación de servicios puede resultar de un evento físico, tal como un corte de fibra óptica, o un evento lógico, como un error en las tablas de ruteo en Internet o un ataque de bombardeo al servicio con gran cantidad de solicitudes hasta el grado que el servidor se satura y deja de recibir petición alguna.

En la actualidad, no hay forma práctica para que un individuo pueda defenderse contra los ataques de negación de servicios, aunque la seguridad multinivel, conexiones de alta capacidad y sistemas de respaldo pueden ayudar a minimizar el impacto.

Una vez que se han visto las medidas necesarias para asegurar el equipo de cómputo y el servicio web, veremos otro eslabón en el proceso de aseguramiento que es la computadora del usuario final.

2.2.4. Asegurar la computadora del usuario

En los primeros cinco años de la existencia del Web, la seguridad en esta área fue propiamente un ejercicio académico. Mientras compañías como Netscape, Microsoft y Macromedia distribuían software de navegadores de web, los investigadores en universidades como Berkeley y Princeton encontraron errores en estos programas. Cada nueva vulnerabilidad en un navegador generaba una historia en los periódicos alertando de cómo el error podría ser explotado por un sitio web "hostil". Pocos días después el vendedor distribuía una actualización. Sin embargo, la realidad era que un pequeño porcentaje de usuarios bajaban la actualización y la gran mayoría permanecía vulnerable. De cualquier manera, pocas pérdidas eran atribuibles a los errores del navegador.

Al mismo tiempo, millones de usuarios de computadoras sufrieron pérdidas de billones de dólares a causa de ataques reales experimentados en Internet. Muchos de los daños fueron causados por virus que se movían rápidamente y por gusanos que viajaban por medio del correo electrónico, o que relacionaban explotación automatizada de errores en los programas de servicio de red. Desafortunadamente, en la actualidad se siguen manteniendo los mismos errores, por ejemplo, el programa Internet Explorer de Microsoft es uno de los programas que ha tenido más vulnerabilidades desde mediados del 2003, poniendo en constante riesgo a los usuarios.

Asegurar la computadora del usuario no es una tarea fácil, debido a que cada usuario tiene necesidades distintas, sin embargo, las organizaciones que brindan un servicio a través de Internet deberían hacer las recomendaciones mínimas de seguridad para hacer uso del servicio. Como ejemplo, la figura 2.6 muestra la recomendación que hace el banco Banamex a sus usuarios de banca electrónica.

Figura 2.6: Mensaje de seguridad de un banco a sus usuarios.

Como usuario de este servicio puedo decir que Banamex utiliza el método de autenticación de usuarios mediante contraseñas. Anteriormente manejaban contraseñas de cuatro dígitos pero a partir del 2003 este banco implementó contraseñas más fuertes utilizando ocho caracteres alfanuméricos.

2.3. Criptografía

Como se mencionó en la sección anterior, la criptografía es el método más eficaz para proveer cierta seguridad en el servicio de web. En esta parte, se abordará de manera general la criptografía y los distintos sistemas criptográficos.

La criptografía es una colección de técnicas matemáticas para proteger la información. Usando criptografía, es posible transformar palabras escritas en otros tipos de mensajes, de tal manera que sean difíciles de entender para cualquiera que no posea una *llave* matemática específica necesaria para desbloquear el mensaje. Al proceso de usar criptografía para modificar un mensaje se le conoce como **cifrado**[16]. Y al proceso de obtener el mensaje original se le conoce como **descifrado**.

[16]Frecuentemente se utilizan erróneamente las palabras **criptar o encriptar** las cuales no existen en la lengua española.

La criptografía es usada para prevenir que la información sea accedida por alguien no autorizado para ello. En teoría, una vez que una pieza de información es cifrada, la información puede accidentalmente ser descubierta o interceptada por una tercera parte sin comprometer la seguridad de la información, debido a que la llave necesaria para descifrar la información no ha sido revelada, y que el método de cifrado resistirá intentos de descifrar el mensaje sin la llave.

En la antiguedad, muchos sistemas criptográficos se basaron en dos técnicas: **sustitución** y **transposición**.

- *Sustitución*. Esta técnica está basada en el principio de reemplazar cada letra en el mensaje que se desea cifrar por alguna otra letra. El cifrado *César*, por ejemplo, sustituye la letra **a** por **d**, la letra **b** por la letra **e**, etcétera.
- *Transposición*. Está basada en cambiar los caracteres que están en el mensaje. Un sistema de transposición relaciona la escritura de un mensaje dentro de una tabla renglón por renglón, entonces lee columna por columna.

Actualmente se utilizan técnicas de cifrado basadas en: *factorización* y *curvas elípticas*. De la primera, se desprenden los algoritmos de cifrado simétrico y cifrado asimétrico, también conocidos como cifrado de llave secreta y cifrado de llave pública, respectivamente.

2.3.1. Sistemas de cifrado simétrico

En estos algoritmos, se utiliza la misma llave para cifrar y descifrar el mensaje. Los algoritmos simétricos son conocidos frecuentemente como *algoritmos de llave secreta*. Los algoritmos simétricos más comunes son [17]:

- **DES**. El **D**ata **E**ncryption **S**tandard fue adoptado como un estándar por el gobierno de los Estados Unidos en 1977 y como un estándar ANSI en 1981. El DES es un algoritmo que utiliza cifrado de bloque y utiliza una llave de 56 bits. En 1998, este algoritmo fue roto por una computadora de propósito general.
- **Triple-DES**. Triple-DES es una forma de hacer más seguro al DES usando tres veces el algoritmo DES con tres llaves diferentes, haciendo una llave de longitud total de 168 bits. También llamado "**3DES**", este algoritmo ha sido utilizado ampliamente por instituciones financieras y por el programa Secure Shell (ssh).
- **Blowfish**. Este algoritmo utiliza cifrado de bloque, tiene la característica de ser rápido y compacto. Fue inventado por **Bruce Schneier**. El algoritmo permite una llave de longitud variable, por arriba de 448 bits y ha sido optimizado para procesadores de 32 y 64 bits.
- **IDEA**. El **I**nternational **D**ata **E**ncryption **A**lgorithm fue desarrollado en Zurich, Suecia por **James L. Massey** y **Xuejia Lai** y publicado en 1990. IDEA utiliza una llave de 128 bits y es utilizado por el programa *PGP* para cifrar archivos y correo electrónico.

[17] Garfinkel, Simson, Pretty Good Privacy, pag. 43

- **RC2**. Este cifrado de bloque fue desarrollado originalmente por **Ronald Rivest** y mantenido como un secreto de marca por RSA Data Security. RC2 permite llaves entre 1 y 2048 bits. La longitud de la llave fue limitada a 40 bits en software que era exportado para permitir el descifrado por la Agencia de Seguridad Nacional (NSA) de los Estados Unidos.

- **RC4**. Este cifrado de flujo fue desarrollado originalmente por **Ronald Rivest** y mantenido como un secreto de marca por RSA Data Security. RC4 permite llaves entre 1 y 2048 bits.

- **RC5**. Este cifrado de bloque también fue desarrollado por **Ronald Rivest** y publicado en 1994. RC5 permite al usuario definir la longitud de la llave, el tamaño del bloque de datos y el número de veces que se repite el cifrado.

- **AES**. Este cifrado de bloque fue desarrollado por **Joan Daemen** y **Vincent Rijmen** y fue elegido en octubre del 2000 por el Instituto Nacional de Estándares y Tecnología para ser el nuevo Estándar de Cifrado Avanzado (AES[18]) de los Estados Unidos. Rinjdael tiene la ventaja de ser extraordinariamente rápido y compacto, utiliza llaves con longitud de 128, 192 o 256 bits.

2.3.2. Sistemas de cifrado asimétrico

Con estos algoritmos, una llave es utilizada para cifrar el mensaje y otra llave para descifrarlo. En algunos algoritmos, la llave de cifrado es llamada comúnmente *llave pública* porque puede ser disponible públicamente sin comprometer el secreto del mensaje o de la llave de descifrado. La llave de descifrado, también es conocida como *llave privada*. Los sistemas que utilizan estos algoritmos son conocidos como *sistemas de llave pública*.

Algunas veces, la gente llama a los sistemas de llave asimétrica, sistemas de llave pública, pero esto no es correcto, debido a que no es obligatorio que la llave se haga pública.

Los algoritmos de llave asimétrica más utilizados son [19]:

- **Diffie-Hellman**. Este es un sistema para intercambiar llaves criptográficas entre dos partes activas. Diffie-Hellma no es actualmente un método de cifrado y descifrado, pero si un método que desarrolla e intercambia llaves privadas compartidas sobre un canal de comunicación público.

- **DSA/DSS**. El **D**igital **S**ignature **S**tandard fue desarrollado por la NSA y adoptado como un Estándar de Procesamiento de Información Federal (FIPS[20]).

- **Curvas elípticas**. Los sistemas de llave pública han sido basados tradicionalmente en factorización (RSA), logaritmos discretos (Diffie-Hellman) y el problema de la

[18] Advanced Encryption Standard.
[19] *ibdem*
[20] Federal Information Processing Standard.

"mochila". Los criptosistemas de curvas elípticas son sistemas de cifrado de llave pública basados en una curva elíptica, mas que en funciones logarítmicas tradicionales; esto es, son basados en soluciones a la ecuación:

$$y^2 = x^3 + ax + b$$

- **RSA**. Es un sistema criptográfico de llave pública bien conocido, desarrollado en 1977 por tres profesores en el MIT: **Ronal Rivest**, **Adi Shamir** y **Leonard Adleman**. Este sistema puede ser utilizado para cifrar información y como base de los sistemas de firmas digitales. La llave puede ser de cualquier longitud, dependiendo de la implementación particular utilizada.

Dos de los usos mas comúnes de la criptografía de llave pública son *cifrado de mensajes* y *firmas digitales*:

- Con el cifrado de mensajes, una persona que desea enviar un mensaje cifrado a un destinatario particular, cifra el mensaje con la llave pública de la persona y el mensaje sólo puede ser descifrado por la persona autorizada para ello y que tenga en su poder la llave privada.
- Con las firmas digitales, el remitente del mensaje usa el algoritmo de llave pública y la llave privada para firmar digitalmente un mensaje. Cualquiera que reciba el mensaje puede validar la autenticidad del mismo al verificarlo con la llave pública del remitente.

2.3.3. Sistemas de cifrado híbrido

El cifrado asimétrico tiene la gran ventaja de poder distribuir llaves en un canal inseguro, pero tiene la desventaja de consumir muchos recursos de procesamiento, en cambio el cifrado simétrico permite mayor velocidad pero no resuelve el problema de distribución de llaves. Dadas las ventajas de uno y otro, se ha optado por utilizar una combinación de ambos, denominados sistemas de cifrado híbridos.

Un sistema de cifrado híbrido usa tanto cifrado simétrico como asimétrico. Funciona mediante el uso de cifrado de clave pública para compartir una clave para el cifrado simétrico. El mensaje que se esté enviando en el momento, se cifra usando la clave y enviándolo al destinatario. Ya que compartir una clave simétrica no es seguro, la clave usada es diferente para cada sesión.

2.3.4. Firmas digitales

Una función *hash* es una función múltiple que asigna su entrada a un valor dentro de un grupo finito. Por regla general este grupo es un rango de números naturales.

La firma digital de un documento es el resultado de aplicar una función *hash* al mismo. Para que sea de utilidad, la función *hash* necesita satisfacer dos propiedades importantes:

1. Debe ser difícil encontrar dos documentos cuyo valor para una función *hash* sea el mismo.

2. Dado un valor *hash* debe ser difícil de recuperar el documento que produjo ese valor.

Algunos sistemas de cifrado de clave pública pueden usar funciones hash para firmar documentos. El firmante cifra el documento con su clave privada. Cualquiera que desee comprobar la firma y ver el documento, sólo tiene que usar la clave pública del firmante para descifrarla. Este algoritmo satisface las dos propiedades necesarias para una buena función *hash*, pero en la práctica este algoritmo es demasiado lento.

Como alternativa está el uso de funciones *hash* designadas para satisfacer estas dos importantes propiedades. **SHA** y **MD5** son dos ejemplos de este tipo de algoritmos. Al usar uno de estos algoritmos, un documento se firma con una función *hash*, y el valor del *hash* es la firma. Otra persona puede comprobar la firma aplicando también una función *hash* a su copia del documento y comparando el valor *hash* resultante con el del documento original.

2.3.5. Criptoanálisis

Si bien es cierto que la longitud de la llave es un factor importante en la fuerza de una algoritmo de cifrado [21], no lo es todo. Si así fuera, cualquiera que desee intercambiar mensajes secretos simplemente utilizaría códigos con llaves de 128 bits o más.

Los criptoanalistas (personas que rompen códigos) ráramente utilizan ataques en la búsqueda de las llaves necesarias para descifrar el contenido de un mensaje. En su lugar, muchos algoritmos de cifrado han sido vencidos al utilizar una combinación de matemáticas sofisticadas y poder de cómputo. El resultado es que muchos mensajes cifrados pueden ser descubiertos sin conocer la llave.

2.3.6. La criptografía y el Web

En la actualidad, el cifrado es la tecnología fundamental utilizada para proteger la información que viaja a través de Internet. Cada día, el cifrado es utilizado para proteger el contenido de las transacciones de web, correo electrónico, grupos de noticias, chat, videoconferencias y las llamadas telefónicas que viajan por Internet. Sin el cifrado, cualquier delincuente, ISP, compañía de teléfonos o cualquiera que tanga acceso físico a los cables que llevan los datos pueden "escuchar" el contenido. Con el cifrado es posible proteger un mensaje de tal forma que todas las computadoras del mundo no podrían descifrar dicho mensaje.

La criptografía puede ser utilizada para más que ocultar mensajes. Si se va más allá, los sistemas que emplean técnicas criptográficas son utilizados para el control de acceso

[21] A mayor número de bits, más combinaciones posibles.

a los sistemas de cómputo y para firmar digitalmente los mensajes. Los sistemas criptográficos han sido también visualizados para permitir el intercambio anónimo de dinero digital y más aún, para facilitar los sistemas de elección a través del voto digital.

Un *sistema* criptográfico es una colección de software y hardware que puede cifrar o descifrar información. Un sistema criptográfico típico es la combinación de una computadora de escritorio, un navegador web, un servidor web remoto y la computadora en la cual el servidor web se está ejecutando. En cambio, un *protocolo* criptográfico describe cómo la información se mueve a través del sistema criptográfico. En el caso de la seguridad en web, el navegador y el servidor remoto de web se comunican utilizando el protocolo criptográfico Secure Socket Layer.

Más de una docena de protocolos criptográficos se han desarrollado para la seguridad y el comercio en Internet. Estos sistemas se pueden dividir en dos categorías. La primera categoría de programas y protocolos criptográficos es utilizada para el cifrado de mensajes que no están en línea. La segunda categoría de los protocolos criptográficos es usada para la confidencialidad, autenticación, integridad y no repudiación de los mensajes en línea.

Sistemas de cifrado fuera de línea

Los sistemas de cifrado fuera de línea están diseñados para tomar un mensaje, cifrarlo y almacenar el texto cifrado o transmitirlo a otro usuario sobre Internet. Algunos de los programas que utilizan este tipo de cifrado se muestran en la tabla 2.6[22].

Protocolo	Función	Desarrollo	Programas que lo utilizan
PGP/OpenPGP	Cifrado y firmas digitales para correo y documentos electrónicos.	Amplio	PGP (Netwok Associates) Hushmail Veridis Highware GNU Privacy Guard
S/MIME	Cifrado y firmas digitales para correo electrónico	Poco	Netscape Comunicator Outlook (Microsoft) Outlook Express (Microsoft)

Cuadro 2.6: Programas que utilizan cifrado.

Sistemas y protocolos de cifrado en línea

Los protocolos criptográficos en línea generalmente necesitan interacción en tiempo real entre un cliente y un servidor para trabajar apropiadamente. El protocolo en línea más popular es SSL, el cual es utilizado para proteger la información y cómo ésta es

[22]Garfinkel Simson, Web Security, Privacy & Commerce, pag. 82

enviada entre un navegador y un servidor web. Algunos de los sistemas que están en esta categoría se muestran en la tabla 2.7[23].

Protocolo	Función	Desa.	Programas que lo utilizan
DNSSEC (Secure DNS)	Provee seguridad en la traducción de IP a nombre de equipo y viceversa.	Poco	BIND, ver. 9 (ISC)
IPsec e IPv6	Tráfico seguro de IP	Poco	
Kerberos	Provee autenticación segura e intercambio de llave criptográfica para protocolos de alto nivel	Medio	Kerberos (MIT) Windows 2000 (Microsoft)
PCT(Private Communications Technology)	Proporciona privacidad para transacciones en el web	Poco	Internet Explorer(Microsoft) Internet Information Server (Microsoft)
SET (Secure Electronic Transactions)	Provee privacidad y no repudiación para transacciones de tarjetas de crédito	Poco	Capital One Wallet
SSH (Secure Shell)	Proporciona acceso remoto seguro y adicionalmente provee cifrado para otros protocolos	Amplio	SSH Version 1.x, 2.x OpenSSH Putty SecureCRT
SSL (Secure Socket Layer)	Cifra las comunicaciones. Es utilizado frecuentemente para descargar páginas web y correo electrónico	Amplio	Internet Information Server (Microsoft) Commerce Server (Netscape) Muchos navegadores de web

Cuadro 2.7: Sistemas y protocolos de cifrado en línea.

[23]Garfinkel Simson, Web Security, Privacy & Commerce, pag. 84 y 85

2.4. Secure Socket Layer (SSL)

SSL debe entenderse por Secure Socket Layer, un protocolo de propósito general para enviar información cifrada sobre Internet. Este protocolo fue desarrollado por Netscape y fue popularizado por el navegador y el servidor de web de la misma compañía. Surgió con la idea de estimular las ventas de los servidores habilitados criptográficamente de Netscape al distribuir libremente el cliente que implementaba el mismo protocolo criptográfico.

Desde entonces, SSL ha sido incorporado dentro de muchos otros servidores web y navegadores, y en la actualidad, el soporte de SSL no es sólo una ventaja competitiva, es una necesidad. SSL ha evolucionado a través de dos grandes versiones. EN 1996, la *Internet Engineering Task Force* (IETF), estableció el Transport Layer Security (TLS), como un estándar de cifrado abierto. El grupo empezó con SSL 3.0 y en 1999 publicó el RFC 2246, "TLS Protocol Version 1.0". El RFC 2712 agregó la autenticación a través de Kerberos y el RFC 2818 aplica el TLS usando HTTP/1.1. En la siguiente sección se verá una introducción al SSL y sus principales características.

2.4.1. ¿Qué es SSL?

SSL es una capa que existe entre el protocolo TCP/IP y la capa de aplicación. Mientras el estándar del protocolo TCP/IP simplemente envía un flujo de información no autenticada y libre de errores entre dos computadoras (o entre dos procesos corriendo en la misma computadora), SSL agrega otras características a ese flujo de información, incluyendo:

- Autenticación del servidor, utilizando firmas digitales.
- Autenticación del cliente, utilizando firmas digitales.
- Confidencialidad de los datos, a través del uso de la criptografía.
- Integridad de los datos, por medio del uso de códigos de autenticación de los mensajes.

La criptografía es un campo que se mueve rápidamente, y los protocolos criptográficos no funcionan sin que ambas partes de la comunicación usen el mismo algoritmo. Por esta razón, SSL es un protocolo muy flexible. Cuando un programa usando SSL intenta conectarse a otro, los dos programas electrónicamente comparan sus capacidades para determinar el protocolo criptográfico más fuerte que comparten en común. A este intercambio se le conoce como *SSL Hello*.

El protocolo SSL fue diseñado por Netscape para usarlo con Netscape Navigator. La versión 1.0 del protocolo fue usada dentro de Netscape. La versión 2.0 del protocolo acompañaba a las versiones 1 y 2 de Netscape. Después de que la versión 2.0 de SSL fue publicada, Microsoft creó un protocolo similar llamado PCT, el cual contenía los mismos defectos que SSL 2.0. Las siguiente versión de SSL fue la 3.0 que incluía muchas

mejoras. La versión 3.0 de SSL es la base del protocolo TLS desarrollado por la IETF. Las implementaciones de SSL 3.0/TLS están presentes en Netscape Navigator, Windows de Microsoft y en las bibliotecas de OpenSSL.

2.4.2. Características de SSL y TLS

SSL/TLS ofrecen muchas características de interés práctico y teórico:

- *Separación de funciones.* SSL/TLS utilizan algoritmos distintos para el cifrado, autenticación e integridad de los datos con diferentes llaves (llamados *secretos*) para cada función. La principal ventaja de esta separación de responsabilidades, es que la llave utilizada para la autenticación e integridad de datos puede ser más grande que la llave utilizada para la privacidad. Esto fue muy útil para aquellos programas que se exportaban de los Estados Unidos, debido a que las regulaciones federales ponían límites sobre la longitud de las llaves usadas para la confidencialidad pero no para aquellas llaves utilizadas para la integridad y la autenticación.

  SSL v3 y TLS permiten las conexiones que no son cifradas pero son autenticadas y protegidas contra intentos deliberados de un atacante sofisticado. Esto puede ser útil en circunstancias donde el cifrado está prohibido o es severamente restringido por las leyes.

  La elección del algoritmo y la longitud de la llave son determinados por el servidor SSL, pero es limitado tanto por el servidor como el cliente.

Características de SSL/TSL

- *Eficiencia.* El cifrado y descifrado de llave pública es una operación que consume tiempo. En lugar de repetir este proceso para cada comunicación entre un cliente y un servidor, la implementación de SSL/TLS puede almacenar temporalmente un "secreto maestro" que es preservado entre las conexiones. Esto permite que nuevas conexiones inmediatamente comiencen comunicaciones seguras sin tener que volver a realizar las operaciones de llave pública.

- *Autenticación basada en el Certificado.* SSL/TLS proveen la autenticación tanto para el cliente como para el servidor a través del uso de certificados digitales. SSL v3 y TLS usan los certificados X.509 v3. La autenticación es una parte opcional del protocolo, aunque los certificados de los servidores son solicitados por las implementaciones de los clientes.

- *Independencia del protocolo.* Aunque SSL fue diseñado para ser ejecutado en lo alto del protocolo TCP/IP, éste puede ser ejecutado en cualquier protocolo confiable orientado a conexión como el X.25 u OSI. SSL no puede ejecutarse en un protocolo no confiable como el User Datagram Protocol (UDP). Todas las comunicaciones

SSL/TLS están dispuestas en un flujo bidireccional. En el caso de TCP, los puertos utilizados para implementar servicios a través de SSL se muestran en la tabla 2.8.

Palabra clave	Puerto	Propósito
https	443/tcp	HTTP (web)protegido
ssmtp	465/tcp	SMTP (envío de correo) protegido
snews	563/tcp	Noticias Usenet protegido
ssl-ldap	636/tcp	LDAP protegido
spop3	995/tcp	POP3 (recepción de correo) protegido

Cuadro 2.8: Puertos que utilizan protección con Secure Socket Layer.

- *Protección contra ataques de* ***hombre en medio*** *y ataques de* ***reenvío***. SSL/TLS fue diseñado específicamente para proteger contra el ataque de *hombre en medio* y contra el ataque de *reenvío*. En un ataque de *hombre en medio*, un intruso intercepta todas las comunicaciones entre dos partes, haciendo que cada una de las partes crea que se está comunicando con la otra.

  SSL/TLS protege contra el ataque de *hombre en medio* al utilizar certificados digitales que permiten al usuario de web validar el nombre del servidor. Desafortunadamente, cada navegador usado en la actualidad esconde esta información, haciéndolo accesible sólo a usuarios quienes usan comandos especiales. Una mejor interfaz de usuario podría desplegar el nombre válido del sitio web en la barra de título del navegador o en algún otro lugar que sea conveniente. Debido a que la información del certificado es normalmente escondida, SSL realiza un trabajo pobre al proteger a los usuarios contra este tipo de ataques.

  En un ataque de *reenvío*, un atacante captura las comunicaciones entre dos partes y reenvía el mensaje. Por ejemplo, un intruso puede capturar un mensaje entre un usuario y una institución financiera indicando que fue realizado un pago electrónico; para reenviar este ataque, el intruso indicaría que se realizaron muchos pagos electrónicos.

- *Soporte de compresión*. Debido a que los datos cifrados no pueden ser comprimidos, SSL/TLS proporcionan la capacidad de comprimir datos del usuario antes de que sean cifrados. El protocolo soporta múltiples algoritmos de compresión.

- *Compatibilidad con SSL 2.0*. El servidor SSL v3 puede recibir conexiones de SSL v2 y automáticamente manejar el mensaje sin forzar la reconexión del cliente.

2.4.3. Protección real de SSL

SSL actualmente protege poco contra ataques reales que los clientes y vendedores han experimentado en Internet. Esto se debe principalmente porque SSL (y Netscape por consiguiente) no intentan resolver problemas difíciles de seguridad del comercio en Internet,

en su lugar, se han centrado en problemas que son fáciles de resolver.

La protección real de los consumidores en Internet vienen de las políticas de compañías como VISA, MasterCard, American Express y otras que implementan dichas políticas. Si se compra algo en Internet con la tarjeta de crédito y se descubre un cargo fraudulento en el estado de cuenta mensual del banco, es posible comunicarse con la compañía de la tarjeta y debatir el cargo. Esta protección no sólo trabaja contra el fraude sino también contra una entrega defectuosa o productos equivocados.

Los consumidores con tarjetas de débito tienen menos protección debido a que el dinero ha sido tomado fuera de la cuenta del banco, y en este caso, el consumidor debe pelear para volver a obtenerlo en caso de que se dé este problema.

2.5. Certificados digitales

SSL/TLS hacen uso extensivo de los certificados de llave pública para autenticar tanto al cliente como al servidor en las transacciones SSL/TLS. Hacen uso de los certificados X.509 v3 y los certificados Diffie-Helman para mantener el par de llaves de cifrado; SSL v3 también soporta un certificado X.509 modificado para mantener las llaves públicas usadas por el protocolo de intercambio de llave Fortezza/DMS del Departamento de Defensa de los Estados Unidos.

2.5.1. Autoridades certificadoras

Una autoridad certificadora (CA[24]) es una organización que avala certificados digitales. Un individuo u organización puede ser una autoridad certificadora: ser un simple CA significa que el individuo o la organización firman certificados con una llave pública. Un CA puede imponer estándares antes de firmar una llave; en el caso de una universidad, ésta podría verificar que la llave que va a certificar realmente pertenezca a un estudiante de dicha institución. El más grande CA del mundo es VeriSign, el cual valida diferentes tipos de certificados. VeriSign firma certificados bajo la Red de Confianza de VeriSign (VTN[25]) para uso público; esta compañía también valida certificados para uso dentro de corporaciones. El más bajo nivel de certificados firmados por VTN no tienen garantía; los más altos niveles vienen con la promesa de que VTN intentará establecer la identidad del dueño de la llave antes de que el certificado sea firmado.

Generalmente un certificado se puede ver como una tarjeta firmada digitalmente. El certificado, firmado con la llave privada de la autoridad certificadora, contiene el nombre del CA, la llave pública del CA, un número serial y alguna otra información.

Existen muchas formas de que una organización proporcione los servicios de certificados digitales:

[24] Certification Authority.
[25] VeriSign Trust Network.

- *CA Interno.* Una organización puede operar un CA para certificar a sus empleados. Los certificados liberados por un CA interno pueden certificar el nombre, posición y nivel de autoridad de un individuo. Estos certificados pueden ser utilizados dentro de la organización para controlar el acceso a recursos internos o controlar el flujo de la información. Como un CA interno, éstas pueden ser las bases para la Infraestructura de Llave Pública de la organización (PKI[26]).

- *Emplear un CA externo para los empleados.* Una organización puede querer tomar los beneficios de usar certificados digitales, pero no tiene las capacidades técnicas para ejecutar sus propios servidores de certificados. Esta organización puede contratar a una firma externa para proveer los servicios de certificación para sus empleados.

- *Emplear un CA externo para los clientes.* Una compañía puede contratar a una firma externa para operar una autoridad de certificación para los clientes de la empresa. Para confiar en las prácticas de certificación de la firma externa, la organización deberá crear sus propios procedimientos.

- *Utilizar un CA externo confiable.* Una compañía o gobierno puede operar un CA que ligue la llave pública con el nombre legal de un individuo o negocio. Como un CA puede ser usado para permitir a los individuos realizar transacciones legales. Las firmas de certificados de un CA tienen una cierta analogía con las licencias para conducir automóviles.

Antes de utilizar un certificado firmado por un CA, es necesario tener una copia de la llave pública del CA. Actualmente, muchos de estos certificados vienen incluidos en navegadores web y sistemas operativos. La llave pública de un CA puede ser agregada manualmente por un usuario.

2.5.2. Tipos de Certificados

Existen básicamente cuatro tipos de certificados digitales en uso sobre Internet:

- *Certificación de CA.* Estos certificados contienen la llave pública de los CA, ya sea el nombre del CA o el nombre del servicio particular a ser certificado. Estos certificados son típicamente de auto-firma, ésto es, son certificados propios de los CA firmados con sus llaves privadas.

- *Certificados de servidor.* Estos certificados contienen la llave pública de un servidor SSL, el nombre de la organización que mantiene el servidor, el nombre DNS del servidor y la llave pública del servidor. Cada servidor web habilitado criptográficamente en Internet debe ser equipado con un certificado del servidor para que el protocolo de cifrado SSL funcione apropiadamente. Aunque el propósito original de estos certificados fue el permitir a un consumidor poder determinar la identidad del servidor web y prevenir ataques de *hombre en medio*, en la práctica los certificados de servidores no son utilizados para este propósito.

[26] Public Key Infraestructure.

- *Certificados personales*. Estos certificados contienen el nombre de un individuo, al igual que su llave pública. También pueden tener otra información como la dirección electrónica del individuo, dirección postal, fecha de nacimiento y alguna otra información de identificación.

  Algunos bancos y casas de inversión liberan certificados digitales a sus depositarios. Los certificados son mantenidos típicamente en las computadoras personales de los depositarios y proveen un nivel adicional de confianza cuando un suscriptor intenta acceder a sus cuentas.

  Muchas corporaciones utilizan certificados digitales para sus empleados. Cada servidor web que pertenece a la organización puede permitir el acceso a cualquiera que tenga un certificado válido, con ésto se evita la necesidad de tener una lista de empleados. Ésto también libera al empleado de tener que recordar muchos nombres de usuario y contraseñas individuales.

  Los certificados personales son sustancialmente una forma más segura de identificar a la gente en Internet que la alternativa de claves de acceso y nombres de usuarios.

- *Certificados de fabricantes de software*. Estos certificados son utilizados para verificar la firma del software que es distribuido, como un componente ActiveX que es descargado en línea y ejecutado. Cada copia del reciente sistema operativo Windows es distribuida con un número del certificado de la compañía de software que puede ser usada para validar la firma de las aplicaciones Windows.

Sin embargo, ésto es sólo el comienzo. Muchos profesionales de la seguridad creen que los certificados digitales serán distribuidos ampliamente a través de Internet y la sociedad en general. Algunos usos propuestos para los certificados digitales incluyen:

- Para los consumidores, los certificados digitales podrían ser utilizados para proveer la membresía en una organización o el derecho de un privilegio legal sin revelar el nombre del consumidor.

- Para reconocer la edad, un certificado podría ser utilizado también para controlar el acceso a la pornografía y cualquier otra información en Internet que está restringida legalmente para los menores de edad.

- Se podrían utilizar los certificados digitales para eliminar el correo basura. Para realizar ésto, se deberá programar al sistema de correo electrónico para rechazar todo el correo que no está firmado digitalmente.

- La llave pública y los certificados digitales incrementarán el uso de contratos comerciales en Internet.

- Los certificados digitales pueden ser usados como la infraestructura de autenticación para una tarjeta de identidad nacional.

2.5.3. Certificados soportados por SSL

SSL soporta los siguientes tipos de certificados:

- Certificados de llave pública RSA con longitud arbitraria de la llave.
- Certificados de llave pública RSA limitados a 512 bits, diseñados para lugares donde está restringido el uso, exportación e importación de criptografía fuerte.
- Certificados de sólo firma RSA, los cuales contienen llaves públicas RSA que son usadas sólo para firmar datos y no para cifrar.
- Certificados DSS.
- Certificados Diffie-Hellman.

2.5.4. Instalación y configuración de SSL en el SIIEM/UAP

Una vez que se ha explicado la importancia de la criptografía en el web y que se ha determinado que SSL es uno de los protocolos más importantes para implementar cifrado, en esta parte de la tesis se verá la forma de instalar y configurar la versión libre de SSL, a saber, **OpenSSL**. Posteriormente se creará el certificado del equipo que aloja el sistema SIIEM/UAP.

Instalación de OpenSSL

OpenSSL es la versión libre de SSL. Para instalar OpenSSL es necesario obtener el código fuente de la siguiente dirección en Internet: http://www.openssl.org/source. Al momento de realizar este trabajo, la última versión es openssl-0.9.7c. A continuación se indican los pasos necesarios para instalarlo.

Desempaquetado del código fuente.

```
% tar zxvf openssl-0.9.7c.tar.gz
```

Ingresar al directorio donde se desempaquetó el código fuente.

```
% cd openssl-0.9.7c
```

Configurar el código de acuerdo a las características del equipo.

```
% ./config
```

Compilar el código fuente.

```
% make
```

Realizar una prueba.

```
% make test
```

Instalar el código ejecutable. Si los pasos anteriores no generaron ningún error, el siguiente paso es instalar el programa compilado, para ello es necesario tener privilegios en el sistema

```
% su            #Cambio a la cuenta de administracion.
$ make install # Instalacion del codigo
```

Verificación del programa. Los programas y bibliotecas quedarán instalados en el directorio /usr/local/ssl. Para verificar la versión de OpenSSL se puede ejecutar el siguiente comando:

```
$ openssl
openssl>version
OpenSSL 0.9.7c 10 Apr 2003
```

Creación del Certificado

Una de las ventajas de OpenSSL es la facilidad que tiene para crear certificados digitales. A continuación se indican la serie de pasos necesarios para crear el certificado para el equipo que aloja al SIIEM/UAP.

1. Crear archivos con datos aleatorios.

```
$ cat /dev/urandom > rand1.dat
$ cat /dev/urandom > rand2.dat
$ cat /dev/urandom > rand3.dat
```

2. Determinar el nombre del dominio para el certificado. Generalmente se crea un certificado para un equipo. En el caso del SIIEM/UAP, el sistema está alojado en una computadora con el nombre **newman.posgrado.unam.mx**.

3. Generar el certificado con OpenSSL utilizando el algoritmo RSA con una longitud de llave de 1024 bits proporcionándole datos aleatorios con los archivos creados en el primer paso.

```
$ openssl genrsa -des3 -rand rand1.dat:rand2.dat:rand3.dat
  -out newman.posgrado.unam.mx 1024

120803328 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.........++++++
.....................++++++
e is 65537 (0x10001)
```

4. Ingresar la frase de paso y confirmarla.

```
Enter pass phrase for newman.posgrado.unam.mx.key:
Verifying - Enter pass phrase for newman.posgrado.unam.mx.key
```

Una vez realizado lo anterior se genera el archivo newman.posgrado.unam.mx.key que contiene la llave privada para el sitio.

5. Realizar una copia de respaldo del archivo que se acaba de generar.

```
$ cp newman.posgrado.unam.mx.key new.pos.unam.mx.key.BAK
```

6. Generar una solicitud de firma de certificado (CSR - Certificate Signing Request) en el que se ingresan los datos de la organización.

```
$ openssl req -new -key newman.posgrado.unam.mx.key
  -out newman.posgrado.unam.mx.csr

Enter pass phrase for newman.posgrado.unam.mx.key:
You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:MX
State or Province Name (full name) [Some-State]:D.F.
Locality Name (eg, city) []:Ciudad de Mexico
Organization Name (eg, company) [Widgits Pty Ltd]:UNAM
Organizational Unit Name (eg, section) []:DGAE,UAP
Common Name (eg, YOUR name) []:UAP
Email Address []:webmaster@uap.unam.mx

Please enter the following 'extra' attributes
```

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

Se genera el archivo **newman.posgrado.unam.mx.csr** después de completar la información solicitada.

7. Generar el certificado firmado por el mismo servidor. En este punto es importante que la firma sea realizada por una Autoridad Certificadora como Verisign, sin embargo, dadas las características académicas del SIIEM/UAP, será la llave del mismo sistema con la que se firme el certificado.

```
$ openssl x509 -req -days 365 -in newman.posgrado.unam.mx.csr
          -signkey newman.posgrado.unam.mx.key
          -out newman.posgrado.unam.mx.crt
Signature ok
subject=/C=MX/ST=D.F./L=Ciudad Universitaria/O=UNAM,DGAE/
         OU=Unidad de Administracion del Posgrado/CN=UAP/
         emailAddress=webmaster@uap.unam.mx
Getting Private key
Enter pass phrase for newman.posgrado.unam.mx.key:
```

Ahora ya se tiene el certificado en el archivo **newman.posgrado.unam.mx.crt**.

Habilitar el servidor de web para utilizar SSL

Una de las características de Apache, el servidor web del SIIEM/UAP, es la posibilidad de agregar módulos de manera dinámica. **mod_ssl** es el módulo que agrega el soporte de SSL al servidor web. Para instalar este módulo, se deben llevar a cabo los siguientes pasos.

1. Obtener la última versión del código fuente de mod_ssl del sitio http://www.modssl.org/source.

2. Desempaquetar mod_ssl y apache para instalarlos de manera conjunta.

```
$ tar zxvf mod_ssl-2.8.14-1.3.27.tar.gz
$ tar zxvf apache_1.3.27.tar.gz
```

3. Configurar mod_ssl.

```
./configure \
 --with-crt=/RUTA/certificado/newman.posgrado.unam.mx.crt \
 --with-key=/RUTA/certificado/newman.posgrado.unam.mx.key \
 --with-apache=../apache_1.3.27 \
 --prefix=/usr/local/apache \
 --enable-shared=ssl
```

4. Configuración de apache.

```
$ cd ../apache_1.3.27
$ PATH=PATH:/usr/sbin
$ ./configure --enable-module=ssl
```

5. Compilación e instalación de apache.

```
$ make
$ su
$ make install
```

6. Abrir el archivo de configuración de apache **/usr/local/apache/conf/httpd.conf** y editar las siguientes líneas.

```
SSLCertificateFile /RUTA/newman.posgrado.unam.mx.crt
SSLCertificateKeyFile /RUTA/newman.posgrado.unam.mx.key
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:
               +LOW:+SSLv2:+EXP:+eNULL
```

7. Iniciar el servidor de web apache.

```
/usr/local/apache/bin/apachectl startssl
```

Verificación del certificado aplicado al sitio.

Existen diferentes pruebas para verificar que el servidor web ya está habilitado criptográficamente, algunas de estas pruebas se muestran a continuación.

- Verificación de puerto abierto. El servidor web habilitado criptográficamente abre de forma predeterminada el puerto 80 y el puerto 443. El comando nmap, que sirve para realizar la verificación de puertos abiertos en un equipo, muestra la siguiente información:

```
nmap localhost

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on lapfer (127.0.0.1):

Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
443/tcp    open        https
```

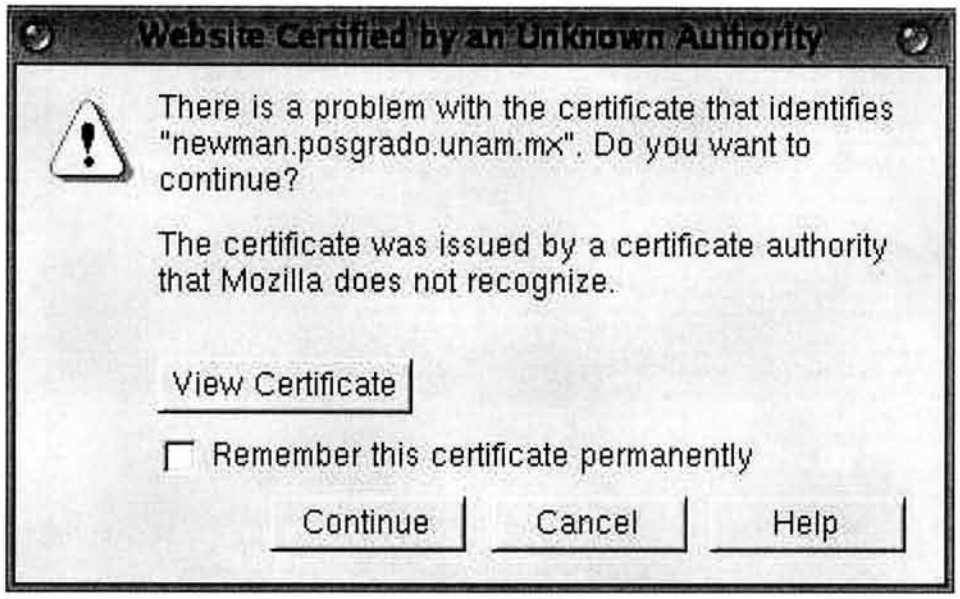

Figura 2.7: Mensaje de aviso del certificado digital del SIIEM/UAP.

- Navegador de Internet. Accediendo a la página del SIIEM/UAP utilizando el protocolo *https*, muestra un aviso de que existe un certificado para el sistema como se muestra en la figura 2.7.
- Ícono de seguridad. Navegadores como Internet Explorer y Netscape muestran el ícono de un candado cerrado indicando que el sitio está habilitado critográficamente, como se muestra en la figura 2.8.

Figura 2.8: Ícono indicando que el SIIEM/UAP está habilitado criptográficamente.

Para obtener información más detallada del certificado se pueda dar doble clic en este ícono de candado como se aprecia en la figura 2.9.

Se deben observar dos cosas importantes:

- Las huellas digitales, *fingerprint*, son el medio con el cual se identifica cada certificado, por lo que son únicas.
- La duración del certificado es de 365 días.

Con la implementación del certificado digital se habilita al servidor web del SIIEM/UAP para el manejo de cifrado a través de SSL, que como se vió antes proporciona las siguientes ventajas:

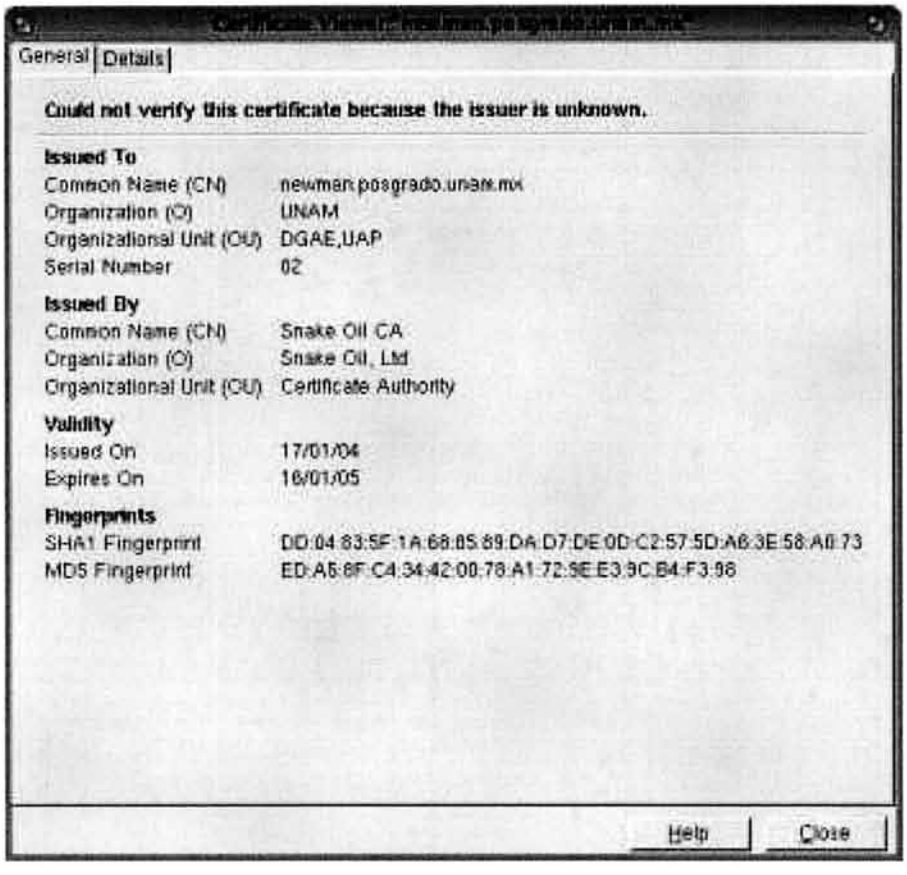

Figura 2.9: Certificado Digital del SIIEM/UAP.

- Confidencialidad
- Integridad
- Confianza entre el cliente y el servidor
- Transferencia de datos cifrados

Sin embargo, como se analizó a lo largo de este capítulo, el cifrado no es la única medida que debe ser tomada para proporcionar el servicio de web seguro, ente otras cosas se debe considerar:

- Asegurar el equipo de cómputo. Esto incluye:
  - Actualización del sistema operativo y de aplicaciones.
  - Proporcionar los servicios mínimos necesarios.
  - Definición de mecanismos de autenticación de usuarios.
  - Habilitar los mecanismos de auditoría del sistema.

  - Realización de respaldos.
  - Restricción de acceso físico.

- Seguridad del servidor web. En esta parte se analizaron los siguientes elementos:
  - Aislamiento del servidor de redes públicas.
  - Definición del control de acceso a los objetos del servidor.
  - Habilitar los mecanimos de registro del servidor web.
  - Definir la seguridad en elementos añadidos al servidor como programas externos.
  - Configurar el servidor para el manejo de cifrado.
  - Mantener una copia de respaldo del servidor web (páginas html, programas, archivos de configuración, etc.).
  - Protección contra ataques comúnes.

- Asegurar la información en tránsito. En este aspecto se puntualizó que la mejor manera de lograr este objetivo es a través del cifrado de datos.

- Asegurar la computadora del usuario. Asegurar el equipo del cliente es algo que está fuera del alcance del administrador del servidor web, sin embargo, se pueden hacer las recomendaciones básicas para que el usuario esté protegido a través del uso de medidas como:
  - Utilizar un navegador de Internet que soporte cifrado.
  - Uso y actualización de antivirus.
  - Uso y configuración de firewalls personales.
  - Cambio periódico de contraseñas.

Con esto se cubre de manera general e integral la seguridad del servidor web, no obstante, los intrusos constantemente están buscando nuevas formas de vulnerar un sistema, por lo que el administrador de sistemas debe estar en constante actualización sobre la información que se genera al rededor de la seguridad en cómputo e implementar las medidas necesarias para reducir el riesgo.

En el siguiente capítulo analizaré otro elemento muy importante dentro de la seguridad del SIIEM/UAP que es la base de datos.

Capítulo 3

Seguridad en la Base de Datos

En este capítulo se abordarán los temas más importantes sobre la seguridad en una base de datos relacional. Se identificarán las amenazas de seguridad en una base de datos y se realiza un análisis de la seguridad que implementa el manejador de base datos en el SIIEM/UAP. Los temas considerados dentro de este capítulo son:

- *Introducción a la seguridad en bases de datos*
- *Problemas que afectan la seguridad en bases de datos*
- *Seguridad en el manejador de base de datos del SIIEM/UAP*

3.1. Introducción a la seguridad en bases de datos

La amplia confiabilidad ofrecida actualmente por las tecnologías de hardware y software, la continua reducción de costos, el incremento de especialistas y la disponibilidad de herramientas de soporte, son factores que han contribuido al amplio uso de los servicios de cómputo.

Esto ha significado que los datos actualmente sean almancenados y manejados por sistemas de cómputo, o por herramientas y técnicas capaces de soportar y manejar los requerimientos de las aplicaciones. Tales requerimientos han sido ampliamente cubiertos por las tecnologías de bases de datos empleando *Sistemas de Manejo de Bases de Datos o DBMS* [1].

Una base de datos es una colección de datos manejados por el software del DBMS. El diseño de las metodologías de bases de datos han sido desarrolladas para soportar los diferentes requerimientos de la información y los ambientes de operación de las aplicaciones. Han sido estudiados modelos conceptuales y lógicos de datos, con los lenguajes asociados y las herramientas para la definición, manipulación y consulta de datos. El objetivo ha sido la producción de DBMS capaces de acceder eficientemente a los subconjutos de datos definidos por las aplicaciones en una base de datos.

Una característica básica de un DBMS es su capacidad para manejar transacciones simultáneamente por aplicaciones concurrentes; así, cada aplicación tiene la visión de una base de datos dedicada. Esta es una característica muy importante, por ejemplo, para una base de datos de un banco y sus cientos o miles de usuarios en línea, o en una base de datos de una línea aérea.

Aunque el uso generalizado de las bases de datos, ya sean centralizadas o distribuidas, y el soporte que han dado a las funciones de las organizaciones, las bases de datos también poseen serios problemas en la seguridad de los datos. De hecho, el daño en un ambiente de base de datos no sólo afecta a un simple usuario o aplicación, sino también a todo el sistema de información: por lo tanto, las consecuencias son impredecibles.

La complejidad en el diseño e implementación de un sistema de información seguro depende del número de factores que son tomados en cuenta, tales como la heterogeneidad de los usuarios del sistema, la granularidad y la extensión territorial de los sistemas de información (ya sea a nivel nacional o internacional), las consecuencias incontrolables e impredecibles de la pérdida de la información y la dificultad en el modelado, especificación y verificación en la seguridad de los datos.

[1] Database Management Systems.

3.1.1. Seguridad en base de datos

La seguridad de la información en una base de datos incluye tres aspectos principales: *confidencialidad, integridad,* y la *disponibilidad.*

Asegurar la **confidencialidad** significa prevención/detección/disuasión de la revelación no autorizada de la información. De manera general, la confidencialidad se refiere a la protección de datos relacionados a los altos niveles de protección como lo son los ambientes militares, gubernamentales y algunos comerciales. La *Privaciadad* se refiere a la información de los individuos, y es definida algunas veces como *el derecho de un individuo, grupo o institución de determinar cuándo, cómo y para qué propósito la información que le concierne puede ser recolectada, almacenada y liberada a otra gente o entidad.*

Asegurar la **integridad** de la información significa prevención/detección/disuasión de la modificación no autorizada de la información. En un ambiente comercial, la integridad de los datos es un aspecto relevante: el buen funcionamiento de una organización depende de la operación adecuada sobre datos correctos y coherentes. Por ejemplo, un empleado no debería ser capaz de modificar su salario o modificar los datos de un pago electrónico.

Asegurar la **disponibilidad** del sistema (esto es, evitar la *negación del servicio*) significa prevención/detección/disuasión de la negación del acceso a los servicios que proporciona el sistema. Por ejemplo, en un ambiente comercial, el pago de órdenes de impuestos deben ser realizados en los términos que marca la ley.

3.2. Problemas de seguridad en las bases de datos

En ambientes de bases de datos, las diferentes aplicaciones y usuarios de una organización se refieren a un conjunto único e integrado de datos a través del DBMS. Por una parte, ésto resuelve problemas como la duplicación, inconsistencia de datos o dependencia entre los programas y las estructuras de datos; por otra parte, las amenazas de la seguridad vienen en un tema más serio e importante en ambientes de bases de datos.

Realizar seguridad en un ambiente de base de datos significa identificar las amenazas y elegir las *políticas* apropiadas ('qué' es lo que se espera que el sistema de seguridad haga) y los *mecanismos* ('cómo' el sistema de seguridad alcanzará sus objetivos). Este proceso también involucra la *garantía del sistema de seguridad* ('qué tan bien' el sistema de seguridad alcanza los requerimientos de protección y ejecuta las funciones esperadas). Estos elementos se definieron en el SIIEM/UAP durante el primer capítulo, sin embargo, es importante entrar a más detalle en lo referente a la base de datos.

3.2.1. Amenazas a la seguridad de la base de datos

Una *amenaza* puede ser definida como un agente hostil, ya sea casual o por utilizar una técnica especializada, que puede revelar o modificar la información manejada por un sistema.

Las violaciones a la seguridad de la base de datos consisten en lectura, modificación o eliminación de datos sin autorización. Los sucesos que traen las violaciones a la base de datos son llamadas amenazas. Las consecuencias de las violaciones pueden ser agrupadas en tres categorías.

- *Emisión no autorizada de la información* causada por la lectura de datos de un acceso intencional o accidental por usuarios no autorizados. Se incluye en esta categoría las violaciones a la confidencialidad derivadas de la observación autorizada de datos que pueden ser usados para inferir información no autorizada.
- *Modificación no autorizada.* Esto involucra todas las violaciones a la integridad de los datos a través del manejo o modificación no autorizados. La modificación no autorizada no necesariamente involucra la lectura no autorizada ya que los datos pueden ser modificados sin que sean leídos.
- *Negación de servicio.* Esto es causado por aquellas acciones que impiden a los usuarios válidos acceder a los datos o usuar los recursos.

Las amenazas de seguridad también pueden ser clasificadas de acuerdo a la forma en que pueden ocurrir, ésto es, amenazas *no fraudulentas* (accidentales) y amenazas *fraudulentas* (intencionales).

Las amenazas no fraudulentas son accidentes ocasionales, estas amenazas involucran:

- *Desastres naturales o accidentales.* Tales como terremotos, inundaciones o fuego. Estos accidentes pueden dañar el hardware del sistema y los datos almacenados; este tipo de desastres siempre causan una violación a la integridad o negación de servicio.
- *Errores o fallas en hardware o software.* Ésto puede conducir a la aplicación incorrecta de las políticas de seguridad, y por consiguiente al acceso, lectura o modificación no autorizados de datos o la negación del servicio.
- *Errores humanos.* Estos errores pueden causar violaciones no intencionales como la entrada incorrecta de datos o el uso inadecuado de las aplicaciones, las consecuencias son similares a las ocasionadas por errores en hardware o software.

Los factores fraudulentos o intencionales denotan un fraude explícito y determinado que causará un daño. Este tipo de violaciones involucran a dos tipos de usuarios:

- *Usuarios autorizados* quienes pueden abusar de sus privilegios y autoridad.

- *Agentes hostiles*. Son usuarios hostiles (internos o externos a la organización) que realizan acciones de vandalismo al software y/o al sistema de hardware, lecturas o escrituras incorrectas de datos. En ambos casos las actividades 'legales' de las aplicaciones pueden enmascarar el propósito real. Virus, caballos de troya y puertas traseras son ataques comúnes de agentes hostiles.

  - Un virus es un código capaz de copiarse así mismo y de dañar permanentemente, y por lo regular sin reparación, el ambiente donde éste se reprodujo.
  - Un Caballo de Troya es un programa el cual, bajo una aparente utilidad, recolecta información para un posible uso fraudulento. Este puede ser software instalado sin el conocimiento del usuario autorizado, el cual explota los privilegios legítimos del usuario para causar un problema de seguridad.
  - Una puerta trasera es un segmento de código escondido en un programa; una entrada especial iniciará este segmento de código y permitirá a su dueño el acceso al sistema más allá de sus privilegios evadiendo los mecanismos de protección.

3.2.2. Requerimientos de protección en una base de datos

La protección a una base de datos de posibles amenazas significa proteger los recursos, particularmente los datos almacenados, de lecturas o modificaciones accidentales o intencionales no autorizadas. Los requerimientos de protección de una base de datos pueden ser resumidos como sigue.

Protección de acceso incorrecto

Esto consiste en garantizar el acceso a la base de datos sólo a los usuarios autorizados. Las solicitudes de acceso deben ser revisadas por el DBMS contra las autorizaciones de los usuarios o aplicaciones. Los controles de acceso son más complejos para una base de datos que para archivos manejados por un sistema operativo. Los controles necesitan ser aplicados a los objetos en una granularidad fina como son registros, atributos y valores. Adicionalmente, el dato dentro de una base de datos es semánticamente relacionado, así se permite a un usuario conocer el valor de un dato sin accederlo directamente, pero por inferirlo de un valor conocido.

Protección de inferencia

La inferencia denota la posibilidad de obener información confidencial a partir de datos no confidenciales. En particular, los problemas de inferencia afectan las bases de datos estadísticas donde los usuarios deben ser impedidos de trazar la información sobre entidades individuales iniciando en información estadística total. Por ejemplo, supóngase a un usuario que primero consulta la base de datos para obtener el promedio del salario de los empleados del genero femenino, y después obtiene el número de empleados del mismo sexo. Si este último valor es 1, el usuario está en la posibilidad de determinar (in-

ferir) el salario de una mujer usando sólo consultas estadísticas (promedio y conteo).

Integridad de la base de datos

Este requerimiento concierne a la protección de accesos no autorizados de la base de datos que podrían modificar el contenido de los datos, también de errores, virus, sabotaje o fallas en el sistema que puedan dañar los datos almacenados. Este tipo de protección es parcialmente cubierto por el DBMS a través de sus controles del sistema propios, copias de respaldo y procedimientos de recuperación, y la otra parte a través de procedimientos de seguridad a la medida.

Los respaldos y procedimientos de recuperación son ampliamente investigados. En caso de una falla, el estado de la base de datos puede no ser consistente. Para preservar la consistencia, se necesita que cada transacción sea atómica. Atomicidad significa que cada transacción sólo puede:

1. Terminar de forma correcta, modificando los datos accedidos.
2. Terminar de forma incorrecta sin modificar los datos accedidos.

Después de que una transacción ha terminado adecuadamente, las modificaciones a los datos son permanentes.

El *sistema de recuperación* utiliza un registro diario, a saber, un archivo que contiene una secuencia de registros dentro de un almacenamiento estable. Para cada transacción, se registran las operaciones que han sido realizadas a los datos (lectura, escritura, inserción, eliminación) también como el control de operaciones de la transacción ('inicio de la transacción','asignación' -terminación correcta, 'cancelación' - terminación incorrecta, 'fin de la transacción'), también el valor anterior y el nuevo de los registros involucrados.

Los procedimientos de seguridad a la medida tienen como objetivo proteger los datos de modificaciones, alteraciones, inserciones y eliminaciones no autorizados. El modelado, diseño y fortalecimiento de estos procedimientos es uno de los objetivos de la seguridad lógica de una base de datos.

Integridad operacional de los datos

Este requerimiento tiene como objetivo asegurar la consistencia lógica de los datos en una base de datos durante las transacciones actuales. El **manejador de concurrencia** es el subsistema del DBMS que cubre este requerimiento.

El manejador de concurrencia asegura las propiedades de *serialización* y *aislamiento* de las transacciones. La *serialización* consiste en que el resultado de un conjunto de transacciones ejecutándose al mismo tiempo es el mismo que el producido por una secuencia estricta de esas transacciones. El *aislamiento* significa la independencia mútua entre transacciones, así se evita el *efecto dominó*, donde una transacción abortada causa que otras

transacciones sean abortadas en cascada.

El problema de asegurar que el acceso concurrente al mismo dato a través de diferentes transacciones no conduzca una inconsistencia de los datos es resuelta comúnmente a través de las técnicas de **bloqueo**.

Las técnicas de bloqueo y desbloqueo consisten, respectivamente, en congelar el dato durante el tiempo necesario para realizar una operación, y es liberado el elemento una vez que la operación ha sido terminada. De esta manera, una transacción puede bloquear un elemento, haciéndolo inaccesible a otras transacciones. El dato es accesible nuevamente al tiempo de liberarlo.

Integridad semántica de los datos

El problema es asegurar la consistencia lógica de los datos modificados, controlando los valores de los datos en un rango permitido. Las restricciones sobre los valores de los datos son expresados como *restricciones de integridad*.

Las restricciones pueden ser definidas para toda la base de datos (condiciones que definen el estado correcto de la base de datos), o para tansiciones (condiciones que son verificadas al realizar una modificación a la base de datos).

Contabilidad y auditoría

Este requerimiento consiste en la posibilidad de registrar todos los accesos a los datos, tanto para operaciones de lectura y escritura. Auditar y contabilizar son herramientas útiles de disuasión para la integridad física de los datos, y también para el análisis subsecuente de secuencias de acceso a las bases de datos.

Autenticación de usuario

Este requerimiento concierne a la necesidad de identificar de manera inequívoca a los usuarios de la base de datos. La identificación de usuarios es la base para cada mecanismo de autorización. Los usuarios tienen acceso a los datos cuando han sido identificados como usuarios 'autorizados' por el sistema.

Manejo y protección de datos sensitivos

Las bases de datos pueden contener *datos sensitivos* que no deberían ser hechos públicos; algunas bases de datos contienen sólo datos sensitivos (por ejemplo, bases de datos militares), mientras otras son completamente públicas (como lo puede ser la base de datos de una biblioteca). Las bases de datos que contienen datos mezclados, esto es, tanto datos sensitivos y ordinarios, presentan problemas de protección más complejos. Un dato es sensitivo en varias circunstancias: en si mismo, cuando es combinado con otros datos, por estar contenido en registros que son declarados como sensitivos o por ser declarado

sensitivo por el *Administrador de la Base de Datos* (DBA[2]).

Protección multinivel

La protección multinivel consiste en un conjunto de requerimientos de protección. Puede ser necesario que la información sea clasificada en varios niveles de protección: por ejemplo, en una base de datos militar, donde es necesaria una clasificación fina que simplifique los datos 'sensitivos' y 'ordinarios'. En estos ambientes, los niveles sensitivos pueden ser diferentes aún entre los elementos de un mismo registro, o valores de los mismos atributos. En este sentido, la protección multinivel tiene la intención tanto de asignar los niveles de clasificación a diferentes elementos de la información y de asignar los diferentes accesos a elementos simples de acuerdo a su clasificación.

Confinamiento

El confinamiento tiene como propósito evitar la transferencia indeseada de información entre programas del sistema. La transferencia de información ocurre por canales autorizados, canales de memoria y por canales secretos. Los canales autorizados proporcionan una salida de información por medio de acciones autorizadas: por ejemplo compilar o editar un archivo. Los canales son áreas de memoria donde se encuentran almacenados los datos por un programa y puedan ser leídos por otros programas. Un canal secreto es un canal de comunicación basado en el uso de recursos del sistema que no son utilizados normalmente para la comunicación entre los sujetos (procesos) del sistema.

3.3. Seguridad en el manejador de la base de datos del SIIEM/UAP

Una vez que se ha determinado la importancia de la seguridad en la base de datos y las amenazas a las cuales se enfrenta, en esta parte del capítulo se verá la forma de aplicar estos conceptos y mecanismos de seguridad en la base de datos específica.

El Sistema de Inscripciones por Internet de Especializaciones del Posgrado tiene como manejador de base de datos a PostgreSQL. Este es un manejador robusto de mediana capacidad, suficiente para los requerimientos del sistema. A continuación se abordarán los elementos de seguridad que utiliza este DBMS. La figura 3.1 muestra el logotipo de PostgreSQL.

3.3.1. Introducción a PostgreSQL

PostgreSQL es una Sistema de Manejo de Base de Datos Relacional (RDBMS) que ha sido desarrollado desde 1977. Su inicio fue a través del proyecto Ingres en la Universidad de California en Berkeley. Ingres fue desarrollado posteriormente de manera comercial

[2]Database Administrator.

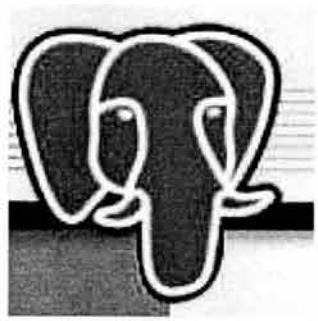

Figura 3.1: El elefante como símbolo de robustez de PostgreSQL.

por Relational Technologies/Ingres Corporation.

En 1986, el equipo lidereado por Michael Stonebraker de Berkeley continuó con el desarrollo del código de Ingres para crear una base de datos relacional llamada Postgres. En 1996, debido al esfuerzo de la comunidad del software libre y en la mejora en la funcionalidad, Postgres fue renombrado como PostgreSQL, después de un breve período como Postgres95. El proyecto PostgreSQL sigue en constante desarrollo a través del mundo por un equipo de programadores de software libre y muchos patrocinadores.

PostgreSQL es considerado como el sistema de base de datos más avanzado de su tipo. Provee muchas características que generalmente sólo son vistas en productos comerciales.

Versión libre de fuente abierta

PostgreSQL es un proyecto de fuente abierta. Fuente abierta, por definición, significa que es posible obtener el código fuente, usar el programa, y modificarlo libremente sin restricción alguna por el propietario del software. En el mundo de las bases de datos, fuente abierta significa que se tiene acceso a pruebas de rendimiento y estadísticas de desempeño a diferencia de otras compañías como Oracle que no lo permiten. El código de fuente abierta también significa que las personas son libres de modificar PostgreSQL para adaptarlo a necesidades específicas.

Sin embargo, existe un malentendido debido a que si el software de fuente abierta es libre de restricciones de distribución, entonces significa que es gratis. Esto no necesariamente tiene que ser así. Si bien es cierto que es posible descargar e instalar software de este tipo, siempre existirá un costo asociado con el tiempo y energía que la organización concentra para el soporte e investigación de la aplicación. De tal manera que si no se cuenta con estos recursos, existen muchos vendedores comerciales y consultores que trabajan específicamente con PostgreSQL.

Soporte

La comunidad de PostgresSQL provee soporte activo a los usuarios a través de listas de correo. Existe una gran cantidad de listas de correo para usuarios que cubren diferentes tópicos (general, desarrollo, etc.). Las listas de correo pueden ser accedidas a través de la página web de PostgreSQL: www.postgresql.org

3.3.2. Autenticación de clientes

La administración y manejo de PostgreSQL son temas que quedan fuera de este trabajo. En esta tesis se dió prioridad a la forma en como PostgreSQL implementa mecanismos de seguridad para el acceso a los datos.

Cuando una aplicación cliente se conecta a un servidor de base de datos, éste especifica con cuál usuario se quiere conectar.

La *autenticación* es el proceso mediante el cual el servidor de base de datos establece la identidad del cliente y determina si la aplicación (o el usuario quien ejecuta la aplicación) está autorizado para conectarse con el nombre de usuario que solicita. La autenticación de los clientes es una característica importante en PostgreSQL. Sin la autenticación, sería necesario elegir entre no permitir la conexión remota o permitir que cualquier usuario acceda a la base de datos.

PostgreSQL tiene diferentes métodos de autenticación de clientes. El mecanismo utilizado para autenticar a un cliente particular puede ser seleccionado en base a la dirección del cliente, la base de datos y el usuario.

Los usuarios dentro de PostgreSQL están separados de los usuarios del sistema operativo en el que es ejecutado el servidor de base de datos. Si los usuarios de un servidor particular tienen cuentas en la máquina que tiene este propósito, es lógico que los nombres de usuarios dentro de la base de datos sean iguales a los del sistema operativo. Sin embargo, un servidor que acepta conexiones remotas pueden tener muchos usuarios que no tiene una cuenta local, en tal caso no es necesario que exista una relación entre las cuentas de usuarios de la base de datos y del sistema operativo.

Archivo pg_hba.conf

En PostgreSQL el control de acceso de los clientes es definido en el archivo de configuración **pg_hba.conf**. Los derechos y restricciones definidos en este archivo no deben ser confundidos con los derechos que se tienen sobre los objetos en una base de datos (*hba* debe entenderse como *host-based authentication*). Un archivo predeterminado es instalado cuando el directorio de datos es inicializado con el comando *initdb*.

El formato general del archivo **pg_hba.conf** es mediante registros, uno por línea. Un registro está compuesto de campos separados por espacios o tabuladores. Los campos

pueden contener espacios en blanco si están encerrados entre comillas. Los registros no pueden estar compuestos de varias líneas.

Cada registro se compone de:

- Tipo de conexión.
- Rango de direcciones IP (si es necesario para el tipo de conexión).
- Nombre de la base de datos.
- Nombre de usuario.
- Método de autenticación.

El primer registro que coincida con estos elementos será utilizado para realizar la autenticación. Si un registro es elegido y la autenticación falla, los subsecuentes registros no son considerados. En caso de que ningún registro coincida, el acceso es negado.

Un registro puede puede tener alguno de los tres formatos

```
local   Base-Datos  Usuario  Metodo-de-autenticacion
                                [opciones-autenticacion]

host    Base-Datos  Usuario  Direccion-IP  Mascara
           Metodo-de-autenticacion  [opciones-autenticacion]

hostssl Base-Datos  Usuario  Direccion-IP  Mascara
           Metodo-de-autenticacion  [opciones-autenticacion]
```

Los campos significan:

local
Este registro intenta la conexión usando sockets Unix. Sin un registro de este tipo, las conexiones a través de sockets están deshabilitadas.

host
Este registro es comparado con las conexiones intentando usar redes TCP/IP. Estas conexiones sólo son habilitadas si el servidor fue iniciado con la opción **-i** o está habilitado el parámetro **tcpip_socket** en el archivo de configuración **postgresql.conf**.

hostssl
Este registro coincide con intentos de conexión usando SSL sobre TCP/IP. El registro **host** coincide con conexiones SSL o no SSL, pero **hostssl** requiere conexiones SSL.

Base-Datos
Especifica la base de datos. El valor **all** indica que coincide con todas las bases de datos. El valor **sameuser** especifica que el registro coincide si la base de datos solicitada tiene el mismo nombre que el usuario que solicita la conexión. El valor

samegroup especifica que el solicitante debe ser miembro del grupo con el mismo nombre de la base de datos requerida. De otra manera, este es el nombre de una base de datos específica de PostgreSQL. Es posible indicar múltiples bases de datos separadas por comas. También es posible especificar un archivo precedido por una @ que contenga las bases de datos. El archivo debe estar en el mismo directorio donde se encuentre **pg_hba.conf**.

Usuario

Indica el usuario para conectarse a la base de datos. El valor **all** especifica que este registro coincide con todos los usuarios. De otra manera, este es el nombre de un usuario específico de PostgreSQL. Al igual que las bases de datos, es posible indicar múltiples nombres de usuarios separados con comas. Los nombres de grupo pueden ser definidos por preceder el nombre del grupo con el caracter +. También se puede especifcar un archivo conteniendo los nombres de usuarios, precediendo el nombre del archivo con @.

El archivo debe estar en el mismo directorio donde esté ubicado **pg_hba.conf**.

Dirección IP y Máscara de red

Estos dos campos contienen el valor de la dirección IP y de la máscara de red en notación estándar (números separados por punto). Las direcciones IP deben ser especificadas numéricamente, no como dominios o nombres de equipos. Los dos valores juntos indican la dirección IP de la máquina cliente que coincide con el registro.

Las implicaciones del engaño de la dirección IP a través de *ip spoofing* están fuera del alcance de PostgreSQL.

Método de autenticación

Especifica el método de autenticación que debe ser utilizado. Las posibilides son:

- trust. La conexión es permitida incondicionalmente. Esté método le permite a cualquier cliente conectarse a una base de datos de PostgreSQL sin la necesidad de una contraseña. *Esta opción puede ser muy últil desde el punto de vista de desarrollo, sin embargo, desde el punto de vista de seguridad no es conveniente para conexiones remotas.*
- reject. La conexión es rechazada incondicionalmente. Esta opción es útil para rechazar las conexiones de ciertos grupos de equipos. *Sin embargo, desde el punto de vista de seguridad, si esta opción está mal configurada puede generar una negación de servicio.*
- md5. Solicita que el cliente proporcione una contraseña cifrada con el algoritmo MD5 para que sea autenticado. Este es el único método que permite almacenar contraseñas cifradas en la tabla *pg_shadow*.
- crypt. Similar al método md5, sólo que utiliza un algoritmo de cifrado más antiguo el cual es necesario para versiones 7.2 y anteriores.
- password. Parecido a md5, pero la contraseña es enviada en texto claro. Esta opción no debe ser utilizada en redes de poca confianza.

- krb4. Se utiliza Kerberos V4 para autorizar al usuario.
- krb5. En este caso se utiliza Kerberos 5 para autenticar al usuario. Este método está disponible sólo para conexiones TCP/IP.
- ident. Obtiene el nombre de usuario del sistemaa operativo del cliente (en conexiones TCP/IP se obtiene al contactar el servidor ident del cliente, para conexiones locales se obtiene desde el sistema operativo) y revisa si el usuario está autorizado para conectarse a la base de datos consultado en el archivo de mapa especificado después de la palabra reservada **ident**.
- pam. La autenticación a través del uso del servicio PAM (Pluggable Autentication Modules) proporcionado por el sistema operativo.

opciones-autenticación
El significado de este campo depende del método de autenticación.

Debido a que los registros en **pg_hba.conf** son examinados secuencialmente para cada intento de conexión, el orden de los registros es importante. Es recomendable que los métodos de conexión más débiles estén al principio, y los métodos de autenticación más robustos se encuentren al final del archivo.

A continuación se muestra parte del archivo de configuración pg_hba.conf que propongo para el SIIEM/UAP:

```
# TYPE  DATABASE  USER      IP-ADDRESS        IP-MASK       METHOD

local     all     postgres                                  md5
local     all     all                                       trust
host      all     all     132.248.73.13   255.255.255.255   md5
host      all     all     132.248.73.16   255.255.255.255   md5
```

Esta configuración indica lo siguiente:

1. Las conexiones locales que intente hacer el super usuario *postgres* requieren de una contraseña.
2. Cualquier solicitud local de cualquier otro usuario a cualquier base de datos no solicita contraseña.
3. Las direcciones IP 132.248.73.13 y 132.248.73.16 son permitidas pare realizar conexiones remotas mediante el uso de una contraseña.

Debido a que los programas cgi acceden a la base de datos de manera local, no es necesario que se solicite contraseña. En el caso de los equipos con las direcciones IP definidas, se confía en la conexión porque estas dos máquinas realizan respaldos remotos de la base de datos, sin embargo, estos respaldos, independientes a los respaldos realizados en las políticas definidas en el primer capítulo, se llevan a cabo a final de semestre, por lo que el mayor tiempo se realizan sólo conexiones locales, de tal manera que el servidor de la

base de datos no acepta solicitudes TCP/IP.

Para llevar a cabo el inicio de PostgreSQL con conexiones TCP/IP o sin ellas, se modificó el archivo de inicialización **/etc/init.d/postgresql**, definiendo dos opciones:

- *start*. Esta opción define que sólo serán permitida las conexiones locales, esta es la forma predeterminada en la cual inicia PostgreSQL. En seguida se muestra la definición de esta opción:

```
start)
$ECHO_N "Iniciando PostgreSQL sin soporte TCP/IP: "$ECHO_C
if [ ! -r $PGLOG ]; then
su - $PGUSER -c "touch $PGLOG"
fi
su - $PGUSER -c "$DAEMON start -D '$PGDATA'  -s -l $PGLOG"
echo "ok"
;;
```

- *start_con_tcp*. Esta opción agrega el parámetro *-o -i* para que PostgreSQL acepte conexiones TCP/IP. A continuación se muestra esta opción:

```
start_con_tcp)
$ECHO_N "Iniciando PostgreSQL con soporte TCP/IP: "$ECHO_C
if [ ! -r $PGLOG ]; then
su - $PGUSER -c "touch $PGLOG"
fi
su - $PGUSER -c "$DAEMON start -D '$PGDATA' -o -i -s -l
     $PGLOG"
echo "ok"
;;
```

Con estas modificaciones es posible iniciar PostgreSQL para que no acepte conexiones TCP/IP, a menos que explícitamente se indique lo contrario. En caso de que se acepten conexiones remotas se aplicarán las reglas definidas en el archivo *pg_hba.conf*.

Problemas en la autenticación

Las fallas de autenticación o problemas relacionados, generalmente son indicados a través de mensajes de error como los que se muestran a continuación:

```
No pg_hba.conf entry for host 132.248.124.184, user petrohs,
database spam
```

Este mensaje implica que la conexión hacia el servidor se llevó a cabo, sin embargo el servidor no puede responder. Este mensaje sugiere que el servidor rechazó la solicitud de la conexión debido a que no encontró la entrada correspondiente en el archivo de configuración.

```
Password authentication failed for user 'mcherrera'
```

Mensajes como éste indican que se ha contactado el servidor, y éste ha respondido, pero la contraseña proporcionada no fue la correcta.

```
FATAL 1: user ''mabemem'' does not exist
```

Este error indica que el usuario no fue encontrado.

```
FATAL 1: Database ''prueba'' does not exist in the system
catalog.
```

La base de datos a la cual se está intentado acceder no existe. Se debe poner atención que si no se especifica el nombre de la base de datos, se utiliza el nombre del usuario como nombre de la base de datos de manera predefinida, lo que puede o no ser correcto.

Para mayores detalles acerca de la información de fallas de autenticación es conveniente revisar el archivo de bitácora de PostgreSQL.

3.3.3. Manejo de usuarios y grupos

Como se vió en la sección anterior, la autenticación de los clientes es el mecanismo de seguridad que PostgreSQL implementa para los intentos de conexión a las bases de datos. En esta parte se verá el otro mecanismo de seguridad que PostgreSQL utiliza para el manejo de objetos dentro de la base de datos mediante usuarios y grupos.

Como en muchos sistemas de bases de datos, los *usuarios* y *grupos* tienen un papel importante dentro de PostreSQL. Usados adecuadamente, los usuarios y grupos permiten granuralidad y control de acceso flexible a los objetos de la base de datos.

Cada base de datos contiene usuarios. Estos usuarios están separados de los usuarios manejados por el sistema operativo sobre el cual el servidor está montado. Los usuarios pueden ser dueños de objetos (por ejemplo tablas) y pueden asignar privilegios sobre estos objetos a otros usuarios para controlar quién tiene acceso a tal objeto.

PostgreSQL almacena los datos de usuarios y grupos dentro de sus propios catálogos de sistema. El control de usuarios define los derechos de quienes están autorizados para realizar funciones dentro del sistema (y qué acciones pueden realizar). Los grupos existen como una forma de simplificar la definición de esos derechos.

Manejo de usuarios

Para establecer una conexión a PostgreSQL se debe proporcionar una forma básica de identificación, ésto se realiza a través del *nombre de usuario*, el cual identifica al usuario

que desea ingresar al sistema. Cada usuario tiene un identificador (ID) de usuario interno, conocido como *sysid*, también tiene una contraseña.

La información de los usuarios está almacenada en la tabla de sistema **pg_shadow**. Esta tabla sólo puede ser accedida por los superusuarios, aunque existe una vista de esta tabla llamada **pg_user** que está disponible para usuarios normales. La figura 3.2 muestra la información contenida en pg_shadow.

```
postgres=# select * from pg_shadow;
usename | usesysid | usecreatedb | usesuper |      passwd         |
--------+----------+-------------+----------+---------------------+
postgres|        1 | t           | t        | md5c53c2656d1c556   |
fzaragoz|      100 | t           | f        | md54ae637972132b1   |
```

Figura 3.2: Tabla de usuarios pg_shadow de PostgreSQL

Los usuarios de la base de datos son globales, esto es, no se crean usuarios para bases de datos individuales. Para crear un usuario se utiliza el comando SQL **CREATE USER** como se muestra a continuación:

```
CREATE USER nombre
```

Donde **nombre** corresponde al nombre del usuario y éste no debe contener caracteres especiales. Para eliminar a un usuario se utiliza el comando **DROP USER**.

```
DROP USER nombre
```

Por facilidad, PostgreSQL proporciona los programas **createuser** y **dropuser** para que realicen la misma función que los comandos SQL, pero éstos son ejecutados desde la línea de comandos:

```
createuser usuario
dropuser usuario
```

Generalmente, cuando se crea una base de datos, el sistema siempre contiene un usuario predefinido. Este usuario tendrá el identificador ID 1, y de forma predeterminada (almenos que se altere cuando se ejecute el comando **initdb**) tendrá el mismo nombre del usuario del sistema operativo que inicializó la base de datos. Normalmente, este usuario tendrá el nombre de **postgres**. De tal manera, que para crear mas usuarios, primero es necesario conectarse como el usuario inicial. Esto puede representar un riesgo de seguridad, debido a que el usuario postgres tiene todos los privilegios sobre la base de datos, por lo que se recomienda asignar una contraseña a este usuario para que pueda acceder a cualquier base de datos.

Debido a que PostgreSQL asume que el usuario con el cual se quiere conectar el cliente tiene el mismo nombre que en el sistema operativo, PostgreSQL proporciona la opción **-U** en la línea de comandos para indicar el nombre del usuario con el que se intenta la conexión.

Atributos de usuarios

Un usuario de la base de datos tiene diferentes atributos que definen sus privilegios e interacción que tiene con el sistema de autenticación.

super usuario. Un super usuario de la base de datos pasa todas las revisiones de permisos. También, este usuario es el único que puede crear nuevos usuarios. Para crear un super usuario, se usa el comando
CREATE USER nombre CREATEUSER.

Creación de bases de datos. A un usuario se le debe dar explícitamente el permiso para crear bases de datos (excepto para super usuarios). Para crear un usuario con estas características se utiliza el comando
CREATE USER nombre CREATEDB.

Contraseña. Una contraseña es importante sólo si el método de autenticación del cliente requiere que el usuario proporcione una para conectarse a la base de datos. Los métodos de autenticación *md5, password* y *crypt* utilizan contraseñas. Las contraseñas de la base de datos están separadas de las contraseñas del sistema operativo.

Para especificar una contraseña al crear un usuario se utiliza el comando
CREATE USER nombre PASSWORD 'cadena'. Para modificar los atributos de un usuario después de que éste sea creado se utiliza el comando **ALTER USER**.

Manejo de grupos

Los grupos sirven para simplificar la asignación de derechos. Generalmente los privilegios deben ser asignados a un usuario individualmente, uno a la vez. Esto puede llegar a ser tedioso si es necesario asignar el mismo acceso a la gran diversidad de objetos en la base de datos.

Los grupos se crean para evitar estos problemas. Un grupo en primer estancia está vacío y requiere sólo de un nombre. Una vez creado, los usuarios que van a compartir los privilegios se agregan al grupo, y de esta manera pertenecen a él. Entonces, los derechos sobre los objetos de la base de datos son autorizados al grupo, en lugar de que sea a cada miembro.

Cualquier super usuario puede crear un nuevo grupo con el comando **CREATE GROUP** como sigue:

```
CREATE GROUP nombre_grupo
```

nombre_grupo corresponde al nombre de grupo que se desea crear. El nombre del grupo debe iniciar con un caracter alfabético y no debe exceder 31 caracteres. Se puede agregrar la palabra clave **USER** para incluir uno o mas usuarios, separados por coma, al tiempo de crear el grupo. La figura 3.3 muestra la tabla de sistema **pg_group** donde se almacenan los grupos de la base de datos.

```
postgres=# select * from pg_group;
 groname | grosysid | grolist
---------+----------+---------
 with    |      100 | {100}
(1 row)
```

Figura 3.3: Tabla de grupos pg_group de PostgreSQL

La eliminación de un grupo puede ser realizada por cualquier super usuario con el comando SQL **DROP GROUP**. No hay forma de poder recuperar un grupo cuando es eliminado, y debido a que no pregunta por la confirmación de la eliminación se debe tener cuidado al usar este comando. El modo de uso de este comando es:

```
DROP GROUP nombre_grupo
```

Si por equivocación se elimina un grupo, para restaurarlo a la funcionalidad previa se debe crear un nuevo grupo con el mismo identificador del grupo eliminado. Este involucra a la palabra reservada **SYSID** al momento de crear el grupo.

Una vez que ha sido creado el grupo, los usuarios pueden ser agregados y eliminados de los grupos en PostgreSQL mediante el comando SQL **ALTER GROUP**, como se muestra en seguida:

```
ALTER GROUP nombre_grupo {ADD|DROP} USER nombre_usuario [, ...]
```

En este caso *nombre_grupo* corresponde al nombre de grupo que va a ser modificado, y *nombre_usuario* es el nombre del usuario que va a ser agregado o eliminado, dependiendo de que opción se especifique **ADD** o **DROP**.

Asignación y revocación de permisos

PostgreSQL mantiene un conjunto de listas de control de acceso (ACL por sus siglas en inglés). Esta información describe a los usuarios con autorización de seleccionar, actualizar o modificar los objetos en una base de datos. Existe un conjunto de privilegios y restricciones para cada base de datos en PostgreSQL. Los super usuarios y propietarios de los objetos de la base de datos mantienen estas ACLs a través de los comandos GRANT y REVOKE.

En lugar de que los super usuarios (quienes pueden manipular cualquier objeto en la base de datos) hagan la asignación de privilegios, sólo los propietarios de los objetos están autorizados para permitir y restringir los privilegios sobre los objetos de su propiedad. Aunque cualquier usuario puede conectarse a la base de datos, si desean acceso a los objetos, deben tener esos privilegios explícitamente garantizados.

Las listas de control de acceso aplican sobre tres tipo de objetos en la base de datos: tablas, listas y secuencias. Para estos objetos, existen cuatro privilegios generales que pueden ser autorizados o negados a un usuario o grupo. Se niega un derecho sólo para revertir la acción de haberlo dado con anterioridad, esto es, desde que se crea un usuario, éste no tiene ningún derecho sobre algún objeto en la base de datos.

El comando \z despliega los permisos en la base de datos donde se encuentre conectado el usuario. La tabla 3.1 muestra la simbología utilizada para los permisos.

Palabra clave	Símbolo	Descripción
SELECT	r	Permite a un usuario obtener datos de una tabla, vista o secuencia. También se le conoce como permiso de *lectura*.
INSERT	a	Permite a un usuario insertar nuevos registros en una tabla.
UPDATE, DELETE	w	Permite a un usuario modificar o eliminar registros de una tabla. Al ceder uno de los derechos, se están cediendo los dos automáticamente. También se conoce como derecho de *escritura*.
RULE	R	Permite a un usuario crear o reescribir una regla sobre una tabla o vista.
ALL	arwR	Simplifica la forma de ceder o quitar todos los derechos al mismo tiempo. ALL no es un derecho en sí mismo, sino la suma de SELECT, INSERT, UPDATE, DELETE y RULE.

Cuadro 3.1: Simbología utilizada en los permisos de los objetos de la base de datos.

GRANT

Para asignar un privilegio a un usuario o grupo, se utiliza el comando GRANT, con la siguiente sintáxis:

```
GRANT privilegio [, ...] ON objeto [, ...]
      TO { PUBLIC | nombre_usuario | GROUP nombre_grupo }
```

Donde:

- *privilegio*. Es cualquiera de los privilegios en la tabla 3.1.
- *objeto*. Es el nombre del objeto de la base de datos (tabla, vista o secuencia) al que es permitido el derecho.

- *TO*. Es la palabra reservada que describe a quién se proporciona el privilegio.

Si después de TO se indica la palabra **PUBLIC**, los privilegios son compartidos por todos los usuarios. Especificando un nombre de usuario se asigna el permiso al usuario específico. De la misma manera, si se especifica un grupo, se autorizará el privilegio al grupo específico y todos sus miembros.

REVOKE

Cuando es creado un usuario normal, no tiene ningún privilegio sobre ninguna base de datos que no le pertenezca. El dueño de un objeto o super usuario puede negar explícitamente un derecho cuando éste ha sido garantizado. El comando para negar un privilegio es **REVOKE**, a continuación se muestra la sintaxis:

```
REVOKE privilegio [, ...] ON objeto [, ...]
       FROM { PUBLIC | nombre_usuario | GROUP nombre_grupo }
```

La sintáxis del comando REVOKE es similar al comando GRANT, excepto que el comando sustrae un privilegio, por lo que se utiliza **FROM** en lugar de **TO**. Si se utiliza el grupo especial que es público, al revocar un privilegio sólo afecta a este grupo. Anular el derecho de PUBLIC no afecta a los usuarios a los que se les ha garantizado el privilegio explícitamente.

Definición de usuarios y permisos en el SIIEM/UAP

Propongo la siguiente definición de usuarios y permisos en el SIIEM/UAP:

- *postgres* es un usuario del sistema operativo que a su vez es un super usuario en la base de datos. Debido a los privilegios que tiene este usuario, no es utilizado por los programas cgi para acceder a la base de datos del SIIEM/UAP.
- *medico* es el usuario que tiene el privilegio de crear bases de datos. Este usuario es dueño de la base de datos *medicina* y tiene como función principal la administración de esta base de datos.
- El usuario *cweb* tiene todos los privilegios de lectura y escritura sobre la base de datos *medicina*, a excepción de la tabla *permiso*, en la cual se definen los usuarios a nivel de aplicación que tienen acceso al sistema.

Se hace notar que el usuario *medico* y *cweb* no son usuarios válidos en el sistema operativo, de tal manera que si se desea ingresar a PostgreSQL con estos usuarios, se debe utilizar la opción *-U* del comando *psql*, como sigue:

```
psql [-h EQUIPO_REMOTO] -U {medico|cweb} medicina
```

La opción -h se utiliza cuando la solicitud se realiza de manera remota y con la opción -U se le indica a PostgreSQL con cuál usuario se solicita la conexión.

Vistas para el control de acceso

Debido a que no existe un mecanismo explícito para controlar el acceso a columnas específicas o renglones de una tabla, es posible hacer ésto de manera indirecta a través de las vistas. Al crear una vista sobre una tabla y obligar a los usuarios a acceder a los datos de la tabla a través de la vista se permite que sólo sean seleccionados las columnas o renglones deseados.

Se puede limitar columnas específicas en una vista con el comando SELECT cuando se crea la vista. De esta manera la vista regresará sólo las columnas especificadas. Para limitar registros se utiliza la cláusula WHERE. La vista mostrará sólo aquellos registros que coinciden con esta sentencia.

Como los privilegios de las listas de control de acceso pueden ser aplicados a vistas y tablas, se puede asignar el derecho de lectura a la vista, pero no a la tabla en sí misma. Entonces, los usuarios seleccionarán datos de la vista aunque no tengan derechos para hacerlo de la tabla.

En el SIIEM/UAP se mantiene la tabla **permiso** donde se encuentran los usuarios que pueden acceder al sistema. Esta tabla, entre otros campos, incluye:

- *nombrec*. Nombre corto del usuario.
- *contra*. Contraseña del usuario.
- *nombrel*. Nombre largo del usuario.
- *permite*. Permisos para el acceso a módulos.

Como se puede observar, existen campos como *contra* y *permite* que no deben ser visualizados por otro usuario que no sea *medico*. Para resolver este problema propongo la creación de la vista **usuarios** que restringe el acceso a la tabla *permiso*. La figura 3.4 muestra la creación de la vista.

```
medicina=# CREATE VIEW usuarios
medicina-#              AS SELECT nombrec,nombrel,enlace
medicina-#              FROM permiso where nombrec <>'UAPOS';
CREATE VIEW
```

Figura 3.4: Creación de la vista *usuarios* para el control de acceso a la tabla permiso.

Una vez definida la vista, se asignan permisos de lectura al usuario *cweb*, como se muestra en la figura 3.5.

En la figura 3.6 se puede ver la prueba que se hace de los permisos. Primeramente se hace el cambio al usuario *cweb* y se intenta seleccionar los datos de la tabla *permiso*, lo

```
medicina=# GRANT SELECT ON usuarios TO cweb;
GRANT
medicina=# \z
public | usuarios    | {=,medico=arwdRxt,cweb=r}
```

Figura 3.5: Asignación del permiso de lectura a la vista *usuarios*.

cual es rechazado; posteriormente se realiza una consulta a *usuarios* mostrando los datos de acuerdo a la definición de la vista.

```
medicina=# \c - cweb
You are now connected as new user cweb.

medicina=> select * from permiso;
ERROR:  permiso: permission denied

medicina=> select * from usuarios;
 nombrec  |    nombrel     |  enlace
----------+----------------+----------
 FERNAN   | FERNANDO       | DESAROLL
 GERVI    | GERVACIO       | DESAROLL
 JIMALZJ9 | ALEJANDRO      | MEDICINA
 MAEUGAZP | MARIA EUGENIA  | MEDICINA
 IVONNECA | IVONNE         | MEDICINA
 CARMED5M | CARMEN         | MEDICINA
 IRENEMEN | IRENE          | MEDICINA
(7 rows)
```

Figura 3.6: Consulta a la tabla *permiso* y la vista *usuarios*.

3.3.4. Conexiones TCP/IP seguras

Como se mencionó en el capítulo anterior, la privacidad y la integridad de datos han llegado a ser dos de los tópicos más relevantes en el área de seguridad en cómputo. Es frecuente saber que sistemas en los cuales se ha confiado por mucho tiempo han fallado o que se ha descubierto un nuevo hueco de seguridad.

Sin embargo, el cifrado de datos es cada vez más utilizado por los usuarios de computadoras. Cada sitio que mantiene comercio electrónico usa Secure Socket Layer para proteger los datos de los usuarios mientras transmiten información personal como lo son números de tarjeta de crédito a través de Internet.

Existen protocolos como FTP, Telnet y POP entre otros que transmiten información como nombres de usuario y contraseñas en *texto claro*, en otras palabras, no son cifrados para que viajen en Internet.

La transmisión de datos en texto claro significa que cualquiera usando un *sniffer* puede potencialmente obtener acceso a información personal. No hay mucha diferencia si se traslada esta situación al área de las bases de datos.

La conexión remota a PostgreSQL sin el uso de tecnología de cifrado implica un riesgo potencial. Si un intruso utiliza un *sniffer* en la red, o sobre una red entre el cliente y el servidor de base de datos, puede obtener acceso completo a la información almacenada en PostgreSQL.

PostgreSQL cuenta con tres métodos de cifrado de datos entre el cliente y el servidor:

- SSL interno. El soporte interno de SSL que tiene PostgreSQL es habilitado con la bandera *–with-ssl* al momento de la compilación, permite que *psql* (o cualquier cliente programado para conectarse a PostgreSQL a través de SSL) se conecte de manera segura.
- SSH/OpenSSH. Puede ser utilizada una sesión *SSH*(Secure Shell) para crear un tunel al servidor remoto, proporcionado por el demonio de SSH. Esto requiere acceso al shell del sistema que está ejecutando PostgreSQL para cada usuario que desee conectarse.
- Stunnel. Es una aplicación que crea un tunel cifrado entre un cliente y el servidor. Este método requiere acceso al shell para iniciar, pero puede ser configurado para ser ejecutado sobre un sistema cliente para un usuario que no tiene acceso directo al shell en el servidor remoto.

3.3.5. Monitoreo de la base de datos

Una tarea importante dentro de la administración y seguridad de un servicio es la revisión de lo que está haciendo dicho servicio, esto se logra a través del monitoreo. Existen diferentes métodos para realizar esta tarea, algunos métodos involucran herramientas proporcionadas por el sistema operativo, y otros son a través de la misma base de datos.

Herramientas del sistema operativo

Los sistemas operativos tipo Unix incluyen herramientas para el monitoreo de procesos como **ps** y **top**.

PS

El comando ps (*process status*) muestra una vista actual de los procesos al momento de su ejecución. Este comando es útil en el SIIEM/UAP debido a que se puede verificar si

está en ejecución el servidor de la base de datos. La figura 3.7 muestra la forma de obtener esta información.

```
ps -auxw|grep postgres
postgres   804  0.0  0.2  4892  520 ?        S      2003   0:05
          /home/postgres/bin/postmaster -D /home/postgres/data
```

Figura 3.7: Comando ps para obtener el estado del proceso de postgres

TOP

A diferencia del comando ps, el comando top muestra la actividad en tiempo real del procesador. Este comando muestra las tareas en orden de consumo del CPU, además de que provee una interfaz para la manipulación de procesos. Este comando puede mostrar las tareas ordenadas por uso de CPU, uso de memoria y tiempo de ejecución. La figura 3.8 muestra el estado de los procesos para el usuario postgres.

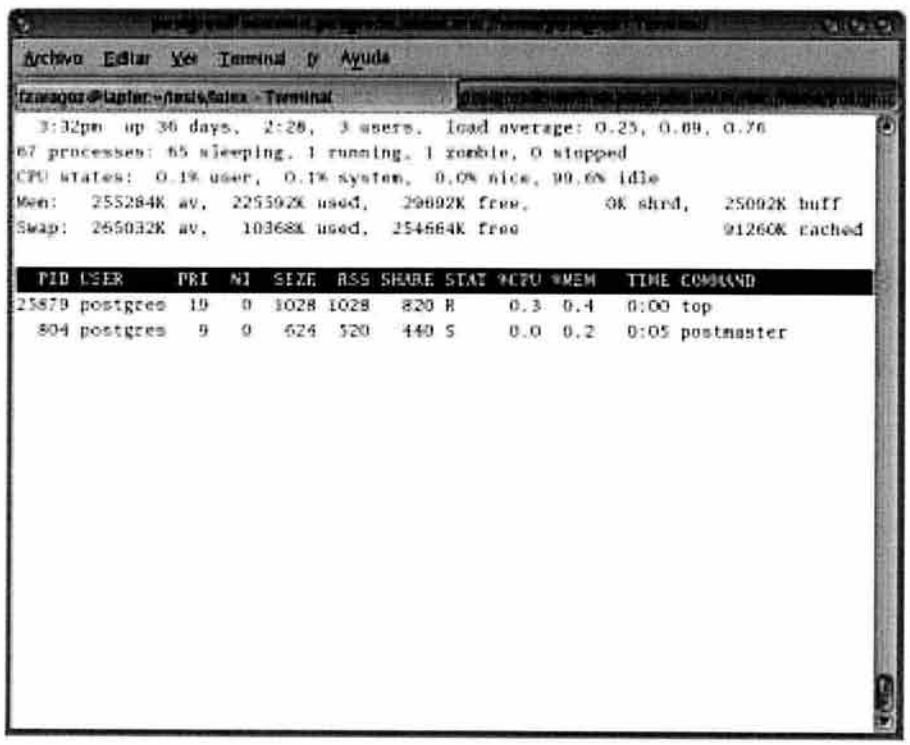

Figura 3.8: Comando top utilizado para monitorear el servidor de la base de datos.

Herramientas de la base de datos

PostgreSQL cuenta con herramientas que son útiles en el monitoreo de la base de datos, en seguida se explican dos de las herramientas más importantes para este propósito.

RECOLECTOR DE ESTADÍSTICAS

El *recolector de estadísticas* de PostgreSQL es un subsistema que tiene la función de obtener y reportar información de la actividad del servidor. Actualmente, el recolector puede contar los accesos a tablas e índices tanto en bloques de disco como en registros individuales. También determina la consulta en ejecución por otros procesos del servidor.

Existen muchas vistas predefinidas para mostrar los resultados de la recolección de estadísticas. También es posible construir vistas usando las funciones existentes para la colección de datos. Cuando se usan las estadísticas para monitorear la actividad actual, es importante que la información no sea actualizada constantemente. Cada proceso del servidor individual transmite nuevos accesos al recolector antes de esperar por otro comando de un cliente, de tal manera que una consulta que está en progreso no afecta los totales desplegados. También, el recolector emite nuevos totales en el período definido en su configuración.

VISTAS DE BLOQUEO

Otra herramienta útil para monitorear la actividad de la base de datos es el catálogo de sistema **pg_locks**. Éste permite al administrador de la base de datos ver información generada por el manejador de bloqueo.

PostgreSQL provee varios métodos para controlar el acceso concurrente a los datos en las tablas. Gran cantidad de comandos de PostgreSQL automáticamente solicitan bloqueo para asegurar que las tablas referenciadas no son eliminadas o modificadas en formas incompatibles mientras el comando se ejecuta. (Por ejemplo, el comando ALTER TABLE no puede ser ejecutado al mismo tiempo con otras operaciones en la misma tabla).

Monitorear la información de bloqueo puede ser utilizado para:

- Ver todos los bloqueos sobre relaciones en bases de datos particulares, en una relación particular o en todos los bloqueos mantenidos por una sesión particular.
- Determinar el efecto de la contención de bloqueo sobre todo el desempeño de la base de datos.

3.3.6. Respaldos

El respaldo y la restauración de datos es un elemento importante en la seguridad de un sistema. Ningún sistema es inmune de la falla en un disco duro, de usuarios descuidados, o de cualquier problema que pueda degenerar los datos almacenados en PostgreSQL. Los procedimientos de respaldos deben ser acentados a través de las políticas y procedimientos[3]. Existen fundamentalmente dos métodos diferentes para el respaldos de la base de datos:

- Respaldos en formato SQL.
- Respaldo a nivel del sistema de archivos.

Respaldos en formato SQL

El procedimiento para realizar los respaldos se explicó detalladamente en el primer capítulo, no obstante, en esta parte retomo la utilería de PostgreSQL utilizada para realizar dichos respaldos.

La idea de realizar respaldos a través del método de vaciado SQL es generar un archivo de texto que, cuando sea recuperado, regenerará la base de datos en el mismo estado que cuando fue creado. PostgreSQL proporciona la utilería **pg_dump** para este propósito. El uso básico de este comando es:

```
pg_dump base_datos > archivo_salida
```

Si no se especifica un archivo de salida, la información será enviada a la salida estándar del sistema, esto es, a la pantalla.

pg_dump es una aplicación cliente de PostgreSQL. Esto significa que es posible hacer un respaldo desde cualquier acceso remoto a la base de datos. De tal manera que no opera con permisos especiales, sólo se debe tener permiso de lectura sobre todas las tablas que se desean respaldar, así que en la práctica, los super usuarios de la base de datos realizan esta tarea.

En seguida se muestran las opciones disponibles para este comando:

- `-a, --data-only`. Sólo toma las sentencias SQL COPY o INSERT (dependiendo si la bandera *-b* está presente). Esto resulta en un respaldo de datos, y no de objetos de la base de datos.
- `-b, --blobs`. Genera un vaciado de objetos grandes también como de datos normales. Esta opción requiere que la bandera *-F* sea pasada con el formato *t* o con *c*. De forma predeterminada, los objetos grandes no son respaldados.

[3]Las políticas de respaldo para el SIIEM/UAP fueron definidas en el primer capítulo.

- `-c, --clean`. Especifica que las sentencias SQL para eliminar los objetos existentes estarán antes de los comandos SQL para crear los objetos. Esta opción es útil para reiniciar una base de datos existente.

- `-C, --create`. Indica que las sentencias SQL para crear la base de datos deben ser incluidas en el respaldo.

- `-d, --inserts`. Causa que las sentencias INSERT sean respaldadas por cada registro de datos, en lugar de las sentencias predefinidas con COPY. Esto puede ser más seguro, debido a que un registro corrupto puede generar una falla en COPY, aunque es más lento el proceso de agregar un solo registro a la vez durante la restauración.

- `-D, --attribute-inserts`. Como la bandera *-d*, provoca que las sentencias INSERT sean resguardadas; sin embargo, con la bandera *-D*, cada instrucción INSERT es creada con una lista explícita entre paréntesis de las columnas, precediendo a los valores de esa lista.

- `-f nombre_archivo, --file=nombre_archivo`. En lugar de enviar la salida del comando *pg_dump* a la pantalla, lo envia a *nombre_archivo*.

- `-F {c|t|p}, --format {c|t|p}`. Determina el formato del archivo de salida:

  - *c (comprimido gzip)*. Crea un archivo comprimido (gzip) y empaquetado (tar).
  - *t (tar)*. Se crea un archivo con formato del comando tar.
  - *p (texto plano)*. Este es el formato predeterminado el cual genera un archivo en texto plano.

- `-h nombre_equipo, --host=nombre_equipo`. Especifica el *nombre_equipo* al cual debe conectarse, en lugar de hacerlo localmente. Se utiliza este comando cuando la base de datos se encuentra en otro servidor.

- `-i, --ignore-version`. Verifica la versión de *pg_dump* y la versión que se está ejecutando de PostgreSQL.

- `-n, --no-quotes`. Suprime cualquier identificador de doble comilla (").

- `-N, --quotes`. Indica explícitamente que debe ser utilizado el caracter doble comilla(") al rededor de los identificadores.

- `-o, --oid`. Causa que los identificadores de objetos (oid *object identifiers*) para cada registro de datos sean respaldados también. Esto puede ser de utilidad si alguna aplicación está basada en el uso de OID.

- `-O, --no-owner`. Hace que el dueño no sea tomado en cuenta al realizar el vaciado. De tal manera que los objetos serán propiedad del usuario que restaure la base de datos.

- `-p PUERTO, --port=PUERTO`. Especifica que la conexión a la base de datos deberá ser realizado a través del puerto *PUERTO*, en lugar del puerto predeterminado (5432).

- `-R, --no-reconect`. Suprime cualquier sentencia **\connect**, las cuales son utilizadas para reforzar la propiedad cuando el respaldo es restaurado. Es similar en la práctica a la opción *-O*.

- `-s, --schema-only`. Causa que el esquema sea almacenado como sentencias SQL (objetos de la base de datos como tablas, secuencias, índices y vistas), ignora la recreación de datos. Esto puede ser útil para mover la estructura general de una base de datos de una máquina de desarrollo a una de producción.

- `-t TABLA, --table=TABLA`. Se respalda una tabla específica.

- `-u, --password`. Proporciona un mecanismo para solicitar un usuario y contraseña.

- `-v, --verbose`. Causa una salida detallada de las funciones que *pg_dump* está realizando.

- `-x, --no-acl`. Suprime cualquier sentencia GRANT o REVOKE, las que son utilizadas para preservar los derechos al realizar el vaciado. Se utiliza esta opción cuando no se desea mantener los derechos en la recreación de la base de datos.

- `-Z, --compress {0-9}`. Realiza una grado de compresión entre 0 y 9 cuando se utiliza la opción *-F c*.

pg_dumpall

PostgreSQL proporciona otro comando alterno a *pg_dump* llamado **pg_dumpall**. El uso principal de esta aplicación es permitir un respaldo de todas las bases de datos en el sistema al mismo tiempo, en lugar de tener que ejecutar *pg_dump* para cada base de datos.

La sintáxis de *pg_dumpall* es la siguiente:

```
pg_dumpall [ -c ] [ -h EQUIPO_REMOTO ] [ -p PUERTO ] [ -g ]
```

Respaldos a nivel del sistema de archivos

Otra forma de realizar los respaldos, aunque se recomienda la vista anteriormente, es mediante la obtención de los datos del sistema de archivos. En este caso se pueden utilizar programas que proporciona el mismo sistema operativo como tar, dump, zip, cpio, etc.

PostgreSQL instala las bases de datos en un directorio que se llama *data* de forma predeterminada. Para respaldar este directorio utilizando la herramienta tar sería de la

siguiente forma:

```
tar zcf ddmmmaaaaMedicina.tgz data
```

De esta forma se crea el archivo comprimido con extensión .tgz del directorio donde se encuentran los archivos de base de datos.

Restauración de la base de datos en formato SQL

Los datos de un respaldo son restaurados a una base de datos de dos formas distintas. Si el respaldo fue creado con *pg_dump*, en texto plano, puede ser pasado al comando *psql* directamente como un archivo de entrada. De otra forma, si el formato de salida elegido fue otro, se debe utilizar el comando *pg_restore*.

Un archivo de texto plano creado por *pg_dump* es pasado como archivo de entrada a *psql*. Éste ejecuta cada instrucción secuencialmente. Dependiendo de cómo haya sido creado el respaldo, y para qué propósito, existen diferentes formas de invocar el comando.

En el SIIEM/UAP, debido al procedimiento de respaldo definido en las políticas para este propósito, el procedimiento para restaurar la base de datos se describe a continuación:

1. Obtener los archivos de respaldo y firma digital, de acuerdo a la fecha y hora deseados, preferentemente los más recientes. Ya sea de una tabla específica o de toda la base de datos.

```
24Mzo2004medicina.sql.gpg  Archivo cifrado
24Mzo2004medicina.sql.md5  Archivo con  la firma digital
                        del archivo 24Mzo2004medicina.sql
```

2. Descifrar el archivo de respaldo. Al descifrar el archivo con extensión .gpg, se obtiene el archivo de sentencias sql generado por *pg_dump*.

```
gpg 24Mzo2004medicina.sql.gpg
gpg: Warning: using insecure memory!

You need a passphrase to unlock the secret key for
user: "BD Admin de medicina <postgres@newman.posgrado.unam.mx>"
1024-bit ELG-E key, ID AA528DCA, created 2002-10-22 (main key
ID 08621DA2)

gpg: encrypted with 1024-bit ELG-E key, ID AA528DCA, created
     2002-10-22 "BD Admin de medicina
     <postgres@newman.posgrado.unam.mx>"
```

3. Obtener la firma digital del archivo de respaldo.

```
md5sum 24Mzo2004medicina.sql
6694f81b83793d0608f45ae8f0635577  24Mzo2004medicina.sql
```

4. Verificar la firma con la firma almacenada en el archivo .md5 que corresponde al respaldo.

```
cat 24Mzo2004medicina.sql.md5
6694f81b83793d0608f45ae8f0635577  24Mzo2004medicina.sql
```

5. Restaurar el respaldo el la base de datos con el comando *psql*

```
psql -f 24Mzo2004medicina.sql medicina
```

Restauración de la base de datos a nivel del sistema de archivos

Restablecer la base de datos de este modo dependerá también del comando o herramienta utilizada para generar el respaldo. Siguiendo el ejemplo anterior, si se utilizó el comando tar para crear el respaldo, para recuperarlo se utilizará el mismo comando pero cambiando la opción `c` que crea, por la opción `x` que extrae lo que se encuentra en el archivo .tgz. El comando necesario para realizar este proceso se muestra en seguida:

```
tar zxf ddmmmaaaaMedicina.tgz
```

Con el comando anterior se regenera el directorio *data* donde se almacenan los archivos de la base de datos. De esta manera se restaura íntegramente el respaldo de la base de datos de medicina del SIIEM/UAP.

Con la restauración de la base de datos del SIIEM/UAP finalizo con el procedimiento que mantiene en condiciones aceptables de seguridad esta parte del sistema.

En este punto puedo concluir que si siendo las bases de datos el método mediante el cual se almacenan los datos de un sistema de información de una organización, en la cual es posible almancenar grandes cantidades de datos que son de relevancia y que en muchos casos es lo más importante, entonces deben implementarse las medidas necesarias para asegurar la integridad y confiabilidad de la base de datos.

Como se ha podido ver a lo largo de este capítulo, la seguridad en la base de datos es un elemento importante que debe ser considerado en cualquier sistema, no importando la magnitud de éste. Para realizar esto el administrador de la base de datos debe auxiliarse

de las herramientas que los mismos DBMS proporcionan para realizar esta tarea. Actualmente es difícil encontrar un manejador de base de datos que no cuente con herramientas propias de monitoreo, respaldo y administración de usuarios.

Una vez que se he abarcado la seguridad desde la creación de políticas, definiendo medidas para asegurar el equipo de cómputo, el servidor web y la base de datos, en el siguiente capítulo doy una explicación más detallada de las herramientas que se utilizan en el SIIEM/UAP que ayudan en la tarea de la seguridad.

Capítulo 4

Herramientas de software libre y la seguridad en el SIIEM/UAP.

En este capítulo se abordará el Software Libre y cómo fue utilizado en el Sistema de Inscripciones por Internet de Especializaciones de Medicina de la Unidad de Administración del Posgrado. Se consideran los elementos más importantes del software libre y la filosofía que está detrás de él, además de que se verá el uso de este tipo de programas sobre un sistema en producción.

Se abordarán los siguientes temas:

- *Introducción al Software Libre.*
- *Linux como sistema operativo.*
- *Herramientas libres de seguridad utilizadas en el SIIEM/UAP.*

4.1. Introducción al Software Libre

COMIENZOS DEL SOFTWARE LIBRE

Mientras trabajaba en el laboratorio de inteligencia artificial en el Instituto Tecnológico de Massachussets, Richard Stallman pertenecía a una comunidad donde interactuaban con un sistema operativo denominado Sistema incompatible de tiempo compartido (ITS por sus siglas en inglés). Compatir el software era una actividad natural, de tal manera que si alguien en otra universidad se interesaba en algún programa realizado por la comunidad de hackers en el laboratorio, se le entregaba el código fuente de dicho programa.

Aunque estos programas no eran considerados como software libre, por que no existía tal concepción, en realidad si lo eran, debido a la naturaleza en su desarrollo y forma de compartirlo.

Sin embargo, se suscitaron ciertos hechos que hicieron que la comunidad de hackers en el laboratorio de Iteligencia Artificial se desintegrara y por ende el trabajo que venían realizando:

- En 1980, debido a las características de nuevas computadoras, la compañía Digital decide descontinuar la serie PDP-10, esto significó que los programas creados para el ITS se volvieran obsoletos.
- La compañía Symbolics, en 1981, contrató a la mayoría de los hackers en el laboratorio.
- Cuando en 1982 el laboratorio adquiere una nueva PDP-10, los administradores deciden instalar el sistema operativo no libre proporcionado por Digital.
- Las computadoras de ese tiempo tenían sus propios sistemas operativos para lo cual los usuarios debían firmar un acuerdo de no revelación, de tal suerte que era imposible compatir con otros usuarios el software, de otra manera, se incurría en piratería según los propietarios del software.

Después de ver el desvanecimiento de su comunidad, y de haber liberado una batalla interna, Richard Stallman decide iniciar un proyecto en el cual se pudiera generar nuevamente una comunidad de programadores en el desarrollo de software libre. Richard concluyó que si realizaba un programa interesante, éste podría llamar la atención y decidió iniciar con un sistema operativo libre con las características de UNIX. A este proyecto le asignó el nombre **GNU** (*GNU's Not Unix - GNU no es Unix*), siguiendo una costumbre hacker de realizar acrónimos recursivos.

El software libre nació formalmente en 1984 como una iniciativa de Richard Stallman de compartir el código fuente de los programas de computadoras a la comunidad de programadores que se pudieran interesar en él. Con la gran ventaja de que pudiera ser modificado y adaptado a necesidades específicas.

Debido a la magnitud del proyecto, los primeros programas desarrollados fueron un compilador de lenguaje C, conocido como GCC y el editor de textos Emacs. A medida que el uso de Emacs se iba extendiendo, Richard se vió en el problema de la distribución del programa a los interesados que no tuvieran acceso a través del FTP del equipo en el laboratorio, asi que decidió, debido a que no tenía un trabajo fijo, pedir una cooperación por el envío del programa mediante correo tradicional, asi inició un negocio en la venta y distribución de software libre, como un antecedente a los negocios que actualmente se relacionan a las distribuciones de software GNU.

Fundación para el Software Libre

Conforme se incrementó el interés de programadores en el desarrollo de software fundado en las bases GNU, Richard y otros vieron la necesidad de atraer fondos para mantener la programación de herramientas, fue asi como crearon la Fundación para el Software Libre (*Free Software Fundation - FSF*). Esta organización no lucrativa se encargó de la venta y distribución de Emacs en cintas, y posteriormente agregaron más software GNU, además de la venta de manuales libres.

Las aportaciones que han realizado los empleados de la Fundación han sido muy importantes, donde destacan el desarrollo de bibliotecas C para la comunicación de los programas GNU/Linux con Linux; y el shell Bash utilizado como interfaz de usuario con el sistema operativo.

Licencia GPL y LGPL

Una manera alternativa de software para que sea usado por cualquier persona es liberarlo como software de dominio público, sin embargo ésto trae un problema consigo. Poner un programa como de dominio público permite que cualquier persona pueda utilizarlo y modificarlo para generar un programa propietario, quitándole la libertad que gozaba el usuario inicialmente.

El objetivo del proyecto GNU es dar la libertad de uso, distribución y modificación del software a todos los usuarios. Así que en lugar de liberar el software como dominio público, se protege a través del *copyleft*:

> "Copyleft dice que cualquiera que redistribuye el software, con o sin cambios, debe dar la libertad de copiarlo y modificarlo más. Copyleft garantiza que cada usuario tiene libertad." [1]

Para que un programa sea protegido con el *copyleft*, el procedimiento a seguir es:

1. Se reservan los derechos.
2. Se añaden términos de distribución los cuales son un instrumento legal que le dan a todo el mundo los derechos de utilizar, modificar, y redistribuir el código del pro-

[1] http://www.gnu.org/copyleft/copyleft.es.html#TOCWhatIsCopyleft

grama o cualquier programa derivado del mismo, pero sólo si los términos de distribución no son cambiados.

Debido a que *copyleft* es un concepto muy general, existen diferentes licencias que abordan con mayor detalle los objetivos de éste. La Licencia Pública General GNU (*GNU GPL - GNU General Public License*) define los términos específicos de distribución del software. Y la Licencia Pública General para Bibliotecas GNU (LGPL GNU) se aplica a algunas bibliotecas GNU.

El Anexo 1 de este trabajo contiene una traducción al idioma español de la versión original de la GNU GPL.

Categorías de Software

Generalmente existe una confusión entre los términos utilizados para definir los distintos tipos de software de acuerdo a los derechos que se tienen sobre él. A continuación se definen estos tipos de software tanto libre como no libre[2]:

- Software libre. El software libre es software que viene con autorización para que cualquiera pueda usarlo, copiarlo y distribuirlo, ya sea literal o con modificaciones, gratis o mediante una gratificación. En particular, esto significa que el código fuente debe estar disponible. "Si no es fuente, no es software".

  Si un programa es libre, entonces puede ser potencialmente incluido en un sistema operativo libre tal como GNU, o sistemas GNU/Linux libres.

  Hay muchas maneras diferentes de hacer un programa libre. Algunas de las posibles variaciones son descritas abajo.

  El software libre es frecuentemente más fiable que el software no libre.

- Software de Fuente Abierta. El término software de "fuente abierta" (*open source*) es usado por algunas personas para dar a entender más o menos lo mismo que software libre. Sin embargo lo correcto es utilizar "software libre".

- Software de dominio público. El software de dominio público es software que no está protegido con copyright. Es un caso especial de software libre no protegido con copyleft, que significa que algunas copias o versiones modificadas no pueden ser libres completamente.

  Algunas veces la gente utiliza el término "dominio público" de una manera imprecisa para decir "libre" o "disponible gratuitamente." Sin embargo, "dominio público" es un término legal y significa de manera precisa "sin copyright". Por claridad,

[2]http://www.gnu.org.

se recomienda el uso de "dominio público" para ese significado solamente y el uso de otros términos para transmitir los otros significados.

- Software protegido con copyleft. El software protegido con copyleft es software libre cuyos términos de distribución no permiten a los redistribuidores agregar ninguna restricción adicional cuando éstos redistribuyen o modifican el software. Esto significa que cada copia del software, aún si ha sido modificado, debe ser software libre.

  En el Proyecto GNU, se protege mediante copyleft casi todo el software que se escribe, porque el objetivo es dar a cada usuario las libertades que el término "software libre" implica.

  Copyleft es un concepto general; para proteger actualmente un programa con copyleft, necesita usar un conjunto específico de términos de distribución. Hay muchas maneras posibles de escribir términos copyleft de distribución.

- Software libre no protegido con copyleft. El software libre no protegido con copyleft viene desde el autor con autorización para redistribuir y modificar así como para añadirle restricciones adicionales.

  Si un programa es libre pero no protegido con copyleft, entonces algunas copias o versiones modificadas pueden no ser libres completamente. Una compañía de software puede compilar el programa, con o sin modificaciones, y distribuir el archivo ejecutable como un producto propietario de software.

  El Sistema X Window ilustra esto. El Consorcio X liberó X11 con términos de distribución que lo hacen software libre no protegido con copyleft. Es posible obtener una copia que tenga esos términos de distribución y es libre. Sin embargo, hay versiones no libres también, y hay estaciones de trabajo populares y tarjetas gráficas para PC para las cuales las versiones no libres son las únicas que funcionan. Si se está usando este hardware, X11 no es software libre en estas circunstancias.

- Software abarcado por GPL. La GPL (General Public License/Licencia Pública General) y GNU es un conjunto específico de términos de distribución para proteger con copyleft a un programa. El Proyecto GNU la utiliza como los términos de distribución para la mayoría del software GNU.

- Software GNU. Software GNU es software liberado bajo el auspicio del Proyecto GNU. La mayoría del software GNU está protegido con copyleft, pero no todos; sin embargo, todo el software GNU debe ser software libre.

  Algo de software GNU es escrito por el personal de la Fundación para el Software Libre, pero la mayoría del software GNU es aportado por voluntarios. Parte del soft-

ware aportado está protegido con copyright por la Fundación para el Software Libre; otra parte está protegido con copyright por los aportadores que los escribieron.

- Software semilibre. El software semilibre es software que no es libre, pero viene con autorización para particulares de usar, copiar, distribuir y modificar (incluyendo la distribución de versiones modificadas) sin fines de lucro. PGP es un ejemplo de un programa semilibre.

  El software semilibre es mucho mejor que el software propietario, pero aún plantea problemas y no es posible usarlo en un sistema operativo libre.

  Las restricciones del copyleft están diseñadas para proteger las libertades esenciales de todos los usuarios. La única justificación, según Stallman, para cualquier restricción sustantiva acerca del uso de un programa es prevenir la adición de restricciones por parte de otras personas. Los programas semilibres tienen restricciones adicionales, motivados por fines puramente egoístas.

  Es imposible incluir software semilibre en un sistema operativo libre. Esto obedece a que los términos de distribución para el sistema operativo libre como un todo es la conjunción de los términos de distribución de todos los programas en él. Agregando un programa semilibre haría al sistema como un todo sólo semilibre. Existen dos razones por las que no se desea esto:

  - El software libre debe ser para todos, incluyendo empresas, no solamente para escuelas o para quienes lo tomen como pasatiempo. Se invita a las empresas a usar el sistema GNU completo y por lo tanto no se debe incluir un programa semilibre en él.
  - La distribución comercial de sistemas operativos libres, incluyendo sistemas GNU basados en Linux, es muy importante y los usuarios aprecian poder comprar distribuciones comerciales en CD-ROM. Incluyendo un programa semilibre en un sistema operativo limitaría su distribución comercial en CD-ROM.

  La Fundación para el Software Libre no es comercial y por lo tanto estaría habilitada legalmente para usar un programa semilibre "internamente". Pero no lo hace, porque esto minaría los esfuerzos para obtener un programa que se pudiera incluir en GNU.

- Software propietario. El software propietario es software que no es libre ni semilibre. Su uso, redistribución o modificación está prohibida, o requiere solicitud de autorización o está tan restringida que no pueda hacerla libre de un modo efectivo.

  La Fundación para el Software Libre sigue la regla de no instalar ningún programa propietario en sus computadoras excepto temporalmente para el propósito específico de escribir un reemplazo libre para ese programa. Con excepción de lo anterior,

no hay excusa posible para instalar un programa propietario.

Por ejemplo, se justifica Stallman al haber instalado Unix en sus computadoras en los 80's, debido a que las utilizaban para escribir el reemplazo libre para Unix. En la actualidad están disponibles sistemas operativos libres, y se ha eliminado todos los sistemas operativos no libres.

- Freeware. El término "freeware" no tiene una definición clara aceptada, pero es usada comúnmente para paquetes que permiten la redistribución pero no la modificación (y su código fuente no está disponible). Estos paquetes no son software libre, por lo que no se debe usar "freeware" para referirse al software libre.
- Shareware. El shareware es software que viene con autorización para la gente de redistribuir copias, pero dice que quien continúe haciendo uso de una copia deberá pagar un cargo por licencia.

  El shareware no es software libre, ni siquiera semilibre. Existen dos razones por las que no lo es:

  - Para la mayoría del shareware, el código fuente no está disponible; de esta manera, no es posible modificar el programa en absoluto.
  - El shareware no viene con autorización para hacer una copia e instalarlo sin pagar una cantidad por licencia, ni aún para particulares involucrados en actividades sin ánimo de lucro. (En la práctica, la gente a menudo hace caso omiso a los términos de distribución y lo hace de todas formas, pero los términos no lo permiten.)
- Software Comercial. El software comercial es software que está siendo desarrollado por una entidad que tiene la intención de hacer dinero del uso del software. No confundir "comercial" y "propietario". La mayoría del software comercial es propietario, pero hay software libre comercial y hay software no libre no comercial.

En la siguiente sección se analiza Linux como uno de los programas más representativos e importantes del software libre.

4.2. Linux

Linux es el kernel del sistema operativo desarrollado por Linus Torvalds en la Universidad de Helsinky. Torvalds, partiendo de las bases del sistema experimental conocido como Minix, se interesó en la tarea de crear un sistema operativo con las características del conocido sistema operativo Unix, esto es, básicamente, un sistema multiusuario y multitarea.

Fue a principios de los 90 que inició la programación del sistema, en octubre de 1991 anuncia la versión 0.02 cuando el sistema podía realizar algunas funciones, pero fue hasta diciembre de 1993 cuando se libera la versión 1.0 de Linux con mayor funcionalidad.

Con ayuda de hackers de todo el mundo y bajo las características de un sistema libre, Linux se ha convertido en un sistema operativo utilizado en proyectos de investigación, gobierno y de la iniciativa privada.

4.2.1. Linux y el GNU

Actualmente muchos usuarios de computadoras utilizan alguna versión de software GNU. Sin embargo, debido a ciertas circunstancias, a este conjunto de software se le nombra *Linux* sin considerar la conexión que existe con el proyecto GNU. Estos usuarios frecuentemente piensan que Linus Torvalds desarrolló el sistema operativo completo en 1991 con un poco de ayuda.

Lo correcto es que Linux es el kernel: el programa en el sistema que asigna los recursos de la computadora a otros programas que se ejecutan en la máquina. El kernel es una parte esencial en el sistema, pero por si sólo es inútil; éste funciona sólo en el contexto de un sistema operativo completo. Linux es utilizado en combinación con software GNU: el sistema completo es básicamente GNU, con la funcionalidad de Linux como su kernel.

Muchas personas creen que una vez que Torvalds terminó de escribir Linux, el kernel, sus usuarios buscaron por otro software para trabajar con él, y encontraron que (sin ninguna razón en particular) mucho de lo necesario para hacer un sistema operativo tipo Unix estaba listo para ser utilizado.

Lo que encontraron no fue resultado de la casualidad, fue el sistema GNU. El software libre disponible era un sistema completo debido a que se venía trabajando desde 1984. El Manifiesto GNU se había fijado la meta de desarrollar un sistema operativo libre tipo Unix, llamado GNU. El anuncio inicial del proyecto GNU también indicaba algunos de los planes originales para el sistema. Al tiempo que Linux estaba siendo escrito, el sistema GNU estaba concluido en gran parte.

Muchos proyectos de software libre tienen el objetivo de desarrollar un programa particular para realizar un trabajo específico. Por ejemplo:

- Linus Torvalds desarrolló el kernel tipo Unix conocido como Linux.
- Donald Knuth escribió un formateador de texto[3] (TEX).
- Bob Scheifler implementó un sistema de ventanas (el sistema X Windows).

De tal manera que es natural medir cada aportación al proyecto total. Si se mide la contribución del proyecto GNU de esta manera se puede ver lo siguiente. Un vendedor de CD-ROM encuentra que en su distribución de Linux, el software GNU es el más grande aporte, cerca de la tercera parte del total, y éste incluye algunos de los componentes esenciales sin los cuales el sistema no funcionaría como lo hace. Linux en sí mismo es el 3 %. Asi que si se elige un nombre en base a quien escribió más programas en el sistema, el más apropiado es "GNU". Sin embargo, apunta Stallman, el proyecto GNU no surgió con el objetivo de crear programas específicos, sino desarrollar un sistema operativo libre tipo Unix.

A principios de los 90, el GNU inició el trabajo sobre un kernel, el GNU Hurd, sin embargo el desarrollo ha sido un trabajo arduo. El GNU Hurd ha sido confiable hasta el 2001, sin embargo se trabaja constantemente en nuevas liberaciones del mismo con el sistema GNU.

Afortunadamente, los usuarios no deben esperar hasta la liberación de Hurd debido a que Linux está listo. Cuando Torvalds escribió Linux, llenó un hueco importante. Asi, la gente podía unir Linux con el sistema GNU para obtener un sistema operativo libre completo. Una versión del sistema GNU basada en Linux; el sistema **GNU/Linux** para abreviar.

La simbiosis suena sencilla, sin embargo no fue asi. Algunos componentes GNU requirieron cambios sustanciales para trabajar con Linux. Integrar un sistema completo como una distribución que pudiera funcionar "fuera de la caja" fue un trabajo muy grande también. Esto necesitaba la forma de cómo instalar e iniciar el sistema, un problema que no fue considerado antes. La gente quien desarrolló los distintos sistemas de distribución ayudaron en gran medida a resolver este problema.

4.2.2. El GNU en el SIIEM/UAP

Todas las bondades y la filosofía de desarrollo del GNU/Linux han motivado la decisión de utilizar este tipo de software en la Unidad de Administración de Posgrado, siendo asi que de los 4 servidores utilizados en la Unidad de Cómputo, dos de ellos ejecutan alguna distribución de Linux en computadoras personales. El uso del software libre en el desarrollo de sistemas de la UAP es de la siguiente manera:

- Sistema Operativo: Linux. www.linux.org y www.kernel.org

[3] LATEX fue utilizado para realizar este trabajo de tesis.

- Bases de datos: Postgres www.postgresql.org.
- Servidor Web: Apache www.apache.org.
- Editores: Emacs y GNU vi.
- Comunicación remota: OpenSSH www.openssh.org.
- Lenguaje para CGI: Perl www.perl.org.
- Compilador C: gcc
- Shell: Bourne Again Shell **bash**.
- Herramientas de seguridad: Varias.

Las principales ventajas que proporciona el uso de software libre a la UAP, y a la Universidad Nacional Autónoma de Mexico en general, es la adaptación del software libre a las necesidades de la Institución, libre de restricciones por estas actividades; y la reducción de costos ya que no es necesario pagar licencias de uso, como en el caso de otros sistemas operativos como Windows o DBMS como Sybase u Oracle.

En la siguiente sección describo cada una de las herramientas utilizadas en el SIIEM/UAP.

4.3. Herramientas de seguridad utilizadas en el SIIEM/UAP

Una vez que se han visto los orígenes y características del software libre, en lo que resta de este capítulo muestro de manera sencilla el uso de herramientas de seguridad que ayuden a reducir los riesgos de amenazas concretas y que refuercen la confianza en el sistema.

Por cada herramienta se abordará de manera breve:

- Introducción.
- Características de seguridad que proporciona.
- Proceso de instalación.
- Uso.
- Resultados.

Esta tesis no pretende ser un tutorial de cada herramienta, para mayores detalles de instalación y uso se recomienda consultar la información en línea y/o los archivos README e INSTALL que se proporcionan en la distribución de cada herramienta.

La presentación de las herramientas se hace mediante una clasificación de acuerdo al mecanismo de seguridad que proporciona. Es importante considerar que estas no son todas las herramientas existentes. Las herramientas que se muestran a continuación son aquellas que han sido probradas por miles de usuarios a nivel mundial y que gozan de una amplia popularidad, sin embargo, esto no es garantía de que inclusive las mismas herramientas fallen (el 17 de Abril del 2003 se descubrió una error en el detector de intrusos SNORT que permitía a un intruso ejecutar código con privilegios de administrador [4]) o permitan ejecutar actividades inesperadas (el 16 de Septiembre del 2003 OpenSSH presentó un hueco de seguridad en las versiones anteriores a la 3.7.1 que permitía ejecutar comandos con privilegios de administrador y causar una negación de servicio [5]). También las herramientas deben ser actualizadas constantemente para utilizar mejoras o resolver problemas que puedan surgir en las mismas.

4.3.1. Firewall

Haciendo una analogía con las paredes que se construyen en un edificio para aislarlo del fuego en caso de un incendio, un firewall en el área de seguridad en cómputo es un dispositivo o programa con el propósito de mantener una red privada aislada de las hostilidades existentes en Internet. Debido a su importancia, se considera al firewall como la primera defensa de seguridad en una red por lo que es indispensable tener un firewall configurado en cualquier equipo conectado a Internet.

El primer firewall de computadoras fue un equipo Unix sin ruteo con conexiones a dos redes diferentes. Una tarjeta de red estaba conectada a Internet y la otra a la red LAN privada. Para acceder a Internet desde la red privada era necesario ingresar al servidor (unix) que funcionaba como firewall. De esta forma se usaban los recursos del sistema para acceder a Internet.

En la actualidad los firewalls está encaminados a reforzar las políticas de una organización. Los firewalls tienen dos propósitos fundamentales:

1. Mantener a los intrusos (hackers, crackers, gusanos, etc.) fuera de la red privada.
2. Mantener a los usuarios (empleados, estudiantes, etc.) protegidos dentro de la red privada.

Tipos de firewalls

Existen distintos tipos de firewalls, sin embargo, todos ellos caen dentro de dos categorías:

1. Filtrado de paquetes.
2. Proxys.

[4] Ver boletín de seguridad UNAM-CERT 2003-014 http//www.unam-cert.unam.mx
[5] Ver boletín de seguridad UNAM-CERT 2003-024 http//www.unam-cert.unam.mx

A continuación se describen cada uno de ellos.

Firewalls basados en filtrado de paquetes.

Un firewall de filtrado funciona en el nivel de red. Los datos pueden salir del sistema si las reglas del firewall lo permiten. Cuando un paquete llega son filtrados por su tipo, dirección fuente, dirección destino y puerto, esta información está contenida en cada paquete. Este tipo de firewall está construido en el kernel de Linux.

Muchos ruteadores de red tienen la habilidad de desarrollar algunos servicios de firewall. De esta forma, los firewalls basados en el filtrado de paquetes pueden ser vistos como un tipo de ruteador. Debido a ésto es importante entender perfectamente la estructura de un paquete IP para trabajar con él.

Este tipo de firewalls son muy rápidos debido a que son muy pocos los datos que son analizados y registrados, asi que no afectan el desempeño de la red.

Una desventaja, si se desea ver de esa manera, es que el filtrado de paquetes no provee mecanismos de autenticación. Los usuarios no pueden ser identificados. La única identidad que un usuario tiene es la dirección IP asignada a su computadora.

Los firewalls que utilizan filtrado de paquetes son transparentes para el usuario debido a que no tiene que configurar las reglas en su aplicación para usar Internet.

El CERT/CC ha definido algunos servicios que deberían ser filtrados con un firewall[6] (ver tabla 4.1).

Servicio	Puerto	Protocolo
Transferencia de zonas (DNS)	53	TCP
Tftpd	69	UDP
Link	87	TCP
SunRPC y NFS	111 y 2049	UDP y TCP
Comandos "r" de Unix	512, 513 y 514	TCP
Lpd	515	TCP
uucdp	540	TCP
OpenWindows	2000	UCP y TCP
X windows	6000+	UDP y TCP

Cuadro 4.1: Servicios que deben ser filtrados con un firewall.

Por ejemplo, el CERT/CC sugiere que los sitios, en caso de utilizarlo, filtren el puerto 53/TCP para prevenir la transferencia de zonas del sistema de nombres de dominio. Se

[6]http://www.cert.org/tech_tips/packet_filtering.html

debe permitir el acceso al puerto 53/TCP sólo a los servidores DNS secundarios. Con esto se previene que un intruso obtenga información adicional acerca de los sistemas conectados a la red de área local.

Proxy

A diferencia del firewall de filtrado de paquetes, los servidores proxy son utilizados para controlar o monitorear el tráfico de salida. Algunas aplicaciones proxy almacenan temporalmente los datos solicitados. Existen dos tipos de servidores proxy:

1. Proxy de aplicación.
2. Proxy tipo SOCKS.

Proxy de aplicación

Este tipo de firewall es el más comúnes en su categoría, también se le conoce como firewall de filtrado de aplicación. Cuando una aplicación desea conectarse a Internet, el cliente envía primero al proxy la solicitud. El servidor proxy se encarga de conectarse al servidor con el cual el cliente desea comunicarse y el servidor proxy entonces regresa los datos al cliente.

Debido a que el servidor proxy controla todas las comunicaciones, ellos pueden registrar todo lo que los clientes hagan. Este tipo de firewall depende del servicio, de tal manera que hay servidores proxy para Web (HTTP), servidores proxy para transferencia de archivos (FTP), por lo tanto el proxy de Web registrará todos los URL solicitados, el proxy de FTP registrará todos los archivos transferidos. Estos servidores también pueden ver el contenido de los paquetes, mas que direcciones y puertos fuente/destino, de tal suerte que pueden filtrar palabras "inapropiadas" desde los sitios que se visitan o puede buscar virus en la información que se transfiere.

Proxy tipo SOCKS

Un servidor de SOCKS es muy similar a los switchs. Simplemente realizan una conexión a través del sistema a otro sistema externo. Muchos servidores de este tipo sólo funcionan con conexiones tipo TCP. Y como los firewall de filtrado de paquetes, no proveen autenticación de usuarios, sin embargo, los SOCKS pueden registrar hacia donde se conectan los usuarios.

Iptables

Aunque el servidor que aloja al SIIEM/UAP no tiene el objetivo de filtrar los paquetes que viajan de Internet hacia la red de la Unidad de Administración de Posgrado, hice uso de iptables, el firewall que tiene integrado el kernel de Linux, para filtrar los paquetes que viajan hacia y desde este servidor en particular.

Iptables es utilizado para configurar, mantener e inspeccionar la tabla de reglas de filtrado de paquetes IP en el kernel de Linux. Se pueden definir muchas tablas diferentes. Cada tabla contiene un número de cadenas internas y también pueden contener cadenas definidas por el usuario. La sintáxis de IP tables se muestra a continuación:

```
iptables -[ADC] cadena especificacion-regla [opciones]
iptables -I cadena [numregla] especificacion-regla [opciones]
iptables -R cadena numregla especificacion-regla [opciones]
iptables -D cadena numregla [opciones]
iptables -[LFZ] [cadena] [opciones]
iptables -N cadena
iptables -X [cadena]
iptables -P cadena objetivo [opciones]
iptables -E nombre-cadena-anterior nombre-cadena-nueva
```

Cada cadena es una listra de reglas las cuales pueden coincidir con un conjunto de paquetes. Cada regla especifica lo que se debe hacer cuando un paquete coincide. A esto se le llama un 'objetivo', el cual puede ser un salto hacia una cadena definida por el usuario en la misma tabla.

La regla de un firewall especifica el criterio para un paquete y un objetivo. Si el paquete no coincide, se examina la siguiente regla en la cadena; si ésta coincide, la siguiente regla es especificada por el valor del objetivo, la cual puede ser una cadena definida por el usuario o alguno de los valores especiales ACCEPT, DROP, QUEUE o RETURN.

ACCEPT permite que el paquete pase. **DROP** significa arrojar el paquete. **QUEUE** pasa el paquete a un espacio de usuario, si es que esta característica está soportada por el kernel. **RETURN** detiene el paso de esta cadena y continúa en la siguiente regla en la llamada de la cadena previa.

Existen tres tipo de tablas independientes (las tablas están presentes en cualquier momento despendiendo de las opciones de configuración y los módulos del kernel). Las tablas se definen con la opción `-t, --table tabla`. Las tablas son:

- `filter`. Esta es la tabla predeterminada. Esta tabla contiene la cadenas internas INPUT (para los paquetes de entrada al servidor), FORWARD (para los paquetes que son ruteados a través del servidor) y OUTPUT (para paquetes generados localmente).

- `nat`. Esta tabla es consultada cuando un paquete crea una nueva conexión. Consiste de tres cadenas: PREROUTING (para alterar paquetes tan pronto entren), OUTPUT (para alterar paquetes generados localmente antes de ser ruteados) y POSTROUTING (para alterar paquetes como van saliendo).

- `mangle`. Esta tabla es utilizada para realizar una alteración especializada del paquete. Hasta versiones del kernel 2.4.17 tenia dos cadenas: PREROUTING y OUTPUT.

A partir del kernel 2.4.18, también son soportadas otras tres cadenas: INPUT, FORWARD y POSTROUTING.

Las opciones que reconoce iptables se dividen en distintos grupos:

1. Comandos. Estas opciones especifican la acción que se debe realizar. Sólo puede ser especificada una a la vez en la línea de comandos a menos que otra sea especificada después. Para todas las versiones largas del comando y nombres de opción, se deben utilizar sólo las letras suficientes para asegurar que iptables pueda diferenciarlas de otras opciones.

   `-A, --append cadena especificacion-regla`
   Agrega una o mas reglas al fin de la cadena seleccionada. Cuando los nombres fuente y/o destino resuelven a mas de una dirección, una regla será agregada por cada combinación posible de direcciones.

   `-D, --delete cadena especificacion-regla`
   `-D, --delete cadena numregla`
   Borra una o mas reglas de la cadena seleccionada. Existen dos versiones para este comando: la regla puede ser especificada como un número en la cadena (iniciando en 1 para la primera regla) o una regla a coincidir.

   `-I, --insert cadena [numregla] especificacion-regla`
   Inserta una o mas reglas en la cadena seleccionada como el número de regla dado. Asi, si el número de regla es 1, la regla o reglas son insertadas al inicio de la cadena. Este es el valor predeterminado si no se especifica un número.

   `-R, --replace cadena numregla especificacion-regla`
   Reemplaza una regla en la cadena seleccionada. Si el nombre fuente y/o destino resuelven para múltiples direcciones, el comando fallará. Las reglas son numeradas iniciando en 1.

   `-L, --list [cadena]`
   Lista todas las reglas en la cadena seleccionada. Si no se especifica una cadena, se listan todas. Como todos los comandos de iptables, esto aplica a la tabla especificada (la tabla predeterminada es filter), asi que si se desean ver las reglas NAT, el comando es:

   ```
   iptables -t nat -n -L
   ```

   `-F, --flush [cadena]`
   Es equivalente a borrar todas las reglas una a una.

`-Z, --zero [cadena]`
Pone a cero los contadores de paquetes y de bytes en todas las cadenas.

`-N, --new-chain cadena`
Crea una nueva cadena definida por el usuario con un nombre dado.

`-P, --policy cadena objetivo`
Define la política para la cadena al objetivo dado. Sólo las cadenas predefinidas pueden tener políticas.

`-E, --rename-chain nombre-cadena-anterior nom-cadena-nueva`
Renombra la cadena especificada por el usuario.

`-h`
Ayuda. Muestra una descripción breve de la sintaxis del comando.

2. Parámetros. Los siguientes parámetros definen una regla. Son utilizados en los comandos add, delete, insert, replace y append.

`-p, --protocol [!] protocolo`
Define el protocolo a revisar de la regla o del paquete. El protocolo que se puede especificar es uno de los valores `tcp`, `udp`, `icmp` o `all`, o puede ser un valor numérico, representando alguno de estos protocolos. También es permitido utilizar algún protocolo definido en el archivo `/etc/protocols`. El argumento "!" antes del protocolo invierte la prueba. El número cero es equivalente a `all`. El valor `all` del protocolo concidide con todos los protocolos y es el valor predeterminado en caso de que sea omitido.

`-s, --source [!] dirección[/mascara]`
Define la fuente. La dirección puede ser el nombre de una red, un equipo, la dirección IP de una red (con /mascara), o una dirección IP. La máscara puede ser una máscara de red o un número simple, especificando la cantidad de 1 en el lado izquierdo de la máscara de red. Así, una máscara de `24` es equivalente a `255.255.255.0`. El argumento "!" antes de la especificación de la dirección invierte el sentido de la dirección.

`-d, --destination [!] dirección[/mascara]`
Es similar a la opción anterior, sólo que en este caso define el destino.

`-j, --jump objetivo`
Especifica el objetivo de la regla; por ejemplo, qué hacer si el paquete coincide. El objetivo puede ser una cadena definida por el usuario, o un objetivo interno. Si esta opción es omitida en una regla, entonces cuando una regla coincida no

tendrá efecto sobre el paquete, pero los contadores sobre la regla se incrementarán.

`-i,--in-interfaz [!] nombre`
Con este parámetro se define la interfaz mediante la cual un paquete será recibido (aplica sólo para los paquetes definidos con las cadenas INPUT, FORWARD y PREROUTING). Si el nombre de la interfaz finaliza con "+", entonces cualquier interfaz que comience con este nombre coincidirá.

`-o, --out-interfaz [!] nombre`
Es el nombre de la interfaz de red por la cual será enviado el paquete (aplica para los paquetes definidos en las cadenas FORWARD, OUTPUT y POSTROUTING).

`[!] -f, --fragment`
Esto significa que la regla sólo se refiere a partir del segundo fragmento de los paquetes fragmentados. Cuando el argumento "!" precede a la bandera "-f", la regla sólo coincidirá con el primer fragmento o con paquetes que no estén fragmentados.

`-c, --set-counters PAQUETES BYTES`
Habilita el contador de paquetes y de bytes de las reglas (aplica para las operaciones INSERT, APPEND y REPLACE).

Definición de reglas del firewall para el SIIEM/UAP

Debido a que en el sistema que aloja al SIIEM/UAP sólo se debe permitir el acceso público al servicio Web y correo electrónico y acceso restringido para iniciar sesiones remotas seguras, es necesario refozar estas políticas a través del firewall. La tabla 4.2 resume estas políticas en cuanto al acceso a los servicios y los puertos utilizados. Todos los demas servicios (y puertos por consecuencia) deben ser bloqueados.

Servicio	Puerto	Acceso remoto
Conexión remota segura (Secure shell)	22/TCP	Restringido (no público)
Correo electrónico	25/TCP	No restringido (público)
Web	80/TCP	No restringido (público)
Base de datos	5432/TCP	Restringido (no público)

Cuadro 4.2: Acceso a los servicios en el SIIEM/UAP.

A continuación se definen las reglas del firewall iptables para implementar estas políticas.

```
#Borra todas las reglas existentes
iptables -F

#Permite paquetes ICMP de equipos confiables
iptables -A INPUT -p icmp -s 132.248.73.7 -j ACCEPT
iptables -A INPUT -p icmp -s 132.248.73.10 -j ACCEPT

# Bloquea los paquetes ICMP (evita las respuestas a ping)
iptables -A INPUT -p icmp -j DROP

#Permite conexiones de secure shell (puerto 22) solo a algunos
#equipos autorizados

iptables -A INPUT -p tcp --dport 22 -s 132.248.73.10
                                                -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -s 132.248.73.20
                                                -j ACCEPT

#Permite conexiones tcp a la base de datos de equipos autorizados
iptables -A INPUT -p tcp --dport 5432 -s 132.248.73.3
                                                  -j ACCEPT

#Permite conexiones desde cualquier lugar al servicio web
iptables -A INPUT -p tcp --dport 80 -j ACCEPT
#Permite conexiones desde cualquier lugar al servicio web cifrado
iptables -A INPUT -p tcp --dport 443 -j ACCEPT

#Permite conexiones desde cualquier lugar al servidor de correo
iptables -A INPUT -p tcp --dport 25 -j ACCEPT

#Permite conexiones X11
iptables -A INPUT -p tcp --dport 6000 -s 132.248.73.7 -j ACCEPT
iptables -A INPUT -p tcp --dport 6000 -s 132.248.73.10 -j ACCEPT


#Bloquea cualquier otro puerto privilegiado
iptables -A INPUT -p tcp --dport 1:1024 -j DROP
iptables -A INPUT -p udp --dport 1:1024 -j DROP

#Evita ataques conocidos (SYN, FLOOD SYN)
iptables -A INPUT -p tcp -m tcp --dport 0:1023 --syn -j REJECT
iptables -A INPUT -p tcp -m tcp --dport 2049 --syn -j REJECT
iptables -A INPUT -p udp -m udp --dport 0:1023 -j REJECT
iptables -A INPUT -p udp -m udp --dport 2049 -j REJECT
iptables -A INPUT -p tcp -m tcp --dport 6000:6009 --syn -j REJECT
```

```
iptables -A INPUT -p tcp -m tcp --dport 7100 --syn -j REJECT
```

4.3.2. Integridad del sistema de archivos

El sistema de archivos es la parte encargada de la administración y organización de la información (archivos, directorios, etc.). Dada su importancia, es necesario un monitoreo adecuado del mismo, ésto ayudará en dos sentidos:

1. Mantener la integridad de la información. A través del monitoreo del sistema de archivos, el administrador puede definir elementos que no deben ser modificados (como programas y aplicaciones), información que debe ser modificada regularmente pero no eliminada (por ejemplo bitácoras), bases de datos de usuarios y grupos (como los archivos `/etc/passwd`, `/etc/shadow` y `/etc/group`) y archivos de configuración.

2. Detectar una intrusión. Generalmente cuando los intrusos ingresan a un sistema realizan actividades que alteran el sistema de archivos, esto puede ser: instalación de exploits[7], modificación en la base de datos de usuarios que aseguren su ingreso al sistema con privilegios de administración, descarga de piratería para instalar un sitio warez, instalación de programas IRC, instalación de caballos de troya que oculten la actividad del intruso, etc.

El administrador del sistema no se percatará de un cambio en el sistema de archivos si no cuenta con un mecanismo de monitoreo que permita detectar estos cambios.

Tripwire

Tripwire es una herramienta útil en el monitoreo del sistema de archivos de sistemas tipo Unix. Esta herramienta fue desarrollada por Eugene Spafford y Gene Kim en el año de 1993 en la Universidad de Perdue.

Mediante la obtención de elementos característicos de un archivo y la firma digital del mismo se realiza una base de datos que constantemente es revisada para detectar una posible alteración en el sistema.

Dependiendo de los parámetros de configuración, la base de datos que mantiene Tripwire almacena la siguiente información:

- Propietario del archivo.
- Grupo al que pertenece el archivo.
- Fecha de creación, modificación y de último acceso.
- Número de inodo.

[7] Ver glosario

- Permisos.
- Tamaño.
- Número de ligas asociadas al archivo.
- Firma digital con diferentes algoritmos.

Debido a la información almacenada en la bases de datos de Tripwire es conveniente tener en cuenta lo siguiente:

- Respaldo. La base de datos debe ser respaldada en un medio seguro.
- Actualización. Debe ser actualizada cada vez que exista un cambio válido en el sistema, como la instalación de una nueva aplicación.

Instalación de Tripwire

Existen dos formas de instalar Tripwire: obteniendo el paquete precompilado u obteniendo el código fuente. El paquete precompilado puede ser obtenido desde:
`http://www.tripwire.org/downloads/index.php`
y el código fuente puede ser obtenido desde:
`http://www.sourceforge.net/projects/tripwire`

El proceso de instalación del programa precompilado se indica a continuación:

1. Obtener el paquete precompilado de Tripwire. Debido a las distintas versiones del kernel, Tripwire proporciona el paquete precompilado RPM 3.0 para versiones de Red Hat 5.x a 6.2.x. Y proporciona el paquete RPM 4.0 para versiones de Red Hat 7.x, sin embargo, también funciona en Red Hat 8.0.

2. Verificar la integridad del paquete. Mediante el comando `md5sum` se obtiene la firma digital del archivo y se compara con la firma que se encuentra en la página oficial de Tripwire. Si las firmas coinciden, es un indicio de que el archivo no está corrupto.

```
$ md5sum tripwire-2.3-47.i386.tar.gz
661a54a6429d4ecb0d756de5046da48f tripwire-2.3-47.i386.tar.gz
```

3. Descomprimir el archivo.
```
$ tar zxf tripwire.i386.tar.gz
```

4. Instalación.

```
$ rpm -hiv tripwire-2.3-47.i386.rpm
```

La instalación genera una serie de programas útiles en el manejo de Tripwire.

- `tripwire`. Es el programa para realizar las operaciones básicas entre las que se encuentran la creación de la base de datos y la revisión de la integridad del sistema de archivos haciendo una comparación con la base de datos generada previamente.

- `twadmin`. Es utilizado para crear, modificar y firmar las políticas de Tripwire; también es utilizado para configurar y manejar los archivos de llaves además de otras funciones administrativas.

- `twprint`. Imprime la base de datos de Tripwire y los archivos de reporte en texto plano.

- `siggen`. Es utilizado por Tripwire para generar valores hash de archivos, con esta herramienta, Tripwire realiza parte de la verificación de la integridad de un archivo debido a que si el contenido del archivo cambia, el valor hash devuelto es diferente.

El proceso de instalación también incluye la generación de páginas de manual para mejor referencia de la herramienta.

- `twintro`. Proporciona una breve introducción a tripwire.

- `twfiles`. Provee una revisión de archivos creados y usados por Tripwire, incluye ubicaciones y propiedades predefinidas.

- `twconfig`. Da una explicación de los archivos de configuración.

- `twpolicy`. Explica de forma detallada el formato y configuración del archivo de políticas.

Configuración de Tripwire

Una vez instalada la herramienta, es necesario configurarla para realizar la revisión adecuada en el sistema. Para ello, si es necesario, se puede editar el archivo de configuración `twcfg.txt` y el archivo de políticas `twpol.txt`. Estos archivos se retomarán posteriormente.

Se ejecuta el script `/etc/tripwire/twinstall.sh` para iniciar la configuración de Tripwire . Este script se encarga de generar el archivo de configuración y de políticas firmados en formato binario. Para ésto, el script solicita una contraseña para generar un archivo de llave que será utilizado para firmar el archivo de políticas y otra contraseña para generar el archivo de llave para firmar el archivo de configuración.

Archivo de Configuración `twcfg.txt`

El archivo de configuración almacena información específica la cual incluye la ubicación de los archivos de datos de Tripwire y la configuración para enviar notificaciones

por correo electrónico. Los parámetros en el archivo de configuración son generados durante el proceso de instalación, sin embargo, pueden ser modificados en cualquier momento.

El archivo de configuración define diferentes elementos a través de las variables que se muestran a continuación:

- `POLFILE`. Define el archivo de políticas. `/RUTA/tw.pol`
- `DBFILE`. Determina la base de datos de Tripwire. `/RUTA/$(HOSTNAME).twd`
- `REPORTFILE`. Indica la ubicación de los reportes. `/RUTA/$(HOSTNAME)-$(DATE).twr`
- `SITEKEYFILE`. Define la ubicación de la llave para el sitio. `/etc/tripwire/site.key`.
- `LOCALKEYFILE`. Determina la ubicación para la llave local. `/etc/tripwire/$(HOSTNAME)-local.key`.

En este archivo existen otras variables, sin embargo, las mencionadas anteriormente son de gran importancia para el funcionamiento correcto de Tripwire.

Archivo de Políticas `twpol.txt`

El archivo de políticas describe los objetos del sistema que deben ser monitoreados por Tripwire y especifica las propiedades que deben ser recolectadas y almacenadas para cada objeto en el archivo de la base de datos.

Cada objeto en el archivo de políticas es asociado con una máscara, la cual describe qué cambios a los archivos o directorios deben ser monitoreados por Tripwire y cuáles pueden ser ignorados de forma segura.

Durante la instalación, un archivo de políticas es codificado y firmado (tw.pol) en el directorio `/etc/tripwire` y se genera una copia en texto plano del archivo (twpol.txt) en el mismo directorio.

Una vez que se genera el archivo de políticas inicial, cualquier cambio debe ser realizado con el comando `tripwire -update-policy` en lugar de simplemente sobreescribir el archivo de políticas. Cuando se crea un nuevo archivo de políticas, la base de datos debe ser reinicializada, debido a que si un intruso modifica los archivos después de la última revisión de integridad, estos cambios no serán detectados y serán incluidos como parte de la nueva base de datos.

Los componentes básicos del archivo de políticas son:

- Comentarios. Se consideran como comentarios cualquier texto precedido por un caracter '#'.
- Reglas. Las reglas determinan los archivos particulares y directorios que Tripwire revisará. Existen dos tipos de reglas conocidas por Tripwire:
  - Las reglas normales definen cuáles propiedades deben ser revisadas de un archivo o directorio. Una regla de este tipo se define de la siguiente forma:

    ```
    nombre-del-objeto -> máscara-de-propiedades;
    ```

    Donde nombre-del-objeto es la ruta completa de un archivo o directorio, no se permiten variables de ambiente, y la máscara-de-propiedades especifica las propiedades de un objeto para examinar o ignorar. El símbolo -> separa dos campos. Con el caracter punto y coma ';' se termina la regla. Si el objeto es un directorio, él y todo su contenido será revisado con la máscara. Por ejemplo:

    `/bin ->$(ReadOnly);` Define el comportamiento para el directorio /bin.

    `/etc/hosts ->$(IgnoreNone) -ar;` Revisa todas las propiedades.
  - Los puntos de excepción le indican a Tripwire no revisar cierto archivo o directorio en particular. La sintáxis es:
    `!nombre-del-objeto`

    Por ejemplo:

    ```
    /etc ->$(ReadOnly);
    !/etc/rc.d;
    !/etc/mnttab;
    ```

    Con las reglas anteriores, Tripwire revisará el directorio `/etc` a excepción de los directorios `/etc/rc.d` y `/etc/mnttab`.

La máscara de propiedades, como se mencionó antes, designa las propiedades de un objeto dado que deben ser revisadas. Una máscara consiste de una serie de símbolos, cada uno de los cuales pueden ser precedidos por un signo mas '+' o menos '-'. Cada caracter se entiende por una propiedad particular para que sea tomada en cuenta durante la revisión de integridad. Si el caracter es precedido por el signo '+', se realiza la revisión de dicha propiedad; en cambio, si es precedido por el signo '-', no se toma en cuenta dicha propiedad. La tabla 4.3 muestra las propiedades que pueden ser revisadas por Tripwire y que son utilizadas para definir la máscara.

Tripwire tiene definidas algunas variables para máscaras de propiedades las cuales pueden ser utilizadas en el archivo de políticas. En seguida se muestran estas variables:

Símbolo	Significado
-	Ignora la propiedad
+	Registra y revisa la propiedad
a	Tiempo de acceso
b	Número de bloques asignados
c	Tiempo de ínodo (creación/modificación)
d	Identificador del dispositivo en el cual reside el ínodo
g	GID del dueño del archivo
i	Número de ínodo
l	El archivo incrementa el tamaño
m	Tiempo de modificación
n	Número de ligas
p	Permisos y bits de modo en el archivo
r	Identificador del dispositivo apuntado por el ínodo, esto es válido para los dispositivos.
s	Tamaño del archivo.
t	Tipo del archivo.
u	UID del dueño.
C	Valor hash CRC-32
H	Valor hash Haval
M	Valor hash md5
S	Valor hash SHA

Cuadro 4.3: Simbología de propiedades en Tripwire utilizada para definir la máscara.

- **Readonly**. Es útil para definir los archivos de sólo lectura, su valor es: `+pinugtsdbmCM-rlacSH`
- **Dynamic**. Es bueno para monitorear directorios de usuarios y archivos con un comportamiento dinámico. El valor de la máscara es: `+pinugtd-srlbamcCMSH`.
- **Growing**.Esta variable es útil para verificar archivos que crecen constantemente como lo son bitácoras. Su valor es: `+pinugtdl-srbamcCMSH`.
- **Device**. Se utiliza para dispositivos u otros archivos que Tripwre no debe intentar abrir. Su valor es: `+pugsdr-intlbamcCMSH` .
- **IgnoreAll**. Sólo verifica la existencia del archivo, pero no revisa mas. El valor de la máscara es: `-pinugtsdbmrlacCMSH`.
- **IgnoreNone**. Revisa todas las propiedades y es útil para definir máscaras propias. Su valor es: `+pinugtsdbmrlacCMSH`.

Debido a lo extenso que es el archivo de políticas no se incluye en este trabajo, sin embargo, la tabla 4.4 muestra los elementos considerados en las políticas para su revisión.

Objetos	Máscara
Programas binarios de Tripwire	Readonly
Directorio de la base de datos de Tripwire	Dymanic -i
Archivo de políticas de Tripwire	Readonly -i
Archivo de configuración de Tripwire	Readonly -i
Archivos de llaves	Readonly
Directorios que no deben cambiar de dueño y permisos como `/`, `/home`, `/etc`	+tpug
Programas de uso y administración del sistema de archivos	IgnoreNone -SHa
Programas de administración del kernel	IgnoreNone -SHa
Programas para el manejo de la red	IgnoreNone -SHa
Programas de administración del sistema	IgnoreNone -SHa
Programas para el control de dispositivos y Hardware	IgnoreNone -SHa
Programas de información del sistema	IgnoreNone -SHa
Programas de información de aplicación	IgnoreNone -SHa
Programas relacionados con el shell	IgnoreNone -SHa
Utilerías del sistema operativo	IgnoreNone -SHa
Ligas simbólicas en el sistema	IgnoreNone -SHa
Directorios temporales	+tpug
Directorios donde se encuentran programas binarios	Readonly
Intérpretes de comandos (shell)	Readonly
Archivos de seguridad (`/etc/security`)	IgnoreNone -SHa
Archivos de inicialización (`/etc/profile`)	Dynamic
Bibliotecas del sistema	ReadOnly
Programas cargadores del sistema (lilo y grub)	IgnoreNone -SHa
Archivos y directorios que cambian cuando inicia el sistema	Dynamic
Directorio del usuario root	IgnoreNone -SHa
Archivos de configuración del usuario root	Dynamic
Archivos de configuración del sistema	Readonly
Dispositivos	Device
Directorio de archivos binarios y bibliotecas(`/bin` y `/lib`)	Readonly
Bitácoras del sistema	Growing
Directorio de páginas y programas del sitio SIIEM/UAP	Readonly
Programas binarios de la base de datos	Readonly
Base de datos del SIIEM/UAP	Growing

Cuadro 4.4: Políticas de verificación de integridad de archivos en el SIIEM/UAP.

Uso de Tripwire

Para finalizar con esta herramienta, sólo falta ver el modo de iniciar y comparar la base de datos de acuerdo a las políticas definidas previamente.

Para inicializar la base de datos se ejecuta el comando `tripwire -m i` como se muestra a continuación:

```
# tripwire -m i
Please enter your local passphrase:
Parsing policy file: /etc/tripwire/tw.pol
Generating the database...
*** Processing Unix File System ***
### Warning: File system error.
### Filename: /home/httpd/web
### No existe el fichero o el directorio
### Continuing...
### Warning: File system error.
### Filename: /home/httpd/w
### No existe el fichero o el directorio
### Continuing...
### Warning: File system error.
### Filename: /home/httpd/eb
### No existe el fichero o el directorio
### Continuing...
...
```

De esta forma, Tripwire se encarga de generar la base de datos de acuerdo al archivo de políticas. La base de datos es almacenada en el archivo `/var/lib/tripwire/newman.posgrado.unam.mx.twd`.

Una vez generada la base de datos, es necesario realizar una revisión periódica a través del comando `tripwire -m c` como se ve en el siguiente ejemplo:

```
tripwire -m c
Parsing policy file: /etc/tripwire/tw.pol
*** Processing Unix File System ***
Performing integrity check...
...
```

Una vez realizada la revisión del sistema de archivos y efectuada la comparación con la base de datos, Tripwire emite un reporte que es almacenado en un archivo y también se muestra en pantalla.

El reporte está dividido en cuatro secciones.

- *Resumen del reporte.* Muestra los datos del equipo y los archivos de políticas, configuración y base de datos utilizados.
- *Resumen de las reglas.* Muestra cada una de las reglas, indicando el número de archivos que fueron agregados, eliminados o modificados.
- *Resumen de objetos.* En esta sección se muestran las inconsistencias encontradas en el sistema de archivos de forma detallada por cada regla. Se indica la ruta del archivo agregado, modificado o eliminado.
- *Errores.* En la última sección se muestran los errores encontrados por Tripwire, generalmente son archivos que están considerados en las políticas que no se encontraron en el sistema de archivos.

A continuación se muestra un ejemplo de este reporte:

```
Wrote report file: /var/lib/tripwire/report/newman-20040124.twr


Tripwire(R) 2.3.0 Integrity Check Report

Report generated by:          root
Report created on:            Fri Jan 24 14:07:32 2004
Database last updated on:     Thu Jan 23 21:42:21 2004

===============================================================
Report Summary:
===============================================================

Host name:                    newman.posgrado.unam.mx
Host IP address:              132.248.73.6
Host ID:                      None
Policy file used:             /etc/tripwire/tw.pol
Configuration file used:      /etc/tripwire/tw.cfg
Database file used:           /var/lib/tripwire/lapfer.twd
Command line used:            tripwire -m c

===============================================================
Rule Summary:
===============================================================

---------------------------------------------------------------
  Section: Unix File System
---------------------------------------------------------------

  Rule Name              Severity Level  Added  Removed  Modified

---------------------------------------------------------------
```

	---------	----------	-----	-------	--------
	Invariant Directories	66	0	0	0
	Temporary directories	33	0	0	0
*	Tripwire Data Files	100	0	0	1
	Archivos del SIIEM/UAP (/home/httpd/web)	100	0	0	0
	Binarios de la base de datos (/home/httpd/w)	100	0	0	0
	Base de datos (/home/httpd/eb)	100	0	0	0
	Critical devices	100	0	0	0
*	User binaries	66	1	0	1
	Tripwire Binaries	100	0	0	0
	Libraries	66	0	0	0
*	Bitacoras	100	0	0	13
	File System and Disk Administraton Programs	100	0	0	0
	Kernel Administration Programs	100	0	0	0
	Networking Programs	100	0	0	0
	System Administration Programs	100	0	0	0
	Hardware and Device Control Programs	100	0	0	0
	System Information Programs	100	0	0	0
	Application Information Programs	100	0	0	0
	Shell Releated Programs (/sbin/getkey)	100	0	0	0
	Critical Utility Sym-Links	100	0	0	0
	Critical system boot files	100	0	0	0
*	Critical configuration files	100	0	0	2
	System boot changes	100	0	0	0
	OS executables and libraries	100	0	0	0
	Security Control	100	0	0	0
	Login Scripts	100	0	0	0

```
  Operating System
  Utilities                100              0        0        0
  Shell Binaries           100              0        0        0
* Root config files        100              2        2        1

Total objects scanned:  38764
Total violations found:  23

===============================================================
Object Summary:
===============================================================

---------------------------------------------------------------
# Section: Unix File System
---------------------------------------------------------------

---------------------------------------------------------------
Rule Name: User binaries (/usr/bin)
Severity Level: 66
---------------------------------------------------------------

Added:
"/usr/bin/amsn"

Modified:
"/usr/bin"

---------------------------------------------------------------
Rule Name: Bitacoras (/var/log)
Severity Level: 100
---------------------------------------------------------------

Modified:
"/var/log/dmesg"
"/var/log/gdm/:0.log"
"/var/log/gdm/:0.log.1"
"/var/log/gdm/:0.log.2"
"/var/log/gdm/:0.log.3"
"/var/log/gdm/:0.log.4"
"/var/log/ksyms.0"
"/var/log/ksyms.1"
"/var/log/ksyms.2"
"/var/log/ksyms.3"
"/var/log/ksyms.4"
"/var/log/ksyms.5"
```

```
"/var/log/ksyms.6"

-------------------------------------------------------------------
Rule Name: Tripwire Data Files (/etc/tripwire/tw.pol)
Severity Level: 100
-------------------------------------------------------------------

Modified:
"/etc/tripwire/tw.pol"

-------------------------------------------------------------------
Rule Name: Critical configuration files (/etc/sysconfig)
Severity Level: 100
-------------------------------------------------------------------

Modified:
"/etc/sysconfig/network"
"/etc/sysconfig/network-scripts/ifcfg-eth0"

-------------------------------------------------------------------
Rule Name: Root config files (/root)
Severity Level: 100
-------------------------------------------------------------------

Added:
"/root/.xauthL7MW7c"
"/root/.xauthhIdcxG"

Removed:
"/root/.xauth9MCwEO"
"/root/.xauthaOmSvJ"

Modified:
"/root"

===================================================================
Error Report:
===================================================================

-------------------------------------------------------------------
  Section: Unix File System
-------------------------------------------------------------------
-------------------------------------------------------------------
*** End of report ***
```

```
Tripwire 2.3 Portions copyright 2000 Tripwire, Inc. Tripwire is a
registered trademark of Tripwire, Inc. This software comes with
ABSOLUTELY NO WARRANTY; for details use --version. This is free
software which may be redistributed or modified only under certain
conditions; see COPYING for details.
All rights reserved.
Integrity check complete.
```

Se debe considerar que cada vez que se haga la instalación de una aplicación en el sistema se debe actualizar la base de datos, debido a que el proceso de instalación realiza modificaciones en el sistema de archivos. La forma de actualizar la base de datos es mediante el comando `tripwire -m u`.

4.3.3. Rastreo de puertos abiertos

Los puertos son como las puertas de una casa, ésto es, son los accesos del mundo exterior hacia el sistema. Generalmente los puertos son abiertos para proporcionar un servicio. Como se mencionó en el Capítulo 2, un equipo sólo debe proporcionar los servicios necesarios de acuerdo al objetivo de la organización. Sin embargo, cuando existe una intrusión en un sistema, una de las principales actividades de un intruso es asegurar el acceso al equipo nuevamente y para ello instala programas que abren un servicio para acceder remotamente. Otra de las actividades comunes es la instalación de IRC que le permiten al intruso mantener canales de comunicación con otros intrusos.

El alcance que un intruso puede tener en un sistema es muy amplio, puede ir desde instalar un servicio troyano para que le asegure el ingreso al sistema hasta instalar un sitio que le permita compartir piratería e inclusive pornografía y para ello requerirá instalar un servicio en el sistema vulnerado. Por tal motivo, el administrador de sistemas debe verificar constantemente los puertos que tiene abiertos en el sistema y verificar dos cosas:

1. Que el sistema proporcione los servicios indispensables, ni uno más.
2. Que no esté deshabilitado un servicio importante para el funcionamiento del sistema.

Nmap

Nmap es una herramienta desarrollada por *fyodor* para rastrear los puertos abiertos en un sistema o red. Esta herramienta es compatible con los distintos tipos de escaneos como UDP, TCP connect(), TCP SYN (half open), ftp proxy (bounce attack), Reverseident, ICMP (ping sweep), FIN, ACK sweep, Xmas Tree, SYN sweep y Null scan.

Entre sus principales características se encuentran:

- Detección remota del sistema operativo.

- Rastreo tipo stealth (oculto).
- Rastreo dinámico y cálculos de retransmisión.
- Rastreo paralelo.
- Detección de servidores inactivos por medio de mensajes ICMP paralelos.
- Rastreo con señuelos.
- Detección de filtrado de puertos.
- Rastreo por fragmentación.
- Especificación flexible del destino y el puerto.

Instalación de Nmap

Esta herramienta puede ser obtenida en `http://www.insecure.org`. Una vez que se tiene el código de Nmap, se ejecutan los siguientes comandos:

```
tar zxf nmap-curren.tgz
cd nmap-current
./configure
make
make install
```

De esta forma se copia el programa y los manuales de la herramienta. No es necesario configurar nada, sólo utilizar la herramienta.

Uso de Nmap

Nmap tiene dos formas de uso, una es a través de la línea de comandos y otra mediante una interfaz gráfica.

El modo de uso de línea de comandos se hace mediante la siguiente sintáxis:

```
nmap [Tipo de rastreo] [Opciones] <equipo o red>
```

La tabla 4.5 muestra las opciones más importantes de Nmap.

El siguiente ejemplo muestra el rastreo al equipo local utilizando un rastreo invisible y utilizando la opción `-O` para determinar el sistema operativo.

```
# nmap -sS -O localhost

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on lapfer (127.0.0.1):
```

Opción	Descripción
-sT	Rastreo por medio de la llamada connect().
-sS	Rastreo *stealth* o invisible por medio de paquetes SYN.
-sU	Rastreo de puertos UDP.
-sP	Rastreo mediante ping, encuentra las máquinas de un rango.
-sF, -sX, -sN	Rastreo stealth (FIN, Xmas, Null).
-sV	Obtiene la versión del servicio.
-sR / -I	Opción para RPC/Identd.
-O	Obtiene el sistema operativo del equipo rastreado.
-p <rango>	Rango de puertos a rastrear.
-F	Realiza el rastreo de los puertos definidos en el archivo `/usr/local/share/nmap/nmap-services`.
-v	*Verbose*, realiza un informe detallado.
-P0	Suprime ping (*paquetes ICMP*) al rastrear.
-T <valor>	Tiempo definido para los paquetes.
-n/-R	No resuelve el DNS / Resuelve siempre.
-oN/-oX/-oG	Tipo de impresión del reporte.
-iL	Obtiene las direcciones IP que serán rastreadas.
-S	Determina la interfaz de red.

Cuadro 4.5: Opciones de Nmap.

```
(The 1599 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
6000/tcp   open        X11
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 0.063 days (since Sat Oct 25 19:34:33 2003)

Nmap run completed -- 1 IP address (1 host up) scanned in 6 seconds
```

La salida anterior muestra que fueron rastreados 1599 puertos en 6 segundos, de los cuales sólo se despliegan los que se encontraron abiertos en el sistema. También determinó que el sistema operativo del equipo es Linux y el Kernel se encuentra entre la versión 2.4.0 y la 2.5.20.

En el siguiente ejemplo se realiza un rastreo SYN de los puertos 1 al 65000 sin utilizar pings.

```
# nmap -sS -p 1-65000 -P0 localhost

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on lapfer (127.0.0.1):
(The 64998 ports scanned but not shown below are in state: closed)
```

```
Port        State       Service
22/tcp      open        ssh
6000/tcp    open        X11

Nmap run completed 1 IP address (1 host up) scanned in 72 seconds
```

Obviamente el proceso se vuelve más lento debido a que se rastrean más puertos en el sistema.

Como se mencionó anteriormente, Nmap cuenta con una interfaz gráfica para realizar el rastreo de puertos. La imagen 4.1 muestra la interfaz utilizada por Nmap cuando es ejecutada por un usuario sin privilegios a través del comando `nmapfe`. Nótese que de acuerdo a la selección de opciones en la interfaz, Nmap muestra cual sería el comando utilizado en la línea de comandos.

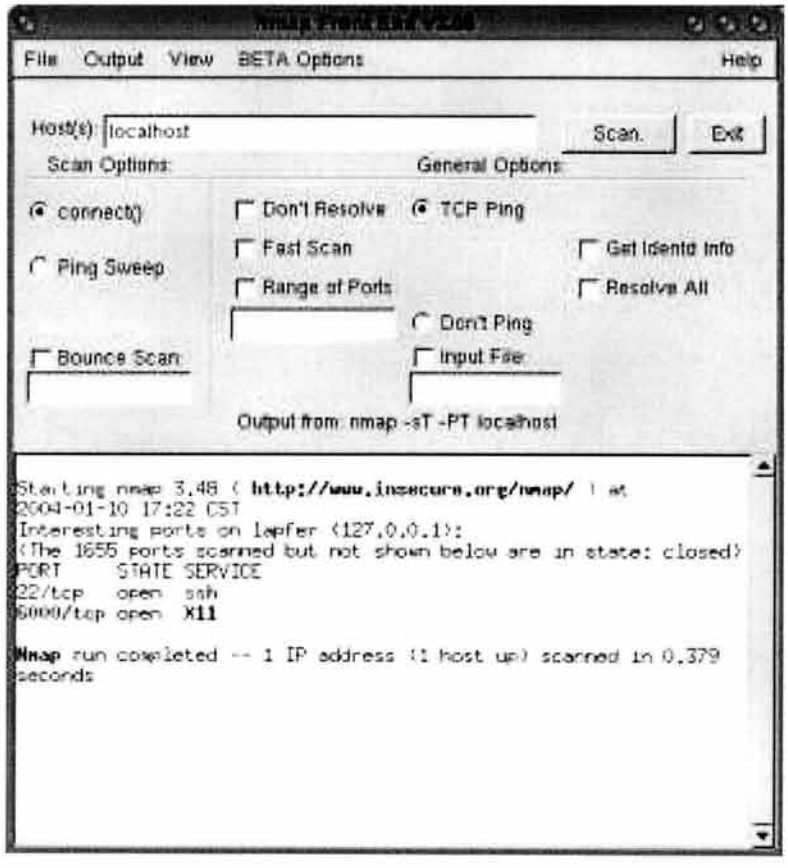

Figura 4.1: Interfaz gráfica de Nmap ejecutada por un usuario sin privilegios.

La imagen 4.2 muestra el uso de la interfaz de Nmap ejecutada por el super usuario del sistema. Nótese que existen más opciones habilitadas para realizar el rastreo.

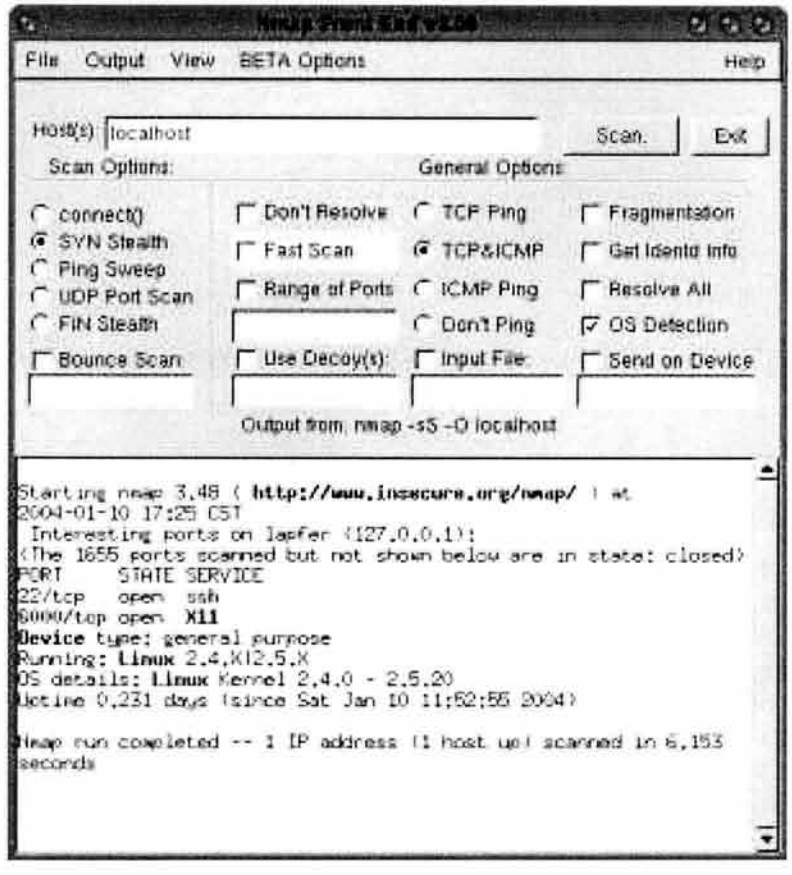

Figura 4.2: Interfaz gráfica de Nmap ejecutada por el usuario root.

El uso de la interfaz gráfica es útil, entre otras cosas, para obtener el comando necesario para ser ejecutado mediante la línea de órdenes en el shell.

Automatización en el SIIEM/UAP

Debido a la ventaja que proporciona Nmap para ser ejecutado mediante la línea de comandos, en el equipo que aloja el SIIEM/UAP desarrollé un programa en lotes que realiza el rastreo de puertos abiertos en el sistema. En la siguiente página se muestra el código utilizado para realizar esta tarea.

```
#!/bin/bash

#################################################################
#Autor: Fernando Zaragoza Hernandez                             #
#Fecha: 1 de Febrero del 2002                                   #
#Ultima modificacion: 27 de Octubre del 2003                    #
#                                                               #
#Este pograma hace un scaneo  de  puertos y lo compara con      #
#un archivo definido. Si existe una  diferencia entre el        #
#original y el que se genere al momento de  hacer el scaneo,    #
#entonces enviara un email, reportando lo sucedido.             #
#################################################################

#Directorio donde se ubica la base de datos y se generaran los
#archivos de salida
DIR=/home/desarrollo/fer/.scan

#Equipo o equipos que se van a rastrear. Se pueden indicar varios
#equipos de la siguiente forma: "equipo_1 equipo_2 equipo_n"
HOST=localhost

#Define las direcciones de correo electronico para enviar el
#aviso de un cambio en los puertos
$EMAIL=fer,fzaraher@hotmail.com
if [ -f $DIR/original.nmap ]
then
  nmap -sS $HOST > $DIR/temp1
  grep -v completed $DIR/temp1>$DIR/temp2
  diff $DIR/original.nmap /$DIR/temp2>$DIR/temp3
  encontro=`wc -l $DIR/temp3 | awk '{print $1}'`
  if [ $encontro -gt 0 ]
  then
    date>>$DIR/error.log
    cat $DIR/temp1>>$DIR/error.log
    Mail -s "Cambios en los puertos" $EMAIL<$DIR/error.log
  fi
  #Borra los temporales
  rm $DIR/temp1 $DIR/temp2 $DIR/temp3
  exit
else
  date>>$DIR/error.log
  echo "No existe el original" >>$DIR/error.log
  mail -s "No existe el orginal del nmap" $EMAIL<$DIR/error.log
  exit
fi
```

La operación del programa se detalla a continuación:

1. Definición de variables.
   - DIR. Esta variable sirve para indicar el directorio donde se encuentran y generarán los archivos durante el proceso de revisión.
   - HOST. Variable útil para indicar el equipo o conjunto de equipos que serán rastreados en sus puertos.
   - EMAIL. Define las direcciones electrónicas para enviar un aviso de correo electrónico cuando haya un cambio en los puertos.
2. Verifica que exista el archivo `original.nmap` el cual es la base de datos con la que se compararán los resultados obtenidos al hacer el rastreo de puertos de los equipos. Si el archivo no existe, se envia un correo electrónico indicando la inexistencia del mismo y el proceso termina.
3. Se realiza el rastreo de puertos a los equipos.
4. Se depura la información obtenida y se compara con la base de datos.
5. Si existe alguna diferencia en el rastreo realizado y la base de datos, se agrega la fecha en el archivo de registro `error.log` y se incluyen los datos obtenidos.
6. Envío de correo electrónico notificando los cambio en los puertos.
7. Se eliminan los archivos generados durante el proceso.

Finalmente, para que el rastreo se lleve periódicamente, el usuario root tiene configurada una tarea mediante el sistema cron para ser ejecutada cada hora. A continuación se muestra la línea de la calendarización de este programa.

```
0 * * * * /bin/bash /home/desarrollo/fer/herra/shell/scan.sh
```

Portsentry

Desde el punto de vista de seguridad, el rastreo de puertos es utilizado para determinar los servicios que puedan estar abiertos en un sistema que presenten una posible puerta de entrada al mismo. Sin embargo, desde el punto de vista de un intruso, el rastreo de puertos puede ser utilizado para determinar las posibles vulnerabilidades que un sistema pudiera tener. Básicamente el intruso determina los puertos que están abiertos en un sistema, obtiene las versiones de dichos servicios e investiga el sistema operativo que está siendo ejecutado en el equipo que desea vulnerar. Teniendo estos datos, un intruso puede buscar un exploit diseñado para vulnerar un servicio específico en un sistema operativo dado.

Portsentry es una herramienta utilizada para detectar el rastreo de puertos hacia el equipo local. La gran ventaja de Portsentry es que es una herramienta proactiva, esto

es, cuando detecta un rastreo de puertos en el sistema, toma una serie de acciones que eviten que un intruso pueda obtener mayor información del sistema e inclusive bloquea mediante distintos mecanismos el acceso al sistema local del equipo remoto que está realizando el rastreo de puertos.

Instalación de Portsentry

La última versión de portsentry es 2.0b1, sin embargo, esta versión no es libre. La versión utilizada en el SIIEM/UAP es la versión 1.0 la cual si es libre. Esta herramienta podía ser obtenida desde `http://www.psionic.org`, desafortunadamente, esta organización fue adquirida por la empresa de telecomunicaciones Cisco y ya no se encuentra disponible, de tal forma que posiblemente no se realicen más actualizaciones a la herramienta. Para obtener la versión 1.0 se puede ir al servidor de ftp del Departamento de Seguridad de la UNAM ftp://ftp.seguridad.unam.mx.

El proceso de instalación es muy similar a las herramientas instaladas con anterioridad.

```
tar xzf portsentry-1.0.tar.gz
cd portsentry-1.0
make linux
make install
```

La instalación genera los siguiente archivos:

- portsentry. Es le programa binario.
- portsentry.conf. Archivo de configuración de la herramienta.
- portsentry.ignore. Archivo donde se definen los equipos que serán ignorados por Portsentry.

Una vez que sea ejecutada la herramienta, se generarán otros archivos donde se lleva un histórico y se tendrá un registro de los equipos que han intentado un rastreo de puertos y son bloqueados.

Configuración de Portsentry

El siguiente paso, antes de utilizar portsentry, es editar el archivo de configuración ubicado en `/usr/local/psionic/portsentry/portsentry.conf`. En este archivo se debe indicar básicamente los puertos que debe monitorear Porsentry y las acciones que deben ser tomadas en caso de que exista un barrido de puertos hacia el sistema, como se muestra a continuación:

- Definir los puertos que deben ser monitoreados. En el archivo de configuración se definen las variables `TCP_PORTS` y `UDP_PORTS`.

- Bloqueo a nivel de kernel. Si se desea que en la tabla de ruteo se obstruya el equipo externo que realiza el rastreo de puertos se debe descomentar la siguiente línea en caso de sistemas Linux:

```
KILL_ROUTE="/sbin/route add -host $TARGET$ reject"
```

o si se cuenta con el firewall iptables, se debe descomentar la línea:

```
KILL_ROUTE="/usr/local/bin/iptables -I INPUT -s $TARGET$
-j DROP"
```

- En caso de no tener habilitado Tcp-wrappers, se debe comentar la línea siguiente:

```
KILL_HOSTS_DENY="ALL: $TARGET$"
```

- También es posible definir un comando propio. Con la siguiente línea se envia un correo electrónico al administrador del sistema.

```
KILL_RUN_CMD="/bin/echo '$TARGET$ ->$PORT$' | /usr/bin/Mail -s
'ATACK $TARGET$' root"
```

Uso de Portsentry

Portsentry utiliza tres modos de operación.

1. Modo clásico. En este modo, a través de las variables TCP_PORTS y UDP_PORTS, Portsentry abre sockets como señuelos en el rastreo de puertos. Aparentemente parecería que están abiertos los puertos, sin embargo, no se proporciona servicio alguno, y cualquier intento de escaneo será registrado y se tomarán las acciones que se hayan configurado en Portsentry. Para ejecutar Portsentry en este modo se ejecuta con las opciones `-tcp` y `-udp`, como se muestra a continuación.

```
/usr/local/psionic/portsentry/portsentry -tcp
```

```
/usr/local/psionic/portsentry/portsentry -udp
```

2. Modo stealth. En este modo se abren sockets crudos, lo que permite a Portsentry verificar un mayor número de puertos. En este modo, si se realiza un escaneo no se verán los puertos como servicios abiertos, sin embargo, Portsentry detectará dichos intentos de rastreo de puertos. Se deben indicar las opciones `-stcp` y `-sudp` para que Portsentry entre en este modo. A continuación se muestran los comandos necesarios.

```
/usr/local/psionic/portsentry/portsentry -stcp

/usr/local/psionic/portsentry/portsentry -sudp
```

3. Modo avanzado. En el modo avanzado Portsentry no abre ningún socket, sino que pide directamente al kernel por la notificación de una petición hacia un puerto. En este modo se utiliza la variable ADVANCED_PORTS_TCP y la variable ADVANCED_PORTS_UDP. Se utiliza el siguiente comando para ejecutar Portsentry en modo avanzado:

```
/usr/local/psionic/portsentry/portsentry -atcp

/usr/local/psionic/portsentry/portsentry -audp
```

Además de realizar todo lo anterior, Portsentry agregará un registro a la bitácora del sistema cuando exista un intento de escaneo de puertos desde el exterior. En sistemas GNU/Linux, el registro se hará en el archivo `/var/log/messages`. A continuación se indican los mensajes registrados en la bitácora cuando existe algún barrido de puertos.

```
Oct 28 09:45:58 lapfer portsentry[1838]: attackalert: UDP scan
       from host:galatea.posgrado.unam.mx/132.248.73.2 to UDP
       port: 138
Oct 28 09:45:58 lapfer portsentry[1838]: attackalert: External
       command run for host: 132.248.73.2 using command:
       "/bin/echo '132.248.73.2 -> 138' | /usr/bin/Mail -s
       'ATACK 132.248.73.2' root"
Oct 28 09:45:58 lapfer portsentry[1838]: attackalert: Host
       132.248.73.2 has been blocked via wrappers with string:
       "ALL: 132.248.73.2"
Oct 28 09:45:58 lapfer portsentry[1838]: attackalert: Host
       132.248.73.2 has been blocked via dropped route using
       command: "/sbin/iptables -I INPUT -s 132.248.73.2 -j DROP"
```

El primer registro indica el intento de escaneo de puertos desde el equipo con la dirección IP señalada. En los registros siguientes se muestran las acciones tomadas, primero envia un correo electrónico, posteriormente bloquea utilizando tcp-wrappers y por último bloquea a nivel del firewall. Se puede observar, a partir de la fecha y hora de los registros, que todas estas acciones se realizaron simultáneamente.

Portsentry es una herramienta muy útil para detectar intentos de rastreo de puertos en el sistema y tomar medidas proactivas para reducir la posibilidad de una intrusión.

4.3.4. Búsqueda de vulnerabilidades conocidas

El software en general es producto de la creación e ingenio humano, específicamente de los programadores que lo desarrollan. Debido a ésto, el software no está exento de errores y la probabilidad de fallas en el código se incrementa en la media que crece el número de instrucciones del programa.

Un problema de los distribuidores de software es que debido a presiones de mercado, la mayoría de las veces liberan código sin ser auditado previamente. Sin embargo, existen miles de usuarios en todo el mundo que constantemente están revisando el código de terceros en busca de problemas de seguridad. Generalmente cuando se encuentra un hueco de seguridad en un programa o aplicación se notifica al creador o distribuidor del programa para que realice las modificaciones pertinentes en sus productos y redistribuyan el software corregido. Desafortunadamente no todos los fabricantes de software responden con la misma rapidez con la que se esperaría, de tal suerte que los usuarios que estén ejecutando una aplicación con errores serán vulnerables mientras no actualicen sus programas.

Si se libera una actualización de un programa vulnerable la recomendación más importante es realizar los cambios pertinentes en el sistema y actualizar las aplicaciones críticas. Ahora imaginemos que no se trata de un equipo, sino de varias decenas de ellos que requieren ser revisados constantemente para verificar las posibles vulnerabilidades que puedan presentar. Esta puede ser una tarea difícil, sin embargo, existen herramientas que permiten identificar vulnerabilidades ya conocidas que ayudan al administrador del sistema a identificar los posibles huecos de seguridad que existen en un equipo o red.

Nessus

El proyecto Nessus provee a la comunidad de Internet una herramienta libre para la búsqueda de vulnerabilidades, con el afán de determinar los posibles huecos de seguridad que un intruso podría utilizar para ingresar de forma ilícita a un sistema.

Nessus, mediante una arquitectura modular, es muy rápido y no supone nada. Ésto es, no da por hecho que en el puerto 80 se encuentra un servicio de Web, debido a que si está siendo ejecutado el servicio de correo electrónico, Nessus lo detectará y realizará las pruebas pertinentes de acuerdo al servicio.

Las principales características de Nessus son:

- Arquitectura modular. Mediante plugins individuales, Nessus realiza las pruebas a equipos específicos.
- NASL (Nessus Atack Scripting Language). Este es lenguaje diseñado para escribir más rápido las pruebas de seguridad.
- Actualización. Debido a que constantemente surgen nuevas vulnerabilidades, nessus actualiza frecuentemene su base de datos de vulnerabilidades.

- Arquitectura cliente/servidor. Nessus funciona a través de un servidor (nessusd) y un cliente (nessus). Este cliente cuenta con un ambiente gráfico y de texto.
- Es posible realizar pruebas a distintos equipos al mismos tiempo.
- Múltiples servicios. Prueba varios servicios, no importa si tiene varios del mismo tipo en diferentes puertos.
- Informes completos. No sólo reporta lo que pueda estar mal en el equipo en cuanto a seguridad sino que proporciona el documento en el cual se está basando y las posibles soluciones. Además de que asigna un nivel de riesgo dependiendo de la severidad del problema.
- Distintos tipos de reportes. Es posible generar distintos formato de reportes: LaTeX, html, html con gráficos, ASCII o xml.

Como se mencionó antes, Nessus se compone de dos elementos: Un cliente y un servidor. A continuación se da una breve descripción de cada uno de estos componentes.

Servidor: `nessusd`

El servidor se encarga de realizar las pruebas al equipo remoto que se desea auditar. `nessusd` intenta inspeccionar el sistema remoto y lista todas las vulnerabilidades y errores de configuración que los afectan. El servidor se ejecuta en la línea de comandos y su sintaxis es la siguiente:

```
nessusd [-v] [-h] [-c arch-conf] [-S ip[,ip2,...]] [-a direccion]
[-p puerto] [-D] [-d]
```

Las opciones más importantes son:

`-c <arch-conf>`. Usa un archivo de configuración distinto a `/usr/local/etc/nessus/nessusd.conf`.

`-a <direccion>`. Le indica al servidor que sólo escuche las peticiones provenientes de `direccion`, la cual generalmente es una dirección IP. Esto es útil cuando se está ejecutando `nessusd` y no se desea que gente externa se conecte a él.

`-p puerto`. Le indica al servidor que escuche en `puerto` distinto al puerto predeterminado 1241.

`-D`. Hace que el servidor sea ejecutado en background o modo de demonio.

`/usr/local/etc/nessus/nessusd.conf` es el archivo de configuración que utiliza `nessusd`. Este archivo contiene algunas de las siguientes opciones:

- `plugins_folder`. Contiene la dirección de los pluggins de nessus. Este archivo generalmente es `/usr/local/lib/nessus/plugins`, pero a través de esta variable se puede cambiar.

- `logfile`. Esta variable indica la ruta del archivo de registro. Se puede indicar syslog si se desea utilizar el registro del sistema via `syslogd`. O se puede indicar stderr si se desea que nessus envie los mensajes a la salida estándar de errores, generalmente la pantalla.

- `max_hosts`. Con esta variable se controla el número máximo de equipos que un cliente puede probar al mismo tiempo. Este valor debe ser calculado de acuerdo al ancho de banda, el número de equipos que se desean probar, memoria y el poder del procesador disponibles.

- `max_checks`. Es el número de plugins que se ejecutarán contra cada equipo probado. Se debe considerar que el número de procesos total que genere nessus será igual a `max_checks x max_hosts` asi que se debe encontrar un balance correcto entre estas dos opciones.

- `cgi_path`. Cuando hace una revisión de CGIs, `nessusd` busca en el directorio /cgi-bin y /scripts. Con esta variable se puede cambiar este directorio para reflejar las políticas de la organización.

- `port_range`. Este es el rango predeterminado que probarán los plugins. La sintaxis de esta opción es flexible, este puede ser un rango simple ("100-10000"), varios puertos ("22, 80, 137"), varios rangos de puertos ("150-500,10000-20000"). También es posible especificar tanto puertos UDP como TCP con U y T respectivamente. En el siguiente ejemplo se define el rango 1 a 1024 para UDP y de 1 a 65535 para TCP: "U:1-1025,T:1-65535".

- `plugins_timeout`. Este es el tiempo máximo de vida en segundos de un plugin. Esta opción es útil para evitar que un plugin no caiga en un ciclo sin fin mientras realiza las pruebas.

- `safe_checks`. Muchas veces, `nessusd` intenta reproducir una condición excepcional para determinar si el servicio remoto es vulnerable para ciertos errores. Esto incluye la reproducción de desbordamientos de pila o formato de cadenas, lo cual puede hacer que el servidor remoto falle. Si se pone esta opción con el valor 'yes', `nessusd` deshabilita los plugins que potencialmente puedan hacer fallar un servicio remoto, en su lugar, intentará hacer muchas revisiones confiando en los mensajes de los servicios probados. Ésto reduce falsos positivos y hace a `nessusd` más amigable através de la red, sin embargo esto puede hacer que se omitan vulnerabilidades importantes.

- `use_mac_addr`. Se debe poner esta opción con el valor de 'yes' si se está probando una red local y cada equipo tiene una dirección IP dinámica (asignada por los protocolos DHCP o BOOTP), y todos los equipos serán referenciados por su dirección MAC.

Cliente: `nessus`

`nessus` es el programa que sirve como interfaz para el usuario con `nessusd`. Este programa es un cliente que puede trabajar en modo gráfico y en modo texto. Este programa tiene las siguientes opciones:

`-c <arch-conf>`. Usa otro archivo de configuración.

`-n`. No utiliza la interfaz gráfica. Esto es útil cuando se está ejecutando nessus en un ambiente sin gráficos.

`-q`. Ejecuta nessus en modo por lotes. Con esta opción se esperan las siguientes opciones:

`-p` Obtiene una lista de los plugins instalados en el servidor.

`-P`. Obtiene la lista de servidores y las preferencias de plugins.

`host`. Es el equipo al cual se conectará el cliente.

`puerto`. Es el puerto en el equipo donde se está ejecutando nessus al cual se conectará el cliente.

`user`. Es el usuario a utilizar con `nessusd`.

`contraseña`. Es la contraseña asociada con el usuario.

`destino`. Es el nombre del archivo que contiene los equipos que serán probados.

`resultados`. Es el nombre del archivo donde se almacenarán los resultados al finalizar las pruebas.

`-r <archivo-reporte>`. Usando la interfaz gráfica, nessus visualiza un reporte desde un archivo en una sesión previa.

`-T <formato>`. Salva los datos en el formato definido, los cuales pueden ser "nbe", "html", "html_graph", "text", "xml", "old-xml", "tex" o "nsr".

Manejo de Usuarios

La herramienta `nessus-adduser(8)` crea un nuevo usuario de nessus.

Instalación

Nessus se compone de varios elementos que deben ser instalados individualmente en cierto orden o se puede utilizar un script de instalación que facilita el proceso. A continuación se describe el procedimiento de instalación de cada uno de los elementos.

1. Instalación de bibliotecas.

```
$ tar zxf nessus-libraries-2.0.8a.tar.gz
$ cd nessus-libraries
$ ./configure
$ make
$ make install
```

Agregar la línea `/usr/local/lib` en el archivo `/etc/ld.so.conf` y ejecutar el comando `ldconfig`.

2. Instalación de NASL

```
$ tar zxf libnasl-2.0.8a.tar.gz
$ cd libnasl
$ ./configure
$ make
$ make install
```

3. Instalación de Programas

```
$ tar zxf nessus-core-2.0.8a.tar.gz
$ cd nessus-core
$ ./configure
$ make
$ make install
```

4. Instalación de Plugins

```
$ tar zxf nessus-plugins-2.0.8a.tar.gz
$ cd nessus-plugins
$ ./configure
$ make
$ make install
```

5. Habilitar las bibliotecas. Se debe verificar que exista la línea `/usr/local/lib` en el archivo `/etc/ld.so.conf`. Si no es así, agregar la línea y ejecutar `ldconfig`.

6. Crear el certificado. Es necesario crear un certificado para el servidor. En seguida se muestra este procedimiento.

```
$ nessus-mkcert

-------------------------------------------------------
            Creation of the Nessus SSL Certificate
-------------------------------------------------------

This script will now ask you the relevant information to
create the SSL certificate of Nessus. Note that this
information will *NOT* be sent to anybody (everything stays
local), but anyone with the ability to connect to yourNessus
daemon will be able to retrieve this information.


CA certificate life time in days [1460]:
Server certificate life time in days [365]:
Your country (two letter code) [FR]: MX
Your state or province name [none]: DF
Your location (e.g. town) [Paris]: CU
Your organization [Nessus Users United]: UNAM, DGAE, UAP



-------------------------------------------------------
              Creation of the Nessus SSL Certificate
-------------------------------------------------------

Congratulations. Your server certificate was properly created.

/usr/local/etc/nessus/nessusd.conf updated

The following files were created :

. Certification authority :
   Certificate = /usr/local/com/nessus/CA/cacert.pem
   Private key = /usr/local/var/nessus/CA/cakey.pem

. Nessus Server :
    Certificate = /usr/local/com/nessus/CA/servercert.pem
    Private key = /usr/local/var/nessus/CA/serverkey.pem

Press [ENTER] to exit
```

7. Agregar un nuevo usuario para que interactue con el servidor.

```
Using /var/tmp as a temporary file holder

Add a new nessusd user
----------------------


Login : fzaragoz
Authentication (pass/cert) [pass] :
Login password : nessus

User rules
----------
nessusd has a rules system which allows you to restrict the
hosts that fzaragoz has the right to test. For instance, you
may want him to be able to scan his own host only.

Please see the nessus-adduser(8) man page for the rules
syntax

Enter the rules for this user, and hit ctrl-D once you are
done : (the user can have an empty rules set)

Login                : fzaragoz
Password             : nessus
DN                   :
Rules                :

Is that ok ? (y/n) [y] y
user added.
$
```

Para crear un usuario se solicita nombre de usuario, tipo de autenticación (certificado o contraseña), si se selecciona contraseña, el programa solicita una contraseña. También solicita que se indiquen las reglas que aplicarán al usuario que se está creando. La regla tiene el siguiente formato: `clave IP/mascara`, donde la clave es alguno de los siguientes valores: deny, accept o default. Además, se puede utilizar el signo de exclamación (!) para negar una o parte de una regla.

Uso de nessus

Una vez que se ha configurado nessus y se ha creado un usuario, ya es posible utilizarlo de la siguiente forma.

- Ejecutar el demonio.

```
nessusd -D &
```

Con esto se ejecuta el demonio en el puerto 1241.

- En la cuenta de usuario, se ejecuta el cliente `nessus` y se abre la ventana como se muestra en la figura 4.3.

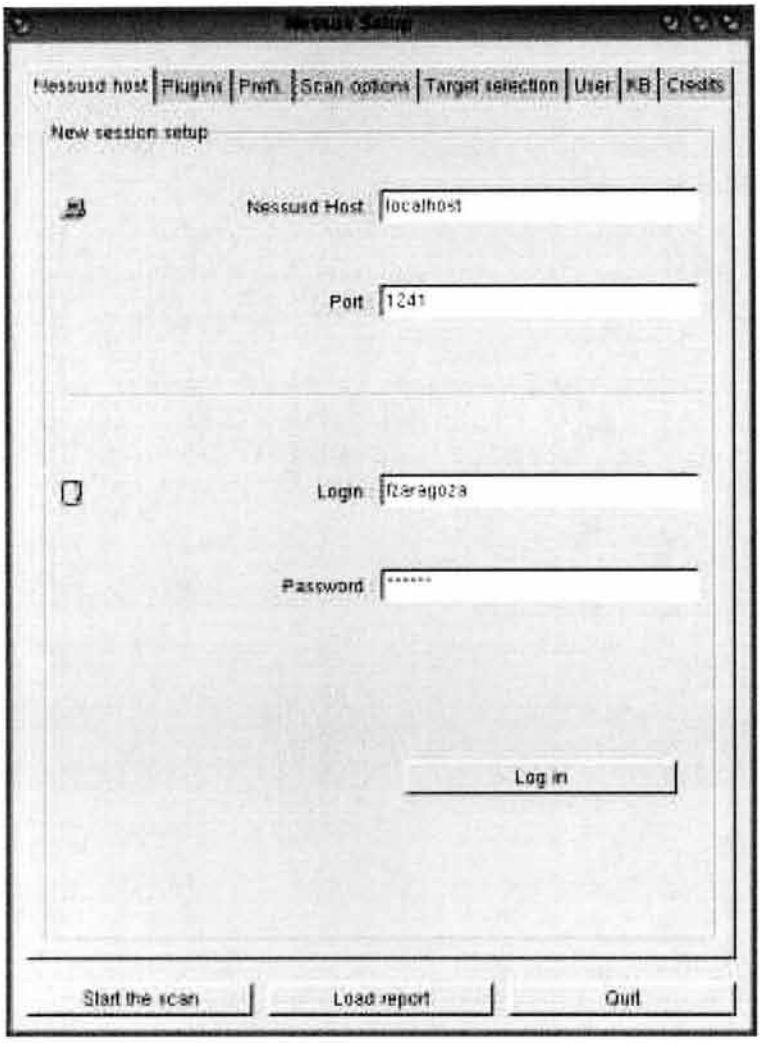

Figura 4.3: Programa cliente de Nessus.

Se debe proporcionar la contraseña y presionar el botón etiquetado como "Log in".

- Si la contraseña proporcionada es correcta, se muestra una ventana para seleccionar qué hacer con el certificado del servidor como se muestra en la figura 4.4.

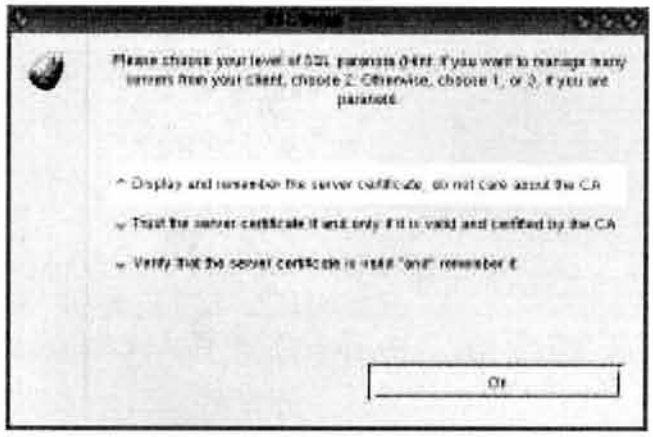

Figura 4.4: Opciones del certificado del servidor de Nessus.

- Posteriormente se muestra el certificado del servidor. Ésto se hace con el fin de autenticar al servidor. (Figura 4.5).
- En seguida se habilitarán los distintos paneles de nessus.

  Los distintos paneles que se muestran son:

  - Sección "Nessusd host". En esta sección se debe ingresar el nombre del equipo ejecutando `nessusd` asi como el puerto. También se debe ingresar el usuario y la contraseña. Una vez hecho ésto, se debe dar un clic en el botón "Log in" el cual establecerá la conexión con el servidor de nessus.
  - Sección "Plugins". Una vez que se ha realizado el registro adecuadamente en el servidor remoto de nessus, esta sección especifica la lista de pruebas que puede desarrollar el servidor. (Figura 4.6).

    Este panel está dividido en dos partes: la familia de plugins y los plugins en sí mismos. Si se da un clic en el nombre de un plugin, entonces aparecerá un diálogo con la descripción del mismo.
  - Sección "Prefs". En este panel se define como hacer ciertas actividades al desarrollar las pruebas. Se indica el puerto al cual se le hace ping, la técnica de rastreo de TCP (connect(), SYN, FIN, Xmas, etc), servidor ftp, servidor de samba. (Figura 4.7).
  - Sección "Scan option". En esta sección se indica un rango de puertos, un número máximo de hilos y el directorio donde se buscarán los CGI y el programa para realizar el rastreo de puertos, el scanner predeterminado es nmap. (Figura 4.8).
  - Sección "Target Selection". En esta sección se debe ingresar el objetivo primario. Un objetivo puede ser un equipo (noether.posgrado.unam.mx), una IP (132.248.10.2), una sub red (132.248.124/24) o una lista de equipos separados por comas (132.248.10.2, 132.248.124/24, www.unam.mx). (Figura 4.9).

Figura 4.5: Certificado del servidor de Nessus.

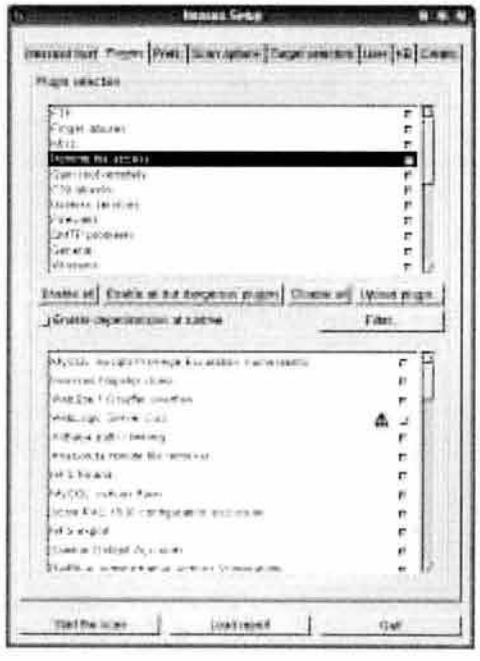

Figura 4.6: Panel *Plugins* de Nessus.

Figura 4.7: Panel *Prefs* de Nessus.

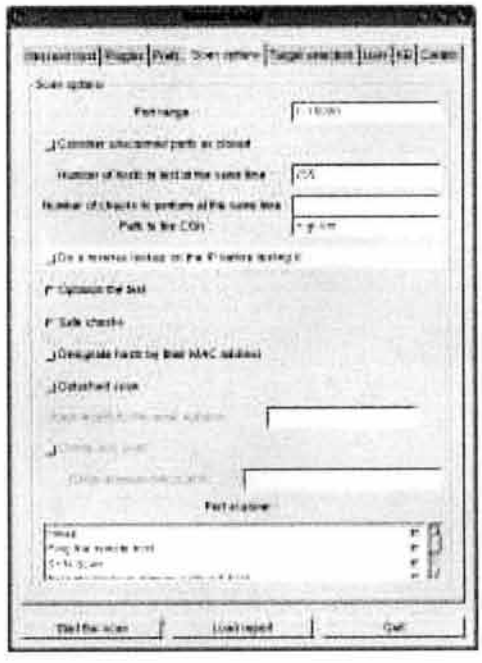

Figura 4.8: Panel *Scan option* de Nessus.

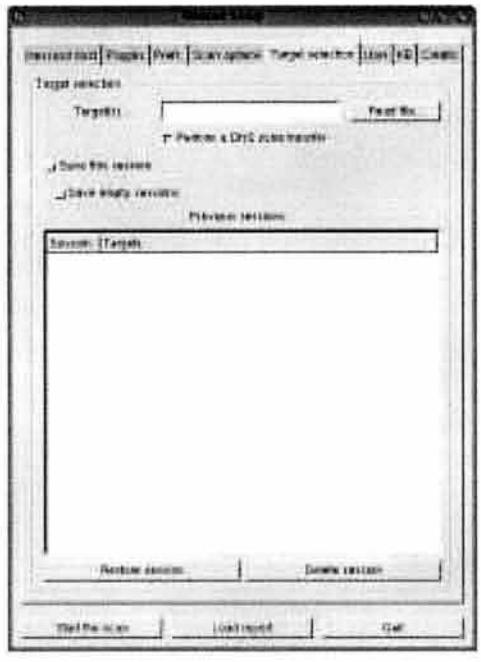

Figura 4.9: Panel *Target Selection* de Nessus.

Este panel permite habilitar la opción "Perform a DNS zone transfer". Esta opción es peligrosa y debe ser habilitada con precaución. Si se desea probar www.nessus.org y la opción está habilitada, nessusd intentará obtener la lista de equipos en el dominio "nessus.org".

- Sección "User". En este panel se puede configurar algunas reglas para el usuario. (Figura 4.10).
- Sección KB (Knowledge Base). La base de conocimiento permite tomar referencia de las búsquedas previas. (Figura 4.11).
- Sección "Credits". Muestra la versión y los autores de Nessus. (Figura 4.12).

- Una vez definido el o los equipos que se desean probar, se da un clic en el botón "Start scan" para iniciar las pruebas. La figura 4.13 muestra la pantalla que indica el avance de las pruebas.
- Una vez finalizada la revisión de los equipos, nessus muestra una ventana con los resultados como en la figura 4.14.
- Estos resultados pueden ser guardados en distintos formatos. La figura 4.15 muestra la ventana para guardar los reportes.
- La figura 4.16 muestra parte del reporte generado en formato html.

A continuación se muestrán algunas de las gráficas del resumen de los resultados obtenidos por nessus.

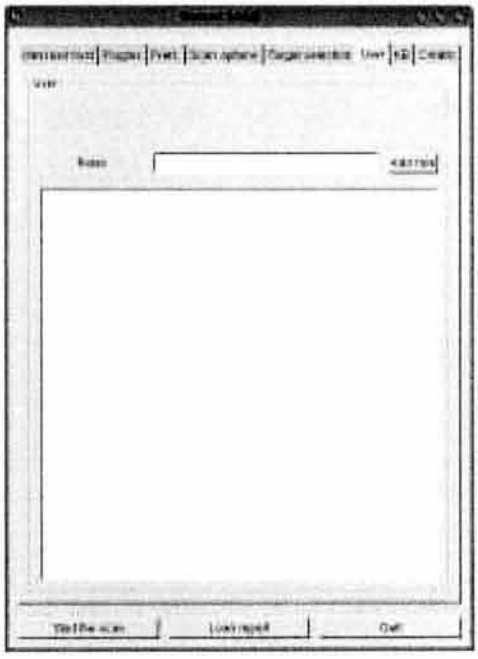

Figura 4.10: Panel *User* de Nessus.

Figura 4.11: Panel *KB (Knowledge Base)* de Nessus.

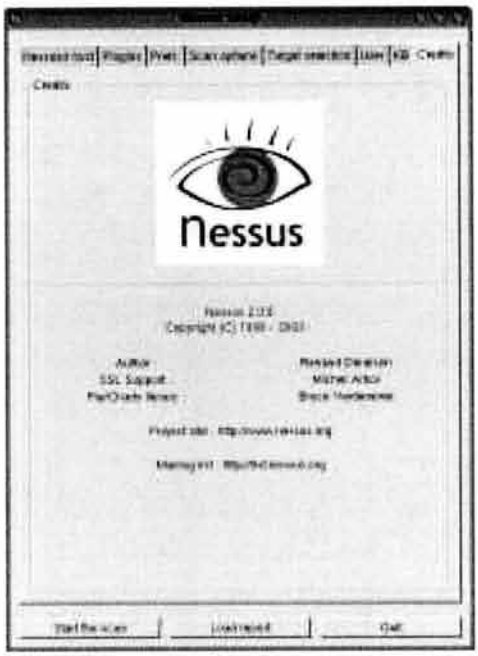

Figura 4.12: Panel *Credits* de Nessus.

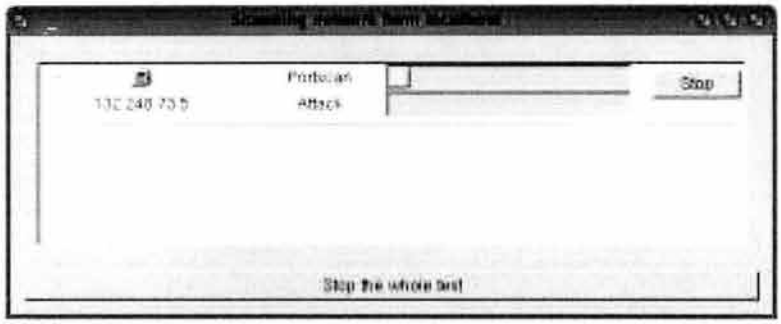

Figura 4.13: Realización de pruebas de Nessus.

Figura 4.14: Resultados de Nessus después de la realización de las pruebas.

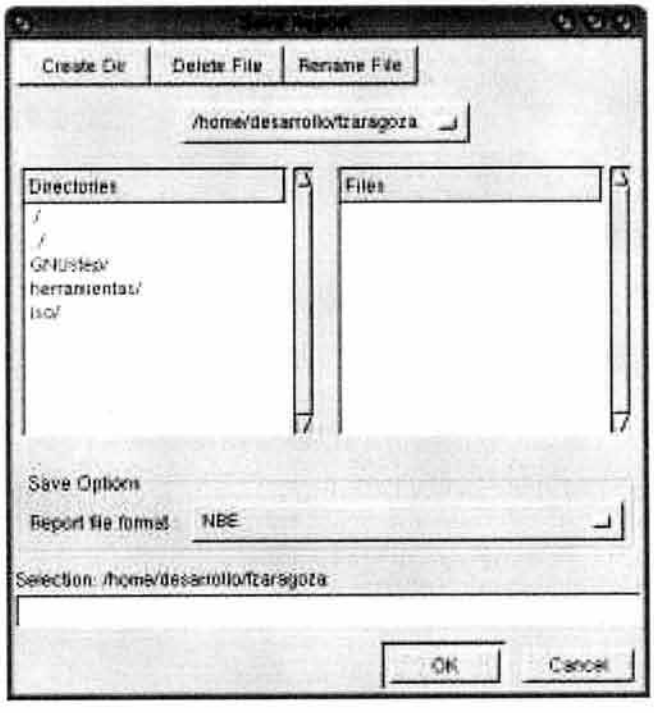

Figura 4.15: Ventana para seleccionar el tipo de reporte de Nessus.

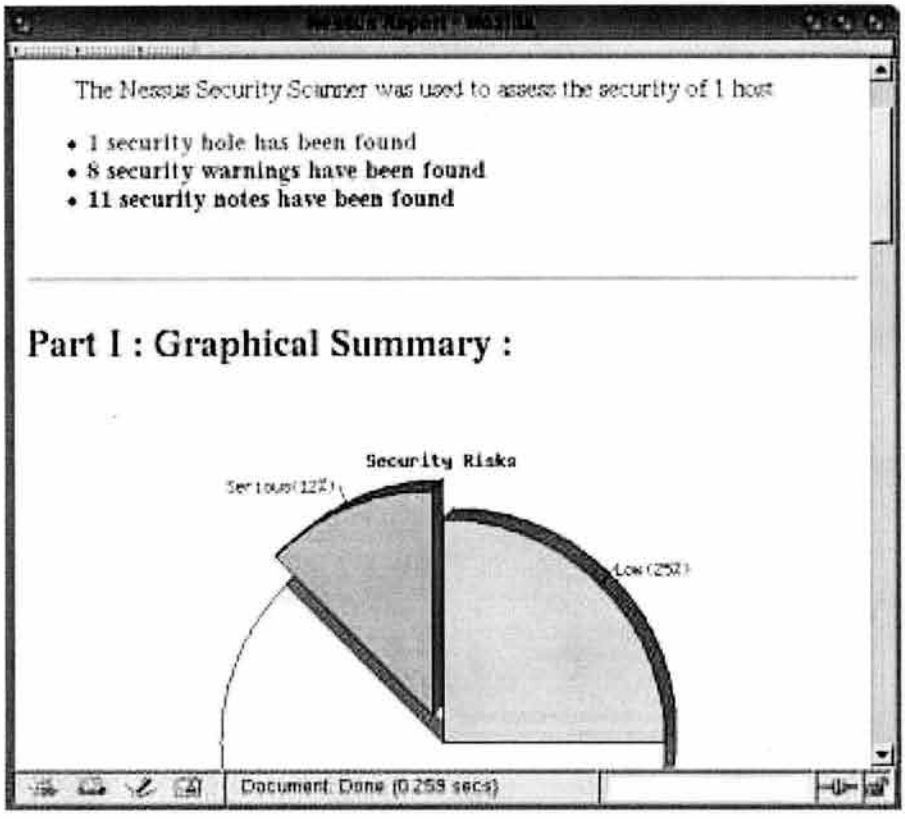

Figura 4.16: Parte del reporte de Nessus en formato html.

- Riesgos de seguridad. (Figura 4.17).
- Servicios mas peligrosos en la red. (Figura 4.18).
- Ocurrencia de servicios en la red. (Figura 4.19)

El Anexo 2 muestra el reporte completo del análisis realizado por nessus al equipo que aloja al SIIEM/UAP.

Como se puede apreciar en el reporte, Nessus no detecta ninguna vulnerabilidad crítica en el sistema, a excepción de un falso positivo en el servidor de https debido a que Nessus no verifica que el parche ya ha sido aplicado.

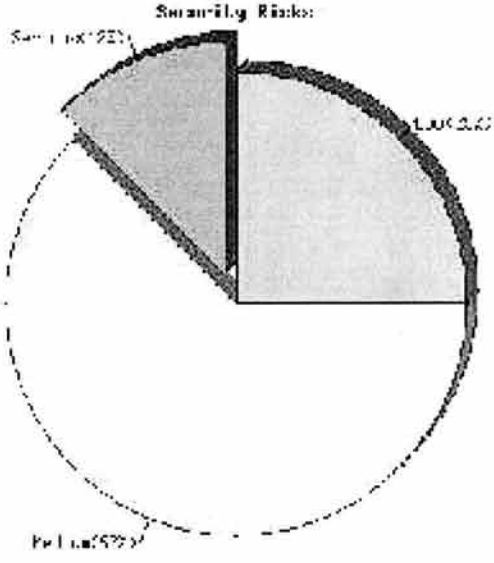

Figura 4.17: Riesgos de seguridad reportados por Nessus.

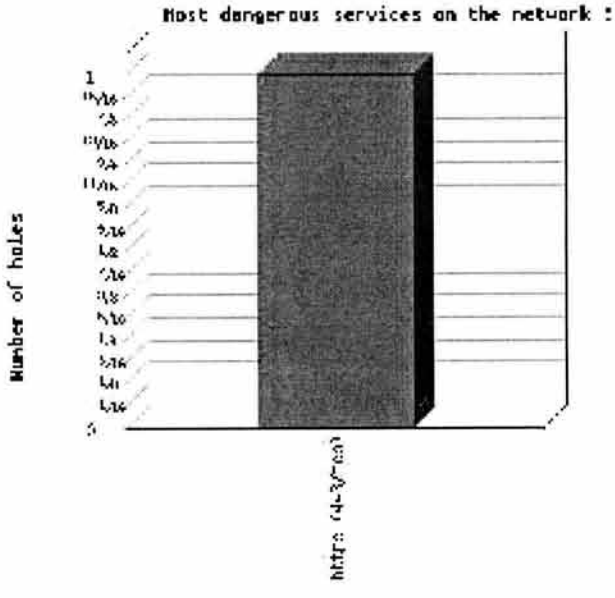

Figura 4.18: Servicios más peligros en la red reportados por Nessus.

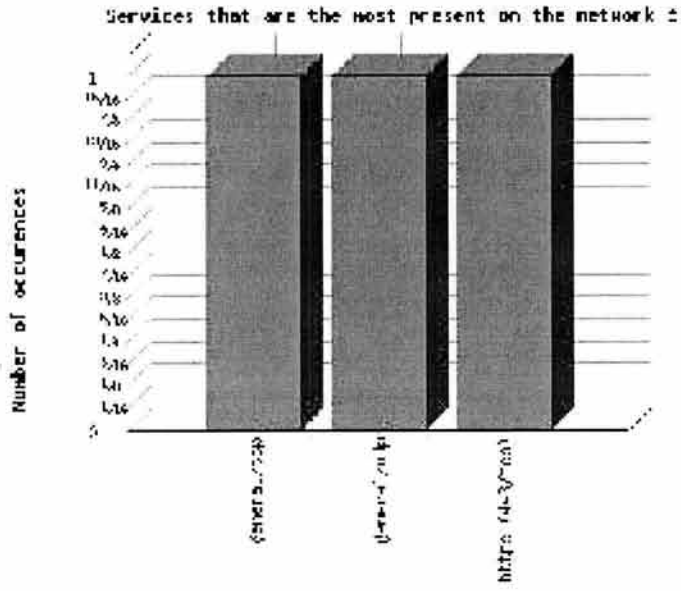

Figura 4.19: Ocurrencia de servicios en la red reportados por Nessus.

4.3.5. Análisis de bitácoras del sistema web

Como se mencionó antes, el sistema de registro `syslog` de los sistemas tipo Unix mantiene un registro de las actividades del sistema en general, sin embargo, el servidor de web Apache mantiene sus propias bitácoras para registrar las actividades en el sistema de páginas web. Estos registros no son mantenidos por `syslog` y por lo tanto las herramientas que permitan analizar las bitácoras de `syslog`, como se verá mas adelante, no están enfocadas a las bitácoras de Apache, por lo que se requiere de una utilería especializada.

Debido a que el acceso público al SIIEM/UAP se realiza mediante el servidor web, es conveniente que se revisen la bitácoras generadas por Apache para comprender desde dónde y cuándo se conectan los usuarios al sistema. Webalizer es una herramienta diseñada específicamente para analizar la bitácora `access_log` y obtener estadísticas de uso del sistema web.

Webalizer

Es un programa para analizar bitácoras de servidores web el cual produce estadísticas de uso en formato HTML para ser vistos mediante un navegador. Los resultados se presentan en formato de tablas y gráficas para su mejor interpretación.

Entre las características mas sobresalientes de Webalizer se encuentran:

- Generación de estadísticas de uso anuales, mensuales, diarias y por hora.

- Despliegue de uso del servidor clasificado por:
  - Sitio.
  - URL.
  - Referencia.
  - Navegador del usuario.
  - Cadenas de búsqueda.
  - Páginas de entrada y salida.
  - Nombre de usuario y
  - País.
- Soporte del formato común de bitácoras (CLF - *Common Log Format*).
- Manejo de bitácoras de otros servicios como FTP (xferlog de wu-ftpd) y Proxy.
- Uso de bitácoras comprimidas con gzip.

Instalación de Webalizer

Como muchas otras herramientas, existen dos formas de instalar Webalizer: compilando el código fuente o utilizando una versión previamente compilada. En este caso veremos el segundo método.

1. Obtener webalizer. Es posible obtener el código precompilado de webalizer y de esta forma se ahorra tiempo en el proceso de compilación. Webalizer puede ser obtenídos desde http://www.webalizer.org/download.html. Para Linux se debe obtener el archivo webalizer-2.01-10-linuxelf-x86-bin.tgz para procesadores Intel.

2. Desempaquetar.
   ```
   $ tar zxf webalizer-2.01-10-linuxelf-x86-bin.tgz
   ```

3. Copiar el archivo de configuración y el archivo binario.

   ```
   $ cd webalizer-2.01-10-linuxelf-x86
   $ cp sample.conf /etc/webalizer.conf
   $ cp webalizer /usr/bin
   ```

4. Modificar el archivo de configuración. Finalmente es necesario hacer algunos cambios menores en los valores predeterminados del archivo de configuración.

   - `LogFile`. Esta variable define el archivo de bitácora que leerá en caso de que no se le proporcione este parámetro. Valor original:
     `#LogFile /var/lib/httpd/logs/access_log`
     Nuevo valor:
     `LogFile /usr/local/apache/logs/access_log`

- `OutputDir`. Define el directorio donde se almacenarán los resultados después del análisis del registro. Valor original:
  `#OutputDir /var/lib/httpd/htdocs/usage`
  Nuevo valor:
  `OutputDir ./usage`

  Con este valor, creará los resultados en el directorio `usage` ubicado en el directorio donde se encuentre posicionado el usuario. Si el directorio no existe, el usuario deberá crearlo.
- `ReportTitle`. Esta variable define el título del reporte. Valor original:
  `#ReportTitle Usage Statistics for`

  Nuevo valor:

  ```
  ReportTitle Estad&iacute;sticas de uso para
  ```

Uso de webalizer

Al ser ejecutado webalizer, lee el archivo de bitácora especificado y produce la salida en formato HTML en el directorio que haya sido configurado. Es posible indicar varias opciones de configuración en la línea de comandos o a través del archivo de configuración como se vió antes. La sintáxis del comando es:

```
$ webalizer [opciones] [archivo_de_bitacora]
```

En seguida se muestran algunas imágenes de los resultados obtenidos con webalizer en el período comprendido entre Octubre del 2002 y Septiembre del 2003.

La figura 4.20 muestra la gráfica anual. En esta gráfica se puede observar que entre marzo y junio se presenta la mayor carga de trabajo en el servidor web según el archivo de bitácora.

Tomando como base el mes de abril del 2003 por ser el que tuvo mayor actividad en todo el año se verán algunos detalles específidos de este mes. La figura 4.21 muestra la estadística detallada de accesos para el mes de abril donde se puede observar que en los primeros días del mes tuvo la mayor cantidad de accesos.

Con webalizer se ha podido determinar que la mayor carga de trabajo durante el día se presenta entre las ocho de la mañana y la una de la tarde como se muestra en la figura 4.22.

Finalmente, la tabla 4.6 muestra las direcciones IP de los primeros cinco equipos que se conectan con mayor frecuencia al servidor web del SIIEM/UAP. Estas direcciones IP pertenecen a la Facultad de Medicina de la UNAM.

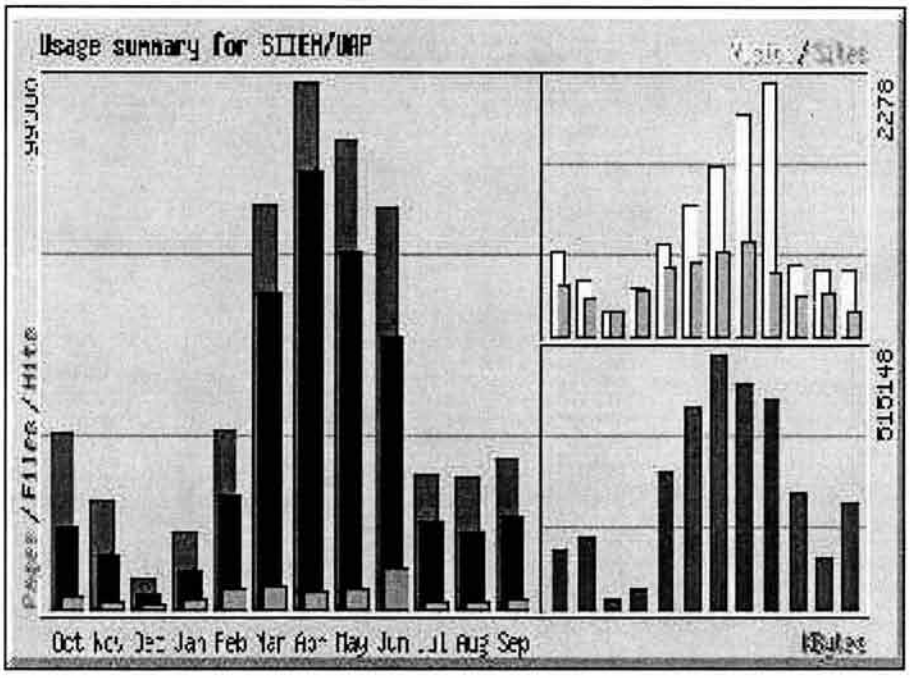

Figura 4.20: Resumen anual de uso del servidor web del SIIEM/UAP.

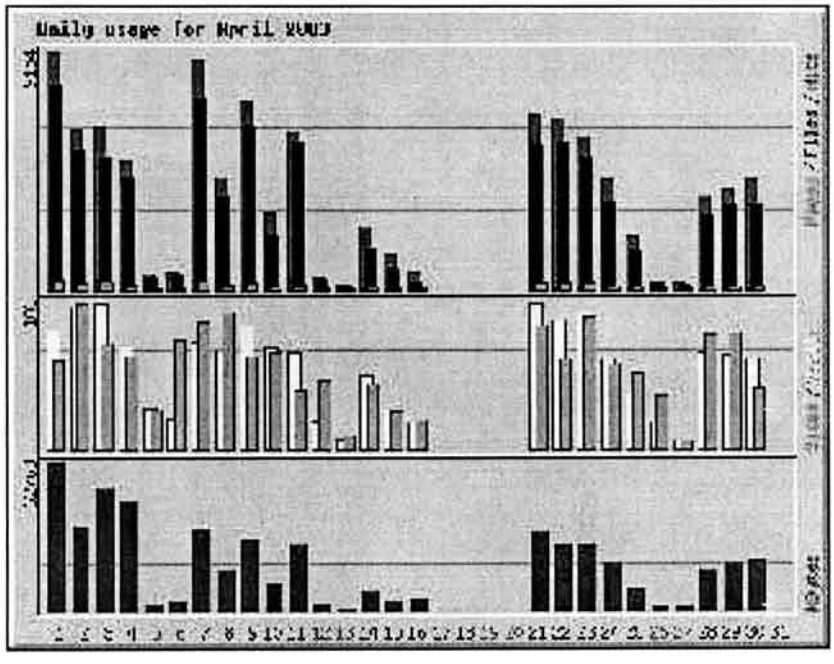

Figura 4.21: Uso diario del servidor web en el mes de abril del 2003 del SIIEM/UAP.

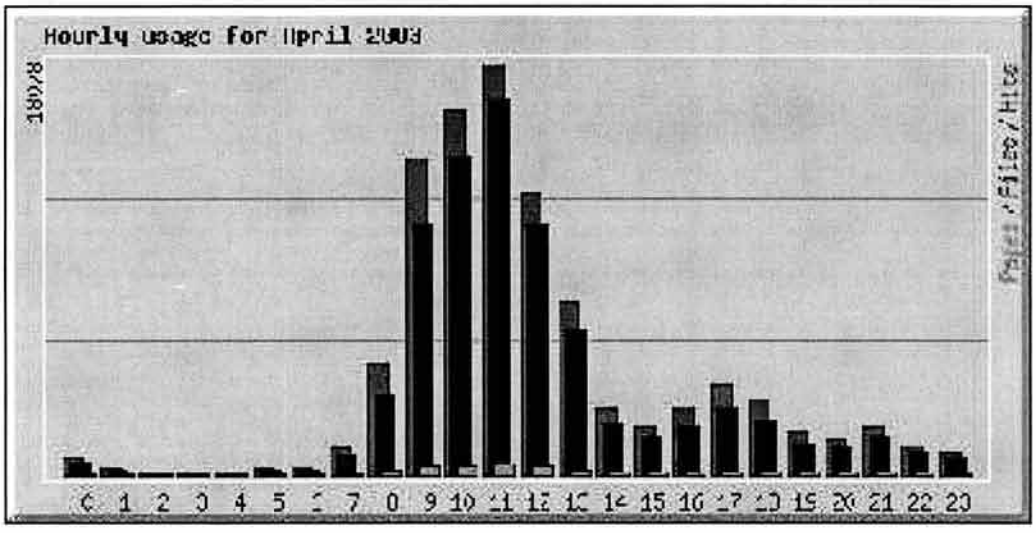

Figura 4.22: Uso por hora del servidor web del SIIEM/UAP.

Número	Dirección IP
1	132.248.250.114
2	132.248.250.109
3	132.248.250.108
4	132.248.250.115
5	132.248.55.198

Cuadro 4.6: Equipos remotos conectados al servidor web del SIIEM/UAP.

4.3.6. Sesión segura entre cliente y el servidor

Un elemento importante dentro de cualquier sistema en Internet es conectarse a él de forma remota, ya sea para obtener información o para realizar el mantenimiento y administración del mismo.

En búsqueda de la mejor forma de realizar la conexión segura hacia al sistema, se deben omitir protocolos inseguros y vulnerables a ataques. Telnet es un protocolo que permite realizar sesiones remotas entre sistemas tipo Unix, sin embargo presenta un gran problema: la información viaja en "claro". Que la información viaje en claro significa que cualquier dispisitivo escuchando en la red (estos dispositivos en el argot de redes y seguridad son conocidos como *sniffers*) pueden capturar la información que viaja en el medio y encontrar información valiosa como contraseñas de usuario. También existen comandos conocidos como comandos "r" que basados en mecanismos de confianza envian información que puede ser utilizada por un intruso para vulnerar un sistema.

Una solución a este problema es Secure Shell, un protocolo que permite la comuni-

cación entre sistemas remotos utilizando cifrado de llave pública y llave privada. La siguiente sección cubre de forma breve la implementación libre de este protocolo: Open Secure Shell.

Open Secure Shell

Open Secure Shell (OpenSSH para abreviar) implementa el protocolo SSH2 definido por la Fuerza de Tarea de Ingeniería de Internet (IETF - *Internet Engineering Task Force*). La estructura general de SSH2 se compone de tres niveles:

- Transporte. Esta capa se encarga de la negociación del algoritmo y el intercambio de la llave. El intercambio de la llave incluye la autenticación del servidor, obteniendo una conexión criptográficamente segura. Esta capa provee integridad, confidencialidad y compresión de datos.
- Autenticación del usuario. Utiliza la conexión establecida por la capa de transporte y utiliza los servicios provistos por ésta. Esta capa incluye varios mecanismos para la autenticación del usuario como contraseña tradicional, así como mecanismos de autenticación de llave pública o basados en el host.
- Conexión. Genera distintos canales concurrentes sobre la conexión autenticada y permite el paso por túneles de las sesiones de ingreso y del reenvío por TCP. Provee un servicio de control de flujo para estos canales, además de que es posible negociar distintas opciones para los canales.

Instalación de OpenSSH

El proceso de Intalación de OpenSSH es muy sencillo si se realiza por medio de la compilación del sistema. A continuación se describen este procedimiento.

1. Obtener OpenSSH y OpenSSL. Es necesario obtener el código fuente de las siguientes direcciones en Internet:
   http://www.openssh.org y http://www.openssl.org
2. Instalar OpenSSL.
   a) Desempaquetar el código fuente.
   ```
   $ tar zxf openssl-version.tar.gz
   ```
   b) Configurar.
   ```
   $ cd openssl-version
   $ ./config
   ```
   c) Compilar y probar.
   ```
   $ make
   $ make test
   ```
   d) Instalar
   ```
   $ make install
   ```

3. Instalar OpenSSH

   - Desempaquetar el código fuente.
     ```
     $ tar zxf openssh-version.tar.gz
     ```
   - Configurar. Se configura OpenSSH indicando la ruta del recién intalado OpenSSL, habilitando el uso de contraseñas md5 y con soporte para tcp-wrappers.
     ```
     $ cd openssh-verison
     $ ./configure --with-md5-passwords --with-tcp-wrappers
       --with-ssl-dir=/usr/local/ssl/
     ```
   - Compilar.
     ```
     $ make
     ```
   - Instalar.
     ```
     $ make install
     ```

Uso de OpenSSH

Ahora que ya está instalado OpenSSH, se puede utilizar para que sea posible el inicio de sesiones remotas seguras en el servidor.

OpenSSH se compone de distintos programas, a continuación se muestra una descripción breve de cada uno de ellos.

- `ssh`. Es el programa básico que se encarga de realizar las sesiones remotas. Es el sustituto a los comandos inseguros `rlogin` y `rsh`.
- `sshd`. Es el demonio de OpenSSH que permite las conexiones remotas al servidor local.
- `ssh-agent`. Es el agente de autenticación con la capacidad de guardar llaves privadas.
- `ssh-add`. Programa útil para agregar llaves en `ssh-agent`.
- `sftp`. Programa tipo FTP que funciona con los protocolos SSH1 y SSH2.
- `scp`. Programa que permite hacer la copia de archivos de forma segura entre sistemas remotos. Este programa sustituye al comando inserguro `rcp`.
- `ssh-keygen`. Herramienta que se encarga de generar las llaves.
- `sftp-server`. Servidor de FTP seguro iniciado automáticamente por el demonio `sshd`.
- `ssh-keyscan`. Programa encargado de recolectar llaves públicas de sistemas remotos.

- `ssh-keysign`. Programa de ayuda para autenticación basada en el host.

En mi experiencia, el usuario no interactua con todos estos comandos, sólo con algunos de ellos. A continuación detallo estos comandos.

`ssh`

Es el cliente de OpenSSH para iniciar una sesión y ejecutar comandos en un sistema remoto. Este programa es el reemplazo de los comandos rlogin y rsh y provee comunicacones cifradas seguras entre dos equipos no confiables sobre una red insegura.

La sintáxis de este comando se muestra a continuación:

```
ssh [-l usuario] equipo_remoto | usuario@equipo_remoto [comando]

ssh [-afgknqstvxACNTVX1246] [-b direccion] [-c tipo_cifrado]
    [-e carac_escape] [-i archivo_de_identidad] [-l usuario]
    [-m espec_mac] [-o opcion] [-p puerto] [-F archivo_conf]
    [-L puerto:equipo:puerto] [-R puerto:equipo:puerto]
    [-D puerto] equipo_remoto | usuario@equipo_remoto [comando]
```

Cuando un usuario se conecta utilizando la versión 2 del protocolo de SSH primero se realiza el proceso de autenticación. En primer lugar el usuario intenta autentificarse usando el método basado en el equipo; si este método falla entonces se utiliza la autenticación de llave pública y finalmente, si este método también falla se utilizará el método interactivo y autenticación a través de contraseña.

Cuando la identidad del usuario es aceptada por el servidor, el servidor ejecuta el comando dado, o le proporciona al usuario un shell en la máquina remota. *Toda la comunicación con el comando remoto o el shell se cifrarán automáticamente.*

La sesión finaliza cuando el comando o el shell en la máquina remota termina y todas las conexiones X11 y TCP/IP son cerradas. El estado de la salida del programa remoto es el estado de salida de `ssh`.

`ssh` mantiene y revisa automáticamente una base de datos que contiene la identificación de todos los equipos a los cuales se ha conectado. Las llaves de los equipos se almacenan en el archivo `.ssh/known_hosts` en el directorio de trabajo del usuario. Además, el archivo `/usr/local/etc/ssh_known_hosts` es revisado por `ssh` en búsqueda de equipos conocidos. Cualquier equipo nuevo es agregado automáticamente al archivo del usuario. Si la identificación de un equipo cambia, `ssh` alerta acerca de ésto y deshabilita la autenticación mediante contraseña para prevenir un caballo de troya que intente obtener la contraseña del usuario. Otro propósito de este mecanismo es la prevención de ataques de hombre en medio.

Algunas de las opciones más importantes son.

- `-c blowfish|3des|des`. Selecciona el algoritmo de cifrado para la sesión. El algoritmo predeterminado es 3des.

- `-p puerto`. Especifica el puerto para conectarse en el sistema remoto.
- `-v`. Ocasiona que `ssh` muestre mensajes de depuración sobre el proceso de conexión.
- `-V`. Despliega la versión.
- `-1`. Provoca que `ssh` utilice la versión 1 del protocolo.
- `-2`. Obliga a `ssh` a usar la versión 2.
- `-4`. Utiliza IPv4.
- `-6`. Usa direccionamiento IPv6.

Un ejemplo para conectarse a un sistema remoto se muestra a continuación:

```
$ ssh -l usuario@equipo_remoto.com
The authenticity of host 'equipo_remoto.com (xxx.yyy.zzz.1)'
can't be established.
RSA key fingerprint is 1b:71:56:80:2f:36:20:ab:10:4f:a0:85.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'equipo_remoto.com' (RSA) to the
list of known hosts.
usuario@equipo_remoto.com password:

Ya entre
-bash-2.05b$
```

El uso del comando ssh es muy frecuente en el SIIEM/UAP debido a que es el medio utilizado para conectarse al servidor y realizar consultas, configuración y administración al mismo.

`scp`

El comando scp realiza la copia de archivos entre equipos a través de la red. `scp` utiliza `ssh` para la transferencia de datos y usa la misma autenticación y provee la misma seguridad que `ssh`. A diferencia de `rcp`, `scp` solicita una contraseña o clave de paso si son necesarias para la autenticación.

La sintáxis de este comando es la siguiente:

```
scp [-pqrvBC1246] [-F archivo_conf] [-S programa] [-P puerto]
    [-c tipo_cifrado] [-i arch_identidad] [-l limite] [-o opcion]
    [[usuario@equipo1:]archivo1 [...] [[usuario2@equipo2:]archivo2
```

Algunas opciones importantes se describen a continuación.

- `-l limite`. Limita el ancho de banda utilizado. El límite se define en Kilo bits por segundo (Kb/s).
- `-p`. Mantiene el tiempo de modificación y acceso asi como los permisos originales del archivo.
- `-r`. Realiza una copia recursiva.
- `-v`. Envia mensajes de depuración.
- `-B`. Selecciona el modo por lotes (previene la solicitud de contraseña).
- `-C`. Habilita la compresión de datos.
- `-P puerto`. Especifica el puerto del sistema remoto.

`sftp`

Este es un programa interactivo para la transferencia de archivos similar a ftp, sin embargo, sftp utiliza todas las características de cifrado proporcionadas por `ssh`.

`sshd`

Este programa es el demonio de OpenSSH que está en espera de conexiones de los clientes. Este es iniciado generalmente al iniciar también el sistema en general. Cada vez que hay una nueva conexión, se crea un nuevo demonio para atender la petición. El demonio recién creado se encarga de manejar el intercambio de llaves, cifrado, autenticación, ejecución de comandos e intercambio de llaves. La implementación de OpenSSH soporta los protocolos 1 y 2.

Una vez validado el usuario, el proceso para iniciar la sesión de un usuario es el siguiente:

1. Si se solicita una sesión de terminal, en lugar de la ejecución de un comando, se muestra la última vez que se conectó el usuario y también despliega el contenido del archivo `/etc/motd`.
2. Si la sesión es sobre una terminal, se registra el tiempo de inicio de sesión.
3. Revisa el archivo `/etc/nologin`, si existe imprime el contenido y termina la conexión.
4. Cambia para ser ejecutado con privilegios de usuario normal.
5. Configura un ambiente básico para el usuario.
6. Lee el archivo `.ssh/environment` del usuario.
7. Cambia al directorio de trabajo del usuario.

8. Si existe, se ejecuta el archivo del usuario `.ssh/rc` ; además, si existe el archivo `/etc/ssh/sshrc` también es ejecutado; de otra manera ejecuta `xauth`.

9. Ejecuta el shell o el comando del usuario.

Con OpenSecure Shell se aseguran conexiones cifradas entre el servidor que aloja el SIIEM/UAP y los sistemas autorizados para conectarse remótamente.

Tcp-wrappers

Tcp-wrappers es una herramienta libre para monitorear y controlar el tráfico de llegada a la red. Como se vió en el proceso de instalación de OpenSSH, se incluyó el soporte de Tcp-wrappers mediante la instrucción `--with-tcp-wrappers`. Con el uso de esta herramienta se reforzará el control de acceso a secure shell (aunque ésto ya lo realiza el firewall, se agregó esta característica al servicio para mayor seguridad).

Mediante reglas, Tcp-wrappers controla el acceso a las aplicaciones de acuerdo al siguiente formato:

```
demonio:equipos[:comando-shell]
```

`demonio` define los nombres de los demonios de red (servicios). `equipos` es una lista de nombres, direcciones o patrones de equipos que serán comparados con el nombre o dirección del equipo cliente que solicita el servicio. `comando-shell` es un programa que puede ser ejecutado cuando exista alguna coincidencia.

Se utilizan distintos comodines para el control de acceso:

- ALL. Este comodín sirve para indicar cualquier nombre de demonio de red y/o equipo.
- LOCAL. Este comodín coincide con cualquier cadena que no contenga el caracter ".", su principal uso se encuentra en la lista de equipo.
- EXCEPT. Este es un operador que se usa de la siguiente manera: lista1 EXCEPT lista2, lo cual significa "lo que está en la lista1 excepto lo de la lista2".

Existen dos archivos en el cual se definen las reglas de Tcp-wrappers.

- `/etc/hosts.allow`. Este archivo especifica los servicios y equipos autorizados. Este es el primer archivo que lee tcp-wrappers para realizar una verificación de acceso.
- `/etc/hosts.deny`. En este archivo se definen las reglas para negar el acceso a los demonios y equipos. Este es el segundo archivo en ser leído por tcp-wrappers.

En el SIIEM/UAP estos archivos tienen el siguiente contenido:

Archivo `/etc/hosts.allow`. En este archivo se permite el acceso al demonio de secure shell (`sshd`) a los equipos con la dirección IP 132.248.73.7 y 132.248.73.10.

```
sshd: 132.248.73.7, 132.248.73.10
```

Archivo `/etc/hosts.deny`. En este archivo se niega el acceso a cualquier otro servicio configurado con tcp-wrappers, en el caso del SIIEM/UAP sólo secure shell tiene esta característica.

```
ALL:ALL
```

Con todo lo anterior se definen los medios necesarios para realizar la comunicación segura con el servidor, principalmente para la administración del mismo. A continuación planteo un tema muy importante que permite identificar anomalías en el sistema a través de la detección de intrusos.

4.3.7. Detección de Intrusos

En la actualidad muchos negocios y hogares instalan alarmas para avisar cuando existe algún intento de robo o destrucción de algún bien valioso, o simplemente para alertar de que alguien está dentro del lugar sin autorización. Estos sistemas de alarma pueden notificar a las autoridades cuando ocurre alguna intrusión. Tales medidas son necesarias para asegurar la integridad de los hogares y la confianza de los dueños.

El mismo nivel de seguridad debe ser aplicado también a los sistemas de computadoras. Internet ha facilitado el flujo de información y la comunicación entre usuarios a través de todo el mundo; sin embargo también presenta amenazas hacia sus usuarios. Crackers o usuarios maliciosos buscan equipos vulnerables tales como sistemas sin actualizar, sistemas infectados con algún troyano y redes ejecutando servicios inseguros. Las alarmas son necesarias para notificar a los administradores y a los miembros del grupo de seguridad cuando ocurra un evento sospechoso para que ellos puedan responder en tiempo real a la amenaza. Los Sistemas de Detección de Intrusos (IDS *Intrusion Detection System*) son diseñados para funcionar como sistemas de alarma.

Un sistema de detección de intrusos (IDS) es un proceso o dispositivo activo que analiza la actividad del sistema y de la red en búsqueda de actividad innesperada e ilícita. La forma en la que el IDS detecta anomalías puede variar, sin embargo, el objetivo es el mismo, agarrar a los intrusos en el acto antes de que hagan un daño real en los recursos.

Los IDS protegen un sistema de ataques, mal uso, y compromiso. También pueden monitorear la actividad de la red, auditar la red y la configuración del sistema en búsqueda de vulnerabilidades, analizar la integridad de los datos, entre otras cosas.

Tipos de Detectores de Intrusos

Detectores de intrusos basados en el anális del equipo

Los detectores de intrusos basados en el equipo analizan muchas áreas para determinar el mal uso (actividad maliciosa o abusiva dentro de la red) o intrusión (intentos de abuso externo). Los IDS de este tipo revisan varios tipos de bitácoras (kernel, sistema, red, firewall, etc.) y comparan los registros contra una base de datos interna de firmas de ataques conocidos. IDS basados en el equipo en sistemas tipo Unix hacen uso del sistema de registros `syslog` y su capacidad para separar eventos registrados por su severidad y funcionalidad (por ejemplo mensajes de error de impresora contra alertas del kernel). Los IDS filtran los registros (como es el caso de algunos eventos como los de red y el kernel que pueden ser muy explícitos), los analizan, revisan los paquetes anómalos con su propio sistema de alertas y tazas de serveridad, y los recolectan en sus propias bitácoras para el análisis por parte del administrador.

Existen distintas herramientas libres para realizar este tipo de análisis como: logcheck, logwatch y swatch.

Los IDS basados en el equipo también pueden verificar la integridad de archivos importantes y programas ejecutables. El IDS revisará una base de datos de archivos sensitivos (y cualquier archivo que se desee agregar) y crear una suma de cada archivo con una firma generada con herramientas como `md5sum` (algoritmo de 128 bits) o `sha1sum` (algoritmo de 160 bits). El IDS almancena la suma en un archivo de texto plano y periódicamente compara cada archivo contra los valores almacenados en el archivo de texto. Si la suma de algún archivo no coincide, entonces el IDS alertará al administrador a través del correo electrónico. Este es el proceso utilizado por Tripwire discutido en la sección *Integridad del sistema de archivos* de este capítulo.

Detectores de intrusos basados en el análisis de la red

Los sistemas de detección de intrusos basados en el análisis de la red (NIDS - *Network IDS*) trabajan de forma distinta a los vistos anteriormente. Este tipo de detectores de intrusos funcionan mediante el análisis de los paquetes de red a nivel del ruteador o del equipo, auditando la información de los paquetes y registrando cualquier paquete sospechoso en un archivo especial con información extendida. En base a los paquetes sospechosos, el NIDS busca en su propia base de datos de firmas de ataques conocidos y asigna un nivel de severidad para cada paquete. Si el nivel de serveridad es lo suficientemente alto, se envia un mensaje de correo electrónico a los miembros del equipo de seguridad para que proceda una investigación de la anomalía.

Los NIDS han llegado a ser muy populares tanto como el crecimiento en tamaño y tráfico de Internet. Aquellas implementaciones que pueden analizar grandes cantidades de tráfico y determinan ataques potenciales son muy utilizados en el área de seguridad.

La mayoría de los NIDS necesitan que el disposivo de red se encuentre en modo promiscuo, lo que permite que el dispositivo capture todos los paquetes a través de la red.

Existen herramientas como `tcpdump` para capturar los paquetes de la red, sin embargo, no es considerado como un IDS debido a que no realiza análisis alguno sobre los paquetes.

Snort [8] es un IDS completo que realiza efectivamente el registro de actividad sospechosa en la red y notifica a los administradores cuando existe algún problema potencial. Snort utiliza la biblioteca estándar `libcap` y `tcpdump` como base para el registro de paquetes.

La característica más importante de Snort no es su funcionalidad, si no su subsistema de firmas de ataques que resulta ser muy flexible. Constantemente se actualiza la base de datos de ataques que pueden ser agregados y actualizados a través de Internet. Los usuarios también pueden crear sus firmas y enviarlas a la lista de correo de firmas de Snort para su análisis, y así todos los usuarios se beneficiarán. Esta ética comunitaria de compartir ha hecho de Snort uno de los detectores de intrusos a nivel de red más actualizado y robusto.

A continuación se analizarán dos IDS que propongo y utilizo en el SIIEM/UAP: LogWatch y LIDS.

LogWatch

LogWatch es un analizador de registros del sistema y reporteador que puede ser catalogado como un detector de intrusos basado en el equipo.

Entre las características de LogWatch se encuentran:

- Configurable. Es posible definir distintos parámetros tanto en la línea de comandos o a través de un archivo de configuración.
- Versátil. Es posible indicar distintos rangos y niveles de análisis de los registros del sistema.
- Fácil de usar. Puede utilizar los paquetes de muchos sistemas tipo Unix, incluido Linux.

Uso de LogWatch

LogWatch se utiliza a través del programa `logwatch` en la línea de comandos.

La sintaxis en la línea de comandos es la siguiente:

[8] http://www.snort.org

```
logwatch [--detail nivel] [--logfile grupo] [--service servicio]
     [--print] [--mailto direccion] [--archives] [--range rango]
     [--debug nivel] [--save archivo] [--help|--usage]
```

A continuación se describen brevemente cada una de las opciones.

`--detail nivel`. Este es el nivel de detalle del reporte. Los niveles pueden ser: `high`, `med`, `low`.

`--logfile grupo`. Esta opción forzará a LogWatch a procesar sólo un conjunto de archivos de registro definidos por `grupo` (por ejemplo messages, xferlog, etc).

`--service servicio`. Con esta opción se obliga a LogWatch a procesar sólo el servicio especificado (por ejemplo login, pam, etc.). Se pueden especificar varios servicios o utilizar el parámetro `All` el cual incluye todos los servicios.

`--print`. Muestra el reporte en la salida estándar (generalmente la pantalla).

`--mailto direccion`. Envia el resultado del análisis por correo electrónico.

`--archives`. Cada grupo de archivos de registro tiene archivos definidos (por ejemplo /var/log/messages) y también tiene respaldos (/var/log/messages.? o /var/log/messages.?.gz). Con esta opción LogWatch buscará a través de los respaldos además de los archivos regulares. Sin embargo, las entradas deben estar en un rango apropiado para ser procesadas.

`--range rango`. Se puede especificar un rango de fechas para procesar. Esta opción está limitada actualmente a los parámetros `Yesterday`, `Today` y `All`.

`--debug nivel`. Esta opción se utiliza para propósitos de depuración. El nivel va de 0 a 100.

`--save archivo`. Guarda el reporte en un archivo en lugar de desplegarlo en pantalla o enviarlo por correo electrónico.

`--usage`. Despliega información de cómo usar el programa.

`--help`. Lo mismo que `--usage`.

El siguiente ejemplo muestra un reporte detallado de LogWatch de un día anterior:

```
# logwatch --detail high --range Yesterday --print


################## LogWatch 2.6 Begin ####################

--------------------- Cron Begin -----------------------
Commands Run:
```

```
**Unmatched Entries**
dic 13 14:26:56 lapfer crontab[2644]: (fzaragoz) LIST (fzaragoz)
dic 13 14:27:03 lapfer crontab[2679]: (fzaragoz) DELETE (fzaragoz)
dic 13 14:27:06 lapfer crontab[2712]: (fzaragoz) LIST (fzaragoz)


--------------------- Cron End -------------------------


###################### LogWatch End #########################
```

LIDS

Actualmente existen distintos problemas en los sistemas GNU/Linux:

- El sistema de archivos no es protegido.
- Los procesos no son protegidos.
- La administración del sistema está desprotegida.
- El superusuario (root) puede abusar de sus derechos.
- La autenticación del acceso al sistema es poco confiable.
- El modelo de control de acceso mediante permisos no es suficiente.

LIDS (**L**inux **I**ntrusion **D**etection **S**ystem) es un detector de intrusos a nivel de kernel de Linux. Con LIDS es posible proteger archivos, directorios y dispositivos importantes. LIDS permite definir listas de control de acceso (ACL - *Access List Control*) para los objetos dentro del sistema, inclusive permite definir controles para el usuario de administración: root.

Las características de LIDS son:

- Protección de archivo. Nadie, incluyendo a root, pueden modificar archivos protegidos por LIDS. Se pueden ocultar los archivos.
- Protección de procesos. Nadie, incluyendo a root, puede matar un proceso protegido. Es posible ocultar procesos.
- Definición de listas de control de acceso granulares.
- Usa y extiende las capacidades para controlar el sistema completo.
- Alertas de seguridad desde el kernel.
- Detector de rastreo de puertos en el kernel.

- Restricción de acceso a la red.

Instalación de LIDS

Para instalar LIDS se requiere aplicar un parche al kernel y recompilarlo. A continuación se explican los pasos para instalar LIDS en un equipo con sistema operativo Linux como el que se está utilizando en el SIIEM/UAP.

1. Obtener el código fuente del kernel de Linux y LIDS.
   - El kenel de Linux puede ser obtenido de la siguiente dirección electrónica: http://www.kernel.org.
   - LIDS puede ser descargado desde: http://www.lids.org
2. Desempaquetar el código obtenido.
   - Desempaquetar el kernel.

     ```
     $ bunzip2 linux-2.4.24.tar.bz2
     $ tar xf linux-2.4.24.tar
     ```

   - Desempaquetar LIDS y sus herramientas.

     ```
     $ tar zxf lids-1.1.2-2.4.21.tar.gz
     $ cd lids-1.1.2-2.4.21
     $ tar zxf lidstools-0.4.3.tar.gz
     ```

3. Agregar el parche al kernel de Linux.

   ```
   $ cd ../linux-2.4.24
   $ patch -p1 <../lids-1.1.2-2.4.21/lids-1.1.2-2.4.21.patch
   ```

4. Instalar las herramientas de LIDS

   ```
   $ cd ../lids-1.1.2-2.4.21/lidstools
   $ ./configure KERNEL_DIR=../../linux-2.4.24
   ```

5. Compilar e instalar las herramientas de LIDS.

   ```
   $make && make install
   ```

   Hasta este punto se han generado los archivos para la configuración y manejo de LIDS.

6. El siguiente paso es configurar el kernel de Linux. Para ello se puede utilizar distintas interfaces para facilitar el procedimiento mediante el comando `make`:

- `make config`. La configuración del Kernel se realiza en modo texto respondiendo a preguntas que hace el programa. A continuación se pone un ejemplo de este método que resulta ser un poco más complicado y requiere de mayor experiencia.

```
$ make config
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in arch/i386/defconfig
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers
(CONFIG_EXPERIMENTAL) [N/y/?]
```

- `make menuconfig`. De esta forma se muestra una interfaz en texto a través de menús como se muestra en la figura 4.23.

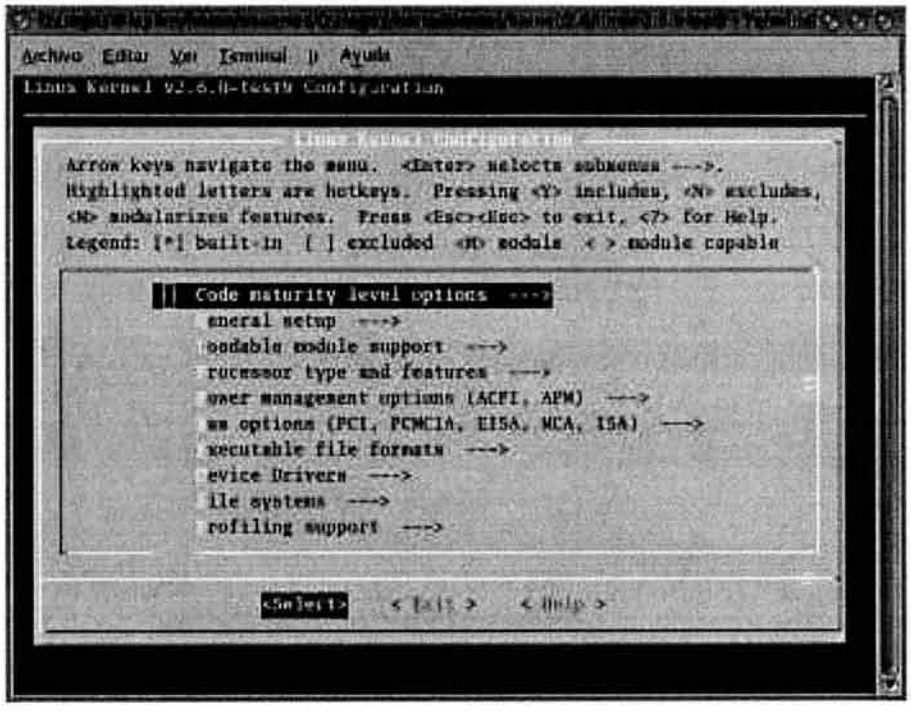

Figura 4.23: Interfaz en texto de configuración del kernel.

- `make xconfig`. Con este parámetro el comando `make` genera una interfaz en modo gráfico como se muestra en la figura 4.24. Esta opción resulta ser muy amigable para la configuración del kernel.

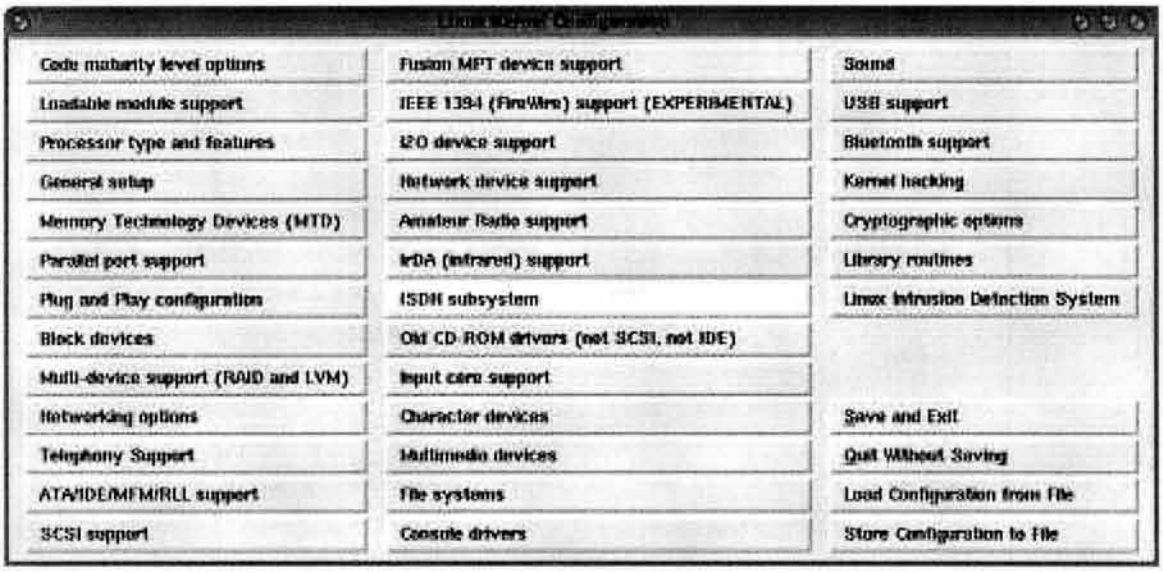

Figura 4.24: Interfaz gráfica para la configuración del kernel.

7. Configuración del kernel. Se configura el kernel utilizando cualquiera de las tres interfaces disponibles.

```
$ cd ../../linux-2.4.24
$ make xconfig
```

Para que funcione LIDS deben habilitarse estas opciones en el kernel:

```
[*]   Prompt for development and/or incomplete code/drivers
[*]   Sysctl Support
```

8. Se habilita y configura la opción de LIDS para ser utilizado en el kernel como se muestra en la figura 4.25.

9. Terminada la configuración se deben guardar los cambios y salir (figura 4.26).

10. Compilación. El siguiente paso es ejecutar una serie de comandos para compilar el kernel y generar sus módulos. Con la siguiente línea se realiza este proceso que puede tardar varios minutos dependiendo del procesador y memoria con la que se cuente.

```
$ make dep && make clean && make bzImage && make modules &&
make modules_install
```

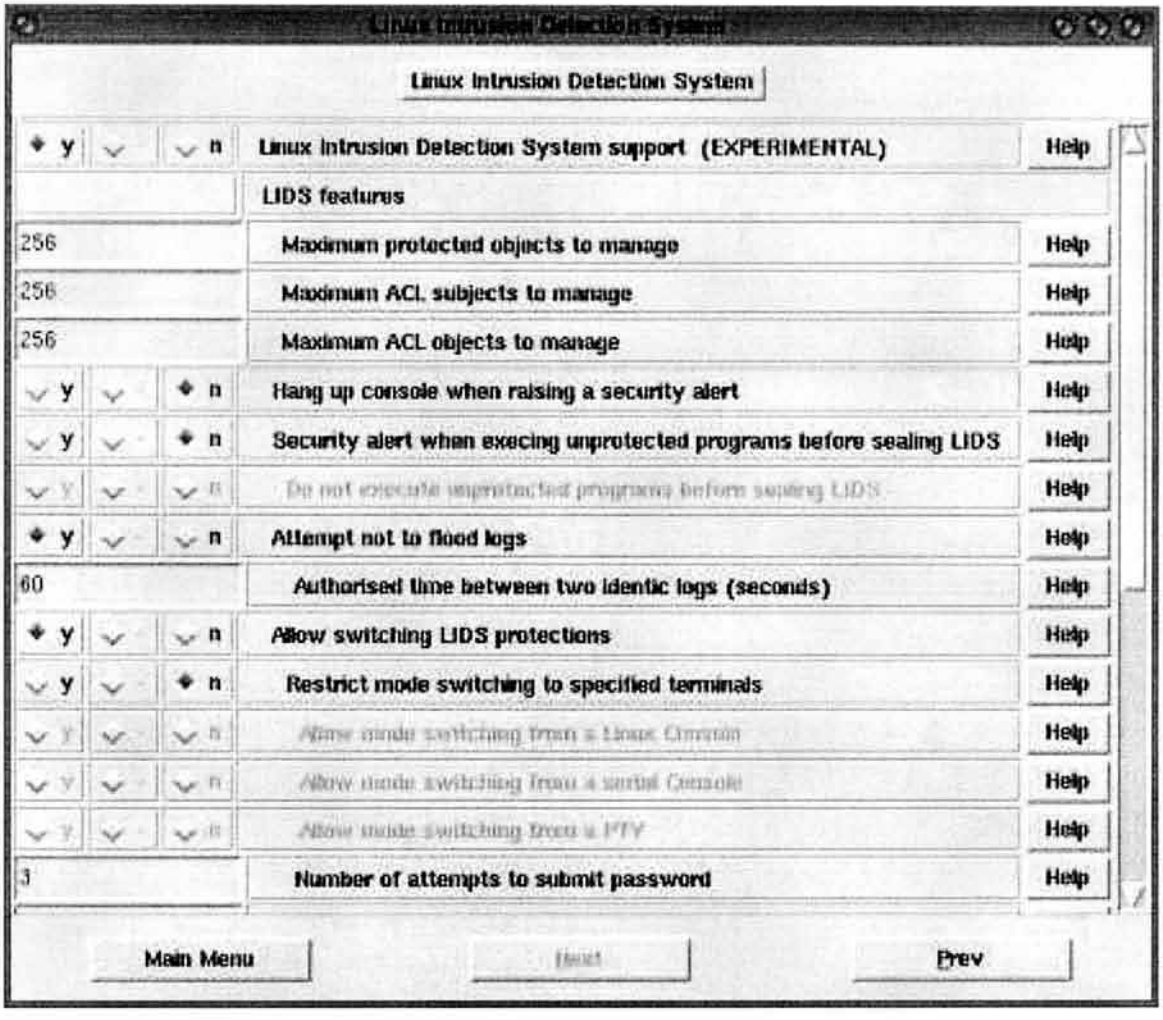

Figura 4.25: Configuración de LIDS en el kernel de Linux.

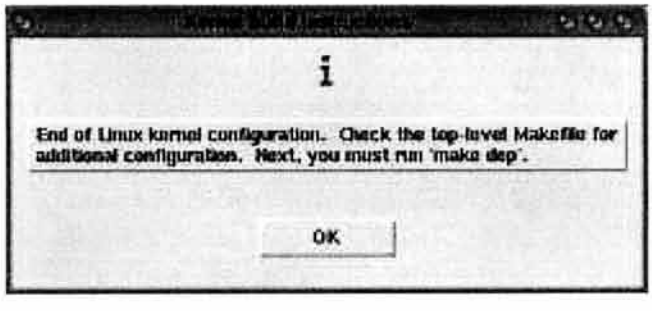

Figura 4.26: Mensaje para guardar la configuración del kernel de Linux.

11. Copia de la imagen del kernel. Se debe copiar la imagen generada del kernel en el directorio `/boot`:

```
cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.24
```

12. Generación de imagenes para los módulos precargados. Este procedimiento es necesario si se utiliza GRUB como el cargador del sistema operativo.

```
$ /sbin/mkinitrd /boot/initrd-2.4.24.img 2.4.24
```

13. Configuración del archivo del cargador. Existen básicamente dos cargadores de linux.

    - GRUB. En versiones recientes de Linux se utiliza el cargador GRUB (**GR**and **U**nified **B**ootloader) para cargar el sistema operativo. Es necesario agregar una nueva entrada para que reconozca el kernel que se acaba de compilar. Para ello se debe editar el archivo `/etc/grub.conf` y agregar las siguientes líneas:

      ```
      title Kernel LIDS (2.4.24)
      root (hd0,4)
      kernel /boot/vmlinuz-2.4.24 ro hdc=ide-scsi
      initrd /boot/initrd-2.4.24.img
      ```

    - Otro cargador muy popular en sistemas Linux es LILO (**LI**nux **LO**ader). El archivo que debe ser modificado es `/etc/lilo.conf` agregando las siguientes líneas:

      ```
      image=/boot/vmlinuz-2.4.24
              label=KernelLIDS
              root=/dev/hda4
              read-only
      ```

      Posteriormente se debe ejecutar el comando `lilo` para que los cambios surtan efecto.

14. Reiniciar el sistema. Una vez realizados los pasos anteriores, se debe reiniciar el sistema para que se cargue el nuevo kernel.

Uso de LIDS

Ya que ha sido instalado el nuevo kernel de Linux y habilitado con LIDS, es posible utilizar este detector de intrusos. LIDS cuenta con dos interfaces para su configuración y administración: `lidsconf` y `lidsadm`. A continuación se describirán brevemente estos dos programas.

`lidsconf`

Este programa sirve para definir las listas de control de acceso (ACL) de LIDS y definir una contraseña. La sintáxis de `lidsconf` es la siguiente:

```
lidsconf  -A  [-s  sujeto]  -o  objeto [-d] [-t de-a]
              [-i nivel] -j ACCION
lidsconf -D [-s archivo] [-o archivo]
lidsconf -Z
lidsconf -U
lidsconf -L [-e]
lidsconf -P
lidsconf -v
lidsconf [-h|H]
```

Comandos:

`-A, --add`. Agrega una entrada.

`-D, --delete`. Borra una entrada.

`-Z, --zero`. Borra todas las entradas.

`-U, --update`. Actualiza los números de ínodo.

`-L, --list`. Muestra todas las entradas.

`-P, --password`. Sirve para definir una contraseña.

`-v, --version`. Muestra la versión.

`-h, --help`. Muestra la ayuda.

`-H, --morehelp`. Muestra la ayuda con un nombre de CAP/SOCKET.

Sujeto: `-s, --subject sujeto`. Puede ser cualquier programa, se debe especificar el archivo.

Objeto: `-o, --object objeto`. Puede ser un archivo, directorio o capacidad.

ACCION: `-j, --jump ACCION`. Las posibles acciones son:

DENY. Niega el acceso.

READONLY. Sólo lectura.

APPEND. Sólo agrega.

WRITE. Permite la escritura.

GRANT. Permite a un sujeto.

IGNORE. Ignora cualquier permiso definido sobre un objeto.

DISABLE. Deshabilita alguna característica.

La ACL en LIDS define cómo un sujeto puede acceder a un objeto. El sujeto puede ser cualquier programa sobre el sistema, el objeto puede ser un archivo, directorio u opción especial (dispositivos MEM, entradas y salidas crudas (RAW IO), un proceso escondido, etc.). La acción define el tipo de acceso que el sujeto tiene sobre el objeto. La sintaxis para definir una ACL es la siguiente:

```
[-s sujeto] [-d|-i TTL] -o objeto [-t escala de tiempo] -j ACCION
```

Cuando no se define un sujeto, la ACL define un acceso predeterminado para el objeto. A continuación se muestran algunos ejemplos:

`lidsconf -A -o /sbin -j READONLY`
Esta ACL protege al directorio `/sbin` como de sólo lectura.

`lidsconf -A -o /sbin/test -j IGNORE`
Especifica que la protección de lectura para el directorio `/sbin` no aplica al archivo `/sbin/test`.

`lidsconf -A -o /etc/passwd -j DENY`
Oculta el archivo `/etc/passwd`.

`lidsconf -A -s /bin/login -o /etc/passwd -j READ`
Permite al programa `/bin/login` leer el archivo `/etc/passwd` aunque el archivo haya sido ocultado en el comando anterior. En este caso, sólo `/bin/login` puede leer el archivo `/etc/passwd`.

Configuración de LIDS en el SIIEM/UAP

A continuación se indican las ACLs utilizadas en el SIIEM/UAP.

Configuración básica

```
# Proteccion de binarios del sistema
#
/sbin/lidsconf -A -o /sbin                                    -j READONLY
/sbin/lidsconf -A -o /bin                                     -j READONLY

# Proteccion de los sistemas de archivos /usr and /usr/local
# (Esto asume /usr/local estan en sistemas de archivos separados).
#
```

```
/sbin/lidsconf -A -o /usr                                   -j READONLY
/sbin/lidsconf -A -o /usr/local                             -j READONLY

# Proteccion de bibliotecas.

/sbin/lidsconf -A -o /lib                                   -j READONLY

# Proteccion del directorio /opt
#
/sbin/lidsconf -A -o /opt                                   -j READONLY

# Proteccion de archivos de configuracion del sistema
#
/sbin/lidsconf -A -o /etc                                   -j READONLY
/sbin/lidsconf -A -o /usr/local/etc                         -j READONLY
/sbin/lidsconf -A -o /etc/shadow                            -j DENY
/sbin/lidsconf -A -o /etc/grub.conf                         -j DENY

# Habilita la autenticacion en el sistema
#
/sbin/lidsconf -A -s /bin/login -o /etc/shadow              -j READONLY
/sbin/lidsconf -A -s /usr/bin/vlock -o /etc/shadow          -j READONLY
/sbin/lidsconf -A -s /bin/su -o /etc/shadow                 -j READONLY
/sbin/lidsconf -A -s /bin/su \
                  -o CAP_SETUID                             -j GRANT
/sbin/lidsconf -A -s /bin/su \
                  -o CAP_SETGID                             -j GRANT

# Protege el sistema de inicio /boot
#
/sbin/lidsconf -A -o /boot                                  -j READONLY

# Protege el directorio hogar de root, pero permite la
#historia de bash
#
/sbin/lidsconf -A -o /root                                  -j READONLY
/sbin/lidsconf -A -s /bin/bash -o /root/.bash_history       -j WRITE

# Proteccion de las bitacoras del sistema
#
/sbin/lidsconf -A -o /var/log                               -j APPEND
/sbin/lidsconf -A -s /bin/login -o /var/log/wtmp            -j WRITE
/sbin/lidsconf -A -s /bin/login -o /var/log/lastlog         -j WRITE
/sbin/lidsconf -A -s /sbin/init -o /var/log/wtmp            -j WRITE
/sbin/lidsconf -A -s /sbin/init -o /var/log/lastlog         -j WRITE
```

```
/sbin/lidsconf -A -s /sbin/halt -o /var/log/wtmp          -j WRITE
/sbin/lidsconf -A -s /sbin/halt -o /var/log/lastlog       -j WRITE
/sbin/lidsconf -A -s /etc/rc.d/rc.sysinit \
                  -o /var/log/wtmp -i 1                   -j WRITE
/sbin/lidsconf -A -s /etc/rc.d/rc.sysinit \
                  -o /var/log/lastlog -i 1                -j WRITE

# Inicio del sistema
#
/sbin/lidsconf -A -s /sbin/hwclock -o /etc/adjtime        -j WRITE


# Baja del sistema
#
/sbin/lidsconf -A -s /sbin/init -o CAP_INIT_KILL          -j GRANT
/sbin/lidsconf -A -s /sbin/init -o CAP_KILL               -j GRANT

/sbin/lidsconf -A -s /etc/rc.d/init.d/halt \
                  -o CAP_INIT_KILL -i 1                   -j GRANT
/sbin/lidsconf -A -s /etc/rc.d/init.d/halt \
                  -o CAP_KILL -i 1                        -j GRANT
/sbin/lidsconf -A -s /etc/rc.d/init.d/halt \
                  -o CAP_NET_ADMIN -i 1                   -j GRANT
/sbin/lidsconf -A -s /etc/rc.d/init.d/halt \
                  -o CAP_SYS_ADMIN -i 1                   -j GRANT

#Otros
/sbin/lidsconf -A -s /sbin/update -o CAP_SYS_ADMIN        -j GRANT
```

Servidor Web (Apache)

En seguida se muestra la configuración de LIDS para proteger apache, las páginas y los programas que le dan soporte al SIIEM/UAP.

```
# Binarios
/sbin/lidsconf -A -s /usr/local/apache/bin/httpd \
                  -o CAP_SETUID                           -j GRANT
/sbin/lidsconf -A -s /usr/local/apache/bin/httpd \
                  -o CAP_SETGID                           -j GRANT

# Archivos de configuracion
/sbin/lidsconf -A -o /usr/local/apache/conf               -j DENY
/sbin/lidsconf -A -s /usr/local/apache/bin/httpd \
                  -o /usr/local/apache/conf          -j READONLY
```

```
# Raiz del servidor
/sbin/lidsconf -A -o /usr/local/apache                -j DENY
/sbin/lidsconf -A -s /usr/local/apache/bin/httpd \
                  -o /usr/local/apache                -j READONLY

# Archivos de registro
/sbin/lidsconf -A -o /usr/local/apache/logs           -j DENY
/sbin/lidsconf -A -s /usr/local/apache/bin/httpd \
                  -o /usr/local/apache/logs           -j APPEND
/sbin/lidsconf -A -s /usr/local/apache/bin/httpd \
                  -o /usr/local/apache/logs           -j WRITE

# Paginas web y programas CGI
/sbin/lidsconf -A -o /home/httpd/www                  -j DENY
/sbin/lidsconf -A -s /usr/local/apache/bin/httpd \
                  -o /home/httpd/www                  -j READONLY
```

Base de datos (Postgres)

Por último, resta definir las listas de control de acceso de LIDS para la base de datos del SIIEM/UAP.

```
#Proteccion de la base de datos
/sbin/lidsconf -A -o /home/postgres/data              -j DENY
/sbin/lidsconf -A -s /usr/local/pgsql/postmaster \
                  -o /home/postgres/data              -j APPEND
/sbin/lidsconf -A -s /usr/local/pgsql/postmaster \
                  -o /home/postgres/data              -j WRITE
```

`lidsadm`

Una vez visto el comando de configuración `lidsconf`, el siguiente comando que se analizará es `lidsadm`.

`lidsadm` es la herramienta de administración de LIDS. La sintaxis de este comando se muestra a continuación.

```
lidsadm -[S|I] -- [+|-][LIDS_FLAG] [...]
lidsadm -V
lidsadm -h
```

Comandos:

`-P`. Cifra una contraseña de LIDS (la contraseña se almacena en el archivo `lids.pw`).

`-S`. Cambia las protecciones de LIDS. Para realizar ésto, LIDS solicitará la contraseña.

`-I`. Cambia una vez las protecciones de LIDS sin solicitar contraseña.

`-V`. Permite ver el estado actual del sistema de LIDS.

`-v`. Muestra la versión de `lidsadm`.

Banderas:

`LIDS_GLOBAL`. Habilita y deshabilita completamente el sistema de LIDS.

`RELOAD_CONF`. Recarga los archivos de configuración y los números inodo de programas especiales.

`LIDS`. Habilita y deshabilita LIDS localmente (el shell e hijos). Esto es conocido como una sesión libre de LIDS (LFS - *Lids Free Session*).

Archivos. Los archivos utilizados por LIDS son:

`/etc/lids/lids.conf`. Archivo de configuración de LIDS.

`/etc/lids/lids.cap`. Define las capacidades globales.

`/etc/lids/lids.net`. Archivo de configuración para las alertas mediante correo electrónico.

`/etc/lids/lids.pwd`. Almacena la contraseña cifrada de LIDS.

Alertas de LIDS

Una característica importante de LIDS es que cualquier anomalía o violación que se intente a las ACLs será notificada por correo electrónico al administrador del sistema, en seguida se muestra un ejemplo de una alerta generada por LIDS.

```
From LIDS_ALERT@lids.sinocluster.com  Sat Jan 10 20:11:36 2004
X-Original-To: root@localhost
Delivered-To: root@localhost.posgrado.unam.mx
From: LIDS_ALERT@lids.sinocluster.com
To: root@localhost.posgrado.unam.mx
Subject:  LIDS ALert
Date: Sat, 10 Jan 2004 20:11:36 -0600 (CST)

LIDS: touch (dev 3:1 inode 385065) pid 26383 ppid 26346 uid/gid
(0/0) on (pts) : Attempt to open /bin/algo for writing,flag=35138
```

Con esta alerta doy por terminado el tema de Detectores de Intrusos. En la siguiente sección propongo el uso de herramientas de cifrado para archivos y correo electrónico mediante algoritmos de llave pública.

4.3.8. Cifrado

En el Capítulo 2 *Seguridad en Web* se habló de criptografía y los distintos sistemas de cifrado. En esta sección se describirá una herramienta perteneciente al software libre que implementa cifrado, tan importante hoy en día en el área de seguridad y que en el caso específico del SIIEM/UAP, es utilizada para el cifrado del respaldo de la base de datos.

GNU Privacy Guard

GNU Privacy Guard (GPG) es la herramienta libre que implementa cifrado de llave pública proporcionando confidencialidad, integridad y no repudiación.

GPG puede ser utilizado para:

- Cifrar archivos. Es posible cifrar archivos con algún algoritmo de llave privada o de llave pública. Una vez que el archivo ha sido cifrado sólo puede ser descifrado por alguien que conoce la frase de cifrado (especie de contraseña) o cuenta con la llave privada que coincide con la llave pública con la cual fue cifrado el archivo.
- Crear llaves. Se pueden crear las llaves necesarias para cifrar, descifrar y firmar mensajes.
- Manejar llaves. Con GPG se puede tener una base de datos de llaves (llamado comúnmente llavero) de diferentes usuarios.
- Enviar y recibir correo electrónico cifrado.
- Utilizar firmas digitales. Es posible crear y verificar documentos firmados electrónicamente.
- Certificar llaves. También se pueden firmar y autentificar llaves públicas de otras personas.

La versión comercial de cifrado es PGP (Pretty Good Privacy). Esta implementación de cifrado tiene la característica de ser multiplataforma. PGP soporta sistemas Unix, Windows y Mac. Además, PGP cuenta también con una interfaz en línea de comandos y una interfaz gráfica. Desafortunadamente, PGP no es software libre y no puede ser utilizado con la libertad con la cual se puede utilizar GPG.

Instalación de GPG

El procedimiento de instalación de GPG es muy sencillo:

1. Obtener el código fuente de GPG del sitio oficial:
   http://www.gnupg.org/download/
2. Desempaquetar el archivo
   ```
   $ bunzip2 gnupg-1.2.4.tar.bz2
   $ tar xf gnupg-1.2.4.tar
   ```

3. Configurar GPG.
   ```
   $ cd gnupg-1.2.4
   $ ./configure
   ```

4. Compilación de GPG.
   ```
   $ make
   ```

5. Instalación de GPG.
   ```
   $ make install
   ```

Uso de GPG

GPG se utiliza en sistemas tipo Unix a través del programa principal `gpg` en la línea de comandos. La sintáxis es la siguiente:

```
gpg [--homedir nombre] [--options archivo] [opciones]
    comando [argumentos]
```

En seguida se detallan algunos de los comandos más utilizados por `gpg`.

- `-s, --sign`. Crea una firma. Puede ser utilizado en conjunto con `--encrypt`.
- `--clearsign`. Crea una firma en texto claro.
- `-b, --detach-sign`. Crea una firma en un archivo anexo.
- `-e, --encrypt`. Cifra datos.
- `-c, --symetric`. Cifra con algoritmos de llave privada utilizando una frase de paso. El algoritmo predeterminado es CAST5, pero se puede modificar utilizando la opción `--cipher-algo`.
- `--decrypt [archivo]`. Descifra un archivo (o la entrada estándar si no se proporciona éste) y escribe a la salida estándar (o al archivo definido con la opción `--output`). También se verifica la firma si el archivo a descifrar está firmado.
- `--verify [[archivo_de_firma] [archivo_firmado]]`. Este comando ayuda a verificar la firma de un archivo. Para ello se debe contar con la llave pública de quien firmó dicho archivo.
- `--list-keys [nombre]`
  `--list-public-keys [nombre]`. Muestra todas las llave públicas del llavero o sólo las que se especifiquen en la línea de comandos.
- `--list-secret-keys [nombre]`. Muestra las llaves del llavero privado, o sólo las proporcionadas como argumentos.
- `--list-sigs`. Muestra las llaves públicas juntos con sus firmas.

- `--check-sigs [names]`. Lo mismo que `--list-sigs` y además se verifican las firmas.
- `--fingerprint [nombres]`. Lista las llaves junto con sus huellas digitales.
- `--gen-key`. Genera un nuevo par de llaves. Este comando es usado de forma interactiva.
- `--edit-key [nombre]`. Muestra un menú para manipular el llavero. Algunos comandos de este menú se muestran a continuación:
  - sign. Crea una firma sobre la llave de un usuario si aún no ha sido firmada por el usuario predeterminado.
  - lsign. Es similar a `--sign` pero la firma es marcada como no exportable y nunca podrá ser utilizada por otros. Esto puede ser utilizado para hacer llaves válidas en un ambiente local.
  - nrsign. La firma es marcada como no revocable.
  - revsig. Sirve para anular una firma.
  - trust. Cambia el valor de confianza del dueño.
  - passwd. Cambia la contraseña de la llave secreta.
  - check. Revisa todos los identificadores de los usuarios seleccionados.
  - save. Guarda los cambios realizados al llavero y sale.
  - quit. Sale del programa sin guardar cualquier cambio realizado.
  - help. Muestra una ayuda de los comandos disponibles.
- `--sign-key nombre`. Firma una llave pública sin la llave privada.
- `--delete-secret-key nombre`. Elimina una llave del llavero privado.
- `--gen-revoke nombre`. Genera un certificado de revocación.
- `--export [nombre]`. Exporta todas las llaves de todos los llaveros, a menos que se proporcione un nombre.
- `--import [archivos]`
  `--fast-import [archivos]`. Agrega llaves al llavero.
- `--version`. Muestra la versión del comando y los algoritmos que soporta.
- `-h, --help`. Muestra la ayuda.

GPG utiliza distintas opciones las cuales pueden ser definidas en un archivo o desde la línea de comandos. Algunas de las opciones más útiles son:

- `-a, --armor`. Genera la salida en formato ASCII.

- `-o, --output archivo`. Escribe la salida a un archivo.
- `-r, --recipient nombre`. Cifra para el usuario determinado por el nombre.
- `-z, --compress-level n`. Define el nivel de compresión. El valor de 0 deshabilita la compresión. El nivel predeterminado de compresión es 6.
- `-i, --interactive`. Solicita la verificación por parte del usuario antes de escribir algún archivo.
- `--for-your-eyes-only`
  `--no-for-your-eyes-only`. Esta opción causa que GPG no permite guardar en un archivo. `--no-for-your-eyes-only` deshabilita esta opción.
- `--cipher-algo nombre`. Usa el algoritmo de cifrado definido por `nombre`. Para ver los algoritmos soportados por gpg es necesario utilizar el comando `--version`.

Antes de poder cifrar, descifrar y firmar archivos y/o mensajes es necesario tener un par de llaves con los cuales poder realizar ésto. En seguida muestro algunos ejemplos para crear las llaves, importar y exportar llaves y por último cómo cifrar y firmar documentos.

- Generar llaves. Si es la primera vez que se utiliza GPG es necesario crear un par de llaves, la forma de crearlas se muestra a continuación.

  1. Ejecutar el comando para crear las llaves:
     ```
     $ gpg --gen-key
     ```
  2. Seleccionar el método de cifrado:

     ```
     Por favor seleccione tipo de clave deseado:
        (1) DSA y ElGamal (por defecto)
        (2) DSA (solo firmar)
        (4) RSA (solo firmar)
     Su eleccion: 1
     ```

  3. Seleccionar el tamaño de la llave.

     ```
     El par de claves DSA tendra 1024 bits.
     Listo para generar un nuevo par de claves ELG-E.
                  el tamano minimo es 768 bits
            el tamano por defecto es 1024 bits
      el tamano maximo recomendado es 2048 bits
     De que tamano quiere la clave (1024)?
     ```

  4. Seleccionar el tiempo de duración de la llave.

     ```
     Por favor, especifique el periodo de validez de la clave.
              0 = la clave nunca caduca
           <n>  = la clave caduca en n dias
     ```

```
        <n>w = la clave caduca en n semanas
        <n>m = la clave caduca en n meses
        <n>y = la clave caduca en n anos
Validez de la clave (0)?
```

5. Definir un identificador de usuario.

```
Necesita un identificador de usuario para identificar
su clave. El programa construye el identificador a
partir del Nombre Real, Comentario y Direccion de
Correo Electronico de esta forma:
"Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Nombre y apellidos: Fernando Zaragoza Hernandez
Direccion de correo electronico: fer@posgrado.unam.mx
Comentario:
Ha seleccionado este ID de usuario:
    "Fernando Zaragoza Hernandez <fer@posgrado.unam.mx>"
```

6. Aceptar todos los datos ingresados o modificar alguno de ellos.

```
Cambia (N)ombre, (C)omentario, (D)ireccion o
(V)ale/(S)alir? V
```

7. Ingresar una frase como contraseña.

```
Necesita una frase contrasena para proteger su clave
secreta.
Introduzca frase contrasena:
```

8. GPG genera las llaves y muestra el ID de usuario asignado y la huella de la llave.

```
claves publica y secreta creadas y firmadas.
clave marcada como de confianza absoluta.

pub  1024D/BC99E1D4 2003-12-30 Fernando Zaragoza Hernandez
<fer@newman.posgrado.unam.mx>
Huella de clave = DF15 0DC5 AD3B A93E B549  CA71 7F91
                  2F5B BC99 E1D4
sub  1024g/DDDF9669 2003-12-30
```

- Exportar llave. Una vez que se han generado las llaves, el siguiente paso es exportar la llave pública para compartirla con otros usuarios, y asi ellos puedan enviar información cifrada utilizando la llave pública.

  Se puede exportar la llave de dos formas:

  - Binario. De esta forma se crea un archivo en formato binario. El comando es:

```
$ gpg --export --output llave.gpg
```

- Texto. Se puede generar la llave en un formato legible que puede ser agregado a la firma en el correo electrónico. Para llevar a cabo ésto se utiliza el siguiente comando.

```
$ gpg --export --armor --output llave.asc
```

El archivo llave.asc contiene algo como lo que se muestra a continuación.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.4 (GNU/Linux)

mQGiBD621UERBADbXvvbsJfygYWPn2p2S0kS5hX+9A5Dj98ByuNapmt
X8/h+zQRFbKbuf9PJNpjENrgaefgD38NJGbaQI4GJ7ipPpFt9ZIUrgm
JPY9kNk7w2jHct4mCtLaKyH7wWMAi21Ij/h3F+Od+uBsrSyg7u2qlqG
...
kS9bvJnh1DRvAJ9xViOeCNrpexcU4Au/jL4UI2y7OwCgyiBCcxAxLLy
oUd2MEQ=
=LzHE
-----END PGP PUBLIC KEY BLOCK-----
```

De esta forma los archivos llave.asc y llave.gpg pueden ser distribuidos a otros usuarios para que agreguen la llave a su llavero.

- Importar una llave. Esto consiste en agregar una llave de otro usuario al llavero propio. Para hacerlo se debe obtener el archivo (ya sea en formato binario o ascii) que contenga la llave pública del usuario y agregarla de la siguiente manera:

```
$ gpg --import llavemem.asc
```

- Visualizar el llavero. Algo importante al utilizar GPG es ver que se cuentan con las llaves de los usuarios para cifrar información o verificar sus firmas. La forma de visualizar el llavero es de la siguiente manera:

```
$ gpg --list-key
/home/usuarios/fzaragoz/.gnupg/pubring.gpg
---------------------------------------
pub  1024D/BC99E1D4 2003-12-30 Fernando Zaragoza Hernandez
                               <fer@newman.posgrado.unam.mx>
sub  1024g/DDDF9669 2003-12-30

pub  1024D/822879BC 2003-12-30 Mabel Entzana Martinez
                               <mabe_mem@hotmail.com>
sub  1024g/C9B8AA0D 2003-12-30
```

- Cifrar un archivo. Como se mencionó en las opciones de GPG, es posible cifrar información utilizando una contraseña o utilizando la llave pública de un usuario. A continuación se muestran las dos formas.

  - Cifrado con contraseña. El siguiente comando muestra como cifrar el archivo /etc/passwd en formato ASCII.

    ```
    $ gpg --output passwd.asc --armor -c /etc/passwd
    Introduzca frase contrasena:
    ```

  - Cifrado con llave pública. De esta forma se cifra un archivo para un usuario utilizando su llave pública. En seguida se muestra un ejemplo.

    ```
    $ gpg --encrypt --recipient mabe_mem --output info.gpg
    /etc/group
    ```

    Con este comando se cifra el archivo /etc/group para el usuario mabe_mem.

- Firma de documentos. Para firmar un mensaje es necesario proporcionar la contraseña debido a que se utiliza la llave privada. En seguida se muestra un ejemplo de cómo firmar un documento.

  ```
  $ gpg --output archivo.sig --sign /etc/hosts

  Necesita una frase contrasena para desbloquear la clave
  secreta del usuario: "Fernando Zaragoza Hernandez
  <fer@newman.posgrado.unam.mx>"
  clave DSA de 1024 bits, ID BC99E1D4, creada el 2003-12-30

  Introduzca frase contrasena:
  ```

- Firmar y cifrar. En seguida se muestra un ejemplo de cómo firmar y cifrar un documento al mismo tiempo.

  ```
  $ gpg --output archivo.sig --encrypt --recipient mabe_mem
        --sign /etc/hosts
  ```

- Descifrar. Finalmente, si se recibe un mensaje cifrado, para descifrarlo se ejecuta el comando gpg y el nombre del archivo como se muestra a continuación:

  ```
  $ gpg info.gpg
  ```

  Si no se cuenta con la llave privada correspondiente para descifrar el archivo, gpg muestra un mensaje como el siguiente:

```
$ gpg info.gpg
gpg: cifrado con clave ELG-E de 1024 bits, ID C9B8AA0D,
creada el 2003-12-30
      "Mabel Entzana Martinez <mabe_mem@hotmail.com>"
gpg: descifrado fallido: clave secreta no disponible
```

GPG fue utilizado en el SIIEM/UAP para cifrar los respaldos de la base de datos debido a que en éstos se encuentra toda la información de alumnos, profesores, grupos, etc.

Deseo que el uso de estas herramientas hayan dado al lector una idea más clara de lo que el Software Libre proporciona y las grandes ventajas que aporta en distintos productos: sistema operativo, plataforma de desarrollo, lenguajes de programación, bases de datos, utilerías de seguridad, aplicaciones varias e inclusive juegos.

Conclusiones

En un sistema de cómputo (sin importar que éste sea académico, comercial o gubernamental) la seguridad debe ser considerada desde la planeación del mismo. Esta planeación incluye la identificación de bienes y amenazas además del análisis costo-beneficio que permitan valorar la importancia y la inversión en la seguridad en cómputo.

La planeación se deberá ver reflejada en la definición de políticas, estándares y procedimientos que determinen el funcionamiento de los sistemas de cómputo y los derechos y obligaciones de los operadores y usuarios del mismo.

No obstante, se debe estar conciente de que un sistema no puede ser totalmente seguro, ésto no se debe a la falta de soluciones técnicas o a la falta de capacidad de los operadores de los sistemas, se debe al hecho de que los sistemas de cómputo evolucionan constantemente, cada vez mas existen computadoras con mayores capacidades técnicas que aunadas al acceso a Internet y a la cada vez mayor cantidad de líneas de código en los programas hacen que se incrementen los riesgos en los sistemas.

Sin embargo, el uso de buenas prácticas, la actualización y la seguridad a profundidad, que cubra la mayoría de las aristas de un sistema, reducirán en gran medida los riesgos de seguridad e incrementarán la confiabilidad del mismo.

Este trabajo de tesis, aplicado al Sistema de Inscripciones por Internet de Especializaciones de Medicina de la Unidad de Administración del Posgrado, alcanzó el objetivo al contemplar los elementos más importantes de seguridad y al definir los mecanismos necesarios para reducir los riesgos. La seguridad en el SIIEM/UAP fue abordada desde cinco ángulos:

1. Evaluación de riesgos y definición de políticas, estándares y procedimientos. Con ésto se definieron los bienes que se deben asegurar y las amenazas que pudieran afectar el sistema. Además se definieron los lineamientos necesarios para el funcionamiento correcto del sistema.

2. Seguridad en el sistema de cómputo. En este punto se definieron los mecanismos necesarios para asegurar el sistema de cómputo mediante el uso de buenas prácticas recomendadas por profesionales de la seguridad.

3. Seguridad en Web. Debido a que el SIIEM/UAP es un sistema al servicio de la comunidad universitaria, se encuentra disponible a través de páginas web. Este servicio

es uno de los más atacados actualmente y por eso fue necesario asegurarlo a través de prácticas recomendadas y probadas a nivel mundial, incluyendo la tecnología de certificados digitales.

4. Seguridad en la base de datos. Siendo la base de datos el almacen de la información, es necesario utilizar los medios necesarios para que los datos sean accedidos por las personas adecuadas utilizando los mecanismos definidos. Después de haber realizado un análisis de los métodos que implementa el manejador de la base de datos se obtuvo la confidencialidad e integridad necesarias para asegurar los datos del sistema.

5. Uso de herramientas. La seguridad no puede estar completa sin el monitoreo constante y la detección de instrusos en un sistema. Sin embargo, estas tareas no pueden ser realizadas sin la ayuda de herramientas que permitan automatizar estas actividades y mantengan alerta al encargado de seguridad o al administrador de sistemas. En el SIIEM/UAP se instalaron y utilizaron herramientas que permiten cubrir un espectro importante de seguridad:

   - Firewalls.
   - Integridad del sistema de archivos.
   - Rastreo de puertos.
   - Análisis de vulnerabilidades conocidas.
   - Análisis de registros en el sistema.
   - Comunicaciones seguras.
   - Detección de intrusos.
   - Cifrado.

Este trabajo de tesis puede ser de gran ayuda para que los administradores de sistemas similares al SIIEM/UAP puedan mejorar la seguridad en el mismo. En el caso de la Universidad Nacional Autónoma de México no existen datos estadísticos del porcentaje de uso de sistemas operativos, sin embargo, es seguro que cada vez son mas los encargados del área de informática que toman mayor interés en plataformas libres debido al bajo costo que representan y a la gran gama de posibilidades que existen.

Asi como los sistemas, lenguajes de programación y generalmente todo lo relacionado a las tecnologías de la información, también los métodos de intrusión evolucionan y surgen nuevas formas de vulnerar y atacar un sistema de cómputo, por eso la importancia de la renovación constante de la seguridad. De tal suerte que un sistema puede ser muy seguro en un momento dado, pero el día de mañana seguramente existirá una nueva amenaza que pueda vulnerar la integridad, confidencialidad o disponibilidad del sistema de información.

La seguridad en cómputo requiere de estudio y capacitación constante, además de un mayor número de profesionales involucrados en esta área. Es necesario hacer conciencia

y cultura entre los estudiantes de ingeniería, informática y cualquier área relacionada al cómputo, sobre la importancia de la seguridad en nuestros días ya que ésto puede generar pérdidas millonarias a una organización, o algo más importante, puede estar de por medio la vida de seres humanos que dependen en mayor medida de los sistemas de información.

Anexo 1
General Public License GPL

VERSIÓN EN ESPAÑOL

Debido a los diferentes connotaciones que puede tener una palabra en el idioma inglés, existen al menos cinco traducciones distintas para la GPL, aquí se agrega una de ellas.

Licencia Pública GNU

Esta es la conocida GNU Public License (GPL), versión 2 (de junio de 1.991), que cubre la mayor parte del software de la Free Software Foundation, y muchos más programas.

Los autores de esta traducción son:

- Jesús González Barahona
- Pedro de las Heras Quirós

NOTA IMPORTANTE:
Esta es una traducción no oficial al español de la GNU General Public License. No ha sido publicada por la Free Software Foundation, y no establece legalmente las condiciones de distribución para el software que usa la GNU GPL. Estas condiciones se establecen solamente por el texto original, en inglés, de la GNU GPL. Sin embargo, esperamos que esta traducción ayude a los hispanohablantes a entender mejor la GNU GPL.

IMPORTANT NOTICE:
This is an unofficial translation of the GNU General Public License into Spanish. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU GPL–only the original English text of the GNU GPL does that. However, we hope that this translation will help Spanish speakers understand the GNU GPL better.

675 Mass Ave, Cambridge, MA 02139, EEUU

Preámbulo

Las licencias que cubren la mayor parte del software están diseñadas para quitarle a usted la libertad de compartirlo y modificarlo. Por el contrario, la Licencia Pública General de GNU pretende garantizarle la libertad de compartir y modificar software libre, para asegurar que el software es libre para todos sus usuarios. Esta Licencia Pública General se aplica a la mayor parte del software del la Free Software Foundation y a cualquier otro programa si sus autores se comprometen a utilizarla. (Existe otro software de la Free

Software Foundation que está cubierto por la Licencia Pública General de GNU para Bibliotecas). Si quiere, también puede aplicarla a sus propios programas:

Cuando hablamos de software libre, estamos refiriéndonos a libertad, no a precio. Nuestras Licencias Públicas Generales están diseñadas para asegurarnos de que tenga la libertad de distribuir copias de software libre (y cobrar por ese servicio si quiere), de que reciba el código fuente o que pueda conseguirlo si lo quiere, de que pueda modificar el software o usar fragmentos de él en nuevos programas libres, y de que sepa que puede hacer todas estas cosas.

Para proteger sus derechos necesitamos algunas restricciones que prohiban a cualquiera negarle a usted estos derechos o pedirle que renuncie a ellos. Estas restricciones se traducen en ciertas obligaciones que le afectan si distribuye copias del software, o si lo modifica.

Por ejemplo, si distribuye copias de uno de estos programas, sea gratuitamente, o a cambio de una contraprestación, debe dar a los receptores todos los derechos que tiene. Debe asegurarse de que ellos también reciben, o pueden conseguir, el código fuente. Y debe mostrarles estas condiciones de forma que conozcan sus derechos.

Protegemos sus derechos con la combinación de dos medidas:

1. Ponemos el software bajo copyright y
2. le ofrecemos esta licencia, que le da permiso legal para copiar, distribuir y/o modificar el software.

También, para la protección de cada autor y la nuestra propia, queremos asegurarnos de que todo el mundo comprende que no se proporciona ninguna garantía para este software libre. Si el software se modifica por cualquiera y éste a su vez lo distribuye, queremos que sus receptores sepan que lo que tienen no es el original, de forma que cualquier problema introducido por otros no afecte a la reputación de los autores originales.

Por último, cualquier programa libre está constantemente amenazado por patentes sobre el software. Queremos evitar el peligro de que los redistribuidores de un programa libre obtengan patentes por su cuenta, convirtiendo de facto el programa en propietario. Para evitar esto, hemos dejado claro que cualquier patente debe ser pedida para el uso libre de cualquiera, o no ser pedida.

Los términos exactos y las condiciones para la copia, distribución y modificación se exponen a continuación.

TÉRMINOS Y CONDICIONES PARA LA COPIA, DISTRIBUCIÓN Y MODIFICACIÓN

1. Esta Licencia se aplica a cualquier programa u otro tipo de trabajo que contenga una nota colocada por el tenedor del copyright diciendo que puede ser distribuido bajo los términos de esta Licencia Pública General. En adelante, "Programa" se referirá

a cualquier programa o trabajo que cumpla esa condición y "trabajo basado en el Programa" se referirá bien al Programa o a cualquier trabajo derivado de él según la ley de copyright. Esto es, un trabajo que contenga el programa o una proción de él, bien en forma literal o con modificaciones y/o traducido en otro lenguaje. Por lo tanto, la traducción está incluida sin limitaciones en el término "modificación". Cada concesionario (licenciatario) será denominado "usted".

Cualquier otra actividad que no sea la copia, distribución o modificación no está cubierta por esta Licencia, está fuera de su ámbito. El acto de ejecutar el Programa no está restringido, y los resultados del Programa están cubiertos únicamente si sus contenidos constituyen un trabajo basado en el Programa, independientemente de haberlo producido mediante la ejecución del programa. El que esto se cumpla, depende de lo que haga el programa.

2. Usted puede copiar y distribuir copias literales del código fuente del Programa, según lo has recibido, en cualquier medio, supuesto que de forma adecuada y bien visible publique en cada copia un anuncio de copyright adecuado y un repudio de garantía, mantenga intactos todos los anuncios que se refieran a esta Licencia y a la ausencia de garantía, y proporcione a cualquier otro receptor del programa una copia de esta Licencia junto con el Programa.

   Puede cobrar un precio por el acto físico de transferir una copia, y puede, según su libre albedrío, ofrecer garantía a cambio de unos honorarios.

3. Puede modificar su copia o copias del Programa o de cualquier porción de él, formando de esta manera un trabajo basado en el Programa, y copiar y distribuir esa modificación o trabajo bajo los términos del apartado 1, antedicho, supuesto que además cumpla las siguientes condiciones:

   a) Debe hacer que los ficheros modificados lleven anuncios prominentes indicando que los ha cambiado y la fecha de cualquier cambio.

   b) Debe hacer que cualquier trabajo que distribuya o publique y que en todo o en parte contenga o sea derivado del Programa o de cualquier parte de él sea licenciada como un todo, sin carga alguna, a todas las terceras partes y bajo los términos de esta Licencia.

   c) Si el programa modificado lee normalmente órdenes interactivamente cuando es ejecutado, debe hacer que, cuando comience su ejecución para ese uso interactivo de la forma más habitual, muestre o escriba un mensaje que incluya un anuncio de copyright y un anuncio de que no se ofrece ninguna garantía (o por el contrario que sí se ofrece garantía) y que los usuarios pueden redistribuir el programa bajo estas condiciones, e indicando al usuario cómo ver una copia de esta licencia. (Excepción: si el propio programa es interactivo pero normalmente no muestra ese anuncio, no se requiere que su trabajo basado en el Programa muestre ningún anuncio).

Estos requisitos se aplican al trabajo modificado como un todo. Si partes identificables de ese trabajo no son derivadas del Programa, y pueden, razonablemente, ser consideradas trabajos independientes y separados por ellos mismos, entonces esta Licencia y sus términos no se aplican a esas partes cuando sean distribuidas como trabajos separados. Pero cuando distribuya esas mismas secciones como partes de un todo que es un trabajo basado en el Programa, la distribución del todo debe ser según los términos de esta licencia, cuyos permisos para otros licenciatarios se extienden al todo completo, y por lo tanto a todas y cada una de sus partes, con independencia de quién la escribió.

Por lo tanto, no es la intención de este apartado reclamar derechos o desafiar sus derechos sobre trabajos escritos totalmente por usted mismo. El intento es ejercer el derecho a controlar la distribución de trabajos derivados o colectivos basados en el Programa.

Además, el simple hecho de reunir un trabajo no basado en el Programa con el Programa (o con un trabajo basado en el Programa) en un volumen de almacenamiento o en un medio de distribución no hace que dicho trabajo entre dentro del ámbito cubierto por esta Licencia.

4. Puede copiar y distribuir el Programa (o un trabajo basado en él, según se especifica en el apartado 2, como código objeto o en formato ejecutable según los términos de los apartados 1 y 2, supuesto que además cumpla una de las siguientes condiciones:

   a) Acompañarlo con el código fuente completo correspondiente, en formato electrónico, que debe ser distribuido según se especifica en los apartados 1 y 2 de esta Licencia en un medio habitualmente utilizado para el intercambio de programas, o

   b) Acompañarlo con una oferta por escrito, válida durante al menos tres años, de proporcionar a cualquier tercera parte una copia completa en formato electrónico del código fuente correspondiente, a un coste no mayor que el de realizar físicamente la distribución del fuente, que será distribuido bajo las condiciones descritas en los apartados 1 y 2 anteriores, en un medio habitualmente utilizado para el intercambio de programas, o

   c) Acompañarlo con la información que recibiste ofreciendo distribuir el código fuente correspondiente. (Esta opción se permite sólo para distribución no comercial y sólo si usted recibió el programa como código objeto o en formato ejecutable con tal oferta, de acuerdo con el apartado b anterior).

   Por código fuente de un trabajo se entiende la forma preferida del trabajo cuando se le hacen modificaciones. Para un trabajo ejecutable, se entiende por código fuente completo todo el código fuente para todos los módulos que contiene, más cualquier fichero asociado de definición de interfaces, más los guiones utilizados para controlar la compilación e instalación del ejecutable. Como excepción especial el código

fuente distribuido no necesita incluir nada que sea distribuido normalmente (bien como fuente, bien en forma binaria) con los componentes principales (compilador, kernel y similares) del sistema operativo en el cual funciona el ejecutable, a no ser que el propio componente acompañe al ejecutable.

Si la distribución del ejecutable o del código objeto se hace mediante la oferta acceso para copiarlo de un cierto lugar, entonces se considera la oferta de acceso para copiar el código fuente del mismo lugar como distribución del código fuente, incluso aunque terceras partes no estén forzadas a copiar el fuente junto con el código objeto.

5. No puede copiar, modificar, sublicenciar o distribuir el Programa excepto como prevé expresamente esta Licencia. Cualquier intento de copiar, modificar sublicenciar o distribuir el Programa de otra forma es inválida, y hará que cesen automáticamente los derechos que te proporciona esta Licencia. En cualquier caso, las partes que hayan recibido copias o derechos de usted bajo esta Licencia no cesarán en sus derechos mientras esas partes continúen cumpliéndola.

6. No está obligado a aceptar esta licencia, ya que no la ha firmado. Sin embargo, no hay hada más que le proporcione permiso para modificar o distribuir el Programa o sus trabajos derivados. Estas acciones están prohibidas por la ley si no acepta esta Licencia. Por lo tanto, si modifica o distribuye el Programa (o cualquier trabajo basado en el Programa), está indicando que acepta esta Licencia para poder hacerlo, y todos sus términos y condiciones para copiar, distribuir o modificar el Programa o trabajos basados en él.

7. Cada vez que redistribuya el Programa (o cualquier trabajo basado en el Programa), el receptor recibe automáticamente una licencia del licenciatario original para copiar, distribuir o modificar el Programa, de forma sujeta a estos términos y condiciones. No puede imponer al receptor ninguna restricción más sobre el ejercicio de los derechos aquí garantizados. No es usted responsable de hacer cumplir esta licencia por terceras partes.

8. Si como consecuencia de una resolución judicial o de una alegación de infracción de patente o por cualquier otra razón (no limitada a asuntos relacionados con patentes) se le imponen condiciones (ya sea por mandato judicial, por acuerdo o por cualquier otra causa) que contradigan las condiciones de esta Licencia, ello no le exime de cumplir las condiciones de esta Licencia. Si no puede realizar distribuciones de forma que se satisfagan simultáneamente sus obligaciones bajo esta licencia y cualquier otra obligación pertinente entonces, como consecuencia, no puede distribuir el Programa de ninguna forma. Por ejemplo, si una patente no permite la redistribución libre de derechos de autor del Programa por parte de todos aquellos que reciban copias directa o indirectamente a través de usted, entonces la única forma en que podría satisfacer tanto esa condición como esta Licencia sería evitar completamente

la distribución del Programa.

Si cualquier porción de este apartado se considera inválida o imposible de cumplir bajo cualquier circunstancia particular ha de cumplirse el resto y la sección por entero ha de cumplirse en cualquier otra circunstancia.

No es el propósito de este apartado inducirle a infringir ninguna reivindicación de patente ni de ningún otro derecho de propiedad o impugnar la validez de ninguna de dichas reivindicaciones. Este apartado tiene el único propósito de proteger la integridad del sistema de distribución de software libre, que se realiza mediante prácticas de licencia pública. Mucha gente ha hecho contribuciones generosas a la gran variedad de software distribuido mediante ese sistema con la confianza de que el sistema se aplicará consistentemente. Será el autor/donante quien decida si quiere distribuir software mediante cualquier otro sistema y una licencia no puede imponer esa elección.

Este apartado pretende dejar completamente claro lo que se cree que es una consecuencia del resto de esta Licencia.

9. Si la distribución y/o uso de el Programa está restringida en ciertos países, bien por patentes o por interfaces bajo copyright, el tenedor del copyright que coloca este Programa bajo esta Licencia puede añadir una limitación explícita de distribución geográfica excluyendo esos países, de forma que la distribución se permita sólo en o entre los países no excluidos de esta manera. En ese caso, esta Licencia incorporará la limitación como si estuviese escrita en el cuerpo de esta Licencia.

10. La Free Software Foundation puede publicar versiones revisadas y/o nuevas de la Licencia Pública General de tiempo en tiempo. Dichas nuevas versiones serán similares en espíritu a la presente versión, pero pueden ser diferentes en detalles para considerar nuevos problemas o situaciones.

    Cada versión recibe un número de versión que la distingue de otras. Si el Programa especifica un número de versión de esta Licencia que se refiere a ella y a "cualquier versión posterior", tienes la opción de seguir los términos y condiciones, bien de esa versión, bien de cualquier versión posterior publicada por la Free Software Foundation. Si el Programa no especifica un número de versión de esta Licencia, puedes escoger cualquier versión publicada por la Free Software Foundation.

11. Si quiere incorporar partes del Programa en otros programas libres cuyas condiciones de distribución son diferentes, escribe al autor para pedirle permiso. Si el software tiene copyright de la Free Software Foundation, escribe a la Free Software Foundation: algunas veces hacemos excepciones en estos casos. Nuestra decisión

estará guiada por el doble objetivo de de preservar la libertad de todos los derivados de nuestro software libre y promover el que se comparta y reutilice el software en general.

AUSENCIA DE GARANTÍA

12. Como el programa se licencia libre de cargas, no se ofrece ninguna garantía sobre el programa, en todas la extensión permitida por la legislación aplicable. Excepto cuando se indique de otra forma por escrito, los tenedores del copyright y/u otras partes proporcionan el programa "tal cual", sin garantía de ninguna clase,bien expresa o implícita, con inclusión, pero sin limitación a las garantías mercantiles implícitas o a la conveniencia para un propósito particular. Cualquier riesgo referente a la calidad y prestaciones del programa es asumido por usted. Si se probase que el Programa es defectuoso, asume el coste de cualquier servicio, reparación o corrección.

13. En ningún caso, salvo que lo requiera la legislación aplicable o haya sido acordado por escrito, ningún tenedor del copyright ni ninguna otra parte que modifique y/o redistribuya el Programa según se permite en esta Licencia será responsable ante usted por daños, incluyendo cualquier daño general, especial, incidental o resultante producido por el uso o la imposibilidad de uso del Programa (con inclusión, pero sin limitación a la pérdida de datos o a la generación incorrecta de datos o a pérdidas sufridas por usted o por terceras partes o a un fallo del Programa al funcionar en combinación con cualquier otro programa), incluso si dicho tenedor u otra parte ha sido advertido de la posibilidad de dichos daños.

FIN DE TÉRMINOS Y CONDICIONES

Apéndice: Cómo aplicar estos términos a sus nuevos programas.

Si usted desarrolla un nuevo Programa, y quiere que sea del mayor uso posible para el público en general, la mejor forma de conseguirlo es convirtiéndolo en software libre que cualquiera pueda redistribuir y cambiar bajo estos términos.

Para hacerlo, añada los siguientes anuncios al programa. Lo más seguro es añadirlos al principio de cada fichero fuente para transmitir lo más efectivamente posible la ausencia de garantía. Además cada fichero debería tener al menos la línea de "copyright" y un indicador a dónde puede encontrarse el anuncio completo.

> <una línea para indicar el nombre del programa y una rápida idea de qué hace.>

> Copyright (C) 19aa <nombre del autor>
>
> Este programa es software libre. Puede redistribuirlo y/o modificarlo bajo los términos de la Licencia Pública General de GNU según es publicada por la Free Software Foundation, bien de la versión 2 de dicha Licencia o bien (según su elección) de cualquier versión posterior.
>
> Este programa se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTÍA, incluso sin la garantía MERCANTIL implícita o sin garantizar la CONVENIENCIA PARA UN PROPÓSITO PARTICULAR. Véase la Licencia Pública General de GNU para más detalles.
>
> Debería haber recibido una copia de la Licencia Pública General junto con este programa. Si no ha sido así, escriba a la Free Software Foundation, Inc., en 675 Mass Ave, Cambridge, MA 02139, EEUU.

Añada también información sobre cómo contactar con usted mediante correo electrónico y postal.

Si el programa es interactivo, haga que muestre un pequeño anuncio como el siguiente, cuando comienza a funcionar en modo interactivo:

> Gnomovision versión 69, Copyright (C) 19aa nombre del autor
>
> Gnomovision no ofrece ABSOLUTAMENTE NINGUNA GARANTÍA. Para más detalles escriba `show w`.

Los comandos hipotéticos `show w` y `show c` deberían mostrar las partes adecuadas de la Licencia Pública General. Por supuesto, los comandos que use pueden llamarse de cualquier otra manera. Podrían incluso ser pulsaciones del ratón o elementos de un menú (lo que sea apropiado para su programa).

También deberías conseguir que su empleador (si trabaja como programador) o tu Universidad (si es el caso) firme un renuncia de copyrightÂ» para el programa, si es necesario. A continuación se ofrece un ejemplo, altere los nombres según sea conveniente:

> Yoyodyne, Inc. mediante este documento renuncia a cualquier interés de derechos de copyright
> con respecto al programa Gnomovision (que hace pasadas a compiladores) escrito por Pepe Programador.
>
> <firma de Pepito Grillo>, 20 de diciembre de 1996
> Pepito Grillo, Presidente de Asuntillos Varios.

Esta Licencia Pública General no permite que incluya sus programas en programas propietarios. Si su programa es una biblioteca de subrutinas, puede considerar más útil el permitir el enlazado de aplicaciones propietarias con la biblioteca. Si este es el caso, use la Licencia Pública General de GNU para Bibliotecas en lugar de esta Licencia.

Anexo 2
Reporte de Nessus

Introduction

In this test, Nessus has tested 1 host and found **1 severe security holes**, as well as 8 security warnings and 11 notes.These problems can easily be used to break into your network. You should have a close look at them and correct them as soon as possible.
Note that there is a big number of problems for a single network of this size.
We strongly suggest that you correct them as soon as you can, although we know it is not always possible.
On the overall, Nessus has given to the security of this network the mark E because of the number of vulnerabilities found. A script kid should be able to break into your network rather easily.
There is room for improvement, and **we strongly suggest that you take the appropriate measures to solve these problems *as soon as possible*** If you were considering hiring some security consultant to determine the security of your network, we strongly suggest you do so, because this should save your network.

newman.posgrado.unam.mx

Open ports (TCP and UDP)
132.248.73.5 has the following ports that are open :

- https (443/tcp)
- general/udp
- general/tcp

You should disable the services that you do not use, as they are potential security flaws.
Details of the vulnerabilities

Problems regarding : https (443/tcp)

Security holes :

- It is possible to execute arbitrary php functions on the remote host by using a flaw in the 'Nuked Klan' package.

  An attacker may use this flaw to leak information about the remote system or even execute arbitrary commands.

  In addition to this problem, this service is vulnerable to various cross site scripting attacks.

  Solution : contact the author for a patch
  Risk factor : Serious
  BID : 6916, 6917, 6697, 6699, 6700

Security warnings :

- Your website allows read access to the CVS/Entries file. This exposes all file names in your CVS module on your website.

  Solution: Change your website permissions to deny access to your CVS directory. Entries contains the following:
  /404.html/1.1.1.1/Thu Mar 7 16:59:20 2002//
  /alumnos.html/1.1.1.1/Thu Mar 7 16:59:20 2002//
  /buzon.html/1.1.1.1/Thu Mar 7 16:59:20 2002//
  /conPost.php/1.1.1.1/Thu Mar 7 16:59:20 2002//
  /conSyb.php/1.1.1.1/Thu Mar 7 16:59:20 2002//
  /creditos.html/1.1.1.1/Thu Mar 7 16:59:20 2002//

```
/cuerpo.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/fer.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/home.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/html.php/1.1.1.1/Thu Mar  7 16:59:20 2002//
/icons.tar.gz/1.1.1.1/Thu Mar  7 16:59:20 2002//
/index.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/index.html.rpmnew/1.1.1.1/Thu Mar  7 16:59:20 2002//
/index.htmlBak13Ago2001/1.1.1.1/Thu Mar  7 16:59:20 2002//
/index.php/1.1.1.1/Thu Mar  7 16:59:20 2002//
/index.shtml/1.1.1.1/Thu Mar  7 16:59:20 2002//
/index.shtml.rpmnew/1.1.1.1/Thu Mar  7 16:59:20 2002//
/indexnswf.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/indexswf.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/info.php3/1.1.1.1/Thu Mar  7 16:59:20 2002//
/medicina.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/menu.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/menu_alumnos.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/phpinfo.php/1.1.1.1/Thu Mar  7 16:59:20 2002//
/planes.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/planes_consulta.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/plantilla.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/plantilla_con_js.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/platform.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/poweredby.gif/1.1.1.1/Thu Mar  7 16:59:20 2002//
/poweredby.png/1.1.1.1/Thu Mar  7 16:59:20 2002//
/prueba.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/prueba.swf/1.1.1.1/Thu Mar  7 16:59:20 2002//
/temas.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
/tutores.html/1.1.1.1/Thu Mar  7 16:59:20 2002//
D/acerca////
D/alumnos////
D/aspirantes////
D/bib_php////
D/credenciales////
D/css////
D/directorio////
D/documentacion////
D/formas////
D/formatos////
D/historias////
D/icons////
D/images////
D/insc2000////
D/inscribe////
D/inscripciones////
```

D/js////
D/manual////
D/medicina////
D/personal////
D/piranha////
D/pruebas////
D/pumas////
D/quimica////
D/sisesc1////
D/swf////
D/web_anterior////

- The remote host seems to be running MyAbraCadaWeb. An attacker may use it to perform a cross site scripting attack on this host, or to reveal the full path to its physical location.

  Solution: Upgrade to a newer version.
  Risk factor : Medium
  BID : 7126, 7127

- Nuked-klan 1.3b has a cross site scripting bug. An attacker may use it to perform a cross site scripting attack on this host.

  In addition to this, another flaw may allow an attacker to obtain the physical path of the remote CGI directory.

  Solution : Upgrade to a newer version.
  Risk factor : Medium
  BID : 6916, 6917

- Mambo Site Server is an open source Web Content Management System. An attacker may use it to perform a cross site scripting attack on this host.

  Solution: Upgrade to a newer version.
  Risk factor : Medium
  BID : 7135

- Your webserver supports the TRACE and/or TRACK methods. TRACE and TRACK are HTTP methods which are used to debug web server

```
connections.

It has been shown that servers supporting this method are
subject to cross-site-scripting attacks, dubbed XST for
"Cross-Site-Tracing", when used in conjunction with
various weaknesses in browsers.

An attacker may use this flaw to trick your legitimate web users
to give him their credentials.

Solution: Disable these methods.


If you are using Apache, add the following lines for each virtual
host in your configuration file :

    RewriteEngine on
    RewriteCond %{REQUEST_METHOD} ^(TRACE|TRACK)
    RewriteRule .* - [F]

If you are using Microsoft IIS, use the URLScan tool to deny HTTP
TRACE requests or to permit only the methods needed to meet site
requirements and policy.

If you are using Sun ONE Web Server releases 6.0 SP2 and later,
add the following to the default object section in obj.conf:
    <Client method="TRACE">
     AuthTrans fn="set-variable"
     remove-headers="transfer-encoding"
     set-headers="content-length: -1"
     error="501"
    </Client>

If you are using Sun ONE Web Server releases 6.0 SP2 or below,
compile the NSAPI plugin located at:
http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F50603


See http://www.whitehatsec.com/press_releases/WH-PR-20030120.pdf
http://archives.neohapsis.com/archives/vulnwatch/2003-q1/0035.html
http://sunsolve.sun.com/pub-cgi/retrieve.pl?doc=fsalert%2F50603
http://www.kb.cert.org/vuls/id/867593

Risk factor : Medium
```

- The SSLv2 server offers 5 strong ciphers, but also
  0 medium strength and 2 weak "export class" ciphers.
  The weak/medium ciphers may be chosen by an export-grade
  or badly configured client software. They only offer a
  limited protection against a brute force attack

  Solution: disable those ciphers and upgrade your client
  software if necessary

- The following files are calling the function phpinfo() which
  disclose potentially sensitive information to the remote
  attacker : /phpinfo.php

  Solution : Delete them or restrict access to them
  Risk factor : Low

Security note :

- A SSLv2 server answered on this port

- A web server is running on this port through SSL

- The following directories were discovered:
  /CVS, /cgi-bin, /css, /html, /icons, /images, /js, /manual,
  /personal,/pruebas, /acerca, /alumnos, /aspirantes, /bib_php,
  /credenciales, /directorio, /documentacion, /formas, /formatos,
  /historias, /insc2000, /inscribe, /inscripciones, /medicina,
  /pumas, /quimica, /sisesc1, /swf, /web_anterior

  While this is not, in and of itself, a bug, you should manually
  inspect these directories to ensure that they are in compliance
  with company security standards

- The following CGI have been discovered :

  Syntax : cginame (arguments [default value])

  /cgi-bin/medicina/acceso.pl (clave[] acceso[] Funcionarios[])
  /cgi-bin/medicina/Entrada_alumnos.pl (cuenta[] dia[] mes[]
  anio [19] Alumnos[])

- This web server was fingerprinted as Apache/1.3.27-9 on Linux w/ mod_fastcgi?E which is consistent with the displayed banner: Apache/1.3.29 (Unix) PHP/4.3.4 mod_ssl/2.8.16 OpenSSL/0.9.7c

- The remote web server type is :

  Apache/1.3.29 (Unix) PHP/4.3.4 mod_ssl/2.8.16 OpenSSL/0.9.7c

  Solution : You can set the directive 'ServerTokens Prod' to limit the information emanating from the server in its response headers.

- An information leak occurs on Apache based web servers whenever the UserDir module is enabled. The vulnerability allows an external attacker to enumerate existing accounts by requesting access to their home directory and monitoring the response.

  Solution:
  1) Disable this feature by changing 'UserDir public_html' (or whatever) to 'UserDir disabled'.

  Or

  2) Use a RedirectMatch rewrite rule under Apache -- this works even if there is no such entry in the password file, e.g.:
  RedirectMatch ^/~(.*)$ http://my-target-webserver.somewhere.org/$1

  Or

  3) Add into httpd.conf:
  ErrorDocument 404 http://localhost/sample.html
  ErrorDocument 403 http://localhost/sample.html
  (NOTE: You need to use a FQDN inside the URL for it to work properly).

  Additional Information:
  http://www.securiteam.com/unixfocus/5WP0C1F5FI.html

  Risk factor : Low
  CVE : CAN-2001-1013
  BID : 3335

- Here is the SSLv2 server certificate:
  Certificate:
      Data:

```
        Version: 3 (0x2)
        Serial Number: 5 (0x5)
        Signature Algorithm: md5WithRSAEncryption
        Issuer: C=XY, ST=Snake Desert, L=Snake Town,
                O=Snake Oil, Ltd,
OU=Certificate Authority, CN=Snake Oil
CA/emailAddress=ca@snakeoil.dom
        Validity
            Not Before: Jan 27 01:01:05 2004 GMT
            Not After : Jan 26 01:01:05 2005 GMT
        Subject: C=MX, ST=Distrito Federal, L=Ciudad de Mexico,
O=UNAM, DGAE, OU=UAP,
CN=newman.posgrado.unam.mx/emailAddress=uap@uap.unam.mx
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
            RSA Public Key: (1024 bit)
              Modulus (1024 bit):
                00:c6:7d:41:5a:d5:9f:83:0a:1e:91:48:f8:d9:ce:
                07:3e:ee:0c:be:f6:53:a8:30:35:46:06:94:08:cc:
                45:35:f9:87:7b:cb:a2:d8:a2:32:ea:3c:cc:9d:ec:
                53:94:10:e1:c6:9f:25:0c:86:3b:69:e9:11:a7:2e:
                da:a1:98:ef:1e:c3:c5:40:0b:e2:c2:7b:fa:16:cc:
                9f:87:ca:6d:4a:0c:9d:c0:91:98:b6:a6:c4:06:c6:
                7f:c1:2e:b7:66:0f:5c:c7:99:1c:b8:26:38:5e:76:
                0f:fd:5f:56:90:3d:0f:50:37:ea:f3:7a:46:94:fe:
                e8:42:94:6b:b4:16:2e:e5:eb
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Subject Alternative Name:
                email:uap@uap.unam.mx
            Netscape Comment:
                mod_ssl generated test server certificate
            Netscape Cert Type:
                SSL Server
    Signature Algorithm: md5WithRSAEncryption
        25:96:f0:98:0c:67:de:8b:c3:43:d9:a5:d7:8b:6f:77:2f:09:
        2b:b1:2f:ac:ad:5f:ca:eb:c6:28:af:99:4a:14:13:09:47:99:
        23:cb:3a:9f:0f:0f:85:c1:8d:12:7d:68:fb:fa:fb:46:71:8a:
        c2:03:ab:e7:0f:4a:f9:65:ba:38:0c:57:55:6a:19:f3:61:f3:
        6a:b3:b0:c6:db:96:bf:ae:f8:81:7f:5c:8e:92:d0:4d:32:ab:
        36:1c:8a:b4:da:4b:d6:06:75:eb:d1:79:5f:92:71:23:08:c5:
        96:7d:83:ec:b4:0b:5f:cd:3b:f0:e4:cb:c5:eb:fd:45:8c:22:
        4b:5f
```

- Here is the list of available SSLv2 ciphers:
  RC4-MD5
  EXP-RC4-MD5
  RC2-CBC-MD5
  EXP-RC2-CBC-MD5
  DES-CBC-MD5
  DES-CBC3-MD5
  RC4-64-MD5

- This SSLv2 server also accepts SSLv3 connections.
  This SSLv2 server also accepts TLSv1 connections.

Problems regarding : general/udp

Security note :

- For your information, here is the traceroute to 132.248.73.5 :
  132.248.124.181
  132.248.124.190
  132.248.120.250
  132.248.255.110
  132.248.255.198
  132.248.73.5

Problems regarding : general/tcp

Security warnings :

- The remote host does not discard TCP SYN packets which
  have the FIN flag set.

  Depending on the kind of firewall you are using, an
  attacker may use this flaw to bypass its rules.

  See also :
  http://archives.neohapsis.com/archives/bugtraq/2002-10/0266.html
  http://www.kb.cert.org/vuls/id/464113

  Solution : Contact your vendor for a patch
  Risk factor : Medium
  BID : 7487

Conclusion

A security scanner, such as Nessus, is not a garantee of the security of your network. A lot of factors can not be tested by a security scanner : the practices of the users of the network, the home-made services and CGIs, and so on... So, you should not have a false sense of security now that the test are done. We recommand that you monitor actively what happens on your firewall, and that you use some tools such as tripwire to restore your servers more easily in the case of an intrusion.
In addition to that, you must know that new security holes are found each week. That is why we recommand that you visit `http://www.nessus.org/scripts.html`, which is a page that contains the test for all the holes that are published on public mailing lists such as BugTraq (see `http://www.securityfocus.com` for details) and test the security of your network on a (at least) weekly basis with the checks that are on this page.
This report was generated with Nessus, the open-sourced security scanner. See http://www.nessus.org for more information

Bibliografía

- **Practical Unix & Internet Security**
  Garfinkel, Simson y **Spafford**, Eugene.
  Estados Unidos, 2003.
  Tercera edición.
  Ed. O'Reilly.
  984 páginas.

- **Web Security, Privacy & Commerce**
  Garfinkel, Simson y **Spafford**, Eugene.
  Estados Unidos, 2002.
  Segunda edición.
  Ed. O'Reilly.
  756 páginas.

- **Database Security**
  Castano, Silvana.
  Estados Unidos, 1995.
  Primera edición.
  Ed. Addison-Wesley.
  456 páginas.

- **Database Security and Integrity**
  Fernandez, Eduardo B., **Summer**, Rita C. y **Wood**, Christopher.
  Estados Unidos,1981.
  Primera edición.
  Ed. Addison-Wesley.
  320 páginas.

- **Practical Postgres**
  Worsley, John, **Drake**,Joshua.
  Estados Unidos, 2002.
  Primera edición.
  Ed. O'Reilly.
  636 páginas.

- **PGP, Pretty Good Privacy**
  Garfinkel, Simson.

Estados Unidos, 1995.
Primera edición.
Ed. O'Reilly.
393 páginas.

- **Building Internet Firewalls, 2nd Edition**
  D. Zwicky, Elizabeth y **Cooper** Simon.
  Estados Unidos, 2000.
  Segunda Edición.
  Ed. O'Reilly.
  890 páginas.

- **Computer Security, Art and Science**
  Bishop, Matt.
  Estados Unidos, 2003.
  Primera Edición.
  Ed. Addison Wesley.
  1084 páginas.

- **IT Security, Risking the Corporation**
  McCarthy, Linda.
  Estados Unidos, 2003.
  Primera Edición.
  Ed. Prentice Hall.
  246 páginas.

- **Hacking Exposed: Network Security Secrets and Solutions**
  Scambray, Joel, **McClure**, Stuart y **Kurtz**, George.
  Estados Unidos, 2001.
  Segunda Edición.
  Ed. McGraw Hill.
  703 páginas.

- **Network Perimeter Security**
  Northcutt, Stephen y otros.
  Estados Unidos, 2003.
  Primera Edición.
  Ed. New Riders.
  678 páginas.

- **Network Intrusion Detection**
  Northcutt, Stephen y **Novak**, Judy.
  Estados Unidos, 2002.
  Tercera Edición.
  Ed. New Riders.
  490 páginas.

Referencias en Internet

- Computer Security Institute
  www.gocsi.com
- Computer Emergency Response Team / Cordination Center (CERT/CC)
  www.cert.org
- SysAdmin, Audit, Networking and Security Institute (SANS)
  www.sans.org
- Departamento de Seguridad en Cómputo de la UNAM
  www.seguridad.unam.mx
- Equipo de Respuesta a Incidentes de la UNAM (UNAM-CERT)
  www.unam-cert.unam.mx
- Forum of Incident Response and Security Teams (FIRST)
  www.first.org
- Security Focus
  www.securityfocus.org
- Security Space
  www.securityspace.com
- Zona - H
  www.zone-h.org
- Servidor Apache
  www.apache.org
- Open Secure Socket Layer
  www.openssl.org
- Módulo SSL
  www.modssl.org
- PostgreSQL
  www.postgresql.org
- Fundación para el Software Libre (FSF)
  www.fsf.org o www.gnu.org
- Tripwire
  www.tripwire.org
- Nmap
  www.insecure.org

- Portsentry
  ftp.seguridad.unam.mx
- Nessus
  www.nessus.org
- Webalizer
  www.webalizer.org
- OpenSSH
  www.openssh.org
- Linux Intrusion Detection System (LIDS)
  www.lids.org
- Kernel de Linux
  www.kernel.org
- GNU Privacy Guard (GPG)
  www.gnupg.org

Glosario

- *Back door*. Ver Puerta Trasera.
- *Bug*. Es una propiedad indeseada o intencionada en un programa o pieza de hardware que causa un mal funcionamiento.
- *Ciber-café*. Establecimiento que cuenta con algunas computadoras en renta con acceso a Internet. Estos lugares son frecuentados principalmente por estudiantes para realizar trabajos escolares.
- *Cracker*. Es una persona que ingresa sin autorización a un sistema en búsqueda de un beneficio personal.
- *DoS*. Denial of Service. La negación de servicio es un tipo de ataque en el cual los usuarios válidos dejan de tener acceso a un sistema.
- *Escanear*. Término utilizado en seguridad para rastrear o buscar un puerto abierto.
- *Exploit*. Programa que aprovecha una vulnerabilidad en un sistema para acceder sin autorización.
- *Firewall*. Programa o dispositivo encargado de proteger una red. Existe dos tipos de firewall: Firewall de Filtrado de Paquetes y Firewall de Filtrado de Aplicaciones, también conocidos como Proxys.
- *Hacker*. **1.**Persona que disfruta aprendiendo los detalles de los sistemas de computadoras y cómo llevarlas hasta el máximo de sus capacidades (contrario a la mayoría de las personas, quienes prefieren aprender sólo lo mínimo necesario). **2.** Persona que programa entusiastamente o que disfruta programando, más que sólo teorizar acerca de la programación.
- *Ingeniería Social*. Tipo de ataque mediante el cual se obtiene información sensitiva a través del engaño a las personas. Este tipo de ataques no requiere de muchos conocimiento ténicos.
- *IP Spoofing*. Esta técnica consiste en enviar una dirección IP falsa para engañar a un equipo.

- *Lamer*. Es un tipo de intruso el cual carece de conocimientos suficientes para entender un programa que explota una vulnerabilidad de un sistema. Este tipo de usuarios no aportan nada y sólo se encargan de descargar programas y utilizarlos "para ver que pasa" y fastidiar a alguien.

- *Navegador*. Interface de usuario utilizada para acceder al servicio de páginas web. Estos navegadores utilizan el protocolo HTTP e interpretan distintos lenguajes como html.

- *No repudiación*. Es la capacidad de un sistema para probar que un usuario, y sólo ese usuario específico, envió un mensaje y que éste no ha sido modificado.

- *OpenBSD*. Sistema operativo tipo Unix que se caracteriza por el código auditado que utiliza, haciéndolo un sistema operativo muy seguro.

- *Puerta Trasera*. Es una herramienta instalada por un intruso después de comprometer un sistema que le garantiza al atacante el acceso al sistema a pesar de los mecanismos de seguridad instalados.

- *RFC*. Request For Comment. Son una serie de documentos técnicos y organizacionales sobre Internet (originalmente ARPANET) que dan comienzo en 1969. Estos documentos discuten muchos aspectos de redes de computadoras, incluyendo protocolos, procedimientos, programas y conceptos. Los documentos de especificación oficial del Protocolo de Internet están definidos por la Internet Engineering Task Force (IETF) y son publicados mediante RFCs. Por lo tanto, el proceso de publicación de RFC juega un papel importante en la definición de estándares de Internet.

- *Ruteador*. Dispostivo de red encargado de dirigir los paquetes a través de redes lógicas basadas en direcciones IP.

- *UPS (**U**ninterruptable **P**ower **S**uply)*. Dispositivo utilizado para proporcionar una fuente de energía eléctrica permanente.

- *URL*. Uniform Resource Locator. Es una forma estándar para especificar la posición de un objeto en Internet, típicamente una página web. El URL está compuesto de tres elementos:

  - Protocolo. Define el esquema de acceso. Pueden ser: http, ftp, gopher, etc.
  - Host. Equipo en Internet donde se encuentra el recurso deseado. Se hace referencia a este equipo mediante un nombre o una dirección IP.
  - Objeto. Es el elemento al cual se desea acceder, ya sea una página web, un archivo de texto o un archivo binario.

El Url se define de la siguiente manera:

protocolo://host/objeto

Por ejemplo:

http://www.midominio.com.mx/imagen.gif

- *Script*. Un script es una serie de líneas de código para generar un programa en lotes. Este tipo de programas son muy utilizados en sistemas Unix para realizar programas utilizando comandos del sistema operativo, se les conocen como *shell scripts*.

- *Script kiddie*. Son personas que lo único que saben es ejecutar programas para acceder ilícitamente a un sistema de cómputo. Los crackes los consideran como personas con poca creatividad debido a que sólo ejecutan programas para atacar huecos de seguridad que otros han descubierto y documentado. Sin embargo, a pesar de su poco conocimiento, son de los intrusos más peligrosos debido a que pueden utilizar herramientas que automatizan la búsqueda en masa de sistemas vulnerables .

Índice de figuras

Índice de cuadros