UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGON

IMPLANTACION DE UN SISTEMA DE SEGURIDAD PARA UNA APLICACIÓN DESARROLLADA EN INFORMIX-SE EN EQUIPOS UNIX SYSTEM V

T E S I S

QUE PARA OBTENER EL TITULO DE

INGENIERO EN COMPUTACION

P R E S E N T A N:

ADELA TORRES PEREZ

CYNTHIA REBECA OLIVARES URESTI

ASESOR: LIC. ISRAEL JUAREZ ORTEGA

México

1999

Agradecimientos

Quisiera aprovechar la oportunidad para agradecer a las personas que hicieron posible este primer gran salto.

A Mi Madre por, el apoyo incondicional, comprensión y confianza brindadas para la realización de mi carrera profesional.

Al claro ejemplo de valentía e inteligencia de mis hermanos y hermanas que ha sido mi motor para seguir adelante.

A mis amigos Juan, Andrés y Cesar por su entusiasmo, conocimientos y facilidades prestadas para realizar este trabajo.

A mis compañeros de clase, profesores y académicos de la Escuela Nacional de Estudios Profesionales Aragón, por la oportunidad de conocerlos, por compartir su sabiduría y la labor que desempeñan.

A la Universidad Nacional Autónoma de México por abrirme sus puertas y darme el orgullo de ser universitario.

... Gracias a ellos he culminado uno de mis más grandes anhelos.

Gracias a ellas quienes me mostraron cómo vivir y cómo no hacerlo,

cuya fortaleza y compasión sostuvieron una antorcha de luz

en mi avance,

cuya crítica, desilusión y falta de fe me llevaron hacia niveles

más profundos de compromiso y resolución.

Tabla Contenido

INTRODUCCIÓN

El Sistema Integral de Información Tributaria (S.I.I.T.) surgió a partir de la necesidad de controlar toda la información que maneja la Secretaría de Hacienda y Crédito Público. Se realizó una junta entre todas las unidades informáticas de la misma para establecer el tipo de políticas a seguir para el desarrollo del sistema que sería capaz de manejar la información de las áreas involucradas.

Como resultado se llegó al acuerdo de que cada Subsecretaría (de Ingresos, de Egresos, etc.) recurriría a sus propios medios para el desarrollo del sistema, siguiendo la estandarización de que la plataforma debía ser UNIX, la programación en Informix 4gl y el manejador de la base de datos sería Informix SE. Esta definición tenía como objetivo el que al termino del desarrollo se unieran todos los sistemas en un solo Macro Sistema que permitiera el intercambio de información eficiente y obtención inmediata de estadísticas para apoyar la toma de decisiones a nivel Subsecretaría.

Resultó ser un proyecto demasiado ambicioso, por lo que muchas de los Unidades Informáticas no terminaron siquiera el análisis de su parte del proyecto.

A diferencia de las demás áreas, la Administración de Informática perteneciente en aquel entonces a la D.G.T.I. (A.G.J.I. actualmente) destinó recursos al desarrollo de su aplicación. Al principio no cubría del todo el flujo de información que existía en la D.G.T.I., únicamente se extendía a las áreas que en aquel entonces desempeñaban las funciones primordiales de la Dirección.

Al cambiar de nombre la Dirección General Técnica de Ingresos por el Administración General Jurídica de Ingresos, se le añadieron nuevas funciones a esta Administración, entre las cuales se encontraba Notificación y Cobranza que debía ser agregada al S.I.I.T. para su control.

Al terminar este módulo, se le agregó otro al sistema que fue bautizado como Contencioso debido a que la Administración Central de lo Contencioso era el área que lo estaba solicitando. Con éste, eran ya tres los módulos que el S.I.I.T. contenía: Control de Gestión, Notificación y Cobranza y lo Contencioso.

Cuando inició su operación en 1993 a nivel nacional, fue implementado en equipos U6000/35 con sistema operativo Unix System V Release 1.1. El equipo no estaba conectado en red por lo que la

única seguridad que era necesaria tener habilitada era la física para que no cualquier persona tuviera acceso a la máquina y por lo consiguiente a la información.

Con la entrada de la red WAN en la S.H.C.P. se obtuvieron muchos beneficios, pero en ese momento también salieron a la luz muchos problemas de diseño que tenia el S.I.I.T. De hecho cualquier usuario experimentado que tuviera una cuenta en el sistema operativo podía insertar, consultar, modificar o borrar desde un registro de la base de datos hasta la base entera.

Fue hasta ese momento en que salió a relucir que el diseño del S.I.I.T. no incluía ningún tipo de seguridad asociado.

Por lo que este trabajo de tesis tiene como propósito implantar seguridad en todos los ámbitos que conlleva la administración de un centro de computo y explotación de la información seguridad, en la Secretaria.

El primer capitulo contiene, las definiciones pertinentes y el enfoque de seguridad que se tratará a lo largo del presente: se menciona teóricamente que medidas de protección y control pueden ser llevadas a cabo en este ámbito. En orden se describe las políticas de seguridad, la seguridad física, la seguridad en software, en hardware, la seguridad tan valiosa que requiere la información, la de las comunicaciones y por ultimo la auditoria.

El segundo capitulo, abarca de manera especifica la seguridad que proporciona el sistema operativo. El S.I.I.T. opera bajo plataforma UNIX versión V, por lo que este capitulo contempla una provechosa descripción de la herramienta, su entorno como ambiente de trabajo y las alternativas de seguridad que nos otorga, como son: monitoreo de accesos, cuentas y passwords, usuario root y el individual sistema de seguridad de UNIX, el Audit Trail.

Los datos de esta aplicación son manipulados por el lenguaje Informix-SE. Por esta razón, el capitulo III nos explica el tipo de lenguaje al que nos enfrentamos, sus características, la ventajosa aplicación de los registros de Transacciones Log y Audit Trail's, así como los niveles de seguridad que nos proporciona por default la Base de Datos.

El cuarto capitulo, hace referencia totalmente a la aplicación, en primer lugar nos expone la situación actual del sistema, se cuenta con el diagrama conceptual del negocio y contiene un análisis de deficiencias de seguridad que han sido identificadas en su operación a la fecha.

Finalmente en el capitulo V, se detallan las precauciones, medidas y controles que se deben implantar en el Centro de Cómputo de esta Subadministración. En cuanto a la explotación del S. I. I. T. se puntualizan que comandos de UNIX y la manera correcta de hacer uso de ellos. Se expone un algoritmo de un sistema Auditor alterno, que permitirá saber con precisión como han sido manipulados los datos en las tablas más importante de la Base de Datos central, detectando asi incoherencias, cuando se posea la duda.

Se consideran los capítulos citados anteriormente suficientes y valiosos, para asegurar que la información que se procesa en el Sistema Integral de Información Tributaria (S. I. I. T.) cumple con los requisitos indispensables de seguridad como son: Autenticidad, Integridad y Confidencialidad.

CAPÍTULO 1

CONCEPTOS DE INDUCCIÓN

1.1. Definición de Seguridad

La seguridad en un sistema de cómputo se define como: la confianza que se tiene en que el sistema se comporte como los usuarios esperan que lo haga. Si la información almacenada en él, se mantiene inalterada y accesible de manera y durante tanto tiempo como el dueño lo desee; se estará logrando gran parte del objetivo de seguridad. Sin embargo, existen grandes y diversas maneras de brindar seguridad a los sistemas de cómputo.

Para conseguir que un sistema de cómputo sea completamente seguro, se necesita tomar en cuenta diferentes aspectos, como son: las instalaciones donde se aloja el Centro de Cómputo (C. C.), el equipo, sistemas operativos, paquetería, respaldos de información, equipo de control de desastres, inclusive reclutamiento y capacitación de personal, etc.

La complejidad de definir a la seguridad en cómputo radica, en que es un tema demasiado subjetivo, pues son diversos los niveles de seguridad que puede necesitar una institución. Estos niveles dependen del tamaño de la institución, de su giro, y hasta de su ubicación. Cada Centro de Procesamiento de Información, necesitará un análisis propio y adecuado para operar de manera segura.

Es importante tomar una actitud realista ante la seguridad, y no caer en la "fiebre tecnológica" de implementar todo lo nuevo solamente por hacerlo, sin realizar algún tipo de examen previo.

De forma general, los pasos que se tienen que seguir se pueden resumir en tres preguntas:

1. **¿Qué se quiere proteger?** Es difícil proteger algo cuando no se sabe qué es. Se tiene que identificar claramente qué sistemas y(o) servicios se desean proteger, basándose en un análisis de las partes más vulnerables, de mayor importancia o de un posible mayor interés para los intrusos.

2. **¿Contra qué se quiere proteger? ¿Cuáles son los riesgos?** Si el sistema que se quiere proteger está desconectado de la red, solamente tiene un usuario autorizado y dicho usuario tiene que entrar personalmente al edificio del sistema, bajo medidas de seguridad físicas muy estrictas,

posiblemente muchos de los riesgos ya están eliminados, y no se necesitan tomar muchas medidas al respecto.

3. **¿Cuánto tiempo, dinero y esfuerzo se puede invertir para lograrlo?** Desgraciadamente, esta pregunta es la que, en la gran mayoría de las ocasiones, determina en última instancia el trabajo que se haga en seguridad. Si se necesitan niveles de seguridad aun más altos, posiblemente sea necesario formar un equipo de gente, darles capacitación, recursos, etc.

Vale la pena mencionar que los niveles de seguridad se pueden incrementar hasta niveles muy altos, si se cuenta con los recursos apropiados, pero se considera imposible eliminar por completo los riesgos.

Se sabe que los niveles de seguridad que es posible obtener, varían en razón directa a los recursos invertidos en ello y se dice comúnmente que la seguridad absoluta solamente se puede obtener a un costo infinito.

Es un hecho que es posible obtener niveles razonables de seguridad, es decir, apropiados para nuestras necesidades, dedicándole un poco de tiempo y esfuerzo como administradores de sistemas.

A continuación en la tabla 1.1, se muestran los aspectos más importantes que se recomienda aplicar a las partes integrales de un Centro de Cómputo, para obtener el control de la seguridad en puntos más singulares.

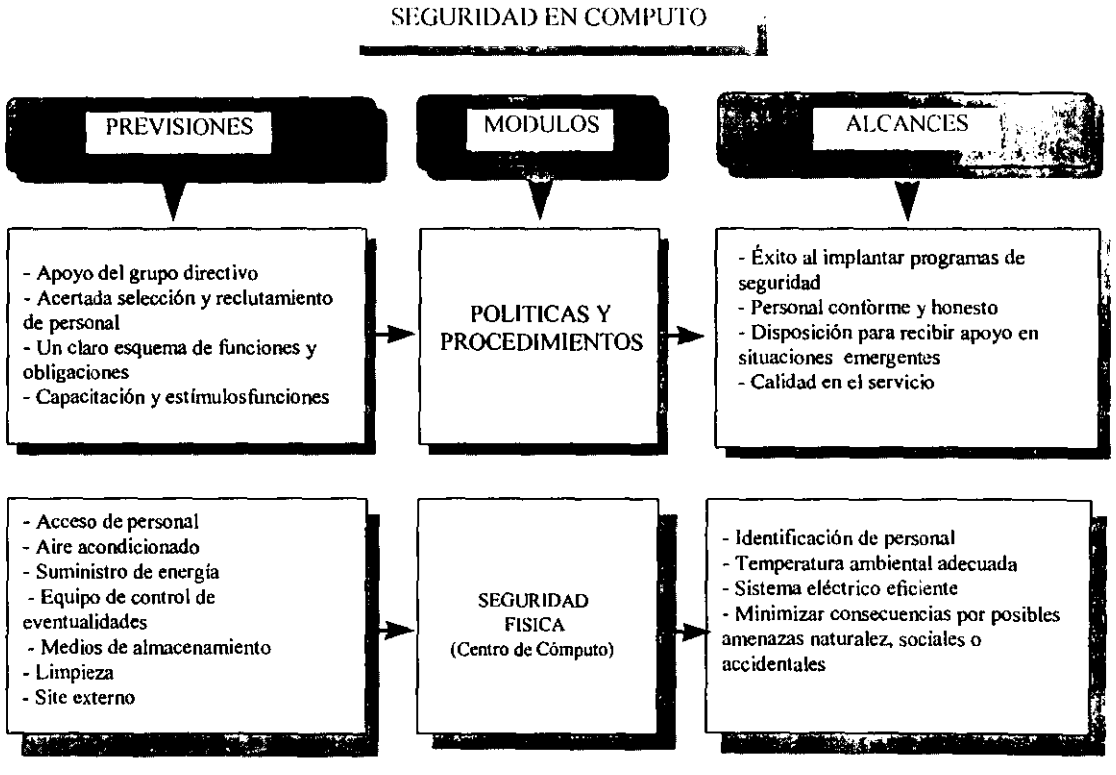

SEGURIDAD EN COMPUTO
PREVISIONES
MODULOS
ALCANCES
- Apoyo del grupo directivo
- Acertada selección y reclutamiento de personal
- Un claro esquema de funciones y obligaciones
- Capacitación y estímulosfunciones
POLITICAS Y PROCEDIMIENTOS
- Éxito al implantar programas de seguridad
- Personal conforme y honesto
- Disposición para recibir apoyo en situaciones emergentes
- Calidad en el servicio
- Acceso de personal
- Aire acondicionado
- Suministro de energía
- Equipo de control de eventualidades
- Medios de almacenamiento
- Limpieza
- Site externo
SEGURIDAD FISICA
(Centro de Cómputo)
- Identificación de personal
- Temperatura ambiental adecuada
- Sistema eléctrico eficiente
- Minimizar consecuencias por posibles amenazas naturalez, sociales o accidentales

tabla 1.1 (a)

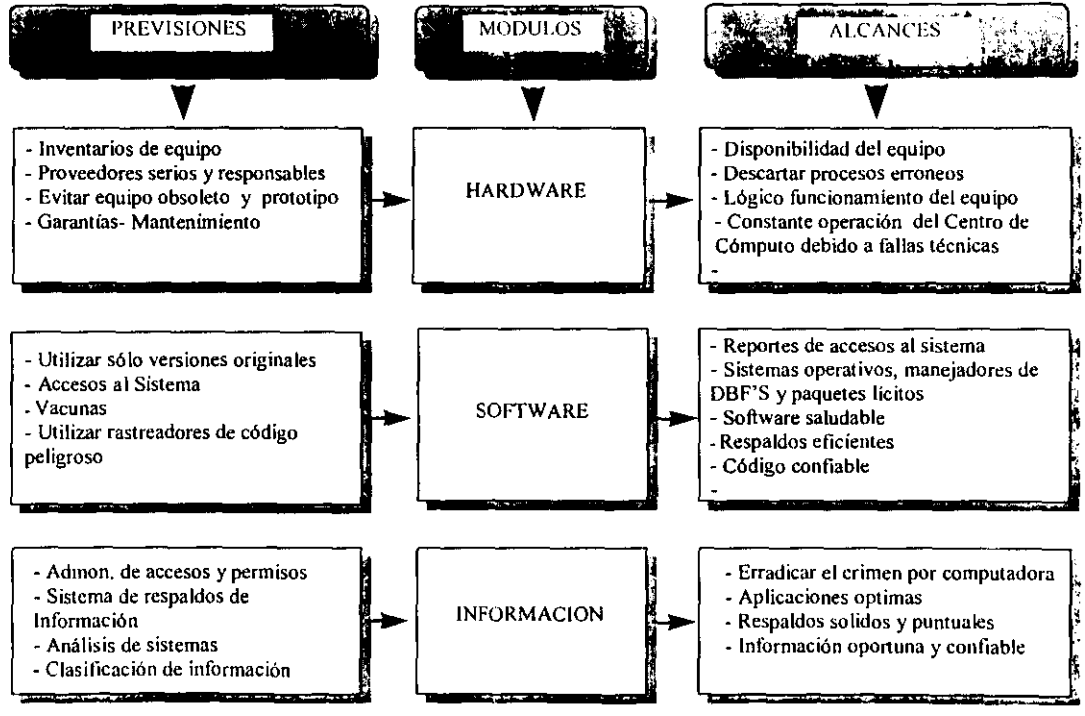

PREVISIONES
MODULOS
ALCANCES
- Inventarios de equipo
- Proveedores serios y responsables
- Evitar equipo obsoleto y prototipo
- Garantías- Mantenimiento
HARDWARE
- Disponibilidad del equipo
- Descartar procesos erroneos
- Lógico funcionamiento del equipo
- Constante operación del Centro de Cómputo debido a fallas técnicas
- Utilizar sólo versiones originales
- Accesos al Sistema
- Vacunas
- Utilizar rastreadores de código peligroso
SOFTWARE
- Reportes de accesos al sistema
- Sistemas operativos, manejadores de DBF'S y paquetes lícitos
- Software saludable
- Respaldos eficientes
- Código confiable
- Admon. de accesos y permisos
- Sistema de respaldos de Información
- Análisis de sistemas
- Clasificación de información
INFORMACION
- Erradicar el crimen por computadora
- Aplicaciones optimas
- Respaldos solidos y puntuales
- Información oportuna y confiable

tabla 1.1 (b)

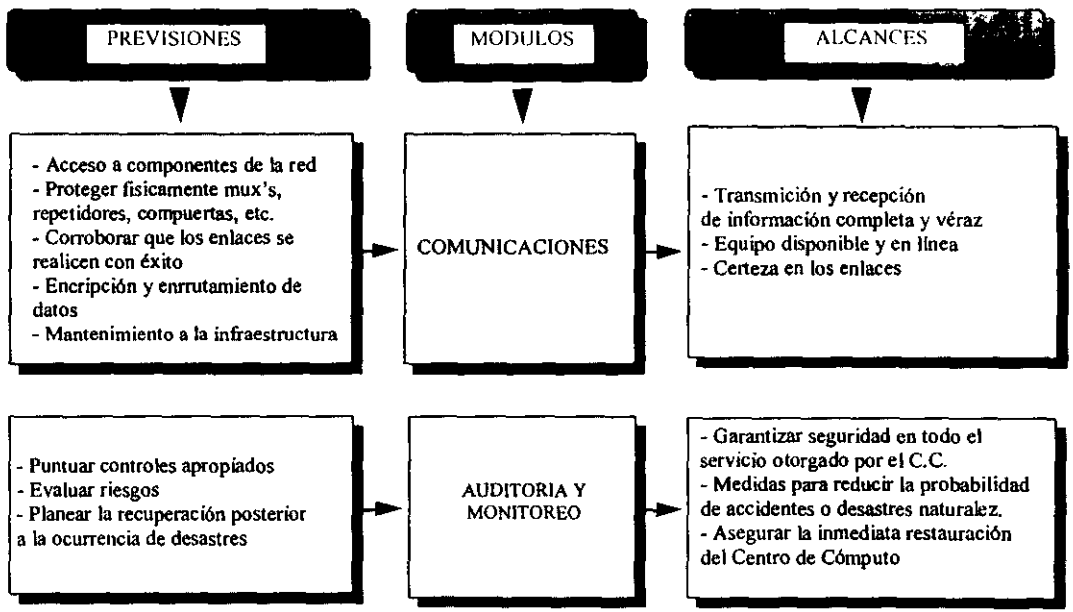

PREVISIONES
MODULOS
ALCANCES
- Acceso a componentes de la red
- Proteger físicamente mux's, repetidores, compuertas, etc.
- Corroborar que los enlaces se realicen con éxito
- Encripción y enrrutamiento de datos
- Mantenimiento a la infraestructura
COMUNICACIONES
- Transmición y recepción de información completa y véraz
- Equipo disponible y en línea
- Certeza en los enlaces
- Puntuar controles apropiados
- Evaluar riesgos
- Planear la recuperación posterior a la ocurrencia de desastres
AUDITORIA Y MONITOREO
- Garantizar seguridad en todo el servicio otorgado por el C.C.
- Medidas para reducir la probabilidad de accidentes o desastres naturalez.
- Asegurar la inmediata restauración del Centro de Cómputo

tabla 1.1 (c)

1.1.1. Aspectos más Importantes de la Seguridad

En la tabla anterior se mostraron de forma general y concreta los niveles de seguridad que se aplican a un Centro de Cómputo. Una breve descripción ayudará a entender el propósito de cada uno de ellos:

1.1.1.1. Políticas y Procedimientos

No cabe duda que el apoyo de los directivos es imprescindible, una buena clasificación de los puestos, procedimientos de contratación de personal bien ejecutados, así como también abandono y transferencia bien diseñados, dar entrenamiento al personal de nuevo ingreso acerca de las medidas de seguridad y mantener actualizado al personal, etc.: reducen la posibilidad de que personas deshonestas dentro de la compañía, en algún momento, puedan hacer mal uso de la información; por eso es conveniente contemplar dichos procedimientos para tratar de evitarlo.

1.1.1.2. Seguridad Física

Incluye los controles de acceso al centro de cómputo. Por ejemplo, por medio de tarjetas de identificación y sistemas biométricos, el suministro de energía eléctrica, condiciones del medio ambiente como temperatura, humedad, polvo, etc., equipo contra incendios, seguridad de los medios de almacenamiento de información, etc.

1.1.1.3. Seguridad en el Hardware

Del funcionamiento apropiado del equipo e infraestructura, de las conexiones de energía. Las ventajas y desventajas del uso de equipo anticuado y/o prototipo. Conservar la temperatura requerida para el equipo, inventarios y códigos de referencia, mantenimientos preventivos y correctivos. La ubicación física del equipo. Garantías.

1.1.1.4. Seguridad en el Software

Se refiere a las herramientas en común, que han de ayudar al procesamiento y en general a la administración de la información, como lo son manejadores de DBF'S, S. O. y paquetería en general. La mayoría de estas herramientas deben contemplar ciertos requerimientos de seguridad. Es

recomendable tener a la mano las últimas versiones de vacunas, a fin de minimizar los riesgos que se producirían debido a una infección de virus.

1.1.1.5. Seguridad de la Información

Clasificarla de acuerdo a la importancia que tiene para la empresa, determinara la seguridad que requiere según el tipo de información, cuanto tiempo necesita ser retenida, a quien se le permitirá tener acceso, si se requieren duplicados o no, etc. Se refiere al acceso a los sistemas de información. Incluye dispositivos y acciones como asignación y cambio periódico de passwords, restricciones de tiempo, log-off automático, sistemas de call-back, detección de intrusos o de varios intentos fallidos de ingreso al sistema. Incluye la asignación de "capacidades" de acceso a ciertos archivos como creación, borrado, modificación, lectura, escritura, etc., además la identificación de "dueños" de los archivos, quienes decidirán a quiénes le otorgan las capacidades antes descritas.

1.1.1.6. Seguridad de las Comunicaciones

Tanto de voz como de datos, vídeo, etc. por cualquier medio. Las amenazas a la seguridad de las comunicaciones son intersección, daño en los medios físicos, daño en la infraestructura de la empresa de servicios portadores, etc. Se pueden tomar diversas medidas de seguridad como encripción, ruteo diverso, call-back modems, etc.

1.1.1.7. Auditoria, Monitoreo y Ajustes

Para asegurar que las operaciones no estén mal representadas, ni amenazadas por procedimientos defectuosos o inadecuados. Se debe evaluar al personal, examinar accesos no autorizados, archivo computarizados, transmisión de datos y procedimientos de recuperación de desastres.

1.1.2. Condiciones que Incrementan la Vulnerabilidad

Existe una serie de contingencias relacionadas con los sistemas de cómputo. Para desarrollar medidas efectivas de seguridad contra éstas, es necesario entender claramente sus causas. En la siguiente tabla 1.1.2, se muestran diversas situaciones a las que queda expuesto cualquier centro de cómputo.[1]

CONTINGENCIAS

	CONTINGENCIA	CONDICIONES QUE INCREMENTAN LA VULNERABILIDAD
Desastres no intencionales		
	Error del programador	Dificultad para prever cómo funcionarán los sistemas y cómo se adaptarán a ellos los usuarios
		Complacencia al suponer que el sistema operará como se espera
		Falta de trabajo y cuidado para asegurar que los sistemas funcionen correctamente
	Falla de hardware	No creer que el hardware puede fallar
		Dificultad para decidir si la falla está en el

[1]Seguridad de la información en sistemas de Cómputo. Luis Angel Rodríguez.
Ventura Ediciones, S.A. de C.V. 1995

		hardware o en el sistema
	Errores de software	Diseño y pruebas inadecuadas
		Factores no esperados que afectan la operación del sistema
	Errores en los datos	Fallas en los procedimientos
		Falta de capacidad en el software para detectar muchos tipos de errores
		Falta de cuidado
		Sistema de respaldos inapropiados
	Daños a las instalaciones y medios de almacenamiento	Seguridad física contra fenómenos naturales no adecuada
		Protección no adecuada contra fallas en los sistemas de apoyo a la computadora
	Desempeño no adecuado de los sistemas	Mal análisis y por lo tanto, mal diseño.
		Demanda de trabajo no prevista
Desastres Intencionales		
	Robo	Diseño incorrecto del sistema de cómputo
		Existencia de muchos objetivos fáciles para el robo
		Sistemas distribuidos

	Vandalismo y sabotaje	Prevención equivocada de accesos no autorizados (a sistemas e instalaciones)
		Procedimientos desatinado de seguridad en toda la empresa

tabla 1.1.2

En forma general, la información computarizada es particularmente vulnerable porque:

- Está más concentrada
- Es más accesible
- Está sujeta a daños no detectables o uso indebido

Una lista completa para tratar la integridad de los datos podría ser muy grande y no muy usual. Esto hace pensar que hay que considerar amenazas en extensas categorías porque las medidas de seguridad son generalmente aplicables a una extenso rango de posibles amenazas.

1.1.3. Virus de Computadoras

Un Virus es una parte de código de un programa de computadora que es capaz no solo de duplicarse así mismo sino también de "pegarse" a otro programa "legítimo" (infectarlo) sin que el usuario lo detecte. Un Caballo de Troya es un programa que parece legítimo pero contiene rutinas que causan daño a otros programas o datos dentro del sistema cuando se corren. Un Gusano es un programa que hace uso del software de red y de alguna instalación de comunicaciones para replicarse a sí mismo

y moverse de sistema en sistema; los gusanos toman ventaja de las fallas de seguridad de los sistemas operativos.[2]

Los ataques de virus son exitosos debido a las siguientes razones:

- Falta de cuidado de los usuarios.
- Ausencia de controles de seguridad o controles inconvenientes.
- Uso ineficiente de los controles de seguridad eficientes.
- Errores y "agujeros" en el software.
- Uso no autorizado de los sistemas.
- Fragilidad de las redes a malos usos.

Definitivamente la destrucción de datos no es lo más grave que puede ocasionar un virus de computadora. La alteración de los datos puede crear problemas mucho mayores. ¿Que pasaría si su registro médico fuera alterado y mostrara un tipo de sangre o factor RH distinto al verdadero?

"Virus de Computadoras". Están colgados de programas. embarrados por otras inserciones de copias de ellos mismos en otros programas. Los virus de microcomputadoras han recibido mayor publicidad pero todas las computadoras basan sus sistemas con un mecanismo de almacenamiento que puede ser infectado. Los efectos de infección de virus puede ser un inconveniente o desastre. Alguna fuente de programas es una fuente potencial de infección de virus.[3]

---

[2] Seguridad de la información en sistemas de Cómputo. Luis Angel Rodríguez. Ventura Ediciones, S.A. de C.V. 1995

"Caballos de Troya". Programas que pueden hacerse pasar como aplicaciones atractivas pero causan daños a los sistemas de información cuando es activado por algún evento. La activación pudiera ser una fecha, hora contador o algún otro cambio en el sistema de información así como también el recibo del nombre de los programadores de la nomina. Programas "Trojan Horses" son algunas veces llamados "Bombas Lógicas" o "Bombas de Tiempo". Esos nombre son derivados de la naturaleza de los eventos que activen la acción del programa. Un programa podría ser ambos un virus o una bomba, puede que sea liberado como parte de un a atractiva aplicación, se copia el mismo por infección de otros programas y llegaría a destruirse cuando es activado por un evento predeterminado.[4]

"Gusanos". Son diseñados para asegurar su propia supervivencia. Difieren de los virus en que ellos pueden existir y replicarse entre ellos mismos sin estar atados al programa principal. Los gusanos se replican en una progresión geométrica como una cadena de caracteres. Cada copia usa algún recurso del sistema y crea más copias mientras el resto del sistema es dedicado para copiar y ejecutar el Worm (gusano). La figura 1.1.3, muestra las principales fuentes de infección de virus.[5]

---

[3] McAfee, J. And Haynes, C. Computer Viruses, Worms, Data Diddlers, Killer Programs, and others Threats to your System. New York: St. Martin's Press, 1989.
[4] McAfee, J. And Haynes, C. Computer Viruses, Worms, Data Diddlers, Killer Programs, and others Threats to your System. New York: St. Martin's Press, 1989.
[5] McAfee, J. And Haynes, C. Computer Viruses, Worms, Data Diddlers, Killer Programs, and others Threats to your System. New York: St. Martin's Press, 1989.

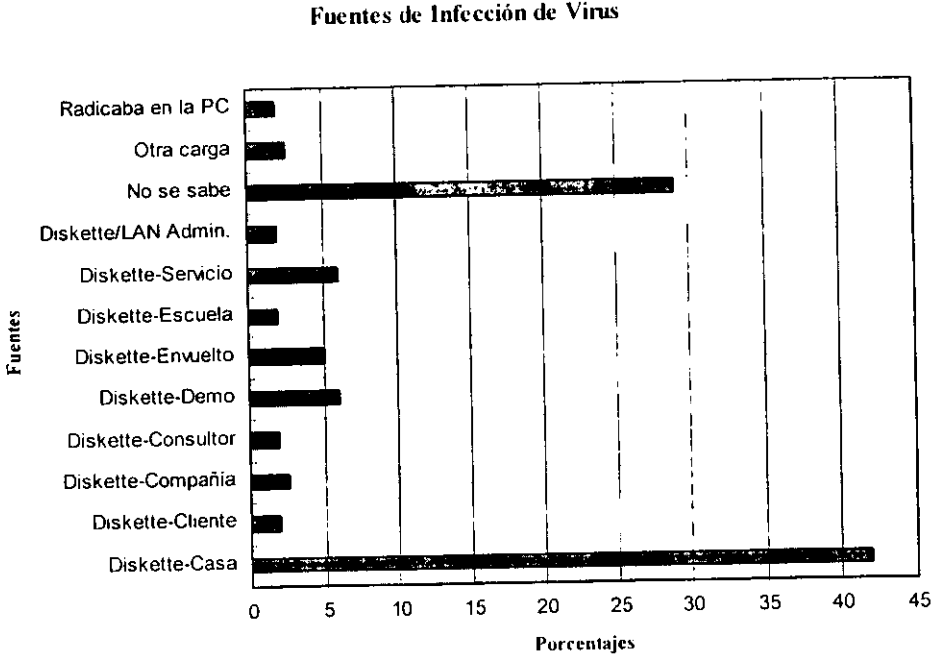
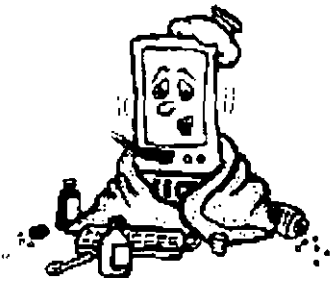

Fuentes de Infección de Virus
Radicaba en la PC
Otra carga
No se sabe
Diskette/LAN Admin.
Diskette-Servicio
Diskette-Escuela
Diskette-Envuelto
Diskette-Demo
Diskette-Consultor
Diskette-Compañia
Diskette-Cliente
Diskette-Casa
Fuentes
0
5
10
15
20
25
30
35
40
45
Porcentajes

figura 1.1.3

1.1.3.1. Medidas para Combatir la Amenaza de los Virus



Prevenir detectar y erradicar virus es una especialidad. Controlarlos es más conveniente para tener éxito. Si un centro es establecido para tratar con el problema de virus. Este contará con personal capacitado para dar consejos y apoyo.

Una práctica sana es hacer regularmente respaldos de los archivos en cualquier caso. La mayoría de los sistemas multiusuarios proveen este servicio en forma automática.

El aspecto de la información debe ser incluido en el temario del curso de capacitación. puesto que puede ser atacada por software maligno. La necesidad de respaldar y revisar la integridad de la información tiene que ser enfatizada.

La alta calidad y controles de cambio deberían usarse para prevenir a los empleados de inserción de "Bombas de Tiempo" o "Bombas lógicas" en sus programas.

La Máquina llave tienen que estar aislada tanto como sea posible; de igual manera la instalación de programas de defensa de virus radicados en ella.

A los usuarios permitirles solo el acceso a un mínimo de programas necesarios para su trabajo.

Los empleados deben tener prohibido instalar software no autorizado en las computadoras de la oficina.

Los centros competentes deberían revisar todo el almacén de dispositivos comprados en las organizaciones y comprobar todos los programas recibidos vía comunicación satelital.

Establecer un registro de incidentes, para inspeccionar todas las fallas que pudiesen suscitares en los sistemas de información. Efectuarse encuentros regulares a fin de promover la ayuda mutua entre los administradores de sistemas de información y los centros de competencia; de esta forma, determinar las causas de errores.

1.2. Políticas y Procedimientos

Así como nuestra sociedad necesita leyes y normas de conducta para lograr cierta seguridad, cualquier organización o empresa requiere de una Política de Seguridad en Cómputo (PSC) para lograr la seguridad, organización y buen uso de sus recursos de cómputo.

Las Políticas y Procedimientos permiten un aprovechamiento más eficiente de los recursos (tanto materiales como humanos) con que cuenta una organización, ya que establecen para administradores y usuarios las acciones permitidas y las no permitidas.

Evitan la toma de decisiones inadecuada por parte del personal inadecuado, pues establecen muy claramente quién tiene la autoridad para tomarlas.

1.2.1. Procedimiento de Contratación

La probabilidad de que una persona llegue a ser deshonesta dentro de grupos pequeños, es la misma así como quien trabaja en grandes grupos.

Los procedimientos de contratación incluye, verificar antecedentes penales, referencias y recomendaciones, investigación socioeconómica de los aspirantes, entrevista previa, etapa de adoctrinamiento de las políticas de seguridad al empezar a laborar, etc.[1]

1.2.2. Confiabilidad del Personal

El mayor daño que puede sufrir un centro de cómputo es el que se hace desde dentro; ni siquiera las fortalezas o los equipos más sofisticados y costosos pueden contra la deslealtad, la deshonestidad o la negligencia. Los empleados descontentos, o los que recientemente han sido corridos de la compañía representan un riesgo mayor. Por lo tanto, la selección del personal es parte importantísima del esquema integral de seguridad.

Es conveniente que jerarquías superiores, traten de solucionar los aspectos que incomodan al personal subordinado o por lo menos estar enterados de estos.

[1] Information System Security, Guidelines for The United Nations Organizations. 1993

Existen persona que obtiene un salario más bajo que el promedio y por lo tanto dentro de esta categoría existen más ladrones que dentro de las que obtienen un salario más alto.

1.2.3. Procedimiento de Terminación

Un grupo de personas entrenado puede manejar cambios que contribuyen a solventar los problemas. Estos cambios podrían ser necesarios y tal vez desagradables para algunos empleados. El llegar a un acuerdo con el personal nos daría una más próxima solución a la perdida de la información. Si ellos ven que se les puede dar una solución a sus incomformidades, pero si no hubiese un acuerdo con ellos, se puede optar por asignarlo a otra tarea mientras mejora su actitud pero, si esto no es posible se puede llegar a despedirlas.[2]

Es necesario notificar al área de seguridad cuando un empleado deja de pertenecer a la empresa. Se deben recoger sus tarjetas codificadas de acceso, credenciales, equipo propiedad de la empresa, eliminar el acceso a los sistemas de información, tratar de conocer los motivos auténticos de la renuncia, etc.

1.2.4. Procedimiento de Transferencia

Cuando un empleado cambia de puesto dentro de la empresa deben redefinirse sus accesos a la información (se deben seguir los pasos de un despido y una contratación).

1.2.5. Entrenamiento del Personal

No olvidemos que el personal es el recurso más importante dentro de las empresas. Se deben enseñar las políticas de seguridad, los estándares, procedimientos, su responsabilidad individual, etc. Simultáneamente el entrenamiento sirve para ir motivando al personal, para que se interese más por su trabajo, para que valore la lealtad y disciplina en el trabajo, con lo que se reducirá la probabilidad de insatisfacción y el riesgo que esto representa. Como parte de las medidas de seguridad, debe entrenarse a los líderes para que sean capaces de reconocer síntomas de insatisfacción en sus subordinados. Una temprana detección y el compromiso individual son llaves estratégicas de control. Una particular atención a la larga se derivará en beneficios para el área de sistemas.

[2] Information System Security, Guidelines for The United Nations Organizations. 1993

Ahora que se ha presentado una serie de generalidades sobre las Políticas de Seguridad en Cómputo, podemos concluir lo siguiente:

Las políticas y procedimientos reducen - no eliminan - el riesgo de que una organización sufra un ataque informático (tanto interno como externo), ya que ofrecen una serie de recomendaciones a las personas que en última instancia son las responsables de proteger la información, con el propósito de que ésta sea resguardada de la mejor manera posible. Son sólo parte del esquema de seguridad que toda empresa (idealmente) debería implantar, el cuál debe complementarse con elementos tales como firewalls, herramientas de software de seguridad, revisión de las bitácoras del sistema, restricción de acceso físico al sistema, actualización constante por parte de los administradores, etc.

Las observaciones de seguridad antes mencionadas se resumen en la siguiente lista:

a) Es imprescindible el apoyo directivo para alcanzar las metas de implantación de seguridad.

b) Es mucho mejor corregir los errores antes de sufrir consecuencias significantes. Esto solo puede ser alcanzado si en el lugar existe un fuerte y debido control.

c) Fallas en el personal seleccionado y entrenado, diseño de sistemas, procedimientos y prácticas no pueden ser valorados o rectificados si los problemas no son detectados y atribuidos a un individuo o grupo definido.

d) La gente desempeña mejor su trabajo si puede ser ayudado individual y específicamente con los errores que suele cometer.

e) El no detectar los errores es deshonesto. Si la gente accidentalmente provoca un error y este no es detectado, ellos pronto verán la potencialidad para cometer más errores que operan a su propia ventaja.

1.3. Seguridad Física

El pleno conocimiento que se posea de los programas de seguridad, debiera de prevenir y proteger al centro de cómputo de intrusos, fuego e inundaciones.

Intrusos, fuego y el agua son las principales amenazas físicas. La guerra, conflictos sociales, huracanes y terremotos podrían significar amenazas físicas dependiendo el lugar donde se tenga localizado el sistema de información. Las maneras necesarias para prevenir, detectar y contrarrestar estos problemas dependen de la localización física de los datos y la sensibilidad de donde reside la información.[8]

1.3.1. Intrusos

El riesgo que sufriría un C. C. como consecuencia de daños provocados por intrusos puede ser disminuido, situando el cuarto de computadora en una área remota de centros de guerra o desobediencias civiles, no anunciándolo públicamente y que reciba mantenimiento por personal confiable. El riesgo de atentado de infiltración es más alto si se cree que el sistema contiene información importante. Los extraños frecuentemente tratarían de dañar las instalaciones si creen que, estas desempeñan una impresionable o emocionante función.

El personal disgustado esta más dispuesto a intentar dañar las instalaciones. Es fundamental percibir rápidamente al personal inconforme y dar solución a las causas que las provocan o en última instancia, reubicarlos antes de que puedan estar informados de los puntos vulnerables de el lugar.

Controles de acceso al centro de cómputo, garantizan que sólo el personal autorizado podrá ingresar al CPD (Centro de Procesamiento de Datos), con lo cual se disminuirá considerablemente el riesgo de robo, destrucción o manipulación no autorizada de equipos e información (desastres producidos por el hombre). [9]

---

[8] Information System Security, Guidelines for The United Nations Organizations. 1993

[9] Tolgo, J.W. Disaster Recovery Planning: Managing Risk and Catastrophe in Information Systems. Englewood Cliffs (New Jersey): Yourdan Press, 1989

Los controles durante los descansos y cambios de turno son de especial importancia. El medio que permite identificar al personal, puede ser a través de teclados y claves numéricas, otros realizan la identificación mediante lectores de tarjetas codificados tarjetas con cintas magnéticas, otros más bien lo hacen con una combinación de medios. Hay otros sistemas de identificación que se basan en quién es la persona y no en qué tiene la persona, como los de reconocimiento de firma, de huellas digitales, de reconocimiento de líneas de la mano, de voz, de la retina, etc., llamados sistemas biométricos. Cuando se utilicen estos sistemas automáticos para las puertas, debe existir una puerta adicional de emergencia. Las aperturas que se usen para recepción y entrega de datos deberían estar en una área separada del centro de cómputo con una división a prueba de fuego.

El acceso físico puede ser controlado por guardias de seguridad, tarjetas llave, chapas digitales, etiqueta de identificación, torniquetes, alarmas y circuito cerrado. Algún nivel de seguridad es deseable aun si este es limitado a recepcionistas quien registran visitantes: la seguridad sin la instalación dependerá de las amenazas y vulnerabilidad de lugares particulares.

1.3.2. El Fuego

Los daños que produce un incendio son generados por el fuego, el calor, los productos de la combustión, el agente extintor y tiene como consecuencia destrucción de construcciones y estructuras. Los incendios son comúnmente considerados como el principal y más temido riesgo en instalaciones de cómputo; sin embargo, estadísticamente el agua es la causa del mayor número de desastres en instalaciones de cómputo.[10]

Evitarse instalar centros de procesamiento electrónico de datos en zonas con fallas geológicas, en general son preferibles zonas suburbanas, se deben encontrar por lo menos a 60 metros de distancia del acceso público más cercano.

El cuarto de computadoras debería situarse en áreas alejadas de riesgos de fuego como, boíles, plantas pesadas, plantas de soldadura, cocinas y almacenes de combustibles e incluso de centros comerciales (por riesgo de bombas).

[10] Information System Security, Guidelines for The United Nations Organizations. 1993

Muchos incendios en sistemas de información ocurren en o cerca del papel que se usa en las impresoras.

Impresoras, guillotinas y envolturas generan electricidad estática y contribuyen al nivel de polvo en la atmósfera. Impresoras que usan deposito térmico (incluyendo las impresoras láser) incrementan el riesgo de fuego.

El nivel de polvo puede ser reducido por unidades de extracción y filtración. El riesgo de combustión puede ser reducido por un regular mantenimiento al equipo y un dispositivo adecuado de descarga estática para el equipo.

Los solventes que son usados para limpiar el equipo de computación son altamente flamables. Deberían ser almacenados en un casillero a prueba de fuego. El techo y el piso deben ser regularmente limpiados para prevenir la acumulación de polvos flamables.

Reducir la oportunidad de una chispa o flama es otro aspecto importante de previsión de fuego. El fumar o comer deberían de ser prohibidos en el lugar. Los cables deberían ser colocados sobre bases para prevenir la humedad.

1.3.3. El Agua

Entre las causas comunes de inundaciones están: fugas de tuberías de agua, aire acondicionado (fugas de agua o condensación), sistemas de enfriamiento por agua, rociadores, etc.

Lugares cerca de depósitos, reservas naturales (ríos) y almacenes de agua deberían ser evitados.

El pobre mantenimiento al sistema de plomería es una de las principales causas de inundaciones. Un mantenimiento planeado al sistema de plomería y la colocación del equipo de cómputo alejado de pipas y cisternas principales previene el riesgo.

La humedad no solo afecta a los equipos sino también a los medios magnéticos y papel por esa razón se deben instalar sistemas de detección de fugas de líquidos. Los daños que pudiera causar la humedad se limitan, si se colocan los cables dentro de una bandeja a lo largo de la instalación.

El sistema de drenaje debe ser adecuado y suficiente (no deben pasar tuberias por encima ni debajo ni a los lados directamente del C. C.), etc. También es necesario contar con cubiertas plásticas anti-inflamables para el equipo de cómputo y detectores de agua, e idealmente con bombas para evacuar rápidamente el agua. Resulta necesario contar con medidas para detectar la intrusión de agua antes de que sea necesario apagar la computadora. Existen sistemas para detectar la presencia de agua bajo de piso antes de que se vuelva peligrosa para el equipo. Además de señalar fugas con una alarma, estos sistemas deben proveer los medios para quitar automáticamente la energia a los equipos.

1.3.4. El Ambiente

La temperatura y la humedad, a la que se someten algunas máquinas tienen que estar dentro de los limites especificados por el fabricante.

Las consecuencias de una interrupción en el suministro de electricidad son proporcionales al grado de dependencia en la computadora.

El sistema de aire acondicionado controla la temperatura y humedad y puede proporcionar filtración de aire. Una falla en este sistema causan un paro inmediato de el procesamiento, ya que la temperatura aumenta rápidamente porque las computadoras, y en especial dispositivos como unidades de disco y cinta (que contienen motores), generan grandes cantidades de calor.[11]

Muchos factores influyen para que el equipo trabaje óptimamente como son: la cantidad de polvo en el aire (tiene que ser bajo). Así como también el enfriador de agua, abastecimiento de energía eléctrica, reguladores de voltaje; necesitan mantenimiento regular.

Information System Security, Guidelines for The United Nations Organizations. 1993

Resulta vital mantener limpio el centro de cómputo. También es importante para la estabilidad de la operación, que el aire esté limpio y libre de partículas. Otras medidas que se pueden poner en práctica para minimizar el impacto de contaminantes que no pueden ser totalmente eliminados son las siguientes:

- Prohibir comer, fumar y beber dentro del centro de cómputo
- Vaciar los basureros y sacar el papel de desperdicio del centro de cómputo
- Poner las impresoras fuera del cuarto donde están las CPU, unidades de cinta, etc.
- Observar medidas apropiadas de mantenimiento del piso falso, etc.
- Asegurar que todas las computadoras estén equipadas con los mejores filtros que el vendedor puede proveer.
- No instalar purificadores de aire generadores de iones y dar un buen mantenimiento al aire acondicionado.

1.3.5. Control de Medios de Almacenamiento Secundario

Los dispositivos magnéticos que hayan sido retirados de uso deben ser reiniciados: ya que aunque hayan sido borrados previamente se puede recuperar la información con procedimientos especiales o cuando hayan cumplido su vida útil deben ser destruidos.

Tener medidas de seguridad en los almacenes de dispositivos magnéticos (no deben estar en el centro de cómputo): controles de acceso, y controles ambientales y tener almacenamiento de respaldos en otro lugar p. ej. en cajas de seguridad contra incendios.

Deben revisarse todos los discos flexibles que son propiedad del usuario antes de entran a la empresa para verificar que estén libres de virus e instalarse detectores automáticos de virus (virus scan) en todas las computadoras.

Una buena planeación de desastres y personal entrenado puede reducir la frecuencia de amenazas fisicas. Esto es un gasto de tiempo y dinero sino existe un soporte de procedimientos y entrenamiento. La planeación y la práctica son la llave del éxito respondiendo a una emergencia. La localización del equipo de cómputo a distancia, lugares inéditos que no están compartidos por otras organizaciones hacen más fácil la seguridad.[12]

A manera de resumen las medidas que se debieran aplicar son:

a) Publicar folletos para asegurarnos que el personal entienda los procedimientos a seguir en caso de algún imprevisto.

b) No deben ser mostradas en áreas publicas, la exacta ubicación de las instalaciones de agua, electricidad u otros servicios.

c) El acceso a áreas delicadas debe ser restringido.

d) Debe ser establecido un plan de contingencia, previamente probado y darle mantenimiento continuo.

e) No debe de haber ningún depósito de agua instalado sobre el centro de cómputo.

f) Proveer al C.C con laminas de plástico y cableado sobre canaletas, para reducir el daño que el agua pueda ocasionar.

g) No alojar material flamable dentro del centro.

h) Prohibir comer y fumar dentro.

[12] Information System Security, Guidelines for The United Nations Organizations. 1993

i) Colocar detectores de humedad, humo.

1.4. Seguridad en el Hardware

El hardware es generalmente confiable, suponiendo que es usado en un ambiente donde se conocen las especificaciones de manufactura y donde se otorga un regular mantenimiento.

Aplicaciones importantes pueden estar protegidas automáticamente por el hardware sin embargo algunos componentes suelen fallar. Es por esta razón que el hardware obsoleto o prototipo es menos confiable y debe evitarse.

Es más seguro evitar combinaciones inusuales de hardware y software, porque puede encontrarse con problemas que aún no se puedan resolver.

Hardware necesita ser protegido de amenazas físicas, robos así como de desastres naturales.

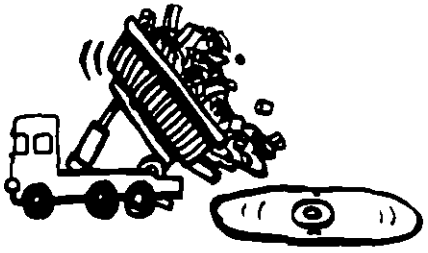
1.4.1. El Medio Ambiente

Algunas máquinas tiene exigentes requerimientos ambientales. La temperatura y humedad deben estar dentro de los limites especificados. La cantidad de polvo en el aire debe ser baja. El enfriador de agua y la fuente de poder necesitan recibir mantenimiento.[13]

El programa de seguridad deberá prever los requerimientos de especificaciones de las máquinas, interrumpible suministro de energía, reguladores de voltaje y unidades de aire acondicionado.

[13] Information System Security, Guidelines for The United Nations Organizations. 1993

1.4.2. Equipo Obsoleto y Prototipo



Adquirir lo último en tecnología, con la idea que solo esta puede satisfacer nuestros requerimientos, puede ser riesgoso. Todavía no ha sido probada y fiablemente aceptada, si llegase a fallar no se han fabricado piezas de refacción para remplazar las partes dañadas.

El equipo obsoleto es más propenso a fallar y es más difícil su mantenimiento porque se dejan de fabricar refacciones. Muchas organizaciones se rehusan a deshacerse de tal equipo, no estando inconscientes que mantenerlo es más costoso; ocupan lugar, consumen más energía, etc. Por si fuera poco el nuevo software no puede ser usado en ese equipo o se necesita adquirir nuevos periféricos para su implantación.

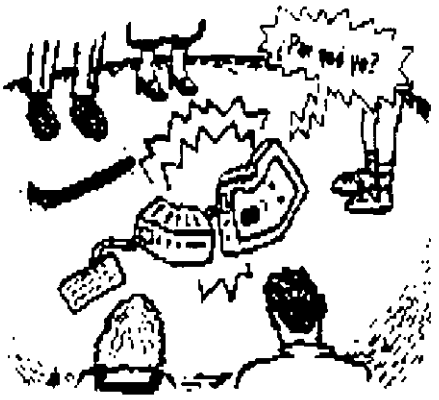
1.4.3. Riesgos de Fallas del Equipo de Cómputo

Aunque la confiabilidad de los equipos de cómputo está mejorando día con día, éstos aún fallan con cierta frecuencia. Esta categoría de riesgo incluye los siguientes peligros:

Falla en el CPU. Una interrupción en la computadora central puede ser el resultado de múltiples factores. Debido a fallas en el hardware o en el software, el sistema puede ser detenido y reinicializado (lo que puede tardar mucho tiempo).

Falla en los Dispositivos Periféricos. Pueden causar una interrupción del sistema para todos los usuarios. Estos dispositivos deben recibir el mantenimiento adecuado.

1.4.4. Tratos Físicos



En los centros de cómputo se encuentran activos de gran valor monetario, que resultan susceptibles de ser sustraídos. Más grave aún puede resultar la sustracción de información (parte de la cual tiene carácter de confidencialidad).

El hardware puede ser movido, robado, dañado o destruido. Una buena seguridad física puede reducir estos riesgos. Todos los recursos informáticos deben ser marcados e inventariados. De esta manera el usuario desistirá de mover el equipo por todo el edificio. Todos los movimientos que se hagan al equipo debe ser informado al personal de soporte de sistemas. Esta particularidad es importante cuando una estrategia de control de acceso depende en particular de máquinas que están atadas a la red.[14]

1.4.5. Mantenimiento

La cobertura otorgada por contrato de mantenimiento varía extensamente. Algunos mantenimientos no garantizan el tiempo de reparación. Tal contrato sería inútil para partes críticas del equipo. Establecer políticas donde se garantice el tiempo de reparación o la reposición del equipo.

[14] Information System Security, Guidelines for The United Nations Organizations. 1993

Las medidas de seguridad que se aplican al hardware son:

a) Solo se debe usar equipo obsoleto y/o prototipo cuando se haya realizado previamente un análisis de factibilidad.

b) Es esencial mantener la temperatura que las máquinas requieren para su buen funcionamiento, por medio de aire acondicionado.

c) Los respaldos de configuración de equipo que regularmente se realicen deben conservarse

d) Todo el equipo debe ser inventariado y asignarle un código de referencia.

1.5. Seguridad en el Software

Existen dos categorías dentro de sistemas de cómputo. El software de "Sistemas" provee servicios a aplicaciones tales como: manejo de tareas, manejo de archivos, comunicaciones, soporte de red y manejo de base de datos. El software de "Aplicaciones", hace uso de las herramientas ofrecidas por el sistema operativo, para hacer que el sistema de cómputo satisfaga los requerimientos del usuario.[15]

Ambos, sistemas y aplicaciones necesitan ser comprobados. El mejor camino para hacerlo es mantener un sistema de desarrollo, que es un espejo; imagen del ambiente de producción. El software tendría que ser comprobado en el ambiente de desarrollo y copiado al sistema "vivo" solo cuando halla pasado la revisión de control de calidad.

El sistema y software de la aplicación se encuentran expuestos a amenazas diferentes. Estas se tratan separadamente a continuación:

1.5.1. El Software de Sistema

Extrañamente los usuarios no ven al sistema operativo a menos que estén trabajando con microcomputadoras. El software de sistema es bien asentado a punto por personal especializado así como también programadores de sistemas, administradores de redes y administradores de bases de datos. Operadores de computadoras suministran día a día necesidades del sistema operativo.

El software de sistema así como la máquina en que corre, es esencial para todas las aplicaciones. Si el sistema falla entonces todas las aplicaciones también fallarán. El primer paso para asegurar la integridad del software de sistema es contar con el sistema operativo correcto. Sin embargo, un nuevo sistema operativo pudiese presentar la misma incertidumbre que sufre un hardware nuevo.

---

[15] Wong, K.K. and Watt S. Managing Information Security: a Non Technical Management Guide. Oxford: Elsevier Advanced Technology, 1990.

El sistema operativo cuenta con recursos diferentes. Debe ofrecer todas las funciones requeridas. Un acceso estratégico de control debe ser cubierto por el S. O. y hacer esfuerzos por asegurar que lo pueda soportar.

Una de las funciones de un sistema operativo es permitir que un programa corra al mismo tiempo. Debería asegurar que cada programa este aislado de otros. A menudo el software ilícito se cubre por medio de una máquina para escribir en un disco o espacio de memoria, que es ocupado por otro programa. Un buen sistema operativo debe parar este acontecimiento.

Una vez que el sistema operativo ha sido instalado por el vendedor, el personal de sistemas se encargara del mantenimiento y optimización. Poca gente en la organización tiene o necesita conocer a detalle la programación del sistema. Las consecuencias de un error provocado por un programador puede ser devastadoras, porque el error puede afectar todas las aplicaciones en el sistema.

El administrador de sistemas de información necesita poseer el suficiente conocimiento de programación para controlar los inconvenientes que pudiesen presentarse en el S. O., del mismo modo que posee conocimientos de aplicaciones de software.

El mando de control, en cooperación con el administrador del sistema operativo, son la llave para asegurar que el sistema operativo continúe funcionando óptimamente.

El personal de sistemas podría hacer cambios en el ambiente de desarrollo, que es una imagen espejo del sistema de producción. Todos los cambios podrían ser revisados por al menos un miembro del personal, con experiencia en programación de sistemas. Dadas al menos dos personas de programación de sistemas, es posible realizar revisiones al ambiente de desarrollo. Una alternativa más sería que cada miembro del personal examine el trabajo de otros antes de ser liberada la aplicación. El supervisor debe responsabilizarse por la autorización de todos los cambios al software en la producción del sistema. Una manera para detectar cambios no autorizados es almacenar una copia de cada una de las versiones autorizadas en otro medio de almacenamiento. Las versiones autorizadas pueden entonces ser comparadas periódicamente con la versión en ambiente de producción.

Dar solución a emergencias en el software de sistema plantea un problema de control. No hay nada de tiempo para comprobar la calidad de control o mando de control. La solución queda al margen de la acción tomada y seguida por un examen retrospectivo de los cambios que fueron hechos.

El personal de sistemas tiene un poderoso acceso a los programas de utilidad que ponen en servicio ellos mismos. El vendedor debiera ser capaz de proveer el nombre de las utilidades que los componen. Estas utilidades necesitaran ser usadas algunas veces pero, es mejor mantenerlas fuera del ambiente de producción. Una manera de controlar es, mantenerlas en algún dispositivo magnético y marcarlo de tal manera que el operador fundamente su uso al responsable de soporte de sistemas.

Todo uso debe ser registrado y referir una razón a detalle.

1.5.2. Aplicaciones de Software

Algunas aplicaciones son compradas como paquetes. El contrato de garantía es usualmente nulo si se efectúan cambios al código del paquete. El personal de sistemas podría realizar cambios, si es instruido por el vendedor. Sin embargo algunos cambios son necesarios y pueden ser permitidos.

Algunas aplicaciones evitan sistemas de control para mejorar su desempeño. En el peor de los casos el usuario de la aplicación puede violar el control de acceso y eludir el registro (login). Cuando se adquiera una nueva aplicación, debemos de percatarnos que no omita los principales controles de seguridad.

Algunos paquetes permiten el acceso a usuarios al sistema operativo directamente. El vendedor debería ser cuestionado si esta facilidad es soportada y como puede ser deshabilitada. Es posible que los usuarios no se les permita acceso al sistema operativo. En ocasiones los usuarios necesitan una función del sistema operativo, más bien que de el paquete. Es mejor incorporar la función dentro de un menú que dejar a los usuarios tener acceso al sistema operativo.

Si el software es escrito en "casa" entonces se requiere de un procedimiento más sofisticado. El ciclo de vida del software comprende especificación, desarrollo, comprobación, funcionamiento y mantenimiento. Toda la fase del ciclo de vida a excepción del funcionamiento debe ocurrir lejos del ambiente de producción.[16]

---

[16] Vasarhelyi, M.; Lin, T. Advanced Auditing: Fundamentals of EDP and Statistical Audit Technology. E.U.A., AddisonWesey, 1990.

A cada fase alguien debe ser responsable por controlar la calidad. Los usuarios finalmente son responsables por confirmar que el software satisfaga sus requerimientos. El analista es responsable por confirmar que los programas satisfagan la especificación. El supervisor del personal es responsable por asegurar que los programas conozcan estándares y estén documentados adecuadamente. Estos procesos deberían ser formalizados. Una serie de documentos deben amparar a un programa de aplicación desde el principio hasta el fin antes de soltar el programa al ambiente de producción puede ver que todos los requisitos a prueba han sido dados. Algún cambio durante la vida de el programa debe ser sujeto al mismo riguroso control de calidad así como la aplicación original.

Algunas medidas de seguridad del software son:

a) Una nueva aplicación siempre debe contemplar los principales controles de seguridad.

b) Usar solo versiones formales de software.

c) Debe estar instalado un ambiente de desarrollo separado y una vez allí probada la aplicación en desarrollo, está estará disponible para que sea trasladada al sistema de producción.

d) Es importante asegurarse que el software de sistema pueda apoyar completamente el conjunto de mandos de acceso requeridos.

e) Un sistema de control fuerte y con calidad debe estar establecido, para supervisar los cambios que pudiese sufrir el software.

f) Los usuarios no deberían tener acceso directo al sistema operativo.

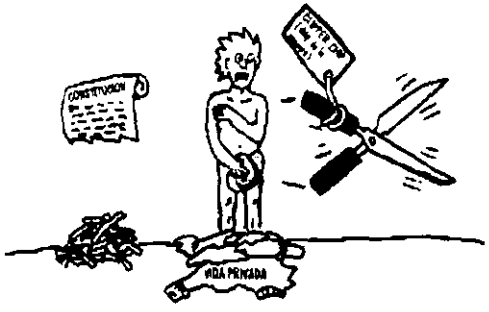
1.6. Seguridad en la Información

Los datos son lo más importante en un centro de cómputo. Los datos pueden ser el resultado acumulado de muchos años de trabajo, y si se pierden, el daño puede ser prácticamente irreparable.

Por lo tanto, es importante asegurarse de tener siempre una copia reciente de los datos importantes del sistema. De esta forma, si algo le sucede a la información en el disco, ésta puede ser recuperada de los respaldos.

Las estrategias y técnicas de realización de respaldos deben ser determinadas por cada sitio de acuerdo a sus necesidades y sus recursos.

Otro aspecto de la seguridad de los datos en un sistema es la verificación de la integridad de los archivos. Es muy común que un intruso (o un error de software, o un administrador descuidado) modifique algún archivo del sistema, y que posteriormente dicha modificación provoque un mal funcionamiento o un hueco de seguridad.



Un asunto que es interesante en el manejo de la información, es el de la privacía. Se refiere al derecho de los individuos y las organizaciones para determinar por ellos mismos cuándo, cómo y hasta qué punto se puede difundir a otros la información acerca de ellos.

Cuando se pone en riesgo la privacía de un individuo o una organización, se le da una mayor atención a la seguridad de la información.

También la falta de seguridad en alguna organización trae consecuencias financieras.

Los costos de las medidas de seguridad deben ser vistos como parte de la inversión total. en la misma forma en que la compañía compra seguros par su maquinaria. Un apropiado nivel de seguridad debe basarse en la clasificación de la información y paralelamente invertiríamos solo los requerimientos (tiempo. empeño y dinero) óptimos.

1.6.1. Crimen por Computadora

Es un término general para describir cualquier uso de sistemas computarizados y llevar a cabo actos ilegales. Se piensa que el crimen por computadora. sabotaje y espionaje son poco frecuentes. De la misma manera en que. la mayoría de los directores de las empresas tienden a pensar que "un desastre nunca me tocará a mí".

Podemos distinguir 2 tipos de crimen por computadora:

- Incidentes donde las computadoras se usan para cometer el crimen, p. ej. fraude. desfalco. etc.
- Incidentes donde las computadoras o los medios de almacenamiento de información son los objetivos. p. ej. instalación de bombas lógicas. intercepción de la información. copia no autorizada. etc.

En la figura 1.6.1. se puede percibir las posibles amenazas a la información. Es por demás mencionar que tienen consecuencias fatales.

Otro problema que se presenta es el de los saboteadores, no tratan de robar nada. En lugar de eso, tratan de invadir y dañar hardware, software o datos. Pueden ser "hackers" (aunque ellos no tratan de hacer daño, en ocasiones lo hacen por error), empleados disgustados o espías.

Amanazas Relacionadas al Crimen por Computadora

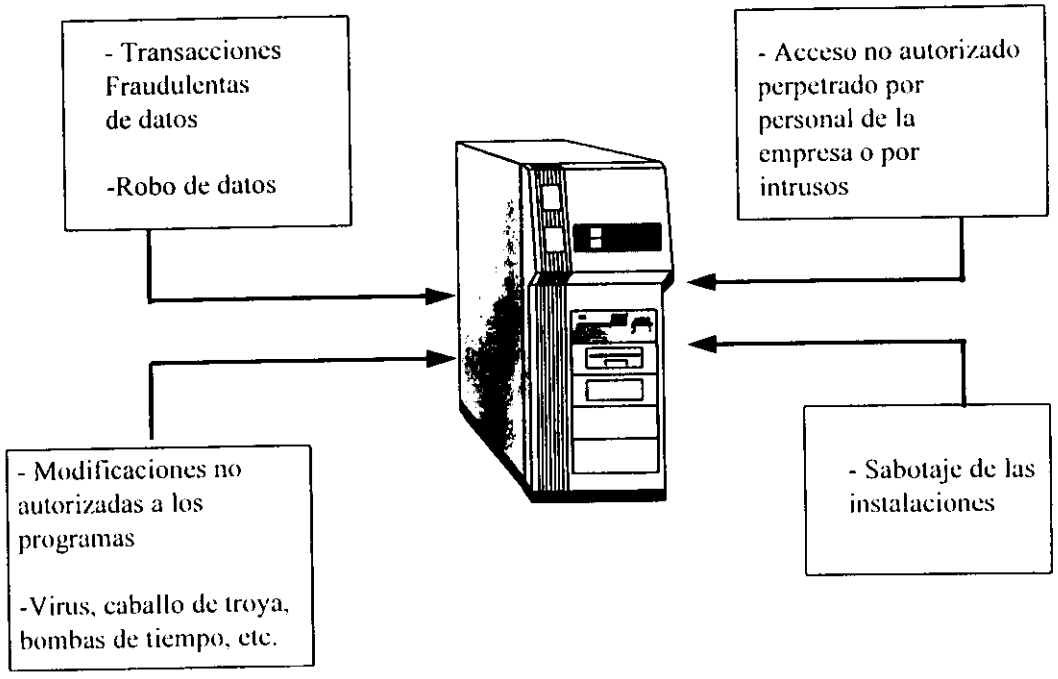


figura 1.6.1

1.6.2. Control de Acceso

Hay típicamente tres niveles de controles de acceso que deberían existir en cualquier sistema de seguridad de la información:

- Identificación de usuario.
- Password (o contraseña) o número de autenticación de usuario. Se usa para autentificar que la persona es quien dijo al identificarse.
- Mecanismo de autorización. Son los derechos a accesar diversos recursos de un sistema de cómputo y están especificados por reglas de autorización. Los mecanismos de autorización o paquetes de control de acceso usualmente contienen las siguientes características:
  - ⇒ Identificación de usuario y password.
  - ⇒ Salir del sistema si el usuario intenta un password inválido más de un número predeterminado de veces.
  - ⇒ Controles de autorización que permiten reglas de acceso a los archivos de datos.
  - ⇒ Reportes de seguridad que resaltan los intentos de acceso no autorizados.

Además el software de control de acceso debe estar diseñado para que las terminales de usuario no puedan tener acceso directo a las funciones del sistema operativo. Incluyen acceso a las librerías de programas y a las diversas tablas del sistema. Debe haber un buen mecanismo para revisar la ocurrencia de cualquier operación inusual.

La seguridad de la información en sistemas de usuario final puede mantenerse a través de la puesta en práctica de medidas como:

- Respaldo frecuente de la información y software, y excluir al centro de almacenamiento de datos, dependiendo de su sensibilidad y valor para la organización.

- Almacenamiento apropiado de medios de almacenamiento, para protegerlos contra calor o frío extremos, polvo, agua o humedad excesiva.

- Almacenamiento en sobres y físicamente alejados de dispositivos magnéticos tales como dispositivos de control de acceso, bocinas, etc.

- Etiquetar apropiadamente y fecharlos para asegurar una identificación positiva de su contenido.

La figura 1.6.2, nos señala medidas importantes que hay que considerar para prevenir accidentes en la información.

Minimizando Accidentes y Crimen por Computadora

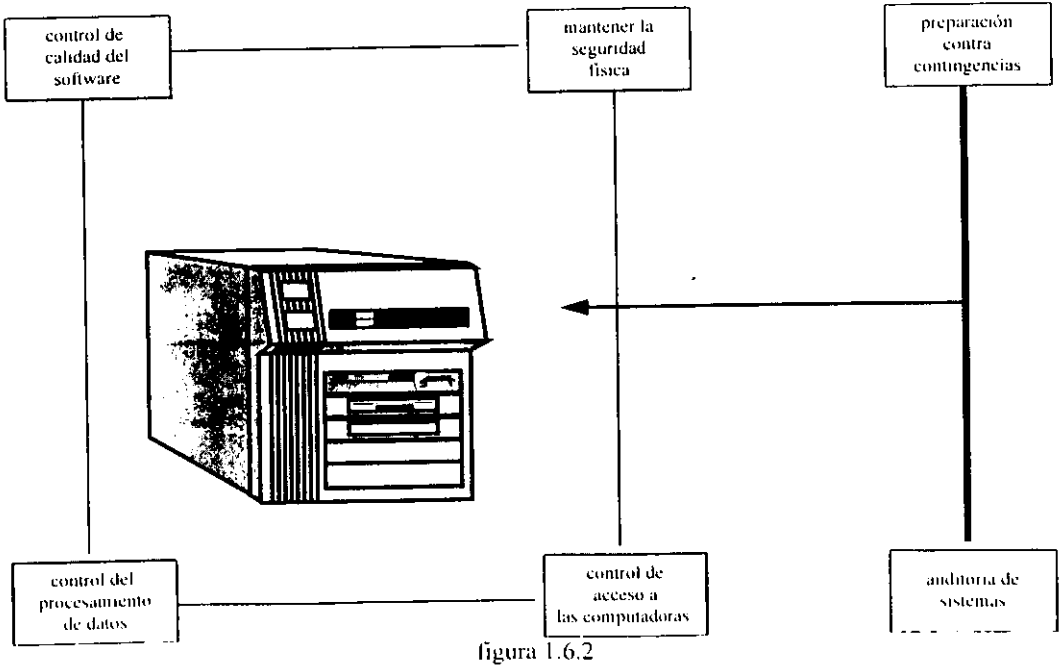


figura 1.6.2

Algunas medidas de seguridad para la información son:

a) Respaldar. con la mayor frecuencia los datos de los usuarios.

b) Respaldar las áreas del sistema dependiendo de su frecuencia de cambio.

c) Alternar respaldos completos (de toda la información) con respaldos increméntales (solamente de lo que haya cambiado desde el último respaldo).

d) Mantener al menos dos juegos de respaldos, alternando los medios magnéticos (o el medio en cuestión) utilizados.

e) Cerciorarse del ciclo de vida de los medios de almacenamiento secundario antes de ser usados. puesto que, crece la posibilidad de errores de grabación.

f) Hacer periódicamente pruebas de recuperación de los respaldos. Pocas cosas son más frustrantes que perder datos, ir al respaldo, y encontrar que el respaldo no se hizo correctamente.

g) Llevar un indice de qué archivos contiene cada volumen de respaldo.

1.7. Seguridad en Redes y Comunicaciones

Actualmente, el gran desarrollo que han tenido las redes de computadoras ha abierto a los usuarios posibilidades nunca antes imaginadas. Ahora, sin moverse de su escritorio, una persona puede tener acceso a información que está físicamente localizada del otro lado del Planeta.[17]

1.7.1. Propósito de la Seguridad en Redes y Comunicaciones

A través de los mismos servicios que nos permiten difundir y obtener la información tan fácilmente, en muchas ocasiones ha sido posible obtener acceso no autorizado a los sistemas de cómputo, permitiéndole a los intrusos en cuestión utilizar recursos a los que no deberían tener acceso, o incluso realizar actos dañinos como robar o destruir información.

Las redes de computadoras presentan formidables problemas de seguridad debido a su naturaleza multiusuaria, multirecursos y multisistemas. La seguridad en redes debe ser tan independiente como sea posible de la seguridad de los nodos separados.

1.7.2. Protección Física de los Medios de Comunicación

Deben protegerse físicamente los medios de comunicación y las "cajas" (hubs, mux's, gateways, etc.) en los lugares donde se manda o recibe información ya que es allí donde se puede interceptar más fácilmente (debido al alto grado de multiplexación utilizado en nuestras comunicaciones, es muy difícil y caro interceptar en otra parte).

En lo que respecta a redes de computadoras, lo que se debe de proteger principalmente son las conexiones de tres tipos:

---

[17] Seguridad de la información en sistemas de Cómputo. Luis Angel Rodríguez. Ventura Ediciones, S.A. de C.V. 1995

El enlace tradicional entre una unidad central de procesamiento y terminales remotas. En este grupo se incluyen ahora las conexiones entre microcomputadoras y computadoras centrales.

- La LAN, que permite a las PC compartir recursos a través de líneas de comunicación
- La red telefónica externa, incluyendo los sistemas de teléfono convencional y otros tipos de servicios públicos de comunicación.

La introducción de sistemas distribuidos y el uso de redes e instalaciones de comunicaciones para transmitir datos es el factor más reciente que afecta la seguridad en forma creciente. La seguridad de redes se necesita, principalmente para proteger los datos durante su transmisión.

1.7.3. Medidas de Prevención y Mitigación

Con seguridad en redes, los datos que han sido encriptados en una computadora, puedan ser desencriptados en otro sistema autorizado. La American National Standards Institute (ANSI) ha emitido varios estándares, que son de los más aceptados.

1.7.3.1. Encripción

Los métodos de cifrado se han dividido en dos categorías: cifradores de sustitución (incluyendo los códigos) y cifradores de transposición. En un cifrador de sustitución, cada letra o grupo de letras se sustituye por otra letra o grupo de letras para disfrazarlas. Los cifradores de sustitución y códigos preservan el orden de los símbolos del texto en claro, pero los disfrazan. Los de transposición no lo disfrazan.

La encripción es un método que oculta el significado transformando mensajes inteligibles en ininteligibles, usando un código o una cifra. Es una alternativa económicamente no muy costosa, pero no esta libre de problemas; sin embrago hay que considerar los costos antes de implementar la encripción.

Es necesario reconocer que, ésta es la herramienta automatizada más importante para seguridad en redes y comunicaciones.

La encripción se lleva a cabo de la siguiente manera: el mensaje original (texto plano) es convertido en algo aparentemente sin sentido (texto cifrado). El proceso consiste de un algoritmo y una llave. La llave es una cadena de bits relativamente corta que controla al algoritmo. El algoritmo producirá una salida diferente dependiendo de la llave usada. Al cambiar la llave, cambia la salida del algoritmo. Después que el texto es encriptado se transmite. En la recepción, el texto cifrado puede ser transformado de regreso al texto original usando un algoritmo de desencripción y la misma llave que usó para la encripción.

1.7.3.2. Manejo de Claves

- Una llave usada para encriptar otras claves se denomina la maestras y es la más importante.
- La clave de encripción de datos debe ser cambiada regularmente, cuando menos una vez al día. Por lo tanto, se requiere un mecanismo para enviar las nuevas claves de encripción de datos.
- Periódicamente, la clave maestra necesita ser enviada a los receptores en forma muy segura.

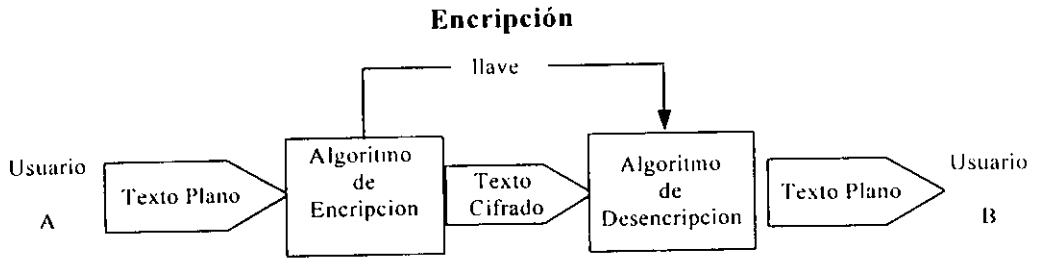
Una de las principales dificultades con la encripción convencional es la necesidad de distribuir las claves en forma segura. Un sistema de encripción que no requiera distribución de las claves sería una buena solución. Se le conoce como criptografía de clave pública.

1. Cada sistema en una red genera un par de llaves para ser usadas en la encripción y desencripción de los mensajes que recibirá.

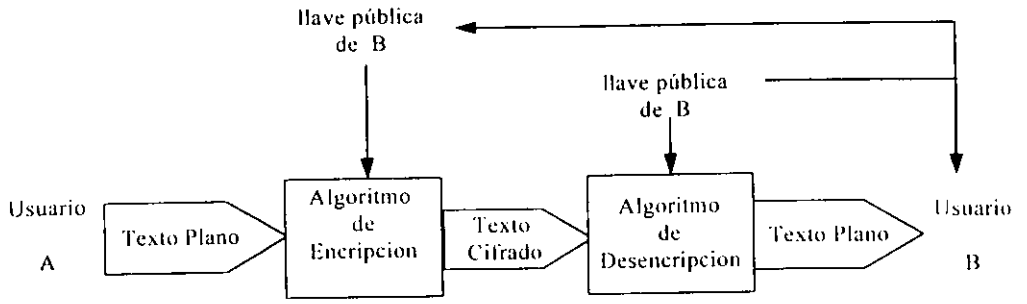
2. Cada sistema publica su llave de encripción, poniéndola en un archivo o registro público. Esta es la llave pública. La llave relacionada se mantiene privada.

3. Si A quiere enviar un mensaje a B, encripta la información a ser enviada usando la llave pública de B.

4. Cuando B recibe el mensaje, lo desencripta usando la llave pública de B. Nadie más que lo recibe puede desencriptarlo porque sólo B conoce su propia llave privada. El proceso anterior se puede observar en la figura 1.7.3.2



Encripción Convencional



Encripción de Llave Pública

figura 1.7.3.2

Como se puede observar, la encripción de llave pública resuelve el problema de la distribución de llaves. Una desventaja es que los algoritmos para la llave pública son mucho más complejos, y otra y la más importante es que, un impostor puede generar un par de llaves pública/privada y diseminar la pública como si fuera de otra persona.

Definitivamente, las ventajas que se obtiene al trabajar en red superan, por mucho, a las desventajas, pero si hacen necesario tomar precauciones especiales para asegurar que las máquinas y los servicios sean utilizados de la forma y por las personas autorizadas.

Las medidas de seguridad más importantes para las redes y comunicaciones son:

a) Solamente los usuarios autorizados deberán tener acceso a la red, controlando el acceso a los componentes de la red y a los passwords.

b) Asegurar también que realmente estamos conectados con quién creemos que estamos, de la misma manera los receptores a su vez, estar seguros de que nosotros somos quienes dijimos que éramos.

c) Asegurarse de que el mensaje permanezca inalterado durante su transmisión, reteniendo la integridad de los datos que están siendo enviados.

d) Preservar la confidencialidad de los datos que viajan a través de cualquier canal de comunicación.

e) Adicionalmente, deben existir formas de probar que un mensaje transmitido ha sido recibido exitosamente.

1.8. Auditoría y Monitoreo

La auditoria en informática, es un conjunto de exámenes y validación de los controles y procedimientos utilizados por el área de informática a fin de verificar que los objetivos de continuidad del servicio, confidencialidad y seguridad de la información así como la integridad y coherencia de la misma, se cumplan satisfactoriamente y de acuerdo a la normatividad externa e interna.

El primer punto de partida en la auditoria a centros de cómputo es contar con un plan bien fundamentado sobre los requisitos o políticas que deben de seguirse para brindar el servicio que ofrece como tal. Contemplar que todas las tareas que emprende un centro de cómputo se realiza sin atentar la seguridad de la información o que pueda llegar a paralizar su funcionalidad.

1.8.1. Auditoría a Sistemas en Desarrollo

El principal desafío en el desarrollo del software es reducir los costos e incrementar la calidad. explotando al máximo los recursos disponibles para lograr el máximo costo-beneficio.

Tipos de Control

- Participación activa y aprobación tanto de directivos como de usuarios.
- Estándares y lineamientos de desarrollo.
- Administración del proyecto.
- Controles en las pruebas y conversiones.
- Revisiones de post-implantación.

1.8.2. Auditoría a Sistemas en Operación

El propósito de los auditores dentro de los sistemas en operación es evaluar la suficiencia y cumplimiento de controles para administrar, operar y utilizar los mismos. Con el objeto de garantizar la seguridad y confiabilidad.

Controles

- Proveer información confiable.
- Promover apego a las políticas prescritas.
- Salvaguardar activos y registro.
- Asegurar el cumplimiento de objetivos organizacionales.
- Promover la eficiencia operacional.

1.8.3. Auditoría a Planes de Contingencia

Verificar la existencia de un plan que permita implementar operaciones de emergencia rápida, eficiente y efectivamente. El auditor puede realizar pruebas de recuperación para evaluar la eficiencia y probar la continuación de las operaciones del centro.

Algunos controles que deben tomarse en cuenta en el diseño y elaboración de un plan de contingencias.

1. Identificar aplicaciones críticas y tiempo que pueden retardarse sin causar problemas.
2. Listar recursos de hardware y software que puedan necesitarse para reiniciar las operaciones.
3. Contar con respaldos de hardware: hacer arreglos para tener capacidad de procesamiento alternativo en caso de que el C. C. se deshabilite.
4. Contar con respaldos de programas de las áreas de: sistema operativo, software de aplicaciones y documentación.

1.8.4. Auditoría a la Administración de Informática

La auditoria a la administración informática tiene como objetivo, verificar y evaluar los controles establecidos dentro de las áreas de informática con el fin de asegurar que la gerencia provea, planee, organicé, dirija y controle los recursos tanto humanos como materiales, de manera eficiente para cumplir los objetivos organizacionales.

- Mantener un nivel de independencia apropiado de las áreas a las que sirve y una máxima separación de funciones.
- Contar con un plan de organización que defina líneas de autoridad y responsabilidades.
- Contar con procedimientos y estándares por escrito.

1.8.5. Auditoría física a Centros de Cómputo

Evaluación de controles del lugar físico donde se lleva a cabo el procesamiento, con el objeto de hacer un mejor uso de los recursos disponibles.[18]

Controles en Centro de Cómputo

- **Controles Preventivos**
  - ⇒ Mantenimiento de instalaciones.
  - ⇒ Mantenimiento de equipo.
  - ⇒ Acceso Físico limitado.

[18] Seguridad de la información en sistemas de Cómputo. Luis Angel Rodriguez. Ventura Ediciones, S.A. de C.V. 1995

⇒ **M**antenimiento del Ambiente.

⇒ **U**bicación física.

- **C**ontroles **Detectivos**

⇒ **S**upervisión.

⇒ **R**eportes.

⇒ **D**etectores de Fuego.

⇒ **R**egistros Cronológicos.

- **C**ontroles **Correctivos**

⇒ **P**lan de recuperación.

⇒ **E**xtinguidores de fuego.

⇒ **R**espaldo fuera de la instalación.

⇒ **S**eguros.

⇒ **E**nergía continua.

Resumiendo, las medidas más importantes son:

a) **D**ebe de existir un grupo de auditoria dentro del área de informática: puede ser propio o contratado externamente.

b) Elaboración de un documento propio donde se listen los controles apropiados que se pretende auditar.

c) Contar con un plan de seguridad, que incluya un plan de contingencia al cual se pueda recurrir en caso de que se paralice o se amenace la funcionalidad del centro de cómputo.

d) El grupo de auditoria debe monitorear organizacional o técnicamente el impacto que tendria cualquier cambio en el programa de seguridad.

e) Incidentes que ataquen la seguridad de un centro de cómputo, deben ser investigados por el auditor externo, personal de soporte de sistemas, auditor interno y usuarios mismos del sistema afectado.

f) Se debe de establecer una estrategia de prueba para cada control identificado en el programa de seguridad.

g) Los sistemas actuales deben contemplar modificaciones, por lo que el software debe estar bien documentado para facilitar cualquier modificación.

h) EL sistema debe cumplir con las expectativas del usuario y no fallar más de lo esperado.

CAPÍTULO 2

SEGURIDAD EN EL SISTEMA OPERATIVO UNIX SYSTEM V

2.1. Antecedentes del Sistema Operativo UNIX System V

El Instituto Tecnológico de Masachussets (MIT), los laboratorios Bell (AT&T) y General Electric se unieron en la segunda mitad de la década de los 60's para desarrollar lo que denominaron "una computadora de servicio público", la cual soportaría miles de usuarios en tiempo compartido, tomando como modelo las redes telefónicas y de energía eléctrica. Dicho sistema, denominado MULTICS (que significa MULTiplexed Information and Computer Services, o Servicios de Información y Cómputo MULTicanalizados), es la base de las redes distribuidas de la actualidad.

Desgraciadamente, el proyecto resultó demasiado ambicioso (mounstruoso, según algunos) y se salió de presupuesto e itinerario. Hacia el fin de los sesenta, la dirección de los laboratorios Bell decidió no continuar con Multics así que retiró a sus investigadores, fue cuando uno de sus Ingenieros, Ken Thompson logró acceso a una computadora Digital PDP-7 para utilizarla como único usuario (fue sin saberlo un precursor de la computación personal). En esa computadora desarrolló un sistema operativo pequeño y simple casi como pasatiempo que incorporaba algunos conceptos importantes de Multics en una versión para un solo usuario. Uno de sus colegas, Brian Kernighan, bromeaba constantemente acerca del UNIplexed Information and Computer Service tildándolo de juguete inútil y pronto se le conoció como el "UNICS" de Thompson.

"Hacia 1970. los Laboratorios Bell requerían de un sistema de desarrollo de aplicaciones. procesamiento de textos y fotocomposición offset que fuera multiusuario y multitareas y debia implementarse en un computador PDP-11/22. Ninguno de los sistemas disponibles comercialmente satisfacía estas necesidades. entonces el UNICS de Thompson fue implementado con éxito y se demostró su alto grado de eficiencia como multitareas y multiusuario, dando lugar a un rápido reconocimiento a este trabajo. El sistema operativo fue tomado en serio y su nombre cambió a UNIX"

Dennis Ritchie se unió a Ken Thompson en 1973 y juntos decidieron escribir una nueva versión del sistema operativo en un lenguaje de alto nivel (rompiendo así con la tradición de que el software de sistemas estuviera escrito en ensamblador) que fue bautizado como lenguaje C.

"Hacia 1974 los Laboratorios Bell otorgaron licencias de UNIX a laboratorios de investigación y fue introducido en las universidades con fines "educacionales", casi gratuitamente y al cabo de pocos años estaba ya disponible para uso comercial. En ese tiempo, los sistemas UNIX prosperaron en los Laboratorios Bell y de allí se difundieron a otros laboratorios, proyectos de desarrollo de software, centros de procesamiento de palabras y a los sistemas de apoyo de operaciones en las compañías de teléfonos en Estados Unidos".[2]

Una cronología condensada de UNIX es la siguiente:

1969 Ken Thompson escribe UNIX para el PDP-7 de Laboratorios Bell.

1971 UNIX fue transportado a PDP-11 donde Dennis Richie comenzó a desarrollar el lenguaje de programación C.

1973 Thompson & Richie recodificaron UNIX en C (Versión 4)

1974 Las licencias de UNIX fueron concedidas a las Universidades. UC Barkeley liberó BSD versión 3.0.

1979 Las licencias de UNIX fueron concedidas para el desarrollo de software (versión 7)

1980 UC Berkeley liberó BSD versión 4.0.

1981 AT&T liberó el System III.

E. Ciurana Macías, "UNIX: ¿Qué es y adonde va?" PC/TIPS

[2] Brian W. Kernighan, Rob Pike "El entorno de programación UNIX"

1982 System III fue transportado a 8086, Microsoft liberó a XENIX.

1983 AT&T libera a System V.

1985 El SVID de AT&T fue establecido.

1987 AT&T libera a System V.3+

1988 UNIX y XENIX se unen en System V.3.2

1996 Procesador a 64 bits.

Gráficamente, se ilustra en la figura 2.1:

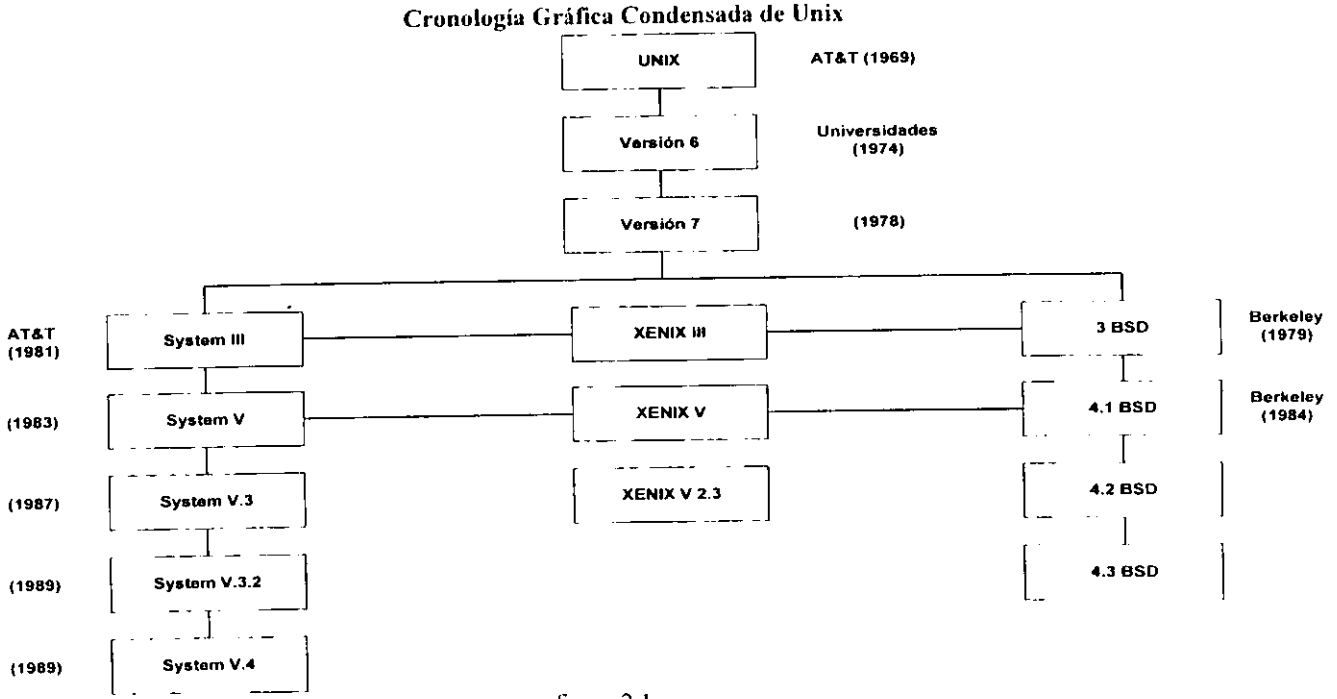


figura 2.1

Este deslizamiento muestra la evolución de las cualidades actuales de UNIX. El sistema UNIX V.3.2 es una unión de UNIX V.3 y XENIX. UNIX V.4, recientemente liberado por AT&T es una fusión de UNIX, XENIX y BSD y brinda compatibilidad binaria para aplicaciones escritas para cada versión de UNIX

2.1.1. Los componentes de UNIX

Unix tiene varios componentes, entre los más importantes están los siguientes:

Kernel

El kernel es el software de la memoria residente que es responsable por la comunicación actual con/y utilizando el hardware del sistema. El kernel es el equivalente al mismo DOS en las PC's y es considerado como el corazón del sistema. Tiene la responsabilidad de monitorear y manejar cada evento que ocurra en el mismo, además de administrar las entradas y salidas, localizar todos los programas actuales y mantener la integridad de los filesystems. El kernel trabaja "hablándole" al hardware de la máquina, cada vez que alguna petición es hecha, tal como transferir datos al disco, imprimir un archivo, utilizar un modem o cualquier otra tarea que involucre el uso de dispositivos físicos, es responsabilidad del kernel atenderlas.

Shell

El shell es una interface que trabaja entre el kernel y los usuarios. Para ejecutar los comandos, las llamadas de bajo nivel del sistema son enviadas al kernel (que solo responde a llamadas del sistema), por esta razón es que los shells fueron desarrollados, para fungir como intérpretes entre el kernel y los usuarios. Proporcionan muchos servicios tales como redirección I/O, procesos background, entubamientos y filtros, ambiente configurable y lenguaje de comandos.

Utilerías

UNIX es un sistema operativo "basado en disco", lo cual significa que las herramientas y los comandos que hacen al sistema útil son almacenados en el disco duro y accesados cuando son necesitados. Existe una gran cantidad de Utilerías para el sistema UNIX, incluyendo aquéllas referentes a las comunicaciones.

En la figura 2.1.1, se aprecia gráficamente a los componentes del sistema operativo UNIX:

Los Componentes de Unix

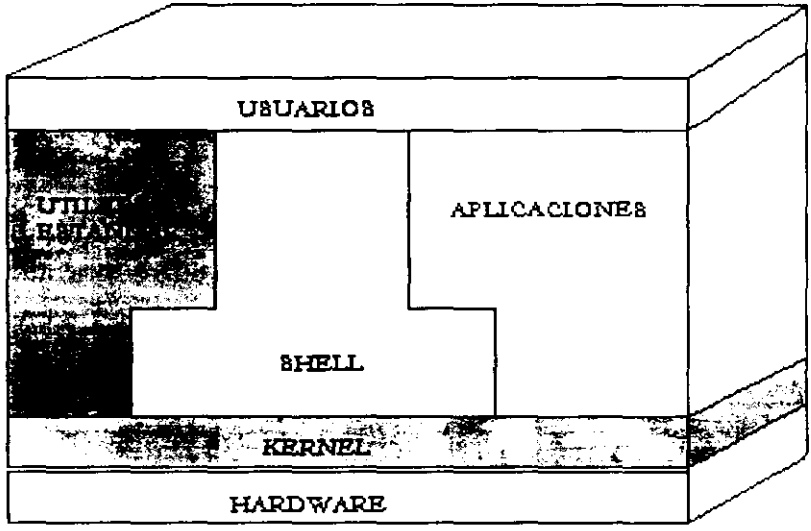


figura 2.1.1

Las Utilerías incluyen:

- Utilerías de Comunicaciones
- Utilerías de Desarrollo de software
- Utilerías de procesamiento de texto
- Utilerías de manejo de archivos

UNIX ofrece muchos intérpretes de comandos diferentes, o shells:

El Bourne shell es el estándar de UNIX y se llama así en honor a su autor, Stephen Bourne, este shell está incluido en todas las versiones de UNIX System V.

El C shell fue desarrollado por la Universidad de Berkeley, ofrece un ambiente que actúa como el lenguaje de programación C. Este intérprete incluye características útiles como alias de comandos e histórico, que permite a los comandos ser reeinvocados después de haber sido ejecutados.

El shell restringido fue desarrollado para brindar algún grado de seguridad en los filesystems. Restringe a los usuarios a un limitado conjunto de comandos y les impide el acceso a rutas específicas del sistema.

El visual shell está incluido en XENIX y representa una orientación visual para poder trabajar con él. No ha sido muy aceptado e incluso ha sido criticado por sus limitaciones. Los sistemas UNIX no incluyen este shell.

El korn shell incorpora características del C shell y el Bourne shell. No está incluido en muchas versiones de UNIX, pero puede conseguirse como un producto separado e instalarse si se desea.

2.1.2. Unix como Ambiente de Trabajo

"Unix está considerado como un sistema operativo diseñado en torno a una manipulación de archivos. Su estructura de archivo es en realidad una propiedad fundamental que ampara muchos de sus mejores aspectos, y en particular esto aparece como una de sus mejores virtudes: simplicidad de estructura y flexibilidad de uso. Como ya se mencionó, una filosofía que ha marcado una obligación en UNIX desde el principio es la de tener una herramienta de software para cada trabajo, es decir, que cada programa que forme parte de él debería ser diseñado para hacer un trabajo, y hacerlo bien, en vez de que haciendo muchos, algunos de ellos pudieran incluso no estar bien relacionados lógicamente".[3]

Como ejemplo de lo anterior, cabe mencionar que UNIX utiliza un programa para listar el contenido de un directorio, otro para listar el de un archivo y otro, incluso, para copiar el contenido de un archivo a otro, mientras que en muchos otros sistemas todas esas tareas pueden ser subtareas u opciones de un gran programa de utilidad.

Existen variaciones de UNIX y sistemas análogos, esto se dió debido a que al principio, los Laboratorios Bell, que fueron los creadores de UNIX, no pudieron comercializarlo a causa de consideraciones técnicas legales. Por lo tanto, el sistema quedó relegado al uso interno de la

[3] D. Budgen "Introducción al Sistema Operativo UNIX"

organización y de centros autorizados dentro del ámbito académico. Las evidentes ventajas de UNIX condujeron a una gran variedad de versiones del mismo y de análogos que eran similares en lo externo, pero diferían en su funcionamiento interno. Estas versiones fueron hechas por empresas de software que no estaban sometidas a las limitaciones legales que los Laboratorios Bell, y que pudieron comercializar tales sistemas con el correspondiente soporte al usuario que por lo general es requerido.

Unix System V Release 4.0 unifica importantes variantes de UNIX en uno, es un producto que conforma la definición de la industria de los estándares de sistemas abiertos. Donde los estándares han sido definidos por cuerpos abiertos de la industria, tales como el comité IEEE P1003. POSIX UNIX System V lo conforma. Cuando los estándares han tenido que ser definidos, UNIX System V incorpora estándares de facto que representan a las características más populares de los sistemas BSD 4.2 y 4.3, SunOS y XENIX.[4]

Para lograr el objetivo de la unificación y la estandarización, un extensivo rediseño de algunos aspectos del sistema UNIX fué necesario, por ejemplo, el tradicional sistema de archivos de UNIX se convirtió en uno de los muchos tipos de sistemas de archivos soportados. Esta versión brinda un alto grado de compatibilidad con previos UNIX System V y aplicaciones.

Los usuarios pueden compartir accesos expansivos, como son impresoras, memoria, almacenamiento de disco, modems y otros periféricos conectados al sistema.

Para mantener la integridad de los datos con acceso múltiple de usuarios, UNIX incluye la habilitación para "mirar" un archivo o un segmento particular de alguno. Mientras esta función es invisible a los usuarios, el sistema en ese momento pertenece sólo a un usuario en un tiempo de acceso a un solo segmento de datos a un tiempo. En la figura 2.1.2, se muestra el concepto de operación de UNIX.

[4] UNISYS UNIX System V Release 4.0 "Migration and Compatibility Guide"

Concepto de Operación de Unix

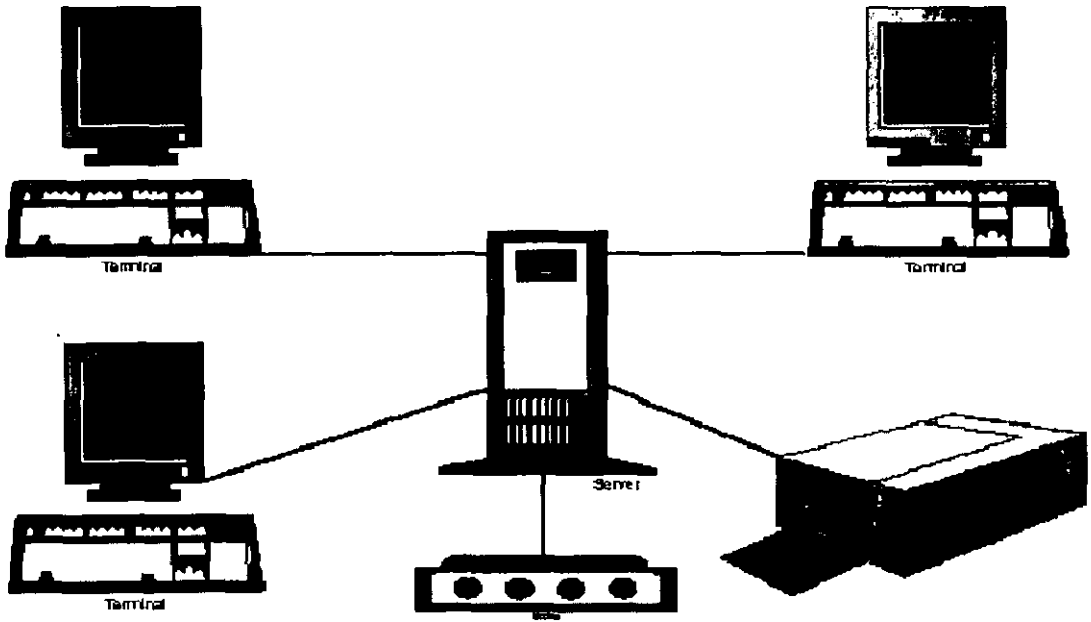


figura 2.1.2

El sistema básico de UNIX incluye correo electrónico, conexión directa usuario a usuario y red de comunicación de sistema UNIX a sistema UNIX. También están disponibles las herramientas para extender las utilerías hacia redes de comunicación heterogéneas que contienen otros sistemas operativos inluyendo MS-DOS y ambiente mainframe de IBM.

Mientras UNIX incluye muchos cientos de comandos, se utilizan de 10 a 20 comandos sobre una base regular. Nadie niega que los comandos de UNIX tienen nombres un tanto extraños, ¿quién se iba imaginar que para ver el contenido de un directorio es necesario teclear "ls" (list directory)?, aunque su nombre es muy difícil de relacionar con la actividad que realiza, su brevedad permite re-teclearlo más brevemente que a los comandos similares de otros sistemas operativos.

La filosofía UNIX es que un comando o rutina tenían que ser diseñados para completar una sola tarea. Una de las características fuertes de UNIX es la habilidad para enlazar comandos, para realizar una manipulación de datos rápida y sofisticada utilizando sólo los comandos del sistema operativo.

2.1.3. La Seguridad en Unix

Intruso

Aunque es utilizado en ambientes en los que concierne la seguridad, UNIX no fue realmente diseñado con ella en mente. Esto no significa que no brinde ningún mecanismo de seguridad, de hecho, varios muy buenos están disponibles. Sin embargo la mayoría de las instalaciones y procedimientos de compañías como Sun Microsystems, Digital Equipment Corporation y AT&T aún instalan el sistema operativo de la misma forma como fué originalmente hecho hace 15 años: con muy poca o ninguna seguridad habilitada.[5]

[5] David A. Curry "UNIX System Security A guide for Users and System Administrators"

Las razones de este estado es larga históricamente. UNIX fué originalmente diseñado por programadores para ser usado por otros programadores. El ambiente en que fue utilizado era el de abierta cooperación, no el de privacía. Los programadores colaboraban típicamente entre ellos en los proyectos y preferían estar habilitados para compartir sus archivos con otros sin tener que saltar obstáculos de seguridad. Debido a que los primeros sitios fuera de los laboratorios Bell en que se instaló UNIX eran laboratorios de investigación donde existía un ambiente similar no se vió necesidad para mayor seguridad hasta tiempo después. En los 80's muchas universidades comenzaron a mover sus sistemas UNIX fuera de los laboratorios hacia centros de cómputo, permitiendo (o forzando) su uso popular y abierto a este maravilloso sistema. Por lo tanto, no fué diseñado desde el principio para ser seguro. Fue diseñado con las características necesarias para poder darle mantenimiento a la seguridad.[6]

Muchos lugares de negocios y del gobierno comenzaron a instalar sistemas UNIX también, a la par en que las estaciones de trabajo se volvían más poderosas y accesibles. Con esto el sistema operativo no fué utilizado más en ambientes donde la colaboración abierta era el objetivo primario. Las universidades requerían que sus estudiantes utilizaran el sistema para sus asignaturas por lo que necesitaban que no pudieran copiarse entre ellos. Los negocios usan sus sistemas UNIX para tareas confidenciales tales como la contabilidad y la nómina.

Para complicar las cosas, nuevas características han sido añadidas a UNIX al paso de los años haciendo de la seguridad más difícil de controlar. Quizá las características más problemáticas son aquéllas relativas a las redes: login remoto, ejecución remota de comandos, transferencia de archivos, sistemas de archivos de Red (NFS) y el correo electrónico. Todas estas características han incrementado la utilidad y usabilidad de UNIX en cantidades inmencionables. De cualquier forma, estas mismas características junto con la conexión de sistemas UNIX a INTERNET y otras redes han abierto muchas áreas de vulnerabilidad para un uso no autorizado del sistema.

Aún si una cuenta está protegida por un password fuerte existen maneras en que un intruso gane acceso a la misma. Es importante proteger su cuenta de estos ataques y puede ser logrado observando varias reglas sencillas.

El shell utiliza una ruta de búsqueda para determinar en que directorios buscar los comandos cuando estos son ejecutados. En el bourne shell y el Korn shell la ruta de búsqueda es insertada usando comandos tales como:

---

[6] Diego Martín Zamboni "Soluciones Avanzadas"

PATH=SHOME/bin:/usr/local/bin:/usr/bin:bin
export PATH

Los directorios son listados en el orden en que están para ser buscados, separados por dos puntos (:). Dos puntos juntos(::) indican el directorio actual. Muchos de los usuarios colocan el directorio actual ".", en algún lugar en su ruta también, usualmente al frente. Esto les permite desarrollar sus propios programas y ejecutarlos fácilmente aún si tienen el mismo nombre que un comando de sistema. Por ejemplo, considere un programa llamado **ls** que hace algo insidioso que el atacante quiere que haga. Si el coloca este programa en algún directorio donde su víctima parece estar trabajando, y la víctima tiene "." en el frente de su ruta de búsqueda, la primera vez que digite ls el programa del atacante será ejecutado antes que el programa del sistema con el mismo nombre.

Para defenderse a sí mismo contra este tipo de ataque, solo debe colocar el directorio actual "." al final de la ruta de búsqueda. De este modo, cuando usted ejecute lo que cree que es un comando del sistema, siempre obtendrá la versión del sistema del comando antes que cualquier versión en el directorio actual.

Muchos programas tales como los shells, editores, lectores de correo y más tienen archivos de inicio asociados con ellos. Estos archivos típicamente residen en el directorio hogar del usuario, y contienen comandos que son ejecutados cada vez que el programa es invocado. Algunos archivos de inicio comunes incluyen **.profile, .cshrc, .login, .mailrc, .exrc** y **.emacs**. Estos archivos son fácil presa para un atacante, debido a que una vez que han sido inicializados rara vez vuelven a ser examinados. Si un atacante puede editar uno de estos archivos y añade comandos a ellos, fácilmente puede tomar una cuenta, modificar sus archivos, etc. Para protegerse contra esto, asegúrese de que todos sus archivos de inicio tienen permisos de escritura solo para usted.

2.1.4. Monitoreo de la Seguridad en los Accesos

Para monitorear la seguridad de las cuentas el administrador debe periódicamente checar dos cosas: usuarios que hayan entrado cuando no deben (tarde en la noche, cuando están de vacaciones, etc) y los usuarios que ejecutan comandos que normalmente no se espera que utilicen (secretarias compilando programas, etc). Ambos hechos, si son detectados pueden indicar que la seguridad del sistema se ha visto comprometida.

El acceso como superusuario es sumamente importante, ya que la entrada al sistema como root debe ser totalmente suprimida. Esto se logra editando el archivo **/etc/default/login** y en la línea que aparece como **ROOTLOGIN=YES**, cambiar este valor por **NO**. Así mismo, los intentos de acceso pueden ser visualizados en consola, si en el mismo archivo se agrega la línea **CONSOLELOG=YES**. De esta forma, cualquier intento frustrado de entrada será desplegado por el script del sistema en pantalla, conteniendo información tal como: cuenta, terminal y/o dirección IP del equipo remoto desde el cual se intentó el acceso.

La mayoría de los UNIX modernos registran la última vez que cada usuario accesó al sistema, usualmente en el archivo **/usr/adm/lastlog**. Esta ocasión es impresa como parte del proceso de login, los usuarios deben ser entrenados para que cuidadosamente examinen esta línea cada vez que ingresen a su cuenta, para reportar entradas inusuales desde equipos remotos desconocidos al Administrador del Sistema. Esta es una manera sencilla de detectar cuentas que han sido comprometidas, debido a que cada usuario debe recordar la última vez que utilizó su cuenta.

El archivo **/etc/utmp** es utilizado para registrar quien está actualmente accesando al sistema. Este archivo puede ser desplegado utilizando el comando **who**.

Para cada usuario, el nombre login, la terminal utilizada y el tiempo de entrada son desplegados. Si el usuario ha entrado remotamente a través de la red de datos, el nombre del equipo remoto del cual procede también es desplegado.

En algunos sistemas, este archivo tiene permisos para todos los usuarios. Esto puede ser un problema de seguridad en muchas formas. Primero, permite a un atacante borrar la entrada que lo muestra dentro del sistema, esto es para "esconderse". Un problema mayor es que el atacante puede cambiar los nombres de los dispositivos de las terminales en los archivos por lo que la próxima vez que un programa como **wall** o **comsat** sea ejecutado, escribirá a algún otro archivo de dispositivo.

El archivo **/usr/adm/wtmp** (etc/wtmp en algunos sistemas) registra cada entrada y salida de los usuarios, tiene el mismo formato que el anterior y puede ser también desplegado con el comando who **/etc/wtmp.**

Una línea que contiene un nombre de cuenta indica la hora en que entró el usuario, una línea sin nombre indica el tiempo en que fué abandonada la terminal. El comando **last** muestra las entradas en el archivo **wtmp**, coincidiendo los nombres de las cuentas con sus tiempos de entrada y salida. Sin argumentos **last** despliega el archivo entero, con un nombre usuario o una terminal como argumento, la salida puede ser restringida al usuario o a la terminal en cuestión. Este comando siempre despliega su salida en orden inverso de la más reciente entrada a la menos reciente:

Por cada sesión login, el nombre de la cuenta, la terminal utilizada, el equipo remoto (si el usuario entró vía red), horas de entrada y salida y la duración de la sesión son mostradas. Adicionalmente, los tiempos de todas las bajas e inicios de sistema (generados por los comandos **shutdown** y **reboot**) son registrados.

En lo concerniente a la ejecución de comandos, ésta se registra en el archivo /usr/adm/acct (o usr.adm/pacct en algunos sistemas) además de quién los ejecutó, cuándo y cuánto tiempo le tomó hacerlo. Bajo UNIX System V existe un comando llamado lastcom, el cual acepta varias opciones, algunas de las más útiles -l línea para restringir información a comandos que corren en una terminal dada, -u usuario para restringir la salida a los comandos que corren bajo un usuario, y -g group para restringir la salida a comandos que corren por miembros de un grupo.

Todos estos conceptos serán utilizados para la implantación de la seguridad del sistema operativo desde línea de comandos en el capítulo 5.

2.2. Cuentas y Passwords de Acceso

En un sistema UNIX, cada usuario es identificado por una entidad conocida comúnmente como "cuenta" del usuario y que está formada por dos elementos: (1) el nombre y número (identificador) de la cuenta, mediante la cual se conoce públicamente al usuario, tanto por parte de la máquina como por parte de otros usuarios y (2) el password o contraseña del usuario, que sirve para que éste, cada vez que quiera hacer uso del sistema, le demuestre a la máquina que se trata efectivamente de él y no de alguien que está queriendo tomar su lugar.[7]

"La contraseña (password) es posiblemente el elemento individual más importante en la seguridad de un Sistema UNIX, normalmente, los ataques comienzan obteniendo una cuenta en alguna máquina. A partir de ese punto, es mucho más fácil obtener cualquier tipo de acceso a los recursos. Si la protección es adecuada, se elimina en un gran porcentaje (80 o 90%) de los posibles ataques".[8]

Como ya se ha señalado, cada usuario del sistema UNIX debe tener un nombre login o cuenta (identificador de usuario) y un password (contraseña). Su nombre y password son asignados por el Administrador del Sistema.

Cuando usted teclea su nombre login, debe éste aparecer en la pantalla tal como lo dígito, de modo contrario, su password NO debe aparecer desplegado en pantalla. Esta es una característica de UNIX que lo ayuda a que otros no aprendan su password y tengan acceso a sus archivos.

NOTA: UNIX es un caso sensitivo, lo que quiere decir que "mipassword" no es lo mismo que "Mipassword".

Respecto a la duración del password, existe el envejecimiento y expiración de contraseña, la cual tiene un máximo tiempo de vida después del cual la misma expira y debe ser cambiada. Usualmente el usuario es obligado a cambiar su password expirado al próximo login. Muchos sistemas también implementan un mínimo tiempo de vida lo cual previene que los usuarios cambien su password cuando expire e inmediatamente cambien al viejo valor. En la versión del password del System V Release 4, los passwords tienen un máximo y un mínimo tiempo de vida asociado a ellos. El valor máximo especifica el número máximo de días que un password puede ser usado antes de que deba

[7] Diego Martín Zamboni "Soluciones Avanzadas"
[8] Diego Martín Zamboni "Soluciones Avanzadas"

ser cambiado. El valor mínimo especifica el número mínimo de días que deben pasar antes de que el password pueda ser cambiado otra vez.[9]

Existe también un tercer número que especifica el número de días antes de la expiración del password en que el usuario debe ser advertido de la próxima expiración.

Los valores para el mínimo y máximo tiempo de vida son insertados como nulos por default bajo System V Release 4. Pero pueden ser modificados a cualquier valor deseado editando los valores MINWEEKS y MAXWEEKS en el archivo **/etc/defautl/passwd**. Utilizando el comando **passwd**, el superusuario puede insertar diferentes valores para usuarios individuales. Un usuario puede ser forzado a cambiar su password (simplemente haciendo que expire) o prevenirlo a hacerlo.

El archivo password estándar de UNIX contiene el password encriptado de cada uno de los usuarios. Este archivo debe estar habilitado para lectura y escritura para varios comandos no privilegiados (como ls) para trabajar. Desafortunadamente, esto significa que cualquier intruso que haya ganado el acceso a su sistema puede copiar el archivo password de su máquina e intentar quebrantar los passwords. Lo que comenzó como un simple quebranto de una cuenta puede rápidamente volverse severo debido a que solo algunos comandos (**login, passwd su**) necesitan ver el password encriptado y todos esos comandos son privilegiados, a los usuarios no privilegiados no se les debe permitir obtener los passwords encriptados de otros usuarios.

El sistema UNIX System V Release 4 ha resuelto este problema con un mecanismo llamado **archivo password sombra**. El archivo password estándar es modificado para que los passwords encriptados no estén almacenados allí. Esto elimina la necesidad de modificar o conceder privilegios extras a programas como **ls**, que necesita el archivo **password** pero no el valor encriptado del mismo. Un segundo archivo, el sombra, es creado para almacenar los passwords encriptados de todos los usuarios. De esta manera, si un atacante copia el archivo password estándar no recibe los passwords encriptados y por lo tanto no puede intentar quebrantar las cuentas de la máquina remota. En System V Release 4, el archivo sombra está almacenado en la ruta **/etc/shadow**.

2.2.1. El Superusuario: root

Dentro de UNIX, todos los usuarios se consideran iguales, con una excepción notable.

[9] David A. Curry "UNIX System Security A guide for Users and System Administrators"

De vez en cuando es necesario que el Administrador del Sistema tenga privilegios especiales, para ese fin, UNIX respalda un identificador de usuario especial denominado **root**.

El superusuario, **root**, es el más poderoso usuario en el sistema. Esta cuenta tiene el poder de dar de baja el sistema, terminar cualquier proceso, crear nuevas cuentas, cambiar el password a cualquier cuenta, y puede leer escribir o borrar cualquier archivo en el sistema entero sin importar sus permisos. Esto quiere decir que esta cuenta debe estar mucho más protegida que las otras cuentas en el sistema. Un solo acceso a la cuenta del superusuario por un intruso puede dejar consecuencias permanentes debido a que puede no ser posible determinar cualquier cambio hecho por el atacante. Si alguna vez recibe un mensaje del superusuario procure ser amable: está hablando con alguien que puede sacarlo del sistema con un solo comando.

Existen algunas reglas comunes para ser usadas cuando se opera como superusuario:

- Nunca coloque el directorio actual (".") en la ruta de búsqueda de **root**. Los programas en el directorio actual pueden ser ejecutados usando la sintaxis ./comando. Asimismo, no coloque los directorios bin privados de los usuarios en la ruta de búsqueda.

- Cuando utilice el comando **su** para convertirse en superusuario, ejecute siempre **/bin/su** en lugar de solo **su**, para evitar un caballo de Troya que pudiera robar el password del superusuario.

- Nunca corra un programa de otro usuario como **root**, podría ser un caballo de Troya.

- Nunca deje el shell del superusuario en su terminal o estación de trabajo sin atender, ni siquiera "por un minuto".

- Cambie el **password** de **root** frecuentemente y sea muy cuidadoso acerca de su selección.

- No le de el **password** de **root** a alguien en quien no confíe para tener acceso a su sistema. Y jamás se lo proporcione a alguien ya sea que confíe en él o no, que no haya demostrado necesitarlo.

- No permita a nadie correr como superusuario, aunque sea por cinco minutos, aún cuando esté mirando sobre su hombro.

- Antes de entrar como **root**, ingrese al sistema como usted mismo y después invoque al superusuario. Esto permite que se lleve un registro de quien se encuentra en la consola o en el sistema accesando a **root** y cuando.

2.2.2. Elección Adecuada de Contraseñas

Como ya se mencionó, si un atacante puede descubrir el password de un usuario puede entonces entrar al sistema y operar con todas las capacidades de la cuenta invadida. Si el password obtenido es el de superusuario, el problema es más serio: el atacante tendrá libre reinado sobre el sistema, con acceso de lectura, escritura y ejecución de cada archivo almacenado en él. Por esta razón "escoger un password seguro es extremadamente importante. Desgraciadamente, esto se dice mucho más fácil de lo que se hace. Normalmente se le permite a los usuarios elegir sus contraseñas y esto ocasiona que muchos de ellos utilicen sistemáticamente contraseñas débiles".[10]

El típico programa **passwd** de UNIX coloca algunas restricciones en la manera en que debe ser usado. Generalmente, requiere que los passwords contengan 5 letras minúsculas, o cuatro caracteres si no utiliza mayúsculas o caracteres no alfabéticos. En el sistema operativo System V Release 4 (AT&T) el **passwd** ha sido modificado para ser de alguna manera más seguro. Los passwords deben tener al menos seis caracteres con al menos dos letras y un dígito o carácter de puntuación. No deben ser iguales al nombre login o alguna versión abreviada del mismo y un nuevo password debe diferir de un password viejo por al menos tres caracteres.

El objetivo al seleccionar un password es hacerlo lo más difícil posible para un atacante el lograr adivinarlo. Esto no le deja otra alternativa que forzar una búsqueda, intentando cualquier posible combinación de letras, números y puntuación. Una búsqueda de este tamaño, aún conducida en una máquina que puede intentar un millón de passwords por segundo (la mayoría de las máquinas que son estaciones de trabajo pueden intentar mil por segundo o menos), requeriría alrededor de cien años completarla.[11]

---

[10] Diego Martín Zamboni "Soluciones Avanzadas"
David A. Curry "UNIX System Security A guide for Users and System Administrators"

Con esto como nuestro objetivo, y utilizando la información anterior, un conjunto de pasos para seleccionar un password puede ser:

- No usar su nombre login de ninguna forma, ya sea al revés, una variación del mismo o doble.
- No utilice su nombre o apellidos de ninguna forma. Tampoco utilice apodos que tenga.
- No use una palabra contenida en diccionarios en cualquier lenguaje ni en listas de palabras.
- No utilice información personal fácilmente obtenible de usted. Esto incluye su número de licencia, número telefónico, número de seguro social, R.F.C., la placa de su automóvil, la calle donde vive, etc.
- No use un password de todos dígitos o la misma letra.
- No utilice un password menor de siete caracteres.
- Utilice un password con letras mayúsculas y minúsculas.
- Utilice un password con caracteres no alfabéticos como dígitos o puntuaciones.
- Utilice un password fácil de recordar para que no tenga que escribirlo.
- Utilice un password que pueda teclear rápidamente, sin tener que ver el teclado. Esto hace más difícil que alguien pueda robar su password viendo encima de su hombro.

El sistema UNIX no asigna usualmente un usuario a una terminal en particular. Por lo tanto muchos usuarios pueden entrar al sistema desde cualquier terminal. Recuerde asignar apropiadamente el tipo de terminal.

El signo "$" es el prompt estándar en el sistema UNIX. Este prompt es el indicador que le deja saber que el sistema está listo para aceptar comandos o entradas. El Administrador de su Sistema puede modificar el prompt que haya en su propio sistema.

El símbolo del indicador significa que UNIX está listo para aceptar sus comandos.

Los comandos son indicados después del símbolo del indicador.

2.2.3. Cuentas de Invitados

Pueden existir en el sistema "**cuentas de invitados**", muchas versiones de UNIX, particularmente aquéllas basadas en System V, brindan un shell restringido llamado **rsh** (puede ser invocado también como **sh -r**). Este shell es útil para una cuenta invitado debido a que le prohibe al usuario accesar partes del sistema y ejecutar acciones tales como:

- Cambiar de directorio hogar (**cd**)
- Insertar el valor de $PATH
- Especificar un comando o una ruta que contenga a la raíz (/)
- Redireccionar la salida de un comando o programa (" > " y " >> ")

Estas restricciones están reforzadas después de la interpretación del archivo .profile. Su propósito es impedir al usuario la examinación del contenido de otros directorios (utilizando **cd** o rutas que contengan a la raiz), ejecutar cualquier programa no aprobado por el Administrador del Sistema (cambiando la variable **$PATH** o usar rutas con raiz) cambiar el contenido de los archivos (redireccionando la salida).

También existen cuentas de los vendedores, muchos de los procedimientos de instalación generan varias cuentas en el archivo password. Este incluye cuentas tales **como deamon, sys, bin, uucp, news** e **ingres** así como otras. Un procedimiento apropiado de instalación deberá crear estas cuentas con un asterisco en el campo del password, para prevenir que nadie intente entrar con ellas. De cualquier forma, muchos procedimientos de instalación de los vendedores instalan estas cuentas sin password, haciéndole posible a cualquiera entrar al sistema.

2.3. Grupos de Trabajo

En UNIX es conveniente crear a los usuarios bajo una estructura organizada de trabajo que esté regida por el tipo de funciones de los usuarios que la componen. Si existen usuarios que accesan alguna aplicación llamada RFC, por ejemplo, es conveniente crear primero un grupo llamado rfc y después crear las cuentas de los usuarios como miembros de ese grupo.

El trabajo de manejar las cuentas de usuarios individuales y grupos de usuarios que trabajan en la computadora consiste principalmente de lo siguiente:

La más importante de las responsabilidades del manejo de cuentas es la seguridad del sistema y el controlar el acceso a éste. La seguridad puede ser implementada en otro nivel una vez que los usuarios han entrado, utilizando el concepto de "membresía de grupo" para controlar el acceso a ciertos archivos y directorios. Cada archivo y directorio es miembro de un grupo (identificado con un código de permiso). Estableciendo y manteniendo asignaciones de grupos de usuarios, se controla el acceso de los mismos a la información del sistema.[12]

El código de permiso es una cadena de diez caracteres que muestra quien puede accesar los datos en cuestión. Esto puede ser observado como parte de un listado de un directorio o archivo corriendo el comando **ls** con la opción **-l** como se muestra en el siguiente ejemplo:

```
dtil_cynthia#ls -l
total 116
-rw-------   1 cynthia  adm          642 May  3  1996 IOMERROR.console
-rwx------   1 cynthia  adm           64 Apr 23  1996 chacl
-rwx------   1 cynthia  adm            5 Apr 26  1996 hola
-rw-rw----   1 cynthia  adm          307 Apr 22  1996 mbox
-rw-rw----   1 cynthia  adm            7 Apr 25  1996 p
-rwx------   1 cynthia  adm           19 Apr 18  1996 prueba
-rw-r-----   1 root     other        706 Dec  5 14:03 sql.regocc
drwxr-xr-x   3 204      adm           96 Nov 26 14:20 verifica
-rw-r-----   1 root     other      51712 Dec  5 14:03 versql.noreste
drwx------   2 cynthia  adm           96 Apr 26  1996 yo
dtil_cynthia#
```

[12] UNISYS UNIX System V Release 4.0 "System Administrator's Guide"

El primer carácter en el listado muestra el tipo de objeto de datos bajo consideración. Los objetos datos incluyen directorios, archivos ordinarios y ligas simbólicas.

Este carácter es seguido de cadenas de tres caracteres que conceden o niegan los permisos de acceso al propietario del archivo, al grupo al que pertenece y a los demás miembros del sistema, en ese orden.

Los caracteres en el código definen los siguientes tipos de permisos de acceso:

r Permiso concedido de lectura (para examinar pero no cambiar) los datos.

w Permiso concedido para escribir (cambiar) los datos.

x Permiso concedido para ejecutar el comando o shell en el archivo.

Si el permiso es negado, un guión (-) aparece en lugar de la letra apropiada en la cadena.

2.3.1. Comandos para la Creación de los Grupos de Trabajo

Por lo anterior, la manera más adecuada de permitir que los usuarios compartan información, es colocarlos en un grupo. Esto es realizado editando el archivo **/etc/group** y crear un grupo nuevo con los usuarios que desean o deben colaborar como miembros. Una línea del archivo group se ve de la siguiente forma:

grupo:password:groupid:usuario1,usuario2,usuario3...usuarion

El **grupo** es el nombre asignado al grupo, muy parecido a un nombre login. Puede ser igual al nombre login de alguien o diferente. La longitud máxima de un nombre de grupo es de 8 caracteres. El campo **password** no es usual en versiones de UNIX derivadas de Berkeley (versiones sin el comando **newgrp**) y deben contener un asterisco en este caso. El **groupid** es un número desde 0 hasta 65535 inclusive. Generalmente los números abajo de 10 o de 100 están reservados para propósitos especiales, pero puede escoger cualquier número sin usar. El último campo es una lista separada por comas (no espacios) de los nombres login de los usuarios del grupo. Si no están listados nombres login, entonces el grupo no tiene miembros lo cual es comúnmente hecho cuando el grupo es usado primariamente por propósitos de propiedad de archivos).

Para crear un grupo llamado trabajo con lestat, louis y claudia como miembros debe añadir una línea como la siguiente en el archivo **group**:

trabajo:*:123:lestat,louis,claudia

Después que el grupo ha sido creado, los archivos y directorios que los miembros quieren compartir pueden ser cambiados para que sean propiedad de este grupo, los bits de permisos en estos archivos y directorios pueden ser insertados para permitir la compartición.

Además de editar el archivo **/etc/group** para crear un grupo de trabajo, existen algunos comandos disponibles en el sistema para llevar a cabo esta tarea: **mkgrp** y **addgrp**, y otros para removerlos: **delgrp** y **rmgrp**. Utilizando estos comandos El usuario que crea el grupo es considerado el propietario del mismo y puede borrar o añadir miembros al mismo.

La utilización de estos comandos está ilustrado en la siguiente tabla

Comandos para la Creación de Grupos de Trabajo

Comandos y Resultados	**Acciones**
$ id uid=203(user1) gid=247(grp1)	**¿Quién soy yo?**
$ mkgrp migrupo /etc/group migrupo:427:usuario1,usuario2,usuario3	**Hacer un grupo con el usuario1 como propietario, usuario2 y usuario3 como miembros.**
$addgrp migrupo usuario4	**Añadir usuario4 a migrupo**
$grep migrupo /etc/grp migrupo:427:usuario1,usuario2,usuario3,usuario4	**Vemos que tenemos**
$ delgrp migrupo usuario1	**Borrar al usuario1 de migrupo**
grep migrupo /etc/grp migrupo:427:usuario2,usuario3,usuario4	**Vemos que tenemos**
$ chgrp migrupo archivo2	**Cambiar de grupo al archivo2 para migrupo**
$ chmod 660 archivo2	**Volvemos leible y escribible a archivo2 para el propietario y para los miembros del grupo.**
$ rmgrp migrupo	**Cuando ya no es necesario, se marca a migrupo para removerlo.**
grep migrupo /etc/group migrupo:<removed>:427:usuario2,usuario3,usuario4	**Se observa al grupo marcado para ser removido**

tabla 2.3.1

El comando **group.cleanup**, que solo puede ser ejecutado por el administrador del sistema, borra los grupos marcados con <removed> del sistema una vez que no existan archivos asociados con el grupo.

2.4. El Sistema de Seguridad de UNIX: Audit Trail

Seguridad Comercial (Comercial Secure CS) es una versión mejorada del sistema operativo UNIX System V. La TBC (Trusted Computing Base) es la colección de hardware, firmware, archivos y programas que son críticos para reforzar los atributos necesarios de seguridad. Preserva colectivamente la seguridad del sistema y asegura el intercambio adecuado de información entre objetos y temas del sistema. El control de acceso discreto protege al TCB de cambios producidos por usuarios no administradores.

El kernel entero de seguridad comercial está incluido en el TCB junto con varios programas de las capas de aplicación del sistema que corren con capacidades administrativas.

Existen algunas definiciones de términos usados frecuentemente discutiendo el control de accesos:

access Una acción tomada por un sujeto en un objeto resulta en la transferencia de información, efectúa un cambio en el estado del objeto (por ejemplo, la creación o el borrado), u ocasiona la creación de sujetos (tales como la ejecución)

object Es una entidad pasiva que contiene o recibe información. Algunos ejemplos de objetos en UNIX son archivos, directorios, inodos, procesos, impresoras, terminales, memoria compartida, estructuras IPC y puertos I/O.

subject Es una entidad activa que ocasiona que la información fluya entre objetos, o cambia el estado del sistema o de un objeto. Algunos ejemplos de sujetos en UNIX son los usuarios y los procesos. Los procesos son sujetos primarios gobernados por el control de acceso

La Seguridad Comercial del sistema operativo audita eventos que son relevantes a la seguridad del sistema, tales como los intentos de entrada, y mantiene una auditación de todos los accesos a los datos en el sistema. El Administrador puede extraer información del **audit trail** para averiguar que archivos han sido accesados o quién a accesado un archivo en particular, puede determinar los eventos

que van a ser registrados, tales como acceso a archivos, ejecución de programas, intentos de acceso al sistema fallidos y cambio de privilegios.[13]

Para preservar un estado seguro de su sistema, el Administrador de la Seguridad del Sistema debe asegurarse que los usuarios operen con sus permisos autorizados en la máquina y debe desalentar cualquier actividad maliciosa. Más específicamente, el Administrador de la Seguridad debe estar habilitado para:

- Cambiar la protección discreta por omisión.
- Configurar los canales del audit trail
- Visualizar los datos del audit trail.
- Recuperar el audit trail de un dump
- Asegurar la integridad de el TCB
- Prevenir compromisos de la seguridad.

La protección discreta por omisión de todo el sistema se inserta en el archivo **/etc/profile**. El Administrador de la Seguridad puede cambiar esto modificando el valor de **umask**. Cuando es liberada, la Seguridad Comercial tiene un **umask** de 022, que transfiere una protección default de lectura, escritura y ejecución para todos los usuarios no privilegiados. Cada usuario recibe el **umask**, a menos que lo cambien. Cambiarlo a un valor menor de seguridad no es recomendable, si se hace al valor 077 que proporciona protección de lectura, escritura y ejecución para los objetos del usuario es altamente recomendable.

El **audit trail** es un sistema que registra todos los eventos de seguridad relevantes, está diseñado para detectar cualquier acción sospechosa o equivocada, esto brinda capacidades vitales de detección y protección. El sistema almacena y mantiene un histórico de los eventos deseados. El **audit trail** es

[13] UNISYS UNIX System V Release 4.0 "Security Features User's Guide"

creado en la transición del nivel monousuario al multiusuario y es mantenido por el TBC a través de operaciones multiusuario siempre y cuando en el archivo **/etc/default/audit** la variable **AUDIT** este en **YES**, por lo tanto, es conveniente ejecutar este cambio por medio del editor del sistema si es que no se cuenta con él.

Los accesos remotos y locales por medio del usuario root pueden ser detenidos de una forma bastante sencilla, en el archivo **/etc/default/login** la variable **ROOTLOGIN** debe estar con el valor **NO**, de esta manera, el superusuario tiene que ser invocado ingresando previamente con una cuenta ordinaria y con lo que respecta al login remoto, no tendrán acceso al superusuario aun conociendo su password. En ese mismo archivo existe otra variable llamada **CONSOLELOG** si esta declarada en **YES**, desplegará en la consola los mensajes de intentos frustrados de entrada al sistema incluyendo el nombre del usuario y la terminal desde la que lo intentó o la dirección IP del equipo remoto.

El **audit trail** puede ser utilizado para contestar las siguientes preguntas típicas:

- ¿Quién accesó el archivo **xyz** entre junio 3 y junio 5?
- ¿Quién fue el último al que se le concedió el derecho de modificar el archivo **abc**?
- ¿Qué objetos acceso el usuario joel el fin de semana?
- ¿Quién accesó el sistema ayer?
- ¿Qué programas ejecutaron y que objetos accesaron?
- ¿Hubo actividades sospechosas ayer: fallas de las cuentas, intentos de acceso fallidos etc?

La instalación, configuración y administración del Audit Trail, será desarrollada a detalle en el capítulo 5, en la implantación de las características de seguridad del sistema operativo.

CAPÍTULO 3

CARACTERÍSTICAS DE SEGURIDAD DE INFORMIX-SE

3.1. Antecedentes de Informix-SE

Informix es un lenguaje de cuarta generación desarrollado por Informix Software. Inc. y diseñado específicamente para aplicaciones de base de datos.

3.1.1. ¿Qué son los Lenguajes de Cuarta Generación?

Los lenguajes de cuarta generación tales como Informix-4gl son diseñados para una particular clase de aplicaciones. Estos son menos complejos que los lenguajes de propósito general como COBOL o C y son más inmediatamente aproximados al "lenguaje natural". Porque se enfocan a un especifico tipo de aplicación. También los lenguajes de cuarta generación son muy poderosos: una simple declaración genera una gran cantidad de código de máquina. Como resultado, los programas escritos en lenguajes de cuarta generación no contienen tantas declaraciones como programas escritos en un lenguaje de propósito general.

Algunas ventajas de los lenguajes de cuarta generación son:

- Son simples. lo que acelera los procesos de construcción y mantenimiento de aplicaciones.
- Son generalmente interactivos, lo que simplifica la depuración de procesos.
- Son atractivos para una gran audiencia porque no requieren un especial entrenamiento.
- El resultado de las aplicaciones son fáciles de usar y pueden resolver problemas eficientemente.

3.1.2. Lenguajes Procedurales y No Procedurales

Los lenguajes de programación son algunas veces referenciados como procedurales y no procedurales. Cuando se usa un lenguaje procedural, se especifica en el programa como se quiere realizar algo. Este paso por paso, hace al lenguaje procedural muy flexible, así que se puede usar para una gran variedad de aplicaciones.

Por ejemplo, si se está diseñando un programa de un manejador de menús usando un lenguaje procedural como COBOL o C, se tiene que especificar, paso a paso, como desplegar el menú y manejar las entradas del usuario. Así como un programa puede incluir instrucciones para desplegar el titulo del menú, opciones del menú y para mover el cursor de una posición a otra. Este pudiera contener instrucciones como **FOR** o **CASE** que ejecutan una serie de operaciones, dependiendo de la entrada del usuario.

Por otro lado, cuando se usa un lenguaje no procedural, se especifican los resultados deseados, y el lenguaje proporciona el procedimiento. Imaginemos que se quiere diseñar un programa de manejador de menús usando un lenguaje no procedural. En este caso, se podrá crear un menú usando un proceso MENÚ de Informix-4gl. No se necesitará usar un proceso de impresión para desplegar el titulo del menú y las opciones porque el **MENÚ** ha construido los procesos que despliegan el menú para nosotros. Igualmente no necesitaremos usar un procedimiento de condición para manejar los requerimientos del usuario, porque el **MENÚ** en efecto crea un procedimiento parecido al CASE.

Informix-4gl combina las características del lenguaje procedural y no procedural. Hemos visto como Informix-4gl ofrece características no procedurales como la característica del **MENÚ** para construir aplicaciones simples. Informix-4GL también ofrece características procedurales como **IF**, **FOR** y **WHILE** así que se puede pensar que el diseño de Informix-4GL podría no ser predicho. Así, Informix-4GL combina la rapidez y simplicidad de los lenguajes no procedurales con la flexibilidad de los lenguajes naturales.

3.1.3. Características de Informix-4Gl.

Informix-4GL es un poderoso lenguaje de cuarta generación, provee todas las herramientas que se necesitan para crear un sistema manejador de base de datos relacional. Es un lenguaje de base de datos que se puede usar para guardar, recobrar, dar de alta y borrar información, además también es un:

- **L**enguaje de programación.
- **U**tilería de construcción de pantallas.
- **U**tilería de construcción de menús.
- **U**n report writer.
- **M**anejador de ventanas.

3.2. Aplicación del Registro de Transacciones

3.2.1. Transacciones: Todos o Ninguno

Una transacción es una serie de operaciones en una base de datos que nosotros queremos completar enteramente o no hacerla.

Si dos acciones envueltas en transferir dinero de una cuenta a otra no son hechas simultáneamente, las cuentas no se balancearán, y la integridad de los datos no se preservará.

Ejemplos de transacción son abundantes en contabilidad y libros de balance donde diversas operaciones en varias cuentas diferentes deberán ser hechas como una unidad, o el libro estará fuera de balance.

El proceso de transferir dinero de una cuenta a otra, por ejemplo, envuelve a dos distintas acciones:

1. Substraer dinero de la primera cuenta.
2. Agregar dinero a la segunda cuenta.

Si una de esas acciones es hecha, pero no la otra, las cuentas no estarán balanceadas y la integridad de los datos no se preservará. Agrupando esas dos acciones juntas como una simple transacción será seguro que ambas acciones se realicen o ninguna se lleve a cabo y la integridad de los datos se preserve.

Los productos Informix soportan la integridad de los datos mediante la implantación de la idea de transacciones. Una transacción es una serie de operaciones de base de datos (declaración de SQL) que nosotros queremos que se complete enteramente o no hacerla.

3.2.2. ¿Qué es un Registro de Transacciones?

Para usar transacciones en una base de datos, se deberá crear un registro de transacciones para la base de datos. Un registro de transacciones es un archivo en el que los productos Informix registran todas las modificaciones a la base de datos.

El registro de transacciones provee dos beneficios:

1. Permite tratar una serie de operaciones como una simple unidad de trabajo.

2. Permite recobrar una base de datos si los datos llegan a corromperse por un disco dañado u otro evento.

Para usar transacciones en una base de datos, se necesita un archivo de transacciones en el cual las modificaciones a la base de datos puedan ser grabadas. Usamos el comando **CREATE DATABASE** con la cláusula **WITH LOG IN** para crear una nueva base de datos y especificar una transacción log.

La sintaxis del comando es la que sigue:

$ dbname palabra_llave pathname llave_opcional

Donde:

dbname	Es el nombre de una base de datos.
WITH LOG IN	Son palabras llave requeridas
pathname	Es la ruta completa, encerrada entre comillas (" "), de un archivo de registro de transacciones.
MODE ANSI	Son llaves opcionales que especifica la base de datos es compatible con ANSI.

El archivo de registro de transacciones puede residir en un diferente disco físico, del que reside la misma base de datos. Esto permitirá recobrar información si existe alguna falla en algún disco.

Si se ha creado una base de datos sin transacciones, podemos adicionar transacciones usando el comando **START DATABASE** con la cláusula **WITH LOG IN**. El comando **START DATABASE** también puede ser usado para cambiar el nombre de un archivo de transacciones.

La sintaxis del comando es la siguiente:

$ dbname palabra_llave pathname palabra_opcional

dbname	Es el nombre de una base de datos
WHIT LOG IN	Son palabras llave requeridas
pathname	Es la ruta completa, encerrada entre comillas ("), del archivo de transacción log
MODE ANSI	Son palabras opcionales que especifican si la base de datos es compatible con ANSI.

El archivo de transacciones podrá residir en un diferente dispositivo, del que reside la misma base de datos .

El **START DATABASE** podrá ser precedida por un **CLOSE DATABASE** si hay una base de datos actual.

El **START DATABASE** podrá ser ejecutada inmediatamente antes de hacer un respaldo de la base de datos, donde el nombre y la localización del archivo de transacciones son guardadas con el respaldo.

El comando **START DATABASE** puede correr solamente por un usuario con privilegio DBA para afectar la base de datos. Un respaldo del archivo de la base de datos podrá ser tomada inmediatamente después de inicializar una transacción log.

Cuando queremos ejecutar un grupo de declaraciones que accesan la base de datos como una unidad, podemos ejecutar esas declaraciones sin una transacción. Si la base de datos tiene un registro de transacción pero no es compatible con ANSI, usamos **BEGIN WORK** antes de la primera declaración en la transacción. Usamos **COMMIT WORK** o **ROLLBACK WORK** después de la última declaración en el grupo para completar la transacción.

BEGIN WORK Marca el inicio de una transacción (Si la base de datos no es compatible con **ANSI)**

COMMIT WORK Marca el final de una transacción autorizando todos los cambios desde donde la transacción empieza.

ROLLBACK WORK Marca el final de una transacción revocando la mayoría de los cambios desde donde la transacción empieza. *

Una base de datos compatible con ANSI es creada incluyendo la cláusula **MODE ANSI** en un **CREATE DATABASE** o en una declaración **START DATABASE**. Una base de datos compatible con **ANSI** puede tener un registro de transacciones, y todas las declaraciones toman un lugar sin una transacción.

No se necesita usar la declaración **BEGIN WORK** con una base de datos compatible con **ANSI** donde la declaración esta implícita. Las transacciones deberán a pesar de eso estar terminadas, sin embargo, con una declaración **COMMIT WORK** o **ROLLBACK WORK**. Cuando una transacción esta terminada, una nueva transacción automáticamente empieza.

Las transacciones de las bases de datos compatibles con **ANSI** son conocidas como transacciones implícitas.

Si las transacciones no son compatibles con ANSI, se deberá utilizar la declaración **BEGIN WORK** antes de ejecutar una serie de operaciones que se quieren considerar como una simple transacción.

Si no se emplea la declaración **BEGIN WORK**, cada declaración que cambie la base de datos será tratada como una simple transacción. Cada declaración, si se ejecuta suficientemente, esta terminada y la base de datos esta permanentemente alterada. Si la declaración falla, hay un automático regreso al estado anterior a la declaración.

Múltiples declaraciones en las transacciones de una base de datos no son compatibles con **ANSI** y son conocidas como transacciones explícitas, así un **BEGIN WORK** deberá ser ejecutado explícitamente.

Cada renglón afectado por una declaración **UPDATE**, **DELETE** o **INSERT**

Una transacción que afecta un gran número de renglones puede exceder el límite impuesto por el sistema operativo en el máximo número de renglones asegurados. Si se encuentra este error . es necesario asegurar la tabla entera después de empezar la transacción.

Si se esta satisfecho de los resultados producidos en la transacción, se termina la transacción con una declaración **COMMIT WORK**. Este depositará todas las modificaciones hechas a la base de datos durante la transacción.

La declaración **COMMIT WORK** cierra todos los cursores abiertos, excepto los cursores declarados **WITH HOLD**. No se deberá usar la declaración **COMMIT WORK** dentro de un loop de un **FOREACH** cuando se usa INFORMIX-4GL, donde este cerrará el loop del cursor.

Todos los renglones y tablas aseguradas están sueltas para la declaración **COMMIT WORK**.

Si no se esta satisfecho con el resultado de una transacción, se termina la transacción con la declaración **ROLLBACK WORK**. Esta declaración regresa a la base de datos al estado en que se encontraba antes de que se empezará la transacción, con una importante excepción.

No se puede regresar la declaración **GRANT** o **REVOKE**, ni tampoco cualquiera de los datos definidos en la declaración. Estas declaraciones alteran el numero o nombres de la tabla o cambian el número, nombres, tipo de datos o índices de columnas.

La declaración **ROLLBACK WORK** cierra todos los cursores abiertos, excepto los **WITH HOLD**, y libera todos los renglones y tablas bloqueadas.

Si se llega corromper una base de datos, se puede recobrar la base de datos restaurando la copia de respaldo y ejecutando la declaración **ROLLFORWARD DATABASE**. Todas las transacciones registradas en el registro de transacciones será explicada a la copia de seguridad de la base de datos, recobrando todas las transacciones completamente.

La sintaxis de la declaración es:

ROLLFORWARD DATABASE *database-name*

donde *database-name* es el nombre de la base de datos que se quiere recobrar.

Inmediatamente después de que se aplica un **FORWARD** a una base de datos, esta en modo **EXCLUSIVO** sin transacciones. El modo **EXCLUSIVO** evita acceso para algún usuario presente.

Después la base de datos es cerrada y reabierta, esta llega a ser accesible a otros usuarios, y las transacciones se pueden reanudar.

Solamente el DBA puede ejecutar la declaración **ROLLFORWARD DATABASE**

3.2.2.1. Mantenimiento del Archivo de Transacciones Log

El archivo de transacciones log puede ser muy largo, y periódicamente, se deseará respaldar este archivo en una cinta e inicializar otro archivo log. Al mismo tiempo, un respaldo de la base de datos deberá ser creado. En general, cada archivo de transacción log deberá tener una copia del archivo de la base de datos.

Después de hacer el respaldo de la base de datos y el archivo log, un nuevo archivo log deberá ser especificado. Para usar el mismo archivo log, se deberá crear un archivo log vacío con el mismo nombre del archivo viejo.

Esto puede hacerse con los siguientes comandos de UNIX.

```
$ cat /dev/null/ >logfile
```

Para cambiar el nombre del logfile, se deberá ejecutar el comando **START DATABASE** antes de hacer un respaldo de la base de datos y especificar el nuevo nombre del archivo log.

La localización del archivo de transacciones log esta guardada en el catalogo systable, y este es tabid = 0. Si se necesita recobrar el archivo, se deberá indicar la localización del archivo de transacciones.

3.2.3. Removiendo un Archivo de Transacciones Log

Se puede remover el archivo de transacciones (log) por las siguientes razones:

- Para incrementar la cantidad del espacio del disco libre.
- Para incrementar la velocidad de la base de datos.
- Para sustituir uno o más audit trail por el archivo de transacciones.
- Para remover las transacciones de una base de datos, ejecutar las siguientes operaciones en el orden mostrado:

1. Con permisos de DBA, accesar la base de datos en modo exclusivo.

$ database database-name exclusive

2. Si este comando falla, una vez usando la base de datos. Esperar y tratar de nuevo con el comando.
3. Localizar y anotar la ruta completa del archivo de transacciones.

$ select dirpath **systables where tabtype = "L"**

4. Remover la entrada para el archivo log de transacciones en la tabla de catalogo del systables (se puede usar Informix).

delete from systables where tabtype = "L"

5. Este comando borra las entradas para el archivo log de transacciones, pero no borra el archivo log de transacciones activas.
6. Cerrar la base de datos.
7. Remover el archivo log de transacciones.

3.2.4. Recobrando un Log de Transacciones Corrupto

Si se descubre que un log de transacciones se ha corrompido se podrá remover este y reemplazarlo con un archivo log vacío siguiendo los procedimientos de la página anterior.

Si esto no puede ser posible, sin embargo, si el archivo de transacciones log esta corrupto e impide el acceso a la base de datos seguir las siguientes operaciones:

- Hacer una copia del archivo systables.dat localizado en el directorio de la base de datos.
- Usar el editor o alguna otra utilería para buscar la entrada del syslog. La línea contiene la palabra llave **syslog** también contiene el nombre del path completo del archivo de transacciones log.
- Localizar y anotar el nombre completo del path del archivo de transacciones log.
- Salir del editor u otra utilería.
- Crear un archivo vacío de transacciones log con el nombre encontrado en el archivo systable.dat. En sistemas UNIX, usar el comando.

$ cat /dev/null > full pathname

Ahora se estará habilitando para accesar la base de datos. Si se escoge, se puede ahora remover el log de transacciones con el procedimiento estándar

Nota: Si una transacción log no puede accesar la base de datos, se presenta el siguiente error

224: cannot open transaction log file

120: ISAM error: cannot open log file

3.3. Utilización de Audit Trails

Un Audit trail es un archivo que contiene un histórico de todas las altas, bajas, actualizaciones y manipulaciones a una tabla de una base de datos. Si la tabla llega a corromperse, se puede usar el audit trail para restaurar la tabla.

Un servicio de audit trail tiene un propósito similar al log de transacciones: cada uno es usado para mantener un registro de modificaciones a la base de datos, y cada uno puede ser usado para actualizar respaldos de una base de datos. El audit trail y log de transacciones son diferentes.

Algunos factores distinguen al audit trail del log de transacciones

a) El log de transacciones contiene un registro coordinado de todas las modificaciones a la base de datos incluyendo actualizaciones, inserciones y bajas que afectan múltiples tablas. El audit trail registra modificaciones a una sola tabla.

b) La transacciones garantizan que las declaraciones SQL estén ya sea completamente realizadas o completamente canceladas. Un audit trail no protege contra parciales ejecuciones de una declaración SQL.

c) Se puede usar el log de transacciones para recobrar una base de datos entera. Se puede usar un archivo de audit trail para recobrar solamente la tabla para la cual esta creado.

d) Con el audit trail, se puede registrar modificaciones a una sola tabla importante sin incurrir en el gasto del sistema de mantener un log de transacción en la base de datos entera.

Se deberá considerar usar un audit trail solo cuando se tiene uno o pocas tablas criticas, y no se necesitan las facilidades adicionales que provee para transacciones. Si se necesita mantener la integridad de base de datos como un todo, o se necesita la garantía de que las declaraciones SQL sean ejecutadas como una unidad ya sea completamente o nada, entonces se tendrá que usar las transacciones.

Si se ha estado usando las transacciones con las base de datos, entonces no se use el audit trail. El log de transacciones recuperado y el audit trail recuperado puede causar conflictos con otro.

Se usa la declaración CREATE AUDIT para crear un archivo audit trail, y empieza escribiendo a el audit trail. La sintaxis para la declaración es la mostrada como sigue.

CREATE AUDIT FOR es requerida como palabra llave

table-name Es el nombre de la tabla para la cual se quiere crear un audit trail

IN Es una palabra requerida.

Pathname Es el nombre de la ruta completa para el archivo del audit trail. Este deberá estar encerrado entre comillas (")

Si un archivo audit trail de la misma ruta ya existe para la misma tabla, el CREATE AUDIT no hace nada. Si un archivo audit trail para la misma tabla existe con una diferente ruta, entonces un mensaje de error es desplegado.

Se deberá hacer una copia de seguridad de los archivos de la base datos para esta tabla inmediatamente después de crear un audit trail . Si es posible, el archivo del audit trail deberá estar guardado en un diferente dispositivo físico reservado para los datos. Así que una falla de uno no daña a otro.

Se deberá ser el dueño de la tabla o tener status de DBA para usar la declaración CREATE AUDIT.

No se puede crear un audit trail para una vista. Tampoco se puede crear un cluster index en una tabla que tiene un audit trail, y viceversa.

Usamos la declaración DROP AUDIT para borrar un archivo audit trail. La sintaxis para la declaración es la siguiente.

DROP AUDIT FOR	Son palabras reservadas
table-name	Es el nombre de la tabla cuyo audit trail se quiere borrar

La declaración DROP AUDIT puede ser usada para remover un archivo audit trail viejo cuando se ha hecho un respaldo de los archivos de la base de datos. Se usa la declaración CREATE AUDIT para inicializar un nuevo audit trail y entonces respaldar la tabla.

Se debe ser el dueño de tabla o tener status de DBA para usar el DROP AUDIT.

En el caso de que el sistema se dañe, se puede usar la declaración RECOVER TABLE para restaurar una tabla usando una copia de la tabla y un archivo audit trail. La sintaxis para la declaración es la siguiente.

RECOVER TABLE	Son palabras llaves requeridas
table-name	Es el nombre de la tabla que se quiere recobrar

Se deberá primero cargar una copia de la tabla para apropiadamente recobrar esta. El respaldo deberá estar en el estado original cuando se inicializó el audit trail. Si no esta en estado original, la recuperación fallará.

Una vez que se recobro la tabla, se usa la declaración DROP AUDIT para remover el contenido del audit trail. Correr la declaración CREATE AUDIT para inicializar un nuevo archivo audit trail. entonces respaldar, la tabla.

Las siguientes declaraciones SQL proveen un modelo para recobrar una tabla, asumiendo que el audit trail empezó desde el último respaldo.

```
{ restore table from last backup }

recover table customer;

drop audit for customer;
```

```
create audit for customer in "/u/safe/customer.aud":

{ make a backup of the recovered table }
```

Nota: Se deberá ser el dueño de la tabla o tener status de DBA para usar la declaración RECOVER TABLE.

3.4. Niveles de Seguridad de la Base de Datos

Los bloqueos que hace INFORMIX-SE los guarda ya sea a nivel de kernel o en el archivo .lok asociado con cada tabla en la Base de Datos. Para determinar que método de bloqueo se usa del INFORMIX-SE. se lista el contenido del directorio .DBS y se ven los archivos .lok; por ejemplo

$ ls -l stores.dbs/* . lok

Si el método de propiedad es usado, la información bloqueada es escrita en el archivo .lok.. La actual fila en disco. no esta bloqueada. pero en lugar de eso una entrada en el archivo .lok indica que filas están bloqueadas.

Si el método de kernel esta implementado. los bloqueos son colocados usando sistema de llamada fcntl. El sistema de llamada fcntl bloquea físicamente los registros en disco

INFORMIX-SE implementa los bloqueos de kernel usando el sistema de llamadas fnctl(2) del Sistema Operativo UNIX. En el método de bloqueo del kernel, el registro esta actualmente bloqueado en el archivo de disco donde éste actualmente reside. En el bloqueo del kernel, el usuario puede esperar durante un registro bloqueado usando el SET LOCK MODE TO WAIT command.

Si un usuario intenta accesar una fila que esta bloqueada. y el comando SET LOCK MODE TO WAIT ha sido usado. el programa que esta accesando la fila bloqueada podrá esperar hasta que la fila haya sido desbloqueada. Si el bloqueo falla para el proceso que esta teniendo el bloqueo. El proceso podría esperar indefinidamente. Primero el bloqueo es liberado. la fila esta libre para esperar al usuario.

INFORMIX-SE implementa el bloqueo del propietario usando un archivo del disco el cual registra todos los bloqueos para una tabla especifica. Hay un archivo .lok creado para cada tabla en la Base de Datos. El sistema de bloqueo de propietario mantiene una estructura para cada tabla en la Base de Datos, y esa estructura esta escrita en el archivo .lok del disco.

Si la implantación de propietario es usada, el comando SET LOCK MODE TO WAIT podrá fallar y regresar un error. Si un usuario intenta accesar una fila que esta bloqueada, la petición fallará y un mensaje de error será regresado.

Los problemas de bloqueos perdidos son usualmente indicados por los mensajes de error ISAM de cuyos estados de los registros o tablas pueden no ser bloqueados. Los problemas de bloqueos perdidos son creados si un proceso del servidor de INFORMIX-SE muere y el manejador del bloqueo no lo libera (cualquiera de los dos el SE o el Sistema Operativo).

Para limpiar los bloqueos perdidos en un sistema usando el mecanismo de bloque del propietario, primero checar que los usuarios no estén accesando la tabla que esta bloqueada. Entonces usar el siguiente comando

$ cat /dev/null > lock_filename

Si se esta en un sistema que usa alguna versión de Unix con bloque del kernel, se debe dar de baja el sistema con shutdown para limpiar el archivo de bloqueo.

INFORMIX-SE permite bloquear a los tres diferentes niveles. Usando el bloqueo a nivel de base de datos, un solo usuario tendrá acceso a esa base de datos. Ningún otro usuario podrá leer o escribir en alguna tabla de la base de datos.

El bloqueo a nivel tabla otorga a un sólo usuario ya sea acceso exclusivo o compartido al especificar una tabla en la base de datos.

El bloque a nivel de fila permite a un usuario bloquear una fila o filas dentro de una tabla en una Base de Datos

El bloqueo a nivel Base de datos es ocasionalmente necesaria o ventajosa para evitar el acceso de usuarios en alguna otra parte de la Base de Datos por algún periodo de tiempo. Este puede ser el caso si se esta:

- Ejecutando un gran número de actualizaciones que envuelven varias tablas
- Guardando los archivos de la Base de Datos para respaldo

- Alterando la estructura de la Base de Datos

- Verificando que una Base de Datos fuera correctamente restaurada después de usar la declaración ROLLFORWARD DATABASE .

La Base de Datos entera puede estar bloqueada usando la declaración DATABASE con la opción EXCLUSIVE.

La opción EXCLUSIVE abre la Base de Datos en un modo exclusivo, y permite solamente el acceso al usuario actual a la Base de Datos.

Para permitir a otros usuarios el acceso a la Base de Datos, se puede ejecutar la declaración CLOSE DATABASE y reabrir la Base de Datos.

La Base de Datos es automáticamente abierta en modo exclusivo siempre que el DBA ejecuta las declaraciones ROLLFORWARD DATABASE y START DATABASE .

Nota: Los usuarios con algún nivel de permisos de la Base de Datos pueden abrir la Base de Datos en modo exclusivo.

Haciéndolo así no obtendrán ellos algún mayor nivel de acceso al que normalmente tienen.

El bloqueo a nivel tabla puede ser usado para evitar que otros usuarios modifiquen la Base de Datos. Se usa el bloqueo al nivel tabla para:

- Evitar conflictos con otros usuarios durante las operaciones en lote que efectúan la gran mayoría de las filas de una tabla.

- Evitar corridas fuera de los bloqueos cuando se están corriendo operaciones como una transacción.

- Evitar que los usuarios hagan querys a una tabla por un periodo de tiempo.

- Evitar el acceso a una tabla mientras se altera su estructura o crean índices.

Se podrá usar el bloqueo a nivel tabla solamente cuando estamos accesando a una tabla en ambiente multiusuario, y cuando simultáneamente interactuamos con otro usuario que puede interferir.

Solamente un bloqueo puede aplicarse a una tabla a la vez. Esto es, si un usuario bloque una tabla, otro usuario no puede desbloquearla hasta que el primer usuario ha desbloqueado esta.

Nota: No se puede bloquear el system catalog tables.

Si la Base de Datos tiene transacciones, las tablas pueden ser bloqueadas sin transacciones. Por lo tanto, estar seguro de que se esta ejecutando un BEGIN WORK (a menos que se este usando una base de datos en modo ANSI) antes de intentar bloquear una tabla. La tabla estará desbloqueada cuando la transacción se complete.

Si se quiere dar a otro usuario acceso de lectura a una tabla, pero prevenirlos de algunas modificaciones de los datos que estas contienen, entonces se deberá utilizar la declaración con la opción IN SHARE MODE.

Cuando una tabla es bloqueada en modo SHARE , otros usuarios están habilitados dar un SELECT a la tabla, pero no están habilitados para insertar, borrar o actualizar filas en la tabla, o alterar la tabla.

Si se quiere prevenir a otros usuarios de no tener acceso a la tabla, entonces se deberá bloquear esta en modo EXCLUSIVE.

En modo EXCLUSIVE, otros usuarios no estarán habilitados para seleccionar, insertar, borrar o actualizar filas hasta que esta sea desbloqueada.

La declaración UNLOACK TABLE restaura accesos a una tabla de la base de datos bloqueada. Se usa esta declaración cuando no se alarga la necesidad de prevenir a otros usuarios del acceso y modificación de la tabla.

Si la tabla fue bloqueada en una transacción, UNLOCK TABLE no es permitido y genera un error. Finalmente la transacción (vía COMMIT or ROLLBACK) desbloqueará la tabla.

Ordinariamente, una fila es bloqueada automáticamente cuando se ejecuta una declaración UPDATE, o cuando se ejecuta una declaración FETCH y el cursor esta declarado con la cláusula FOR UPDATE. Bloqueando una fila se previene a dos programas o usuarios de intentar actualizar la misma fila al mismo tiempo. Un programa o un usuario y puede proseguir con la actualización. El otro deberá esperar para que el bloqueo sea liberado después de actualizar la fila.

Si la declaración UPDATE afecta solo una fila, el bloqueo es liberado inmediatamente después de ejecutar la actualización. Si la declaración UPDATE afecta más de una fila, la misma fila bloqueada estratégicamente es usada. Tan pronto como una fila es actualizada, el bloqueo es liberado y la siguiente fila es bloqueada y actualizada. Cuando la actualización termina , la última fila bloqueada es liberada.

Si se quiere más control sobre la actualización de múltiples filas se puede de declarar un cursor para la actualización. La cláusula WHERE de la declaración SELECT especifica las filas que se pueden actualizar. Después de que se abre el cursor y se coloca el FETCH en una fila, esa fila permanece bloqueada hasta que se cierre el cursor o el FETCH la siguiente fila.

El bloqueo a nivel tabla es ligeramente diferente cuando se esta usando una base de datos con transacciones.

Las filas que se insertan, actualizan o borran dentro de una transacción permanece bloqueada hasta el final de la transacción.

Durante las transacciones que afectan a un gran numero de filas, se puede exceder el limite que el sistema operativo permite en el numero de bloqueos. Este problema puede ser evitado bloqueando la tabla completa al empezar la transacción. Si se bloquea la tabla entera, entonces el nivel de bloqueo a nivel fila no es usado, porque es innecesario. Como resultado, es probable que se alcance el límite de bloqueos simultáneos.

Bloqueando una tabla completa prevenimos otras de alterar datos en la tabla. Bloquear la tabla completa solamente cuando el bloqueo a nivel fila es insuficiente.

Si la base de datos tiene transacciones pero no es compatible con ANSI, se deberá usar un BEGIN WORK antes de que se utilice la declaración LOCK TABLE.

Si la base de datos tiene transacciones la declaración UNLOCK TABLE no puede ser usada y genera un error. Las tablas solo pueden ser bloqueadas dentro de una transacción y pueden solamente ser desbloqueadas terminando la transacción. Todos los bloqueos permanecen en una tabla y son liberados cuando el COMMIT WORK o ROLLBACK es procesado.

La declaración SET LOCK MODE es usada para determinar si las llamadas que alteran o borran una fila bloqueada espera a una fila para llegar a ser desbloqueada.

La opción TO NOT WAIT causa un error que es regresado si una declaración trata de alterar o borrar una fila (o el SELECT o una fila FOR UPDATE) que otros procesos tienen bloqueados. Este es el modo de default.

La opción TO WAIT causara un estado de espera en un intento de alterar o borrar una fila que ha sido bloqueada por otro proceso hasta que la fila bloqueada llega a desbloquearse.

La característica del SET LOCK MODE esta disponible solamente en los sistemas que tienen bloque a nivel kernel y aplica solamente en bloqueos a nivel tabla.

CAPÍTULO 4

ANÁLISIS DEL SISTEMA INTEGRAL DE INFORMACIÓN TRIBUTARIA

4.1. Situación Actual del Sistema Integral de Información Tributaria (S.I.I.T.)

La situación actual de la seguridad para la aplicación, se reduce al manejo de passwords y permisos en los archivos que se generan en el equipo Unix. El S.I.I.T. tiene permisos para todos los usuarios del sistema y su base de datos es de dominio público, por lo que al conectar las máquinas en red, cualquier usuario remoto tiene acceso a los archivos ejecutables de la Aplicación y a la base que genera. La información generada por el sistema, debe ser protegida debido a que contiene datos que son claves para el proceso jurídico tales como nombre del contribuyente, fechas de resoluciones, persona que lleva el caso, montos de adeudos (a veces de miles de millones de pesos), títulos y bienes embargados y estado de los juicios del área contenciosa.

El S.I.I.T. que actualmente opera a nivel nacional, es el encargado de reportar la productividad de las áreas que abarcan las Administraciones Locales Jurídicas de Ingresos.

Esta aplicación se divide en tres módulos:

1. Control de Gestión
2. Notificación y Cobranza
3. Contencioso

Control de Gestión: Este módulo lleva el control integral de las promociones. Una promoción es todo aquello que los contribuyentes piden a la Administración General Jurídica de Ingresos

(A.G.J.I.) y requiere una respuesta. (estado en que se encuentra una resolución o a que área ha sido turnada).

De estas áreas la que tenía prioridad era la de Recursos Administrativos, debido a que la información manejada allí corresponde a las apelaciones de las demandas, que según el contribuyente son injustificadas. El área debe emitir una resolución durante cierto tiempo para favorecer a alguna de las partes implicadas en la demanda (ya sea a la propia Secretaría o al Contribuyente), en caso contrario se favorecerá al contribuyente.

El sistema también cuenta con una opción para la consulta de las promociones (Mesa de Trámite), pero ésta solo muestra una parte de la información capturada para evitar que el contribuyente se entere que abogado maneja su caso y así evitar casos de corrupción. Otra de sus funciones, es la entrega de productos útiles para las áreas mencionadas, tales como estadísticas de control y reportes de supervisión. El mismo se dividió en partes para un mejor entendimiento de la programación y utilizar partes que podían servir en su momento a otros programadores.

Notificación y Cobranza: Notifica el adeudo al contribuyente si en un plazo de 45 días éste no presenta un recurso de revocación o no paga, el área homónima procede a embargar los bienes equivalentes a 5 veces el monto del adeudo.

Contencioso: Lleva el control de los juicios presentados por el contribuyente al Tribunal Fiscal de la Federación (T.F.F.) en contra de la A.G.J.I.

La figura 4.1, muestra el flujo de información que se manejaba en el sistema en su primera versión y la relación de los módulos del S.I.I.T. Actualmente, la función de Notificación y Cobranza fue entregada a la Administración General de Recaudación por lo que la última versión de este sistema, ya no contempla este módulo en su operación.

Sistema Integral de Información Tributaria

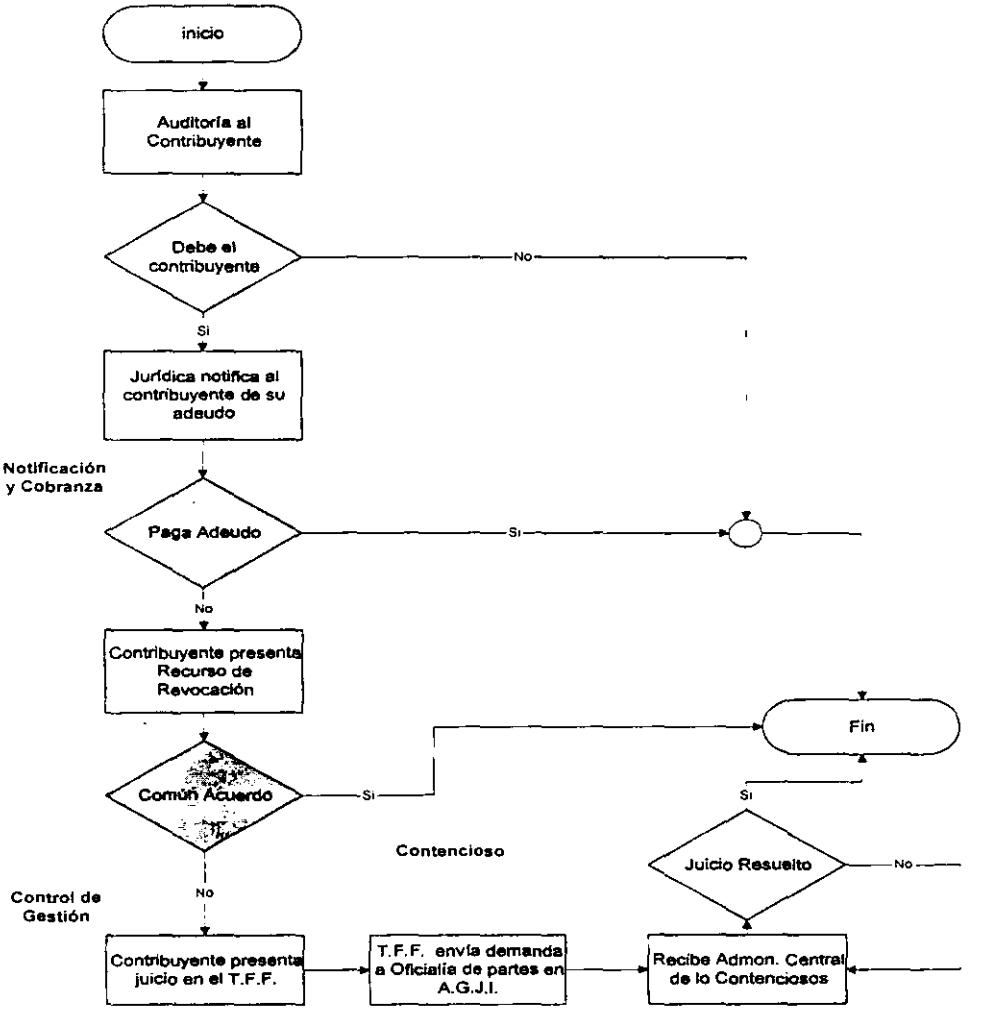


figura 4.1

Oficialía de Partes se caracteriza por ser el área que debe dar solución al problema de la documentación (que desde su ingreso se llama promoción) y la distribuye. Las áreas las reciben y tienen cierto período de tiempo para dar una respuesta a la promoción, que se contabiliza a partir de que Oficialía de partes la recibe. Durante ese lapso de tiempo la promoción atraviesa por ciertas etapas, proceso que se registra en el sistema.

Si la respuesta a la promoción era a través de un oficio, éste se turna a Oficialía de Partes, que a su vez registra la salida en la base de datos.

Como todas las etapas tienen que ser registradas, el sistema es la fuente de información más fidedigna y rápida con que se cuenta para enterar al contribuyente de su promoción, tarea que realiza el área de Mesa de Trámite, además de proporcionar reportes y estadísticas de productividad de la A.G.J.I.

4.1.1. Organigrama de Usuarios del S.I.I.T.

De las ocho Administraciones Centrales que forman a la Administración General Jurídica existen cuatro que utilizan al S.I.I.T. para el desarrollo de sus actividades. En la tabla 4.1.1 se puede apreciar cuales son. Cada una de ellas tiene diferentes usuarios del S.I.I.T en el servidor Unix HP Vectra que a nivel central lo tiene en explotación.

No.	Login Actual	User ID	Group ID	Nombre Real Usuario	Atributo	Path	Ubicación
1	Admon	236	50	Administración de Control de Gestión	Administrador del S.I.I.T.	/u/siit93/siit1	Módulo 6 P.B.
2	admonres	246	50	Administración de Informática	Administrador en Informática	/u/siit93/siit2	Módulo 6 P.B.
3	asisten	240	50	Asistencia al Contribuyente	Consulta	/u/siit93/siit1	Módulo 5 P.B.
4	auditor	245	50	Auditor del Sistema	Audita	/u/siit93/siit2	Módulo 6 P.B.
5	captura	242	50	Notificación y Cobranza	Captura	/u/siit93/siit2	Módulo 5 P.B.

tabla 4.1.1

No.	Login Actual	User ID	Group ID	Nombre Real Usuario	Atributo	Path	Ubicación
6	consulta	249	50	Notificación y Cobranza	Consulta	/u/siit93/siit2	Módulo 5 P.B.
7	exterior	250	50	Comercio Exterior	Consulta	/u/siit93/siit2	Módulo 6 P.B.
8	externos	243	50	Notificación y Cobranza	Captura	/u/siit93/siit2	Módulo 5 P.B.
9	oficial	237	50	Oficialía de Partes	Captura	/u/siit93/siit1	Modulo 6 P.B.
10	recursos	238	50	Central de lo Contencioso	Captura	/u/siit93/siit1	Módulo 6 P.B.
11	servis	239	50	Asistencia al Contribuyente	Consulta	/u/siit93/siit1	Módulo 5 P.B.
12	siit	234	50	Administración de Informática	Desarrollo	/u/siit93	Módulo 6 P.B.
13	tramite	241	50	Asistencia al Contribuyente Mesa de Trámite	Consulta	/u/siit93/siit1	Módulo 5 P.B.
14	siitcen	208	50	Administración de Informática	Consulta	/u/siit93/siit2	Módulo 6 P.B.

tabla 4.1.1

Administraciones Centrales que Operan el S.I.I.T.

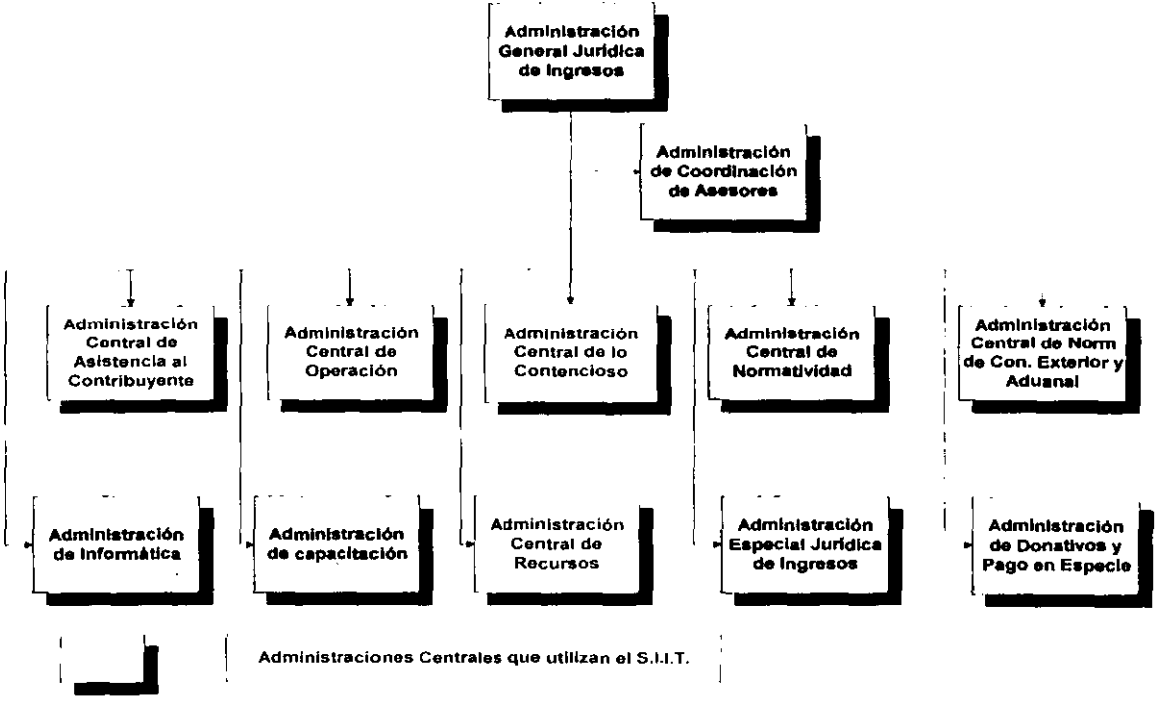
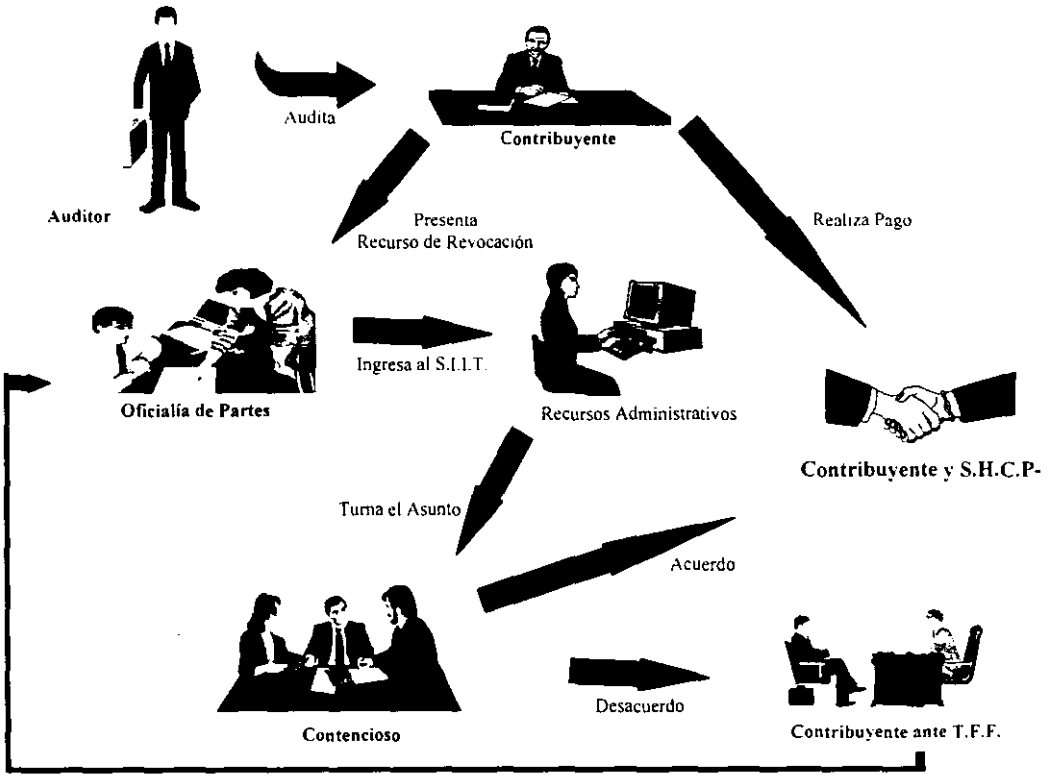


figura 4.1.1

4.2. Presentación del Sistema

Diagrama Conceptual



4.2.1. Módulo de Control de Gestión

Esta sección del sistema fue denominada Control de Gestión, por que permite el control de las promociones o asuntos[1], pero únicamente para las áreas de Servicios al Contribuyente, Recursos Administrativos y Asistencia al Contribuyente.

Los objetivos fundamentales de este módulo son:

- Llevar a través de un computador el registro y control de todas las promociones que ingresan a las dependencias de la A.G.J.I.
- Evitar la alteración de la secuencia cronológica con que cada asunto o promoción es recibido.
- Facilitar el conocimiento del estado de cualquier asunto en todo momento.
- Identificar el vencimiento de los asuntos de acuerdo a su período de conclusión predefinido.
- Permitir identificar y prever los asuntos vencidos.
- Medir y evaluar por individuo o área la productividad.

Como una breve descripción de Control de Gestión, se mencionará que en él se registra toda la documentación ingresada a la A.G.J.I., a través del área llamada Oficialía de Partes.

Oficialía de Partes la registra, captura y clasifica las promociones recibidas, además le asigna un número de folio consecutivo a cada una para llevar el control de las promociones dentro del área técnica, este mismo número se le entrega al contribuyente para que a través de éste, él se informe de su asunto.

---

[1] Promoción.- Toda aquella petición ingresada en la A.G.J.I., que requiera una respuesta por parte de la misma ya fuera esta escrita o por medio de algún tipo de acción.

Cuando las áreas que deben dar solución reciben las promociones de parte de Oficialía de Partes, registran que las recibieron dentro del sistema y a partir de ahí empieza el estudio técnico de la promoción. A partir del ingreso de la promoción en Oficialía de Partes el tiempo de respuesta predefinido empieza a registrarse hasta que el área da una solución a la promoción. Durante ese lapso, la promoción pasa por ciertas etapas de estudio para las áreas, y cada vez que cambia de etapa la promoción se debe registrar dicha información en el sistema.

Si la respuesta a la promoción era a través de un oficio, éste se turnaba a Oficialía de Partes, que a su vez registra la salida en la base de datos.

Como todas las etapas debían se registradas, el sistema es la fuente de información más fidedigna y veloz para enterar al contribuyente de su promoción, tarea que realizaba el área de Mesa de Trámite, pero su programa de consulta solo muestra una parte de la información capturada, para evitar que el contribuyente se entere de que abogado maneja su caso y así evitar la corrupción.

Otra de las funciones del sistema, es la entrega de productos útiles para las áreas mencionadas, como estadísticas de control y reportes de supervisión.

Durante la práctica con el sistema al inicio de su operación, se le detectaron diversos errores de programación y diseño, por lo que se llegó a la conclusión de realizar una reprogramación de ciertas secciones del sistema, mientras la versión de la prueba piloto continuaba en actividad.

Al empezar la reprogramación, se solicitó la documentación del diseño y análisis del sistema, pero no existía, por lo que se requirió que se llevara a cabo, pero las instrucciones recibidas fueron simplemente reprogramar el sistema basados en los programas de la prueba piloto. Esta nueva versión contaba con los cambios requeridos por las áreas que usaban la aplicación, como el de poder medir la productividad al personal a nivel individual. Pero lógicamente el proceso tardó mucho más de lo debido, por no tener el diseño bien estructurado.

El sistema se dividió en módulos para un mejor entendimiento de la programación y utilizar programas comunes a otras partes del mismo.

Una vez liberada la versión nueva del sistema, solo se le realizaron los cambios necesarios en la programación como para interactuar con los otros módulos que serían desarrollados posteriormente.

Control de Gestión es el módulo que hasta la fecha, funcionando a nivel nacional, es el encargado de reportar la productividad de las áreas que abarca, de las Administraciones Locales. Su diagrama entidad-relación se muestra a continuación en la figura 4.2.1

DIAGRAMA ENTIDAD-RELACIÓN DEL SUBSISTEMA DE CONTROL DE GESTIÓN

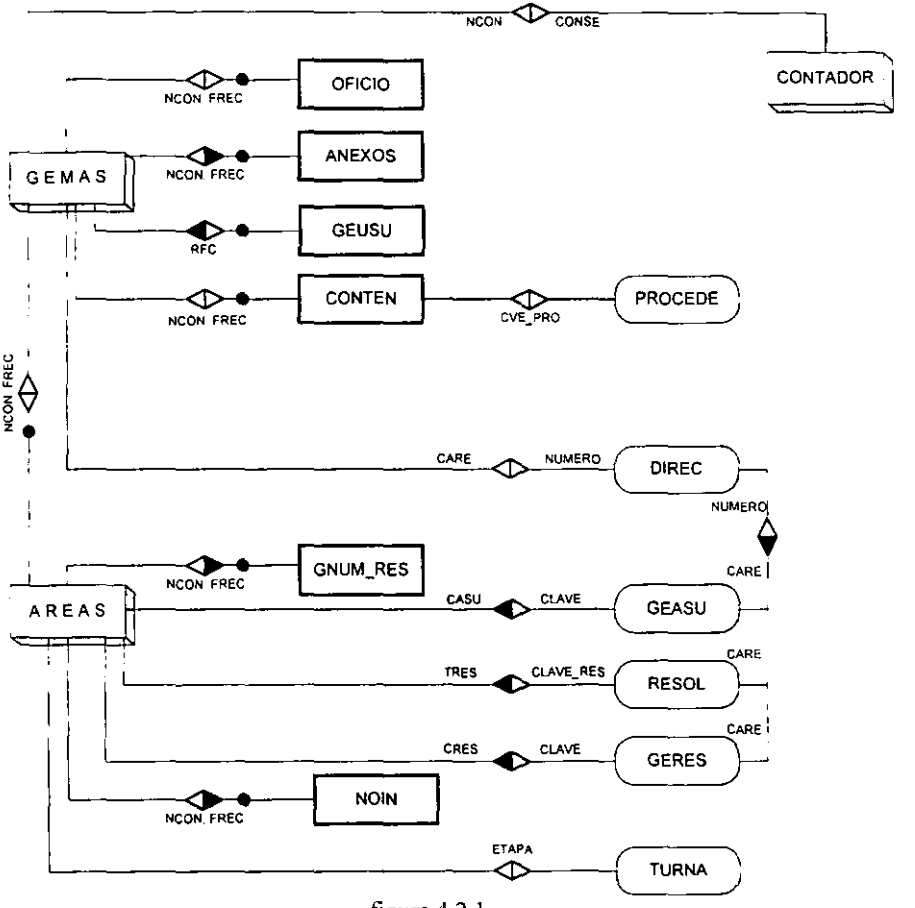


figura 4.2.1

4.2.2. Notificación y Cobranza

Al cambiar de nombre el área jurídica, de Dirección General Técnica de Ingresos al de Administración General Jurídica de Ingresos, le fueron añadidas nuevas funciones las cuales tenían que ser integradas al S.I.I.T. a la brevedad posible.

El área más importante que se agregó fue la de Notificación y Cobranza, por lo que el incluirla dentro del sistema resultó ser de la más alta prioridad.

Cuando alguna Administración de Auditoría (que también se divide en centrales, regionales y locales) realizaba un estudio de los ejercicios fiscales de algún contribuyente, ya fuese este moral o físico, y de este estudio se desprendiera un adeudo con la Secretaría de Hacienda y Crédito Público, la Administración de Auditoría genera una resolución, dentro de la cual fundamenta el adeudo. Esta resolución entonces se turna a la Administración Jurídica de Ingresos correspondiente, para que ésta controle el adeudo, además de notificar al contribuyente del mismo.

El contribuyente al enterarse del adeudo tiene de plazo 45 días para poder realizar cualquiera de la siguientes acciones :

- Pagar el adeudo ya sea totalmente o en parcialidades.
- Interponer un medio de defensa ante un autoridad competente, las cuales pueden ser Recursos de Revocación, Juicios de Nulidad, y algunos otros, con esta opción debe de dejar algún tipo de garantía por el monto del adeudo ante la Secretaría de Hacienda.
- No pagar y no interponer ningún medio de defensa.

En caso de que no se hubiere reportado ningún pago o si interpuso un medio de defensa pero no dejo la garantía correspondiente, el área de Notificación y Cobranza procede a embargar al contribuyente o intervenir su negocio en caso de que éste lo tenga.

Cada uno de estos pasos tiene infinidad de implicaciones, como por ejemplo, las resoluciones tienen un tipo de clasificación donde se capturan datos diferentes, los motivos de la generación del adeudo pueden ser muchos, y además los adeudos generaban un tipo de interés y recargos basados en el Índice Nacional de Precios al Consumidor y dependiendo del tipo de impuesto, multas o recargos del que se trate.

Los tipos de notificaciones son: personal, estrados o edictos. Mientras que los llamados incidentes, que no es otra cosa que la acción que el contribuyente siguió dentro de los 45 días después de que se la notificó que tiene un adeudo, se clasifican por pagos, medios de defensa y garantías.

Los embargos implican 3 tipos de remates, además de la intervención a un negocio que significa el retenerle el 10 % de sus ganancias hasta que pague el monto total.

Además de registrar a los contribuyentes no encontrados, a los que se les embargó y después pagaron, llamado pago espontáneo, etc. Su diagrama entidad-relación se muestra en la figura 4.2.2

Actualmente, la función de Notificación y Cobranza fue entregada a la Administración General de Recaudación, por lo que el S.I.I.T. ya no la incluye dentro de su operación.

DIAGRAMA ENTIDAD-RELACIÓN SUBSISTEMA NOTIFICACIÓN Y COBRANZA

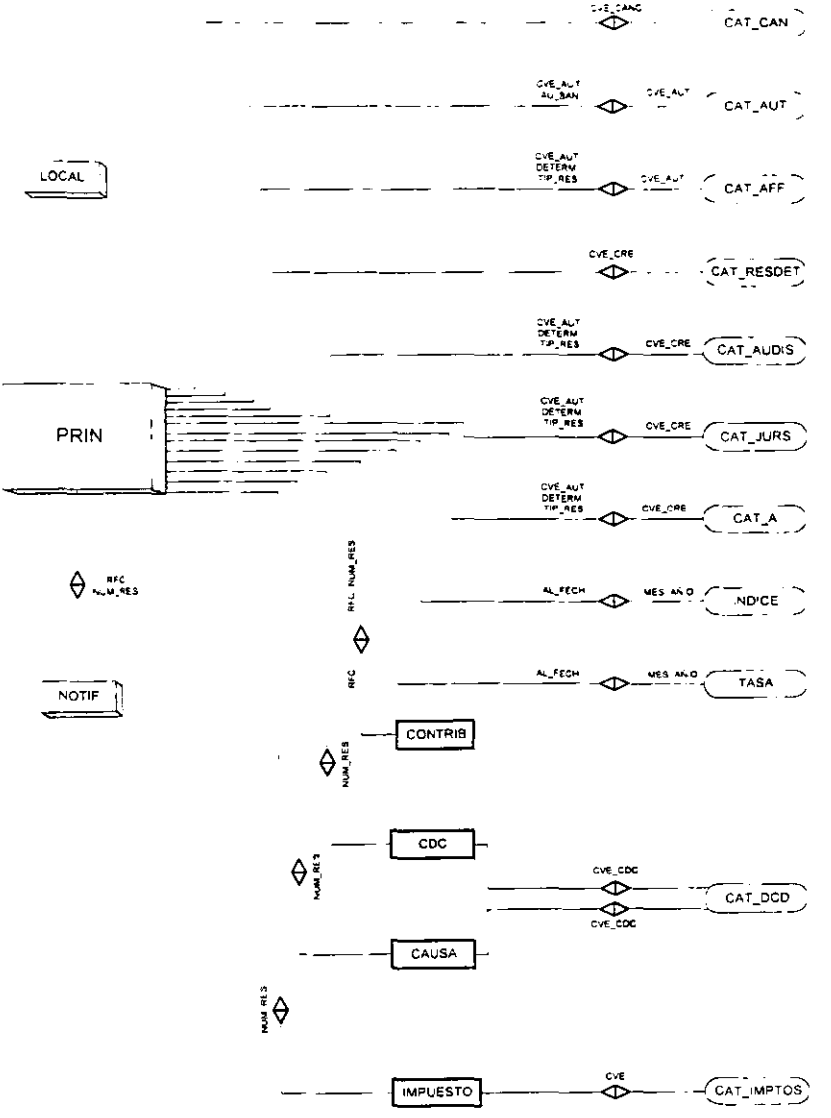


figura 4.2.2

4.2.3. Contencioso

Al terminar el sistema de información de Notificación y Cobranza, se inició un nuevo módulo del S.I.I.T. al que se le llamó Contencioso.

El sistema de información de lo Contencioso pretendía abarcar un medio de defensa del contribuyente, para nulificar una resolución que Auditoría le determinó.

A grandes rasgos se trata de explicar en los siguientes párrafos.

El contribuyente recibe la notificación de que tiene un adeudo con la S.H.C.P., como se explicó en el subtema anterior, puede interponer un medio de defensa, en primera instancia están los medio de defensa de la propia S.H.C.P., a los cuales se les denomina Recursos de Revocación, donde gente que trabaja para la Secretaría investiga el caso (esto se controla en el sistema de Control de Gestión). Al llegar a una resolución el área que maneja los recursos y ésta llega a ser desfavorable al contribuyente, si éste aún no está de acuerdo con la determinación, expone su caso ante el Tribunal Fiscal de la Federación levantando una demanda contra la S.H.C.P.

El área de Oficialía de Partes del T.F.F. estudia la documentación entregada por el contribuyente (a quien a partir de ese momento se le llama Actor) para levantar una demanda, si es el caso notifica a la autoridad (S.H.C.P.) que ha decidido aceptar la demanda. A partir de esta notificación, la S.H.C.P. registra el juicio en el sistema de información de Contencioso, capturando todo lo que a continuación se describe.

Los abogados del T.F.F. estudian a fondo la documentación entregada por el actor, para determinar si desechan o aceptan la demanda. Además en ese paso el actor puede incluir información que no incluyó en la documentación original, por lo que los motivos de la demanda pueden aumentar.

Cuando el T.F.F. acepta estudiar la demanda para dar una solución, examina los documentos entregados por el actor y por la autoridad, enterándoles a ellos y solicitándoles más documentación en caso de ser necesario. Durante ese período se llevan a cabo los incidentes, que son las acciones que determinó el T.F.F. seguir, así como las pruebas presentadas por el actor ante el mismo.

Cuando el T.F.F. emite su sentencia con respecto a la demanda, si el actor o la autoridad no están de acuerdo, acuden a una instancia superior del poder judicial (Suprema Corte de Justicia), donde el actor puede pedir un amparo o la autoridad un recurso de queja, aquí nuevamente empieza el estudio de su caso, presentando pruebas, hasta llegar a una resolución la cual es irrevocable.

Cada uno de estos pasos es contemplado en el sistema de Contencioso, capturando infinidad de fechas de sentencias, de notificaciones, de amparos, de presentación de pruebas, de presentaciones de peritos, sentidos de las sentencias, etc.

El sistema al igual que el de Notificación y Cobranza, salió en su primera versión sin reportes, por lo que nuevamente el rechazo hacia él por parte de los usuarios fue difícil de tratar. Pero además este sistema implicaba captura de campos que para los usuarios son innecesarios, lo que causo infinidad de repercusiones para su operación. Diagrama entidad-relación en la figura 4.2.3

DIAGRAMA ENTIDAD RELACIÓN DEL SUBSISTEMA DE LO CONTENCIOSO

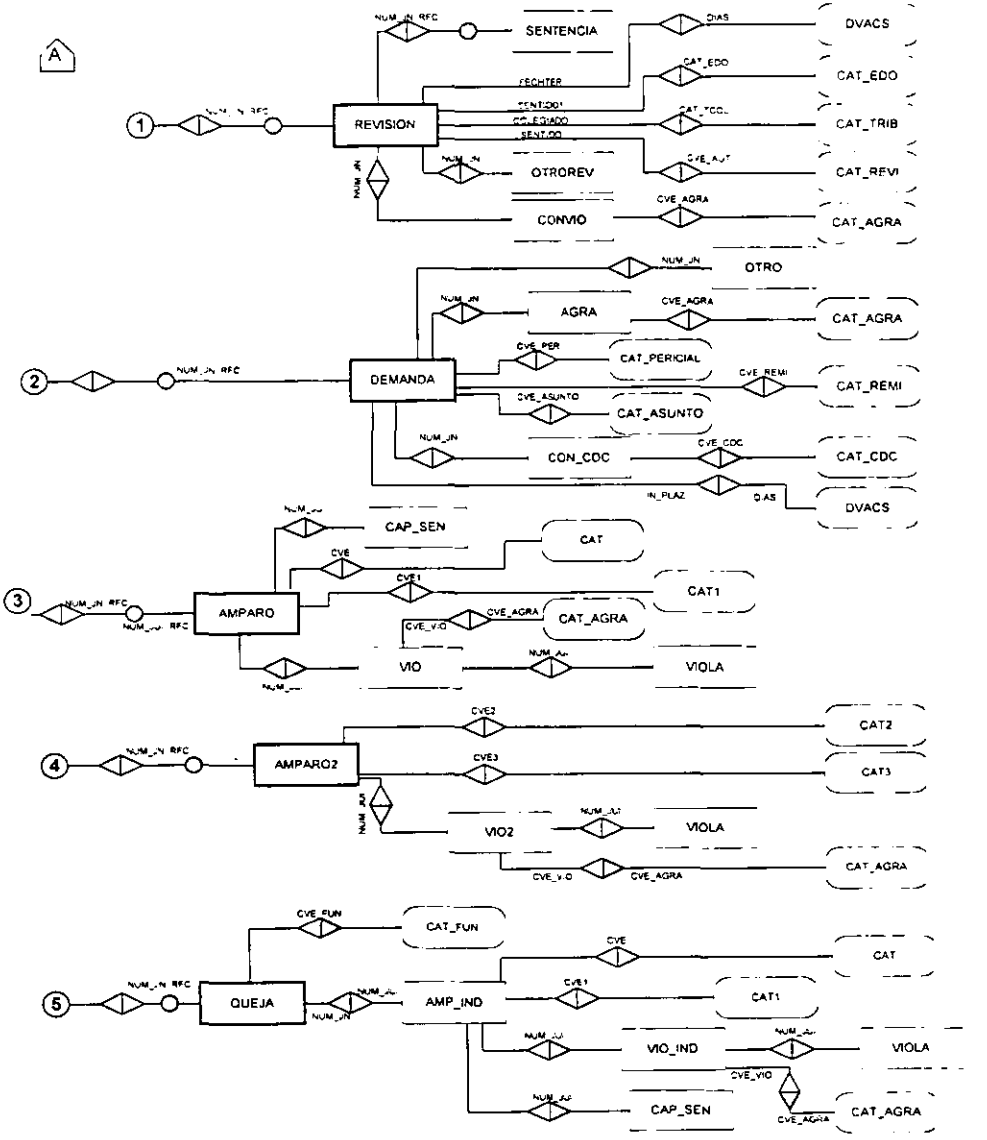


figura 4.2.3

4.3. Deficiencias de Seguridad de la Aplicación

Las deficiencias comienzan, a nivel central, con la estructura funcional de la Administración de Informática. Las Subadministraciones de Diseño, Desarrollo y Explotación de Sistemas trabajan conjuntamente en el S.I.I.T., pero sin restricciones acerca de los programas fuentes de la aplicación. Usualmente ingresan con el mismo usuario (siit) que es el propietario del código y las formas que utiliza y por lo tanto están habilitados para modificar o borrar cualquier archivo.

Otra deficiencia, es que el password de root es utilizado por varios usuarios, por lo que sin el control adecuado, se han suscitado problemas graves de operación en el servidor de Desarrollo del S.I.I.T. Además, cuando alguna empresa externa precisa de realizar alguna prueba o soporte en la máquina se les proporciona de igual manera.

No se tiene ningún sistema de monitoreo o seguridad habilitados en este equipo, ni se lleva un registro adecuado de su funcionamiento. Se crean cuentas temporales (para pruebas) y se dejan ahí por tiempo indefinido, lo cual es una puerta falsa de entrada para un atacante.

A nivel nacional, los programas ejecutables, las formas y las bases de datos tienen permisos de escritura, lectura y ejecución por lo que cualquier usuario en el sistema y excepto los que entren con un shell restringido, todos tienen acceso a ellos. La cuestión del soporte se repite: la contraseña del superusuario es proporcionada a personal ajeno a la A.G.J.I., y como no se ha fomentado suficientemente la cultura Unix a este respecto, los Administradores del S.I.I.T. no solo lo entregan, sino que además dejan en completa libertad de acción a estas personas. Cuando surge algún problema, ellos ni siquiera saben que fue lo que lo ocasionó porque no estuvieron presentes en ese momento.

CAPÍTULO 5

IMPLANTACION DEL SISTEMA DE SEGURIDAD PARA EL SISTEMA INTEGRAL DE INFORMACIÓN TRIBUTARIA

5.1. Implantación de Seguridad Física

La interrupción del funcionamiento de un centro de cómputo puede ser causada por diversas situaciones: ocurrencias de desastres tales como: pérdidas de información, incendios, inundaciones y fallas de electricidad son las más comunes. Ejemplos de otros desastres incluyen sabotaje, descuido y disturbios civiles. La probabilidad de que ocurran tales eventos puede ser reducida a través de un efectivo programa de seguridad. De este programa depende cualquier empresa y no nada más la S.H.C.P.

Hoy en día es claramente palpable que todas las empresas dependen en gran medida de la información que alojan sus centros de cómputo, lo que ha provocado que, en lugar de exponer sus recursos informáticos como se venía haciendo anteriormente ahora se deban de proteger.

Ante la preocupación por la posible pérdida de información que pudiesen sufrir las instituciones, han optado simplemente por planear "un buen sistema de respaldos". Sin embargo, de esta manera lo que se resuelve es recuperar sistemas y datos actualizados (si el plan es bueno) . No obstante nunca nos preguntamos ¿cómo restablecer la operatividad de los sistemas automatizados que son necesarios para el funcionamiento de la empresa?

Existe una gran diferencia entre la reparación de los datos y la recuperación de la operación de los sistemas automatizados. La primera simplemente puede ser atacada con un plan de respaldos, y la segunda involucra aún más; elaborar un plan de recuperación de operaciones mediante la cual, la empresa reanude sus funciones posterior a la ocurrencia de algún desastre.

Mencionado lo anterior, este tema tiene como objetivo, señalar que medidas de seguridad pueden ser llevadas a cabo premeditadamente para reducir la probabilidad de pérdida de información o reducir el impacto posterior que pudiese suscitarse debido a la ocurrencia de algún desastre.

Es nuestro deber informar que un plan de contingencias puede ser desarrollado tan detallada y escrupulosamente se requiera, dependiendo de la filosofía de la empresa y las facilidades que brinde la administración de la misma.

Para abordar el presente, se ilustra en la tabla 5.1, las entidades que se deben de proteger, de los diversos factores internos y/o externos, se listan algunas alternativas y requerimientos propuestos; indispensables en la S.H.C.P. y una última columna para expresar la factibilidad de llevar a cabo lo propuesto.

MEDIDAS DE SEGURIDAD

¿Qué se debe proteger?	¿Contra qué se requiere proteger?	Alternativas
El Inmueble	♦ Manifestaciones civiles ♦ Catástrofes naturales	✓ Site Externo ✓ Equipo de protección civil
El Centro de Cómputo	♦ Intrusos ♦ Accidentes ♦ Ladrones ♦ Medio ambiente	✓ Sistemas biométricos ✓ Circuito cerrado ✓ Extintores y aire acondicionado ✓ Alarmas
El Hardware	♦ Fallas técnicas o/y operacionales ♦ Robo	✓ Mantenimiento correctivo y preventivo ✓ Inventario ✓ Temperatura y ambiente
El Software	♦ Piratería ♦ Infecciones por virus ♦ Alteraciones en la configuración original	✓ Usar solo versiones formales ✓ Revisar cintas magnéticas ✓ Respaldar configuración de herramientas de operación y desarrollo
La Información (S.I.I.T.)	♦ Integridad ♦ Privacia ♦ Perdidas ♦ Virus ♦ Robo	✓ Utilerias del S.O. ✓ Utilerias de informix (SE) ✓ Programa auditor ✓ Respaldos de información ✓ Antivirus ✓ Accesos y permisos
Del Personal	♦ Deshonestidad ♦ Incomformidades ♦ Negligencia	✓ Selección ✓ Entrevistas ✓ Capacitación

tabla 5.1

Las políticas para la contratación, rehubicación, terminación de personal y la capacitación del mismo, son celosamente llevadas a cabo por el área de Recursos Humanos de la propia Secretaria. Sin embargo las recomendaciones y alternativas para el correcto empleo o destitución de personal fueron ya mencionadas a lo largo del capitulo 1.

5.1.1. Protección del Edificio

La S.H.C.P., se encuentra expuesta a frecuentes manifestaciones de inconformidades sociales, debido a su naturaleza y ubicación; incluso tiene como vecina a la P.G.R. que también se encuentra expuesta a tales movimientos.

En el edificio de la Secretaria de Hacienda y Crédito Público (ubicado entre Av. Reforma y Av. Hidalgo), a la fecha solo se cuenta con personal de vigilancia en las diferentes entradas del inmueble, y en ocasiones de conflicto refuerzan al personal en las mismas.

La seguridad propia de esta Secretaria, registra bolsos a la entrada y salida mediante un sistema rastreador computarizado provisto de monitores y otro dispositivo detector de metales por donde atraviesa toda persona que labora o visita las instalaciones. Los empleados federales deben portar en todo momento su credencial que los acredita como tales y los visitantes deben registrarse a la entrada, de lo contrario se les prohibe el acceso.

Adicionalmente el personal de vigilancia visita las instalaciones periódicamente a fin de rastrear bombas con detectores de metal y gases.

En los estacionamientos: El personal muestra pase de entrada que acredite que esta adscrito a la Secretaria de Hacienda, se revisa la cajuela y la parte inferior del automóvil con bastones-espejo. A la salida se realiza el mismo procedimiento.

Todas las tareas mencionadas para brindar seguridad pueden completarse con la instalación de un *site* externo. Es una alternativa más que no solamente resguardaría información sino que además brinda la posibilidad de trabajar en él cuando las instalaciones sean inoperables.

El *site* externo fungiría principalmente como almacén y como centro de operación en caso de situaciones críticas.

En el *site* se deben guardar:

- Hardware mínimo necesario para operar la aplicación.
- Respaldos de los programas, sistemas y software de la base de datos en producción.
- Documentación requerida para restaurar los sistemas e información; esto incluye inventarios, manuales técnicos y de usuario, formas impresas, etc.

Cabe también la posibilidad de instalar cámaras de vídeo en lugares estratégicos dentro de la sala de servidores.

5.1.2. Centro de Cómputo

En el siguiente plano (figura 5.1.2), se muestra la ubicación física de los servidores Unix que contienen información del S.I.I.T., tanto en su etapa de producción como en la de explotación.

Ubicación de la Sala de Servidores

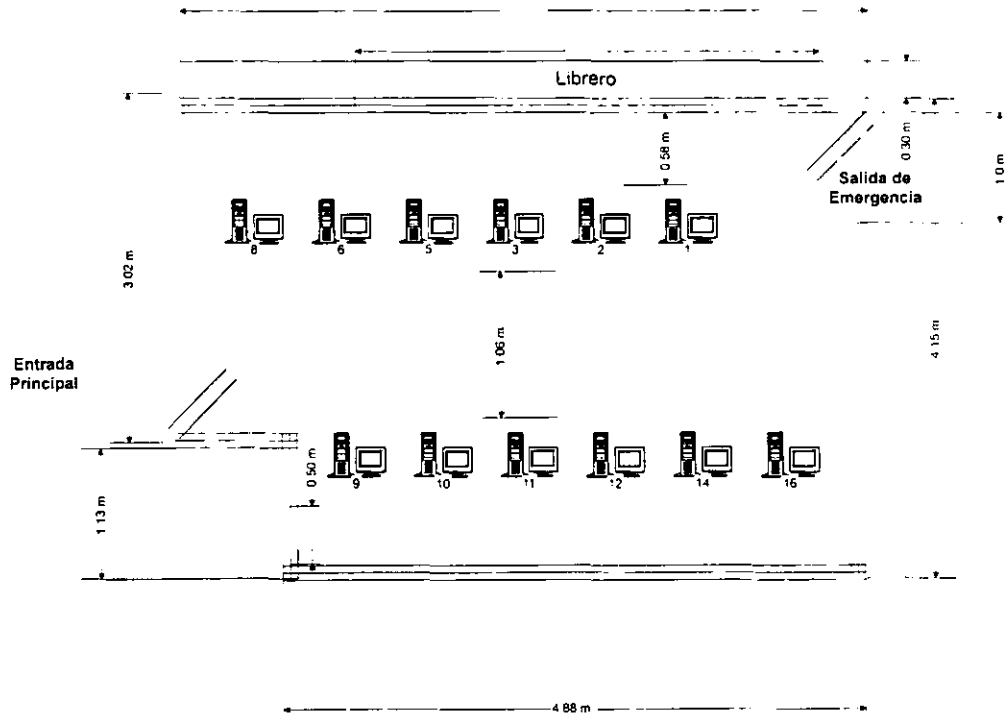


figura 5.1.2

Es importante mencionar que no existe equipo preventivo contra fuego, inundaciones, etc., en este centro de cómputo. Por lo cual, se propone adquirirlo y tomar en cuenta las recomendaciones del capitulo 1.

5.1.3. Protección al Hardware

Es necesario tener un inventario completo y al día del total del equipo concentrado en el área. El registro del equipo se lleva a cabo por el área de soporte informático.

El equipo usado para el S.I.I.T., se lista en el capitulo 4 subcapitulo 4.7.

Equipo de Comunicaciones: El equipo de comunicaciones que hace uso el S.I.I.T., es propiedad de la empresa externa (ISOSA), ésta brinda por completo la administración del servicio a la S.H.C.P.

Plan de mantenimiento. El mantenimiento preventivo así como el correctivo es proporcionado por los proveedores (UNISYS, HP, NCR y CEPRA), según su propio calendario.

Ambiente: Las instalaciones cuentan con aire acondicionado que es mantenido y regulado por el personal de mantenimiento del inmueble.

El responsable o autoridad del sistema periódicamente, debe de recibir un reporte de los factores antes citados; que podrían provocar en determinado momento problemas a la seguridad en el hardware. De esta manera la autoridad podrá tomar alguna determinación cuando así sea necesario.

5.1.4. Protección al Software

El software utilizado o relacionado en el S.I.I.T. , es el siguiente:

- S.O. UNIX System V reléase 4.0
- Informix 4gl

- Informix SE
- X.25
- TCP/IP

El sistema operativo UNIX no puede ser atacado por virus, puede sufrir ataques por programas como caballos de troya, que solo pueden ser detenidos con la implantación de la seguridad a nivel S.O. (consultar este mismo capitulo en el punto 5.2), debido a que no existen vacunas para los caballos de troya.

Una alternativa efectiva y funcional, es mantener las versiones originales de cada componente de la lista de software y un respaldo de la configuración adecuada para cada uno.

Tener a la mano las versiones originales es de gran utilidad, cuando se necesitara crear otra estación de trabajo o restablecer la operación en alguna otra.

5.1.5. Protección de la Información

Como ya se mencionó para que la información sea segura no debe perder su integridad ósea, debe estar completa y ser fiable, también ser mostrada únicamente a quien se debe y procesada por el usuario correcto.

Con el manejo de passwords y accesos de los usuarios que intervienen en su flujo del esquema S.I..I.T. se logra gran parte del objetivo.

Con la creación del Sistema Auditor, que tiene entre otras metas descifrar los archivos audits, estará fortaleciendo el objetivo de otorgar seguridad a la información. Se podrá saber si fueron correctamente manipulados los datos en sus principales tablas de la Base de Datos.

Se propone seguir respaldando la información mensualmente o respaldar con periodos más cortos cuando se ha detectado mayor carga de información en la Base de Datos.

Las cintas resultantes del respaldo, deben ser etiquetadas correctamente con fecha y títulos. No olvidando tener un respaldo del código fuente y del archivo ejecutable de las dos ultimas versiones.

5.2. Implantación de las características de Seguridad del Sistema Operativo Unix System V

En base a lo expuesto en el capítulo 2, se procederá a implantar la seguridad para el S.I.I.T. a nivel Sistema Operativo. La seguridad en este rubro comienza con la creación de grupos de trabajo y es seguida de la creación de cuentas, planeando la creación de contraseñas apropiadas para cada cuenta, instruyendo al usuario acerca de como cambiarla periódicamente, siguiendo las recomendaciones que fueron explicadas en el punto 2.2.2, ya que es mejor invertir cierto tiempo en esta actividad, que asignar contraseñas que corren el riesgo de ser quebrantadas.

Todas las instrucciones y recomendaciones siguientes van dirigidas al Administrador del Equipo donde reside el S.I.I.T., ya sea en su fase de desarrollo (como sucede en la Administración de Informática) o en su fase de explotación (como es el caso a nivel nacional). Aunque estas especificaciones funcionan para este caso particular, pueden ser implantadas en cualquier equipo con el mismo sistema operativo.

5.2.1. Asignación de permisos predeterminados

Los permisos en los archivos y directorios pueden ser controlados predeterminadamente en la creación de todo archivo o directorio en el sistema, asignando un valor que definirá los permisos que tendrá aún sin que el usuario propietario los modifique.

El valor de protección discreta establecida de todo el sistema se inserta en el archivo **etc/profile**, modificando el valor de la variable **umask**. Cuando el sistema operativo es liberado, la Seguridad Comercial del mismo tiene un **umask** de 022, que transfiere una protección predeterminada de lectura, escritura y ejecución para todos los usuarios no privilegiados. Cada usuario recibe el **umask**, a menos que lo cambien. Cambiarlo a un valor menor de seguridad no es recomendable.

El asignar el valor 077 que proporciona protección de lectura, escritura y ejecución para los objetos del usuario es altamente recomendable.

Para lograr lo anterior, se edita el archivo /etc/profile como superusuario y se declara la variable igual a 077.

5.2.2. Duración de los passwords

Como ya se mencionó en el capítulo 2 los valores para el mínimo y máximo tiempo de vida de las contraseñas son insertados como nulos predefinidamente bajo System V Release 4. Se modifican editando los valores MINWEEKS y MAXWEEKS en el archivo **/etc/defautl/passwd**. Utilizando el comando **passwd**, el superusuario puede insertar diferentes valores para usuarios individuales. Un usuario puede ser forzado a cambiar su password (simplemente haciendo que expire) o prevenirlo para que lo haga. En System V Release 4, el comando :

 # passwd -x maxdias -n mindias login

Es utilizado para hacer lo anterior. Si mindias es mayor que maxdias, el usuario es advertido a cambiar siempre su password. Si maxdias es igual a -1, la duración del password esta deshabilitada para ese usuario. El comando:

 # passwd -f login

es utilizado para marcar el password de ese login como expirado, forzando al usuario a cambiarlo la próxima vez que entre a su cuenta. El comando:

 # passwd -w numdias login

inserta el número de días antes de la expiración del password para comenzar a avisar al usuario de su próxima expiración.

El comando:

 # passwd -s login

despliega la información respecto al password en el siguiente formato:

```
# passwd -s root
root  PS  08/06/96  0  168  7
# 
```

login	status passwd	fecha del último cambio	mindias	maxdias	warning
root	PS	08/06/96	0	168	7

Para que esta información incluya todas las cuentas del sistema, se ejecuta:

 # passwd -a -s

```
root  PS  01/08/97  0  168  7
daemon  LK
bin  PS  02/13/96  0  168  7
sys  LK
syss  PS  02/11/97  0  168  7
adm  LK
uucp  LK
lp  LK
nuucp  LK
listen  LK
sync  LK
slip  LK
mlsadmin  PS  02/13/96  0  168  7
install  PS  02/13/96  0  168  7
sysadm  PS  12/06/96  0  168  7
makefsys  PS  12/06/96  0  168  7
checkfsys  PS  12/06/96  0  168  7
mountfsys  PS  12/06/96  0  168  7
umountfsys  PS  12/06/96  0  168  7
setup  PS  12/06/96  0  168  7
vmsys  LK
oasys  LK
```

El archivo password estándar de UNIX contiene el password encriptado de cada uno de los usuarios. Este archivo debe estar habilitado para lectura y escritura para varios comandos no privilegiados (como **ls**) para trabajar. Desafortunadamente, esto significa que cualquier intruso que haya ganado el acceso a su sistema puede copiar el archivo password de su máquina e intentar quebrantar los passwords. Lo que comenzó como un simple quebranto de una cuenta puede rápidamente volverse un problema severo debido a que solo algunos comandos (**login, passwd, su...**)

necesitan ver el password encriptado y todos esos comandos son privilegiados, a los usuarios no privilegiados no debe permitírseles obtener los passwords encriptados de otros usuarios.

5.2.3. Implantación de la Seguridad en los Accesos

Los accesos remotos y locales como usuario root pueden ser detenidos de una forma bastante sencilla.

Editando el archivo **/etc/default/login** a la variable **ROOTLOGIN** se le asigna un valor igual a **NO**, de esta manera, para tener privilegios de superusuario en el sistema, se tendrá que ingresar al mismo con una cuenta ordinaria y después teclear el siguiente comando:

S/bin/su

Nunca invoque a root simplemente con su, ya que un programa residente en su ruta con el mismo nombre podría ser un Caballo de Troya que robe el password del superusuario.

En el caso de los accesos remotos, si no conocen la contraseña de una cuenta cualquiera en el sistema, no tendrán acceso a root ni siquiera conociendo su contraseña.

En el mismo archivo, se declarará la variable **CONSOLELOG** igual a **YES**, de esta forma, los intentos de entrada al sistema fallidos son reportados a la consola, así como aquellos en que un usuario ordinario intente y/o logre convertirse en superusuario. Si los intentos son desde equipos remotos, aparecerá su alias y su dirección IP. Si se trata de equipos PC's de la red local, solo su IP y en todos los casos, la cuenta del usuario implicado.

5.2.4. Monitoreo

La mayoría de los UNIX modernos registran la última vez que cada usuario acceso al sistema, usualmente en el archivo **/usr/adm/lastlog**. Esta ocasión es impresa como parte del proceso de entrada al sistema y que usualmente se ve como las líneas que aparecen abajo:


```
UNIX(r) System V Release 4.0 (ses35)

login: cynthia
Password:
Last login: Mon Feb 10 17:18:08 from 99.96.20.204
 UNIX System V/386 Release 4.0 Version 2
ses35
Copyright (C) 1984, 1986, 1987, 1988, 1989, 1990 AT&T
Copyright (C) 1990, 1992 UNIX System Laboratories, Inc.
Copyright (C) 1987, 1988 Microsoft Corp.
All Rights Reserved
Copyright (C) 1987, 1988, 1989, 1990, 1991, 1992, 1993 Unisys Corp.
All Rights Reserved

$
```

Los usuarios deben ser entrenados para que cuidadosamente examinen esta línea cada vez que ingresen a su cuenta, y para reportar entradas inusuales desde equipos remotos desconocidos al Administrador del Sistema. Esta es una manera sencilla de detectar cuentas que han sido comprometidas, debido a que cada usuario debe recordar la última vez que utilizó su cuenta.

El archivo **/etc/utmp** es utilizado para registrar quien está actualmente accesando al sistema. Este archivo puede ser desplegado utilizando el comando **who**:

 # who

```
siit        term/108      Sep 19 18:31
daniel      pts/0         Sep 19 18:51
siit        pts/1         Sep 19 20:55
siit        pts/2         Sep 19 21:05
```

Por cada usuario, el nombre login, la terminal utilizada y el tiempo de entrada son desplegados. Si el usuario ha entrado remotamente a través de la red de datos, el nombre del equipo remoto del cual procede también es desplegado.

En algunos sistemas, este archivo tiene permisos para todos los usuarios. Esto puede ser un problema de seguridad en muchas formas. Primero, permite a un atacante borrar la entrada que lo muestra dentro del sistema, esto es para "esconderse" del Administrador del Sistema. Un problema mayor es que el atacante puede cambiar los nombres de los dispositivos de las terminales en los archivos por lo que la próxima vez que un programa como **wall** o **comsat** sea ejecutado, escribirá a algún otro archivo de dispositivo.

El archivo **/usr/adm/wtmp** (etc/wtmp en algunos sistemas) registra cada entrada y salida de los usuarios, tiene el mismo formato que el anterior y puede ser también desplegado con el comando who:

 # who /etc/wtmp

root	console	Ago 28	07:48
root	console	Ago 28	17:03
luis	ttyp0	Ago 28	13:27 (dti1)
luis	ttyp0	Ago 28	21:17
siit	ttyp3	Ago 28	07:48

Una línea que contiene un nombre de cuenta indica la hora en que entró el usuario, una línea sin nombre indica el tiempo en que fue abandonada la terminal. El comando **last** muestra las entradas en el archivo **wtmp**, coincidiendo los nombres de las cuentas con sus tiempos de entrada y salida. Sin argumentos **last** despliega el archivo entero, con un nombre usuario o una terminal como argumento, la salida puede ser restringida al usuario o a la terminal en cuestión. Este comando siempre despliega su salida en orden inverso de la más reciente entrada a la menos reciente:

 # last

```
root     pts/0        hp9              Mon Sep  9 19:40   still logged in
.telnet  pts/0                         Mon Sep  9 19:40 - 19:40  (00:00)
root     pts/0        hp9              Mon Sep  9 19:40 - 19:40  (00:00)
root     pts/0        hp9              Mon Sep  9 19:15 - 19:40  (00:24)
.telnet  pts/0                         Mon Sep  9 19:15 - 19:15  (00:00)
daniel   term/107                      Mon Sep  9 19:13
siit     term/108                      Mon Sep  9 19:12
siit     term/108                      Mon Sep  9 19:12 - 19:12  (00:00)
.ttymon1 term/108                      Mon Sep  9 19:12 - 19:12  (00:00)
siit     term/108                      Mon Sep  9 18:46 - 19:12  (00:26)
ttymon   console                       Mon Sep  9 18:43
daniel   console                       Mon Sep  9 18:43 - 18:43  (00:00)
daniel   console                       Mon Sep  9 18:06 - 18:43  (00:36)
daniel   term/107                      Mon Sep  9 17:44 - 19:13  (01:29)
.ttymon1 term/107                      Mon Sep  9 17:44 - 17:44  (00:00)
daniel   pts/0        99.96.20.221     Mon Sep  9 16:27 - 19:15  (02:48)
.telnet  pts/0                         Mon Sep  9 16:27 - 16:27  (00:00)
siit     term/107                      Mon Sep  9 15:21 - 17:44  (02:22)
siit     pts/0        99.96.20.230     Mon Sep  9 14:38 - 16:27  (01:48)
siit     pts/0        99.96.20.230     Mon Sep  9 14:38 - 14:38  (00:00)
.telnet  pts/0                         Mon Sep  9 14:37 - 14:38  (00:00)
root     pts/0        99.96.20.220     Mon Sep  9 14:20 - 14:37  (00:16)
```

Todos estos archivos y comandos deben ser utilizados para monitorear la seguridad de los accesos. La siguiente tabla los muestra en forma condensada:

Comandos Utilizados para el Monitoreo

Archivo	Comando	Aplicación	Utilización
/usr/adm/lastlog		Registra la última entrada de los usuarios al sistema.	Diaria
/etc/utmp		Registra quién está actualmente accesando al sistema	No aplica
	who	Despliega la información contenida en /etc/utmp	Cada media hora
who /etc/wtmp		Registra cada entrada duración de la sesión y salida de los usuarios	Diaria
last		Muestra las entradas en el archivo **wtmp**, coincidiendo los nombres de las cuentas con sus tiempos de entrada y salida	Diaria

tabla 5.2.4

5.2.5. Activación, Configuración y Administración del Audit Trail

Como se menciona en el capítulo 2, el **audit trail** es creado en la transición del nivel monousuario al multiusuario. Para activarlo, se ingresa como superusuario al sistema y se edita el archivo **/etc/default/audit** con el siguiente comando:

 # vi /etc/default/audit

El script despliega la siguiente información:

```
AUDIT=YES
DIRECTORY=/var/sat
NUMBER=10
BLOCKS=50000
RECORD=173177757
DISPLAY=0000140001
FILE0=/var/sat/audit0
FILE1=/var/sat/audit1
FILE2=/var/sat/audit2
FILE3=/var/sat/audit3
FILE4=/var/sat/audit4
FILE5=/var/sat/audit5
FILE6=/var/sat/audit6
FILE7=/var/sat/audit7
FILE8=/var/sat/audit8
FILE9=/var/sat/audit9
~
~
~
```

La variable **AUDIT** debe estar declarada como **YES**, de lo contrario utilice el editor para asignarle ese valor. Una vez activado, el siguiente paso es configurarlo, para lograrlo se invoca al programa sysadm desde línea de comando:

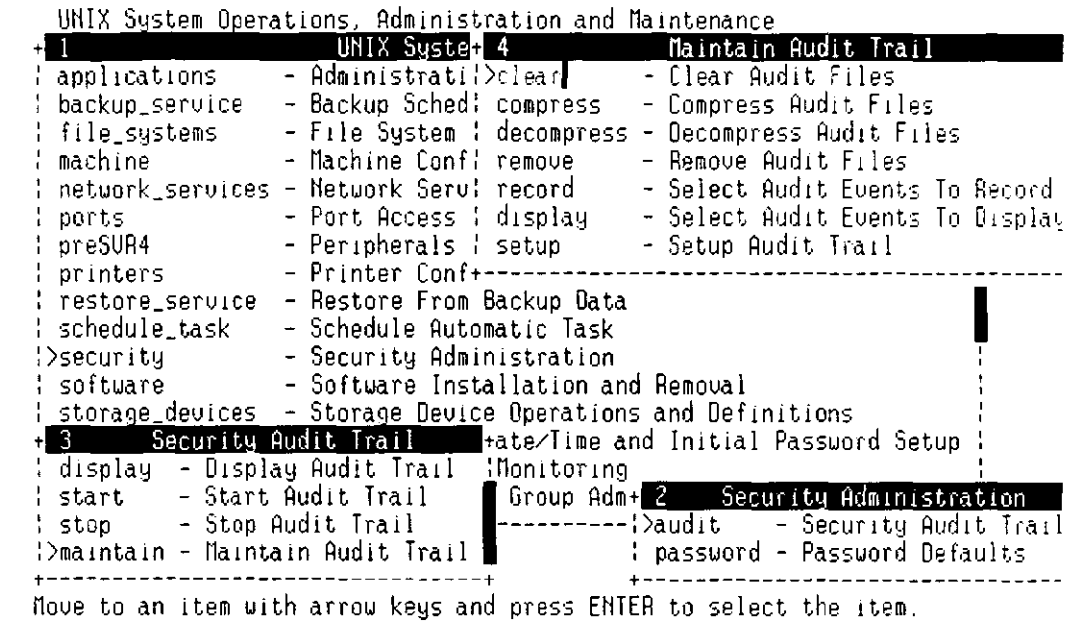
 # sysadm

Se selecciona **security** del menú principal del Administrador de Seguridad como se muestra a continuación:

```
  UNIX System Operations, Administration and Maintenance
+1                    UNIX System V Administration                    +
| applications      - Administration for Available Applications       |
| backup_service    - Backup Scheduling, Setup and Control            |
| file_systems      - File System Creation, Checking and Mounting     |
| machine           - Machine Configuration, Display and Shutdown     |
| network_services  - Network Services Administration                 |
| ports             - Port Access Services and Monitors               |
| preSVR4           - Peripherals Setup                               |
| printers          - Printer Configuration and Services              |
| restore_service   - Restore From Backup Data                        |
| schedule_task     - Schedule Automatic Task                         |
|>security          - Security Administration                         |
| software          - Software Installation and Removal               |
| storage_devices   - Storage Device Operations and Definitions       |
| system_setup      - System Name, Date/Time and Initial Password Setup |
| system_status     - System Status Monitoring                        |
| users             - User Login and Group Administration             |
+---------------------------------------------------------------------+

Move the cursor to the item you want and press ENTER to select it.
```

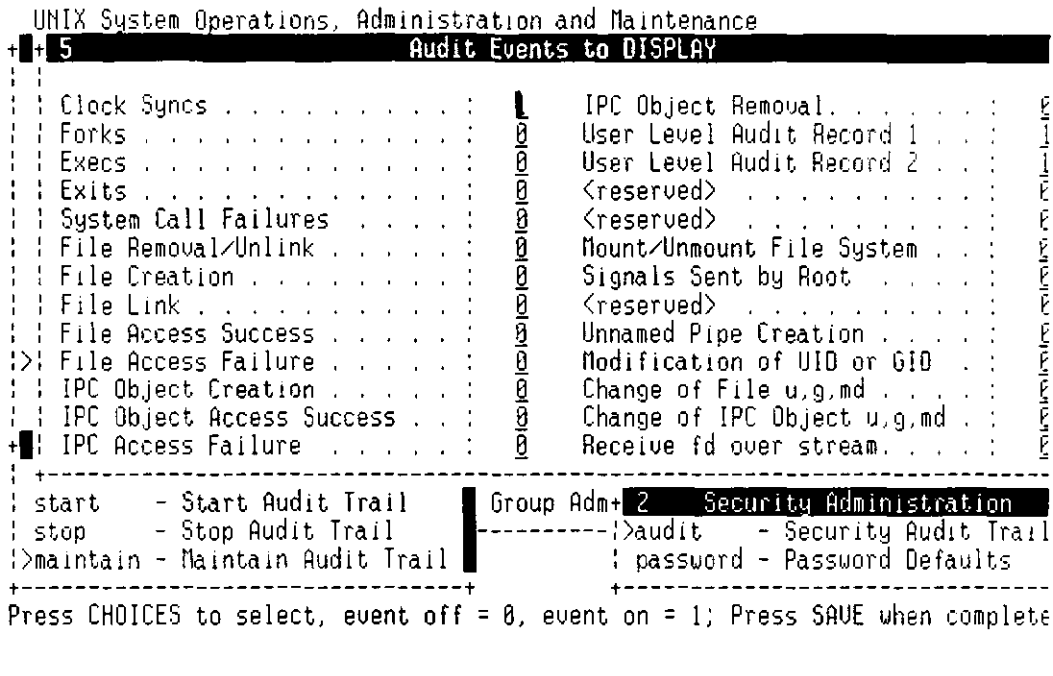
Cada tipo de evento está relacionado con un diferente canal del **audit trail**. Existen 32 canales auditores, aunque solo 23 son utilizados (del 0-15, 18-19, 21-25). Estos canales son registrados juntos en el audit trail. Para escoger qué canales van a ser habilitados o deshabilitados se selecciona **record** del menú de mantenimiento del audit trail que se encuentra debajo de del menú de Security audit trail:



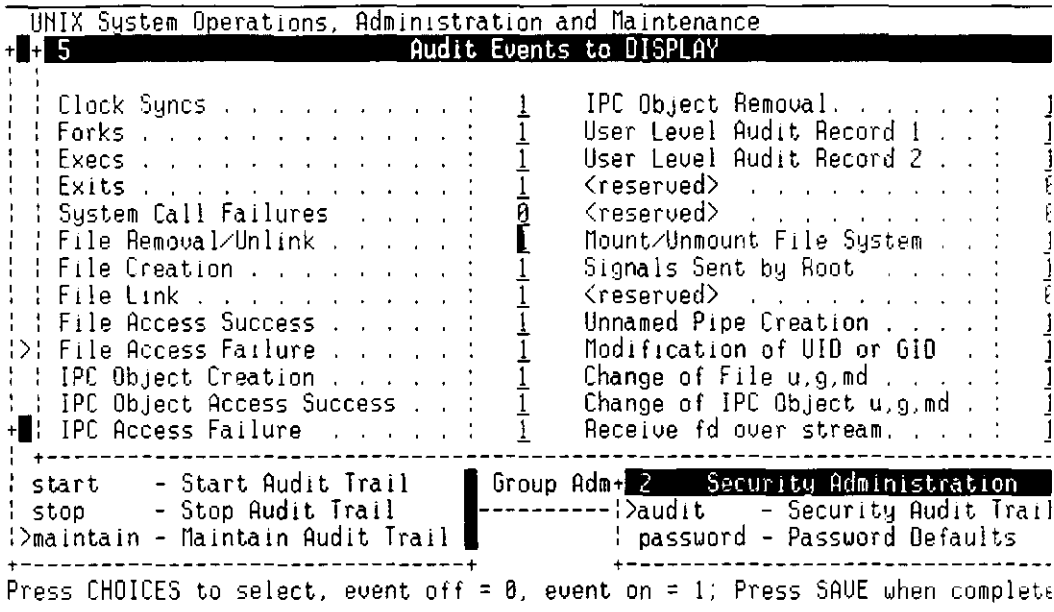
Seleccionando los canales mencionados, procedemos a personalizar la combinación de eventos registrados por el **audit trail**. La figura muestra qué canales auditores serán encendidos para capturar cada tipo de evento. Estas selecciones toman efecto la próxima vez que el programa es inicializado.

Para cambiar los canales que están siendo registrados, debe detener el **audit trail**. Si selecciona **stop** del menú **Security Audit Trail** el sistema seguirá en modo multiusuario y las acciones de los usuarios no serán auditadas. Por lo tanto, debe llevar al sistema a modo monousuario para detener al **audit trail**, una vez que ha sido detenido, puede habilitar/deshabilitar los canales a registrar y restablecer el sistema a modo multiusuario.

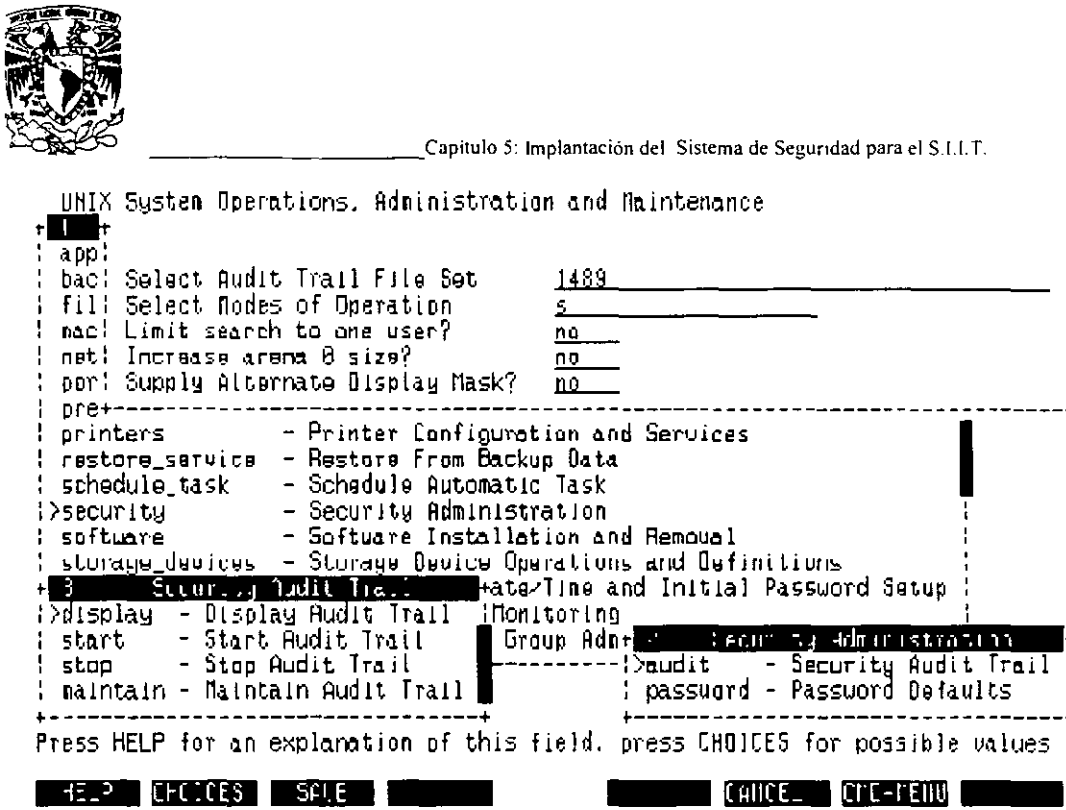
Para capturar toda la información valiosa, se deben habilitar todos los canales excepto uno: **System Call Failures**, este canal es excluido debido a que su información es voluminosa y en general, sin interés. La siguiente figura muestra los canales que pueden ser habilitados desde el menú Audit Events to DISPLAY:



Para habilitar los canales, se cambian los ceros (apagados) por los unos (encendidos). Al terminar la selección se oprime <ctrl><f> 6 para salvar la configuración. Es recomendable activar todos los canales mencionados en la página anterior.



Cada vez que el sistema operativo levanta, un archivo del **audit trail** es almacenado en la ruta **/var/sat**. Para efectos de administración, estos archivos deben ser respaldados en cinta, ya que tienden a crecer mucho por todo lo que se registra en ellos. La visualización de la información es realizada también por medio del sysadm (siguiendo los menúes security, audit, display) y oprimiendo F2 para seleccionar el registro deseado:



El nombre del registro es numérico como puede observarse en la figura. El sysadm despliega la información contenida de la siguiente forma:

```
01/30/97 17:23:19 file chown to/from administrative id: old uid = 0: new uid = 105
        /dev/term/105
01/30/97 17:23:19 root/console  LOGIN: SUCCESS: user: siit uid: 105 gid: 102 on device,
dev/term/105
01/30/97 17:23:19 pid=02089 setgid: old e/rgid = sys/sys: new e/rgid = SIIT/SIIT
01/30/97 17:23:19 siit/term/105 session start
01/30/97 17:23:22 exec, euid != owner: e/ruid = root/siit; owner = siit
        /sbin/su
01/30/97 17:23:31 pid=02090 setgid: old e/rgid = SIIT/SIIT: new e/rgid = sys/sys
01/30/97 17:23:31 exec, euid != owner: e/ruid = root/root; owner = siit
        /sbin/sh
01/30/97 17:23:45 exec, euid != owner: e/ruid = root/root; owner = siit
        /usr/bin/ls
01/30/97 17:23:57 exec, euid != owner: e/ruid = root/root; owner = siit
        /usr/bin/ls
01/30/97 17:24:08 exec, euid != owner: e/ruid = root/root; owner = siit
        /usr/bin/ftp
01/30/97 17:25:17 file chown to/from administrative id: old uid = 105: new uid = 0
        /dev/term/105
01/30/97 17:25:17 file chgrp to/from administrative id: old gid = 7; new gid = 0
        /dev/term/105
01/30/97 17:25:21 file chgrp to/from administrative id: old gid = 0: new gid = 7
        /dev/term/105
01/30/97 17:20:00 pid=02108 setgid: old e/rgid = root/root; new e/rgid = sys/sys
01/30/97 17:20:01 pid=02109 setgid: old e/rgid = sys/sys; new e/rgid = uucp/uucp
01/30/97 17:26:01 uucp/-NONE-   session start
01/30/97 17:40:00 pid=02119 setgid: old e/rgid = root/root; new e/rgid = sys/sys
01/30/97 17:40:01 pid=02120 setgid: old e/rgid = root/root; new e/rgid = sys/sys
01/30/97 17:40:01 sys/-NONE-    session start
```

La primer linea registró que el día 30 de enero de 1997 se cambió el propietario de un archivo que pertenecía a root (id = 0) para el usuario siit (id = 105) a las 17:23:19 del día.

La segunda línea registrada muestra una invocación exitosa al superusuario desde la cuenta siit en la terminal /dev/term/105. y así sucesivamente. Está claro que interpretar este tipo de información. requiere conocimientos avanzados del sistema y además perfecto conocimiento de los identificadores de usuario ya que como se desplegó en la primera línea del audit, no se menciona el nombre de la cuenta. sino el identificador asociado con ésta.

De esta forma se puede tener conocimiento de qué hace quién en el sistema auxiliándose además con los comandos del monitoreo. Los archivos generados en la ruta /var/sat tienen que ser respaldado periódicamente (1 vez al mes, o más si se tiene mucha actividad en la máquina o gran número de usuarios y borrados), ya que crecen bastante y pueden llegar a saturar el fylesystem de root. Con todo lo anteriormente expuesto, la seguridad en el sistema operativo ha quedado establecida para el Sistema Integral de Información Tributaria. En el subtema siguiente, se expondrá el último nivel que a nuestro

criterio era necesario para proteger esta aplicación, que consiste en el desarrollo de un programa en lenguaje C que audita las tablas más importantes del S.I.I.T. y registra las transacciones que se llevan a cabo en ellas.

5.3. El Auditor: programa basado en las características de Seguridad de Informix-SE

Para elevar el nivel de seguridad del S.I.I.T. y aprovechando las ventajas que en este rubro ofrece el Informix-SE , que han sido explicado en el capítulo 3, se decidió desarrollar un programa que auditaría las tablas más importantes de este sistema.

En el último nivel de seguridad de Informix SE, se registran todos los accesos, cambios, consultas, modificaciones y borrados realizados a la base de datos en un archivo llamado archivo de transacciones (Transaction Log), donde la operación que se realizó se registra junto con el identificador del usuario, la fecha en que se hizo la operación y los cambios realizados.

Este nivel a su vez cuenta con dos opciones, una para registrar absolutamente todo lo que sucede en la base de datos, y otro para registrar lo que sucede en tablas específicas.

Los archivos que se generan, solo pueden ser consultados por un usuario con todos los permisos (en este caso el propietario de la base de datos), estos archivos se encuentran encriptados, por lo que se requiere de un programa especial que proporciona el SE para poder leer la información contenida en ellos. Este mecanismo es el que complementa finalmente la seguridad de la aplicación, porque registra las operaciones hechas en la base, y proporciona mucha más información que la que en un principio se creía obtener.

Al realizar pruebas del programa de Informix SE que desencripta los archivos de transacciones fue evidente que solo funcionaba para el caso en que se habilitará la opción para toda la base de datos, además de que los campos del sistema como las fechas y números no aparecían en el formato adecuado.

Por esta razón se deseaba desechar la idea de usar los archivos de transacciones, ya que si aún la revisión de la información capturada era difícil de entender para una persona con conocimiento informático, lo sería mucho más para los abogados que serían designados a esta tarea.

La opción era entonces, que cuando se realizará cada inserción modificación y borrado de los campos de las bases de datos, quedara un registro en ella misma, es decir, insertar en la base la información nueva, dejando intacta la anterior, con lo cual tendríamos que modificar absolutamente todos los programas, además de que el problema de los respaldos se precipitaría más rápidamente.

Los archivos de transacciones eran la mejor respuesta, pero su punto débil era el programa que los desencriptaba, por lo que se tenía que hacer un programa que sustituyera al de desencriptamiento, para que cualquier persona que conociera la operación del sistema pudiera interpretar los resultados de las transacciones. Este programa se ligaría a la base de datos, obteniendo la ruta del archivo de transacciones, con la cual abriría el archivo y lo desencriptaría.

La primera versión de este programa tuvo una falla, el habilitar el archivo de transacciones a nivel base registraba absolutamente todo, hasta las tablas temporales que el SE crea para generar reportes y consultas, por lo que el archivo crecía en forma vertiginosa. Algunos de estos archivos llegaron a ser más grandes aún que las bases de datos completas, registrando solo lo de una quincena de trabajo.

Así que en una segunda versión, se deshabilitó la opción del registro de la base completa y se hizo por tablas, en las cuales únicamente se registraban inserción, modificaciones y borrado.

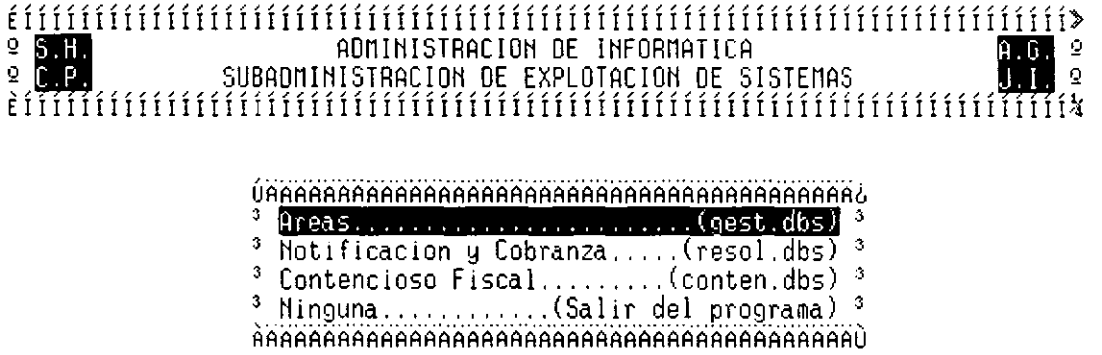
Aunque cualquier seguridad es violable, es nuestro trabajo hacer que esta violación sea lo más dificultosa que se pueda, por lo que es importante aprovechar al máximo las capacidades de seguridad que ofrecen los sistemas operativos y los manejadores de las bases de datos.

5.3.1. Operación

El programa debe tener permisos 777 para su ejecución en el sistema operativo, y es invocado con la siguiente sintaxis:

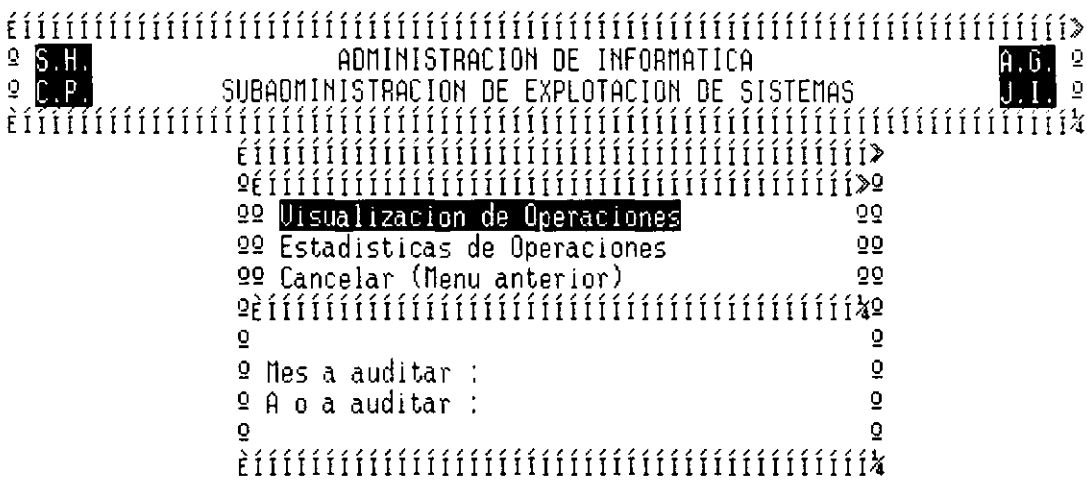
$ ./auditor

Se despliega la pantalla del menú principal que es la que se muestra a continuación:

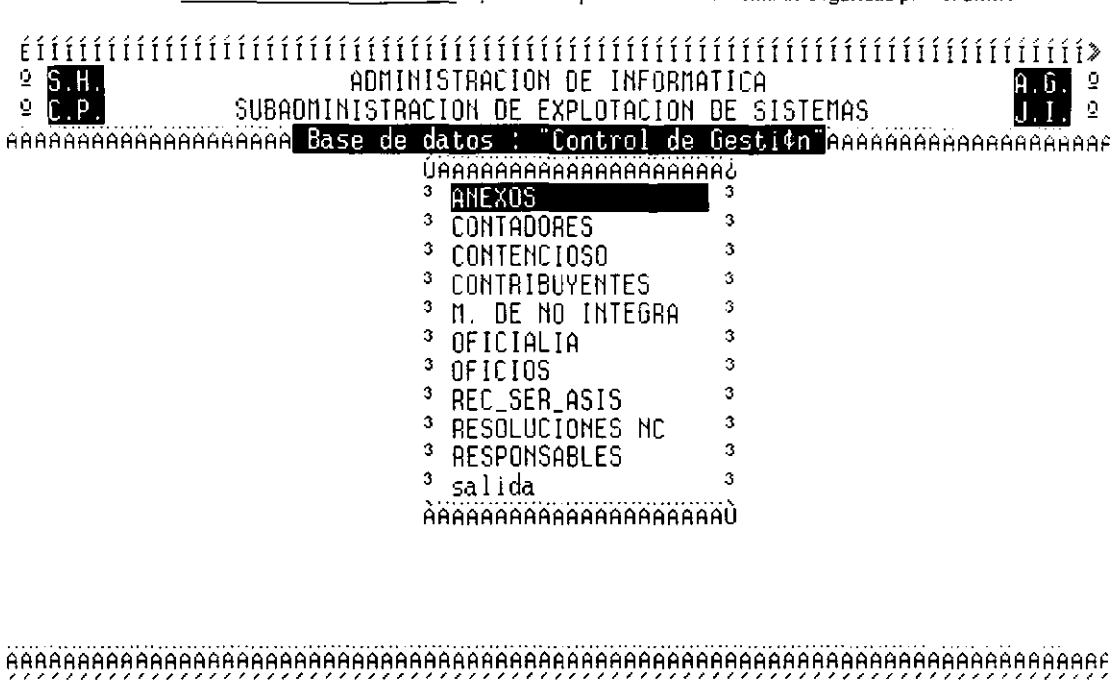


Como se puede apreciar, las opciones están relacionadas a una base de datos en especial (gest.dbs, resol.dbs y conten.dbs) que respectivamente pertenecen a Control de Gestión, Notificación y Cobranza y Contencioso, los tres módulos originales del S.I.I.T. Se pueden consultar las transacciones que fueron realizadas en cualquiera de estas tres bases de datos en sus tablas más importantes. Al elegir cualquiera de estas opciones, el script despliega otra pantalla en la que se permite Visualizar la operaciones hechas, obtener Estadísticas de Operaciones o Cancelar y regresar al menú anterior,

proporcionando el mes y el año que van a ser auditados, si se oprime la tecla <esc> sin digitar las fechas, el auditor proporcionará la información hasta la fecha actual. Para generar toda esta información, el auditor utiliza los archivos .log y .aud que son generados en la misma ruta que las bases de datos:



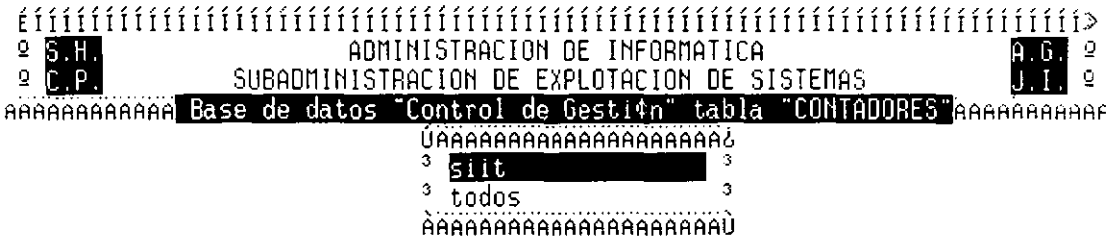
Al seleccionar alguna de estas opciones, en la parte superior al centro, el programa indica que base de datos está trabajando y cual tabla se puede seleccionar de la misma. Para el caso de Control de Gestión las tablas auditadas se muestran en la pantalla. Seleccionando la opción de Visualización de Operaciones, se despliega lo siguiente:



Ingresando a la tabla Contadores, el sistema requiere el tipo de operación que se quiere verificar, Inserciones, Modificaciones, Borrados o Todos como se aprecia en la siguiente pantalla:

Cada módulo del S.I.I.T. tiene usuarios que han sido creados en el sistema para su consulta, captura, etc, dependiendo de sus funciones y área a la que pertenecen. El auditor permite seleccionar a un usuario en particular para verificar las transacciones exclusivas de éste, aunque también pueden ser elegidos todos.

El único usuario que es auditado en los tres módulos es el usuario siit que es utilizado por el Administrador del S.I.I.T. con fines administrativos sobre las bases de datos. Esto se observa a continuación para Control de Gestión:



El formato en que es desplegada la auditoría es como sigue:

```
ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»
º S.H.                    ADMINISTRACION DE INFORMATICA                          A.G. º
º C.P.            SUBADMINISTRACION DE EXPLOTACION DE SISTEMAS                   J.I. º
ÆÆÆÆÆÆÆÆÆÆÆÆ Base de datos "Control de Gesti¢n" tabla "CONTADORES" ÆÆÆÆÆÆÆÆÆÆÆÆÆµ

  Operaci¢n     Usuario      A o  Mes  Dia  Hora  Minuto  Segundo
  Modificaci¢n  recursos      96   12   02    12      09       00

 Conse. Num. Expediente      96337101
 Conse. de Num. de Control  960011967
 Conse. de Recurso          960000917
 Numero de Determinante     325-A-VIII-4-
 Clave del Area             S
 con_Inc                    nulo
 expedie                       1062930

 Se modifica por :

 Conse. Num. Expediente      96337101

Mov. flechas arriba abajo, P g. arriba Ctrl-R, P g. abajo Ctrl-N, Sig. Ctrl-F
```

El reporte indica que tipo de operación se hizo, el usuario que la realizó, la fecha (año, mes, día) y la hora en que fue efectuada (hora, minuto, segundo). En esta opción de Modificaciones, se presenta primero la información que existía en la base de datos (el registro completo) y enseguida el campo modificado y la nueva información. En las opciones de Inserciones y Borrados solo se presenta el registro que fue dado de alta o el que fue eliminado de la base.

Cuando se generan Estadísticas de Operaciones, la información el formato es el de la siguiente pantalla, seleccionando previamente la base de Contencioso y la tabla Amparo del Recurso:

```
ÉÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍÍ»
º S.H.                    ADMINISTRACION DE INFORMATICA                    A.G. º
º C.P.              SUBADMINISTRACION DE EXPLOTACION DE SISTEMAS           J.I. º
ÈÍÍÍÍÍÍÍÍ Base de datos "Contencioso Fiscal" tabla "AMPARO DEL RECURSO" ÍÍÍÍÍÍÍÍ¼
```

Las siguientes estadisticas son del dia 02/03/97 a las 13:39.28
al dia 02/03/97 a las 13:54.50

Columna	Inserciones	Borrados	Modificaciones
R.F.C	6	2	0
Num de juicio de nulidad	6	2	0
Notif demanda de amparo	6	2	0
Audiencia constitucional	6	2	0
Present contest demanda	6	2	0
Cve concepto violacion	6	2	0
Notif sentencia de amparo	6	2	0
Clave de sentencia	6	2	0
Inter rec rev en amparo	6	2	0

```
Mov. flechas arriba abajo, Pág. arriba Ctrl-R, Pág. abajo Ctrl-N, Sig. Ctrl-F
```

Con este tipo de control, alguna negligencia, error, omisión o falsedad por parte de alguna persona con la operación del sistema, es registrada rigurosamente y se deslindan responsabilidades que de otra forma serían difíciles de probar. Esto obedece a que, aunque la Administración de Informática trabaja para apoyar a las áreas en sus procesos, es también responsable de la información que este sistema maneja y cualquiera de las situaciones mencionadas la involucra por lo que este el auditor le proporciona elementos suficientes para delimitar y comprobar las responsabilidades correspondientes.

Conclusiones

A lo largo del desarrollo de este trabajo, nos hemos percatado que hace falta otorgar un mayor esfuerzo, no sólo en difundir la cultura de seguridad en cómputo sino, además aplicarla a todos los nuevos proyectos de computo que se den en todas nuestras empresas Mexicanas.

A partir de hoy los costos de las medidas de seguridad deben ser siempre vistos como parte de la inversión total de un proyecto. Los niveles de seguridad que se propusieron se basaron en la clasificación de los recursos e información que posee esta dependencia; siempre bajo la constante supervisión y valiosas propuestas que pudimos otorgar en base a nuestro desarrollo profesional.

A través de este documento pudimos observar que, aunque los ataques pueden tener objetivos diferentes, estos estarán limitados a la capacidad y empeño que aplique el personal administrativo y los usuarios del sistema S.I.I.T.

Es sabido por la mayoría de los directivos que la falta de seguridad en su ámbito laboral siempre traerá consecuencias que en el peor de los casos pueden ser financieras.

Aunado a la investigación y desarrollo de este material de tesis, se integró el planteamiento de un Sistema Auditor. El enfoque que propone este sistema alterno, permitirá detectar, errores u omisiones que se hayan llevado acabo en la operación, deslindando así responsabilidades

Con el desarrollo de los cinco capítulos que contiene este trabajo de tesis, consideramos que se implantó la seguridad mínima aceptable para el S.I.I.T, contribuyendo con este proyecto a elevar la productividad de las áreas usuarias y desarrolladoras involucradas de la S.H.C.P. a nivel nacional.

Bibliografía

Luis Angel Rodríguez. "Seguridad de la información en sistemas de Cómputo" Ventura Ediciones. S.A. de C.V. 1995

McAfee, J. And Haynes, C. "Computer Viruses, Worms, Data Diddlers, Killer Programs, and others Threats to your System". New York: St. Martin's Press, 1989.

Information System Security, Guidelines for The United Nations Organizations. 1993

Tolgo, J.W. Disaster Recovery Planning: Managing Risk and Catastrophe in Information Systems. Englewood Cliiffs (New Jersey): Yourdan Press, 1989

Wong, K.K. and Watt S. Managing Information Security: a Non Technical Management Guide. Oxford: Elsevier Advanced Technology, 1990.

Vasarhelyi, M.; Lin, T. Advanced Auditing: Fundamentals of EDP and Statistical Audit Technology. E.U.A.: AddisonWesey, 1990.

E. Ciurana Macías, "UNIX: ¿Qué es y adonde va?" PC/TIPS

Brian W. Kernighan, Rob Pike "El entorno de programación UNIX"

D. Budgen "Introducción al Sistema Operativo UNIX"

UNISYS UNIX System V Release 4.0 "Migration and Compatibility Guide"

David A. Curry "UNIX System Security A guide for Users and System Administrators"

Diego Martín Zamboni "Soluciones Avanzadas"

David A. Curry "UNIX System Security A guide for Users and System Administrators"

UNISYS UNIX System V Release 4.0 "System Administrator's Guide"

Bibliografía (Referencias)

Spafford H. Eugene and Garfinkel Simson. "Practical Unix and Internet Security". 2da. Edición. Mayo 1996. Edit O'Reilly.

Spafford H. Eugene and Garfinkel Simson. "Web Security & Commerce". 1a. Edición. Junio 1997. Edit O'Reilly.

Khare Rohit, Rifkin Adam, Spafford H. Eugene, Garfinkel Simson, Stein Lincoln. "Web Security" A matter of trust. Edit. OReilly

Bellovin Steve and Cheswick Bill. "Firewalls and Internet Security". 1ª Edición 1994. Edit. Addison Wesley.

Chapman D. Brent and Zwicky D. Elizabeth. "Building Internet Firewalls". 1a Edición 1995. Edit. O'Reilly & associates.

Hunt Craig. "TCP/IP Network Administration". 2da Edición Octubre 1997. Edit. O'Reilly & Associates.

News Magazine:

http://www.news.com/News/Item/0,4,24962,00.html

CERT-EU:

http://www.cert.org

ftp://ftp.cert.org/pub/tech_tips/UNIX_configuration_guidelines

ftp://ftp.cert.org/pub/tech_tips/root_compromise

ftp://ftp.cert.org/pub/tech_tips/security_tools

ftp://ftp.cert.org/pub/tech_tips/password_file_protection

Mx-CERT

http://www.mxcert.org.mx

ASC - UNAM

http://www.asc.unam.mx

ftp://ftp.super.unam.mx/pub/seguridad/tools