UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES
CAMPUS ARAGON

"ADMINISTRACION REMOTA DE LA SEGURIDAD DE SISTEMAS DE COMPUTO, MEDIANTE INTEGRACION DE HERRAMIENTAS DE MONITOREO"

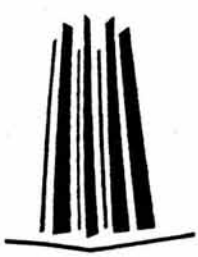
T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION

P R E S E N T A :

MARCO ANTONIO BRAVO RAMIREZ



ASESOR DE TESIS:
M.EN C. LEOBARDO HERNANDEZ AUDELO

SAN JUAN DE ARAGON, EDO. DE MEXICO. 2004

AGRADECIMIENTOS

A Dios,
que guía e ilumina mi camino.

A mi mamá,
por todos sus consejos, dedicación y
esfuerzos invertidos en mi educación.

A mi papá,
de quien he aprendido a
ser una mejor persona.

A mis hermanos,
con todo mi afecto, por
su apoyo y paciencia.

A América y Dayana,
con especial gratitud, por su entusiasmo
y motivación para concretar esta etapa.

A Danae y Julio,
por toda su colaboración para la
integración de este trabajo.

A Leobardo Hernández, mi asesor,
por su guía y orientación en cada
etapa de este proyecto.

A la UNAM y la DCAA,
ya que representan una parte fundamental
en mi formación profesional.

Finalmente, es justo reconocer que este trabajo es resultado del esfuerzo y dedicación de mucha gente que colaboró en mi formación educativa a través de los años, tanto a nivel personal como profesional. Son muchos los nombres para listarlos a todos, pero a todos y cada uno sólo puedo decirles: GRACIAS!

PRÓLOGO

La implantación de las tecnologías de la información en las organizaciones y su aplicación cada vez más extensa en diversos aspectos de las actividades humanas, ha permitido el desarrollo de grandes avances en áreas tan diversas como la economía, los negocios, la cultura, las ciencias, las relaciones humanas y en general la creación y difusión de información.

Cada vez es más común que las organizaciones realicen mayores inversiones en materia de tecnologías de la información para sistematizar sus procesos de negocio; procurando con esto contar con sistemas informáticos que proporcionen altos niveles de disponibilidad, desempeño y una mejor respuesta a las demandas de nuevos servicios.

Por lo tanto, resulta evidente que para las organizaciones la seguridad informática desempeña un papel crucial para garantizar la disponibilidad, integridad y confidencialidad de la información que manejan en sus distintos niveles, ya que tiende a incrementarse la posibilidad de que los equipos, sistemas y servicios de cómputo utilizados puedan presentar vulnerabilidades y quedar expuestos a ataques informáticos.

En virtud de este escenario, los administradores de sistemas se enfrentan a retos tecnologicos cada vez más complejos que requieren el manejo de conceptos y habilidades diversas. Aunado a la expansión de las redes de telecomunicaciones que permiten que los equipos de cómputo puedan estar ubicados en prácticamente cualquier lugar del mundo, se requiere que los administradores de sistemas y encargados de la seguridad informática en las organizaciones cuenten con herramientas que apoyen su labor.

En este trabajo se presenta una alternativa de solución, basada en la integración de metodologías y herramientas de software de libre distribución, que permitan a los administradores de sistemas y seguridad informática contar con información del estado que guardan los equipos y servicios de cómputo. Es importante comentar que la utilización de herramientas de software de libre distribución permite una mayor independencia a las organizaciones para adecuar y desarrollar nuevas caracteristicas en las herramientas, además de los ahorros en costos financieros.

Este documento se encuentra organizado en siete capítulos, en los cuales se abordan las bases de la seguridad informática así como, la problemática identificada, las alternativas de solución disponibles y la propuesta de solución producto de este trabajo.

En el Capítulo 1 se presenta la introducción a los temas de administración de sistemas: origen, concepto, funciones, contexto, fundamentos, modelos y

estándares que se han desarrollado; en materia de seguridad informática se aborda su contexto, fundamentos y modelos; en lo concerniente a criptografía se profundiza en los antecedentes, tipos de cifrados, algoritmos y esteganografía; y también se describe la evolución y clasificación de las herramientas de seguridad de libre distribución disponibles.

En el Capítulo 2 se aborda el tema de la administración segura de sistemas de cómputo, para lo cual se revisan conceptos necesarios para el aseguramiento de los mismos, incluidos la valoración y mitigación del riesgo así como, los procedimientos y consideraciones más importantes que deben tenerse en cuenta para lograr una administración segura de sistemas. También se tratan las consideraciones generales, la actualización de aplicaciones y pruebas a tener en cuenta para llevar a cabo una adecuada administración remota de la seguridad.

El Capítulo 3 plantea el contexto y determinación del problema que se trata de resolver en este trabajo, para lo cual se profundiza en la mayor dependencia de las aplicaciones computacionales en la cual se encuentra inmersa la sociedad actual así como, el incremento de la posibilidad de ataques informáticos y las dificultades presentadas al aumentar el número de equipos de cómputo.

En el Capítulo 4 se exponen las alternativas de solución existentes a la problemática planteada, y se muestsra un panorama de los principales elementos que fueron considerados para la identificación de las diversas soluciones así como, de las metodologías para enmarcar el problema existente: la seguridad integrada.

La propuesta de solución se presenta en el Capítulo 5, mostrando su esquema general, la situación actual y el modelo de solución para la administración integrada de la seguridad, se comentan las etapas necesarias para la implantación de la solución. Así como, el análisis y diseño realizados para delimitar los factores importantes, la estructura de datos y los procesos necesarios que constituyen la solución propuesta.

En el Capítulo 6 se describe el desarrollo de la propuesta, que involucra la construcción e implantación de la solución. En esta parte se comentan los requisitos previos que deben cumplirse para ponerla en funcionamiento, los programas desarrollados, el esquema de funcionamiento, la logística, los procedimientos de instalación y las consideraciones para el mantenimiento.

Finalmente, la evaluación de la solución propuesta se expone en el Capitulo 7, en donde se identifican los factores evaluados: funcionalidad, utilidad, efectividad, seguridad y facilidad de uso; y se presentan los resultados y conclusiones del proyecto, a partir de los cuales se derivan y proponen líneas de trabajo por desarrollar tomando como base la solución propuesta.

ÍNDICE

Capítulo 1. Introducción.

La administración de sistemas, la seguridad informática y las herramientas de seguridad son tres elementos básicos en la implantación exitosa de sistemas de cómputo seguros. Por una parte la administración de sistemas se enfrenta a la operación que día con día debe realizarse para el adecuado funcionamiento de los sistemas de cómputo, mientras que la seguridad informática estudia, de una manera formal, lo relativo a la protección y manejo adecuado de la información dentro de una organización. Ambas disciplinas requieren apoyarse en herramientas que aplicadas a los sistemas de cómputo, colaboren para el óptimo desempeño y seguridad de dichos sistemas.

La primera parte de este capítulo, correspondiente a la administración de sistemas, comprende lo relativo a los antecedentes de esta actividad en relación con la evolución de los sistemas de cómputo, los conceptos de lo que es un administrador de sistemas y las funciones que éste debe desempeñar.

En lo concerniente a la seguridad informática, en primer lugar se pone ésta en un contexto para luego profundizar en los fundamentos de la disciplina, exponer los modelos de seguridad informática más importantes y describir los estándares que en esta materia han establecido diversas organizaciones y gobiernos alrededor del mundo.

Posteriormente, se expone el tema de criptografía en donde se mencionarán sus antecedentes, los tipos de cifrados y los algoritmos más conocidos y utilizados en la actualidad.

Para finalizar, se describen y comentan las herramientas de seguridad disponibles, su evolución y clasificación de acuerdo a las funcionalidades de cada una. Algunas de las herramientas mencionadas serán empleadas en el prototipo que acompañará este trabajo de tesis.

1.1 Administración de sistemas.

La aplicación de los sistemas de cómputo se ha extendido a prácticamente cualquier actividad del ser humano, y la administración de sistemas es una disciplina fundamental en la operación de dichos sistemas, ya que para un adecuado funcionamiento de cualquier sistema de cómputo se requiere la ejecución de una gran diversidad de tareas.

1.1.1 Origen.

La administración de sistemas de cómputo, surge con la creación de las computadoras y se vuelve más necesaria a medida que la industria del cómputo crece y se diversifica. Es debido a que la computadora es un dispositivo de índole general y programable, que se requiere una figura encargada de adecuar dichos

dispositivos a aplicaciones específicas, de vigilar el apropiado funcionamiento y mantenimiento de los equipos, sistemas y aplicaciones informáticas. Las funciones de la administración de sistemas, a través del tiempo, se han ido adecuando a medida que las computadoras han evolucionado.

Los antecedentes de la adopción de máquinas de cálculo a las organizaciones comerciales, datan desde finales del siglo XIX, cuando Herman Hollerith llevó su lectora de tarjetas perforadas al mundo de los negocios y fundó la Tabulating Machine Company en 1896, que luego de una serie de integraciones, llegó a ser conocida como International Business Machines (IBM) en 1924. Otras compañías como Remington Rand and Burroghs también comercializaban lectoras de tarjetas perforadas para uso comercial. Tanto las empresas privadas como el gobierno de Estados Unidos utilizaron tarjetas perforadas para procesamiento de datos hasta los años 60s [7].

Los avances en el desarrollo de las computadoras, han podido ser observados y diferenciados claramente y se han clasificado en las llamadas "generaciones de computadoras". Cada generación de computadoras está diferenciada principalmente por la evolución que han tenido los elementos eléctricos y electrónicos que componen las máquinas, pero también, aunque de forma gradual, por la evolución que han tenido los programas de software que utilizan las computadoras. Cada una de estas etapas ha representado características y posibilidades de uso diferentes de los sistemas de cómputo [7].

Las computadoras de la primera generación se caracterizaron por el hecho de que las instrucciones de operación estaban hechas para cubrir tareas específicas para las cuales las computadoras eran utilizadas. Cada computadora tenía un programa diferente codificado en binario (llamado lenguaje máquina) que le indicaba cómo operar. Esto hacía a la computadora difícil de programar y limitaba su versatilidad y velocidad. Otras características distintivas de las computadoras de la primera generación fue el uso de tubos de vacío (responsables del enorme consumo de energía) y tambores magnéticos para el almacenamiento de datos. En esos primeros días, un solo grupo de personas se encargaban de diseñar, construir, programar, operar y dar mantenimiento a cada máquina. Como ya se mencionó, la programación se realizaba en lenguaje máquina absoluto y con frecuencia se utilizaban conexiones para controlar las funciones básicas de la máquina. Los lenguajes de programación eran desconocidos (incluso el lenguaje ensamblador). El modo usual de operación consistía en que el programador tenía que reservar cierto periodo de tiempo utilizando una hoja de reservación pegada en la pared, iba al cuarto de la máquina, introducía su programa en la computadora (comúnmente esto implicaba realizar diversas conexiones de cables) y pasaba unas horas esperando que ninguno de los miles de bulbos se quemara durante la ejecución. La inmensa mayoría de los problemas eran cálculos numéricos directos, como por ejemplo el cálculo de valores para tablas de senos y cosenos. A principios de la década de los cincuenta, la operación mejoró un poco con la introducción de las tarjetas perforadas. Fue entonces posible escribir los

programas en las tarjetas y leerlas en vez de realizar conexiones de cables; por lo demás el proceso era el mismo [7].

En 1948, la invención del transistor transformó el desarrollo de las computadoras. El transistor reemplazó los enormes y engorrosos tubos de vacío dentro de las televisiones, radios y computadoras. Debido a esto el tamaño de la maquinaria electrónica se redujo desde entonces. El transistor se empezó a utilizar en las computadoras hasta 1956. En conjunto con los primeros avances en la construcción de memoria de núcleo magnético, los transistores permitieron a las computadoras de la segunda generación ser más pequeñas, más rápidas, más confiables y con una mejor eficiencia en el consumo de energía que sus predecesoras [7].

Fueron los programas almacenados y los lenguajes de programación lo que le permitió a las computadoras la flexibilidad para finalmente tener un costo adecuado y ser productivas para el uso en las empresas. El concepto de programa almacenado significó que las instrucciones a ejecutarse en una computadora para una función específica (conocido como programa) eran mantenidas dentro de la memoria de la computadora, y rápidamente podía ser reemplazado por un conjunto de instrucciones diferentes para realizar funciones distintas. Una computadora podía estar imprimiendo recibos para clientes y minutos después funcionar para diseñar productos o realizar procesos de nómina [7].

El ingreso de las computadoras comerciales tuvo que esperar más tiempo, hasta que las condiciones del mercado fueron apropiadas y es hasta 1951, cuando la UNIVAC I (Universal Automatic Computer), construida por Remington Rand, llegó a ser una de las primeras computadoras disponibles comercialmente, al tomar ventaja de los avances tecnológicos logrados hasta entonces. Tanto el Buró de Censos de Estados Unidos como General Electric adquirieron UNIVACs. Uno de los aciertos más impresionantes de la UNIVAC fue predecir el triunfo de la elección presidencial de 1952 que ganó Dwight D. Eisenhower. La UNIVAC, fue la primera computadora diseñada específicamente para aplicaciones comerciales. A partir de esta, otras más fueron lanzadas al mercado y con esto, el cómputo se diversificó hacia una cantidad importante de aplicaciones de negocios [7].

Fue a mediados de la década de los años 60s cuando se crearon nuevas carreras en las universidades (programador, analista y experto en sistemas de cómputo) [7]. Por primera vez hubo clara separación entre los diseñadores, constructores, operadores, programadores y personal de mantenimiento. En un principio, las computadoras habían sido dispositivos de relativo fácil manejo, en cuanto a su funcionalidad, ya que se orientaban a tareas específicas y eran sencillas de operar, pero al aparecer las computadoras programables, todo cambió: se requirió contar con personal capacitado adecuadamente para programar y operar los sistemas de cómputo.

Las computadoras, que al inicio eran una herramienta para científicos, han llegado a diseminarse para ser usadas en las industrias, universidades, oficinas de

gobierno, comercios y hasta los hogares. Las aplicaciones desarrolladas para las computadoras han evolucionado en complejidad y la industria del cómputo ha tenido un crecimiento inusitado, lo que ha provocado una gran demanda de personal calificado para operar las computadoras en ambientes comerciales. Al extenderse el uso de las computadoras en las organizaciones, la figura del administrador de sistemas ha surgido como el especialista encargado de instalar, adecuar, optimizar y mantener funcionando las computadoras, el sistema operativo y los programas de aplicación que se utilizan en las diversas áreas de las organizaciones, implantados con el objeto de proveer servicios o ejecutar procesos de manera más eficiente. Y es debido a estas necesidades y a la difusión de las microcomputadoras y las redes, que la función de los administradores de sistemas ha cobrado mayor relevancia dentro de las organizaciones.

1.1.2 Concepto.

Un sistema es un conjunto de elementos interrelacionados e interdependientes que funcionan como un todo, es relevante identificar algunas propiedades que tienen los sistemas y que resultan interesantes para poder comprender un poco más la labor del administrador de sistemas:

1. Son complejos debido a que tienen muchos componentes, ciclos de retroalimentación, tiempos promedio entre fallas y son dependientes de infraestructura.

2. Interactúan con otros sistemas, formando sistemas aún más grandes.

3. Tienen propiedades emergentes. En otras palabras, los sistemas realizan cosas para las que no fueron diseñados originalmente. Los usos que puede tener un sistema dado puede variar con el tiempo (por ejemplo, el teléfono ha cambiado la forma en que las personas interactúan, aunque originalmente se pensaba que podía ser utilizado para avisar sobre el envío y recepción de telegramas).

4. Tienen fallas muy particulares comúnmente llamadas "bugs"[1]. Un "bug" es una propiedad emergente no deseable de un sistema. Es diferente a un mal funcionamiento. Cuando algo no funciona, simplemente ya no trabaja adecuadamente. Pero cuando un sistema tiene un "bug", sigue funcionando pero se comporta de manera incorrecta en alguna forma particular, posiblemente irrepetible, y posiblemente inexplicable. Los "bugs" son relativos a los sistemas. Las máquinas pueden fallar o no trabajar, pero sólo un sistema puede tener "bugs" [9].

[1] El término "bug" ("bicho" en inglés) es usado -desde finales del siglo XIX- para designar, dentro de un contexto, cualquier falla o problema presentado en las conexiones o funcionamiento de un aparato eléctrico [6]. Este término también ha sido aplicado a la computación para referirse a problemas en el hardware y software.

Es importante identificar un sistema de cómputo, ya que es la parte fundamental de las labores de un administrador de sistemas. De manera general, un sistema de cómputo está compuesto de dos tipos de componentes básicos: hardware y software[2].

A su vez, el hardware puede ser dividido en sus componentes principales: la unidad central de proceso (CPU, por sus siglas en inglés), el almacenamiento principal, el almacenamiento secundario y los dispositivos de entrada y salida (I/O, por sus siglas en inglés).

Por otra parte, el software está compuesto por: software del sistema (todo el software relacionado al sistema operativo[3]) y software de aplicación (aquellos programa diseñados y elaborados para llevar a cabo tareas diversas) [2]. En la Figura 1-1 se puede observar esta relación.

[2] Existen componentes llamados firmware, y son una combinación de hardware y software, por lo que deben ser considerados en ambas categorías.
[3] El sistema operativo es el software que administra los recursos y controla la operación de una computadora [2 p. 26]. Se conforma del núcleo (o kernel), interfase (o shell) y utilerías.

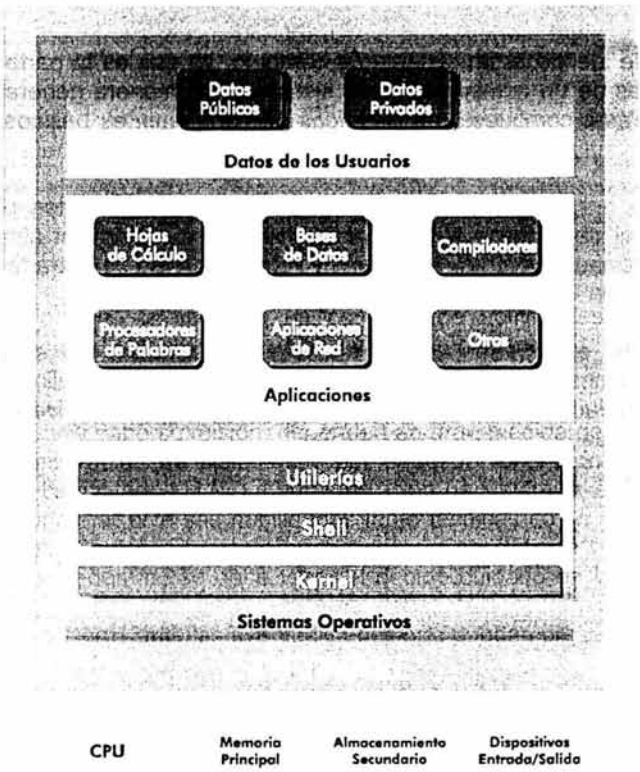


Figura 1-1 Organización de un sistema de cómputo.

La administración de sistemas es un conjunto de actividades relativas a la instalación, configuración, optimización y mantenimiento de los sistemas informáticos y que son realizadas por una figura llamada administrador de sistemas [8]. Las organizaciones que cuentan con sistemas de cómputo y que los utilizan para llevar a cabo su misión, dependen en gran medida de contar con una adecuada administración de sistemas y por ende de tener administradores de sistemas mejor capacitados [11]. La administración de sistemas, es una actividad que requiere concertar diversas habilidades tanto técnicas como de interacción y comunicación personales. El simple hecho de que un sistema "funcione" no es suficiente, ya que además, se debe tener el debido cuidado en la configuración, documentación y resguardo de un sistema dado, para proteger y en su caso recuperar dicho sistema ante las eventualidades que puedan presentarse.

Los administradores de sistemas son los responsables de asegurar la operación apropiada, el soporte técnico oportuno y la protección eficaz de los bienes de cómputo de una organización. A diferencia de otros avances tecnológicos, y por

las aplicaciones que puede proporcionar un sistema de cómputo puesto en producción, las fallas que puedan presentarse en tales sistemas pueden llegar a impactar de manera negativa a millones de personas alrededor del mundo, por lo que esta actividad es más crucial que su equivalente en otras tecnologías.

El aumento de la dependencia de las computadoras en todos los sectores de la sociedad ha permitido a los administradores de sistemas estar en contacto con información cada vez más valorada por las organizaciones para las que prestan servicio, por lo que se incrementa el impacto que puede causar cualquier error realizado, mala decisión tomada o problema presentado. El alcance de las responsabilidades de un administrador de sistemas es amplia. Los usuarios confían en la realización de tareas de previsión, planeación, mantenimiento y reparación de fallas. Además se espera que los administradores de sistemas tengan un buen conocimiento de las opciones de hardware y software existente en el mercado, y que pueden proporcionar mejores soluciones a los requerimientos de procesamiento de información de las organizaciones [12].

El administrador de sistemas, debe enfrentarse día a día con la "realidad" más que con la "teoría" de los sistemas. La teoría trabaja mejor en condiciones ideales y en configuraciones de laboratorio. El mundo real involucra sacrificios en el diseño, variables no previstas e implantaciones imperfectas. "En la teoría no hay diferencia entre teoría y práctica. En la práctica sí existe diferencia" – Yogi Berra [9].

1.1.3 Funciones.

La administración de sistemas no es un fin en sí mismo, sino un proceso continuo. El proceso del mantenimiento de los sistemas es constante. Sin embargo, todo este proceso puede segmentarse en breves periodos en los que las actividades están determinadas y pueden ser claramente identificadas y diferenciadas. Por ejemplo, podemos enumerar como actividades diferenciadas, las siguientes:

- Instalar el sistema operativo.
- Crear y modificar usuarios del sistema.
- Realizar respaldos de información.
- Monitorear el sistema.
- Recuperar respaldos información.
- Actualizar programas.
- Revisar la seguridad del sistema.
- Agregar o reemplazar hardware a los equipos.
- Reconfigurar el sistema.
- Instalar aplicaciones.
- Organizar y asignar el espacio en disco duro.
- Realizar trámites para la adquisición de hardware adicional.
- Interactuar con otras áreas de la organización.
- Reacomodar físicamente los equipos.
- Proporcionar soporte técnico a los usuarios del sisten...

- Realizar ajustes al sistema para mejorar el desempeño.

De manera general, la función de un administrador de sistemas es permitir a los usuarios el uso de los sistemas de cómputo en una manera que beneficie a toda la organización para la cual laboran. Conforme los sistemas informáticos crecen y se diversifican en las organizaciones, las funciones que deben realizarse -relativas a la administración de sistemas- pueden ser divididas, dando como resultado administradores de sistemas especialistas en diferentes rubros, como por ejemplo:

- Administradores de sistemas operativos.
- Administradores de bases de datos.
- Administradores de equipo de cómputo.
- Administradores de equipo de comunicaciones.
- Administradores de servicios de red.
- Administradores de aplicaciones.
- Administradores de la seguridad informática.

Es importante mencionar que las funciones de un administrador de sistemas se diversifican más debido a la versatilidad de aplicaciones que pueden proveer los sistemas de cómputo, y parte fundamental de estas labores están relacionadas con la seguridad informática en mayor o menor medida.

1.2 Seguridad informática.

Conforme se extiende la utilización de los sistemas de cómputo y se incorporan a las diversas actividades del ser humano, la seguridad informática cobra cada vez mayor atención en las organizaciones, debido principalmente al incremento ataques presentados [2]. En la actualidad los sistemas computacionales, pueden llegar a ser tan críticos, que del adecuado funcionamiento de éstos, pueden incluso depender una gran cantidad de vidas humanas.

La seguridad informática involucra diversos aspectos que deben tenerse en cuenta para lograr proteger adecuadamente los sistemas de cómputo, es importante hacer notar que la seguridad informática ha evolucionado -y se mantiene en constante evolución- de acuerdo con los avances en las diversas áreas relacionadas a la informática.

1.2.1 Contexto.

En el inicio del desarrollo de las computadoras, la seguridad de éstas consistía simplemente en la incorporación de mecanismos de protección externos para permitir el acceso a los equipos sólo a las personas autorizadas [2, p. 2]. Esto era así, debido principalmente a los usos y funcionalidades de los sistemas de cómputo en sus inicios, ya que por lo general era necesario estar en contacto directo con los equipos para procesar información.

La seguridad externa a los sistemas de cómputo es un punto muy importante (aunque no el único) para conformar una infraestructura de seguridad informática adecuada en una organización, y está compuesta principalmente por tres elementos que a continuación se describen.

1.2.1.1 Seguridad física.

La seguridad física consiste en las técnicas usadas para asegurar cualquier elemento de valor, las cuales aplican sólo para elementos materiales y tangibles. Tales técnicas son usadas, en el ámbito de los sistemas de información, no sólo para proteger la información almacenada en dichos sistemas, sino también para evitar el robo de los equipos y dispositivos de cómputo, comunicaciones y periféricos.

Las medidas pueden consistir desde candados y guardias, hasta cámaras de vigilancia y sistemas de alarmas. En razón de la importancia de la información resguardada, así mismo deben ser más sofisticados los dispositivos y sistemas utilizados para proteger dicha información [2, pp. 23,24].

La seguridad física es un frente importante para la protección de los sistemas de cómputo, pero no el único, por lo que debe integrarse con otros mecanismos, procedimientos y tecnología, para que en conjunto se logre una protección adecuada tanto de los sistemas de cómputo, como de la información que se almacena, procesa y transmite entre dichos sistemas.

1.2.1.2 Seguridad en las personas.

Es importante señalar que los sistemas de cómputo, por más automatizados que se encuentren, en algún punto de su operación, son manejados por personas, mismas que son propensas a cometer errores, olvidos o descuidos, además de ser "componentes" no predecibles en la operación de sistemas de cómputo, por lo que la seguridad en las personas es un elemento muy importante a tener en cuenta.

Asimismo, quienes son usuarios de los sistemas de cómputo y de la información que éstos contienen, son también personas. Debido a esto, se puede expresar que un sistema es tan seguro como las personas que lo operan y utilizan, por lo que es crucial contar, entre otras cosas, con los procedimientos para determinar el nivel de confianza que una organización puede otorgar a un individuo en particular.

Esto involucra poder determinar el nivel de acceso que la organización ha concedido a un individuo en base a sus conocimientos, experiencia previa y requerimientos del trabajo que desempeña. En un gran porcentaje de los casos de crímenes informáticos se encuentran involucrados personal de la misma organización atacada, lo que ilustra la importancia que debe darse a este aspecto

de la seguridad. Han sido numerosos los casos en los cuales el crimen ocurrió como resultado de que un empleado violó la confianza otorgada y utilizó el sistema de cómputo para obtener un beneficio personal [2, p. 24].

1.2.1.3 Seguridad administrativa.

La seguridad administrativa describe los métodos a utilizar para implantar las políticas de seguridad establecidas, estos métodos detallan, entre otras cosas, cómo deben organizarse los equipos de cómputo, comunicaciones y periféricos; cómo se lleva a cabo el manejo, almacenamiento y destrucción de cintas u otros dispositivos de almacenamiento o de respaldo de información; cómo deben realizarse las instalaciones, actualizaciones o aplicaciones de parches de software; qué debe hacerse cuando un empleado es despedido o se retira voluntariamente de la organización; cómo deben ser conducidas las visitas a los centros de cómputos; y, cómo deben ser las relaciones con empleados temporales [2, p. 24].

La ausencia de seguridad administrativa en las organizaciones es una ventaja para los intrusos, quienes pueden utilizar la ingeniería social como recurso para alcanzar sus objetivos y lograr acceso a los sistemas de cómputo, o en el caso de un empleado despedido, que aún semanas después de haber sido dado de baja de la organización, sigue teniendo privilegios en los sistemas de cómputo y puede usar el sistema para su propio beneficio o en perjuicio de la organización.

1.2.2 Fundamentos.

Conforme los usos y funcionalidades de los sistemas de cómputo han evolucionado, se ha incrementado la complejidad tanto en hardware como en software, así como también la forma en que operan y la manera en que se interconectan los sistemas de cómputo, y se ha requerido desarrollar, adecuar e implantar medidas de seguridad cada vez más sofisticadas,

Es por esto que las medidas de seguridad externas no son suficientes para mantener la seguridad de la información, por lo que se han tenido que incorporar medidas de seguridad internas en los equipos y el software de los sistemas de cómputo. Esto ha generado la necesidad de crear disciplinas especializadas que enfoquen sus esfuerzos a proteger los sistemas de cómputo y sobre todo la información que es procesada, almacenada y transmitida en éstos.

En relación con sus orígenes, la seguridad de la información ha estado presente en la historia del ser humano desde épocas muy remotas, sin embargo, la seguridad informática, como la conocemos hoy, se estructura en los años 50s, con los esfuerzos del gobierno de los Estados Unidos relativos a limitar las emanaciones de radiación eléctrica o electromagnética en los equipos electrónicos, lo que derivó en un conjunto de estándares clasificados, conocidos con el nombre clave de TEMPEST [19]. Pero es hasta la siguiente década, cuando

en 1967 el Departamento de Defensa de los Estados Unidos realiza los primeros estudios sobre las amenazas para sus equipos de cómputo, y a partir de entonces el tema de la seguridad cobra una mayor relevancia y el desarrollo de esta disciplina sale de los reductos militares y gubernamentales y llega también a ámbitos comerciales. Hasta ahora, los avances y aplicaciones de la seguridad informática han sido muchos y muy variados, tanto es aspectos teóricos como prácticos, como se tendrá oportunidad de observar a lo largo de este trabajo.

1.2.2.1 Definición.

En una definición general, la seguridad informática es una disciplina especializada en proteger la información procesada, almacenada o transmitida en los sistemas de cómputo. La protección de la información debe ser entendida como asegurar la confidencialidad, integridad y disponibilidad de la información, mediante la incorporación de procedimientos y tecnología fundamentados en modelos, estándares, algoritmos y protocolos.

Cada uno de los términos vertidos en la definición se describirán más adelante. En principio, la seguridad informática está compuesta por dos aspectos principales, por una parte se encuentran los servicios de seguridad que esta disciplina define y por otra los mecanismos de seguridad mediante los cuales se implementan dichos servicios. Otros conceptos relativos a la seguridad informática los componen la valoración de bienes, vulnerabilidades y amenazas, los cuales son necesarios comprender. A continuación se detalla cada uno.

1.2.2.2 Servicios.

Los servicios de seguridad definen los objetivos específicos a ser implantados mediante un mecanismo de seguridad. Es importante entender exactamente el alcance de cada servicio y la manera en que se relaciona con otros, para que en determinado momento puedan ser implementados adecuadamente. A continuación se describe cada uno.

- **Confidencialidad**. Se refiere a que los datos que procesa o almacena un sistema de cómputo, así como los que son transmitidos entre diferentes sistemas de cómputo, sean dados a conocer sólo a las personas autorizadas para ello.

- **Autenticación**. Es el proceso por el cual es posible identificar a un individuo y validar que efectivamente es quien dice ser, esta validación debe ser indiscutible y demostrable, y puede ser implementada a través de algo que se conoce, se posee o se es.

- **Integridad**. Estipula que los datos que procesa o almacena un sistema de cómputo, así como los que son transmitidos entre diferentes sistemas de

cómputo, se encuentren libres de modificaciones no autorizadas, esto incluye borrar y crear datos de manera no autorizada.

- **Control de acceso**. Como servicio, es el conjunto de procedimientos para garantizar que los sujetos, individuos o procesos, obtengan acceso sólo a los datos que tienen permitido.

- **No repudio**. Es el conjunto de políticas por medio de los cuales una persona no puede negar falsamente su responsabilidad de las acciones que haya realizado o le hayan dirigido en un sistema de cómputo.

- **Disponibilidad**. El objetivo de la disponibilidad es que los sujetos, individuos o procesos autorizados, puedan hacer uso de la información cuando lo requieran.

1.2.2.3 Mecanismos.

Los mecanismos de seguridad consisten en alguna funcionalidad específica para establecer o implementar un servicio de seguridad. Al igual que los servicios, los mecanismos se encuentran interrelacionados unos con otros. A continuación se describen.

- **Cifrado**. Consiste en transformar la información original, llamada *texto en claro*, en información transformada, llamada *texto cifrado*. El texto cifrado normalmente tiene una apariencia de aleatorio e ininteligible. Aunque el texto cifrado es reversible, es decir puede obtenerse el texto en claro a partir del texto cifrado, mediante el proceso inverso conocido como descifrado (siempre y cuando se cuenten con los elementos necesarios para llevar a cabo esto). El mecanismo de cifrado por sí mismo ya que, a través de éste pueden ser implementados los servicios de integridad, confidencialidad y autenticación. En la sección 1.3 más adelante, se profundiza en este tema.

- **Firma digital**. Es una herramienta de seguridad que proporciona evidencia electrónica para autenticar la identidad de quien lo generó. A través de este mecanismo pueden ser implementados los servicios de integridad, autenticación y no repudio. Una firma digital es análoga a una firma manuscrita, dado que a través de estos mecanismos, es posible saber y probar que un documento fue creado por alguien en particular.

- **Control de acceso**. Mediante este mecanismo se regula la entrada a los sistemas de cómputo y a la información almacenada en éstos. Este mecanismo está muy ligado al servicio de autenticación, mismo que identificará a los sujetos ante el sistema, para luego proceder al proceso de autorización de acuerdo a los privilegios que cada sujeto tenga asignados en el sistema.

- **Integridad**. La integridad, como se describió anteriormente es un servicio, pero también es un mecanismo, que implementa algoritmos que garantizan que los datos transmitidos o almacenados en los sistemas de cómputo, no hayan sido alterados de forma no autorizada.

- **Autenticación**. Este mecanismo se implementa mediante diversas técnicas con el objetivo de identificar a los sujetos, individuos o procesos que intervienen en el uso de un sistema de cómputo. Las técnicas utilizadas pueden ser desde el conocimiento de una contraseña hasta dispositivos analizadores de la huella digital, la voz, el iris o los patrones de calor generados por los vasos capilares en la cara de una persona.

En la Tabla 1-1 se muestra la relación que guardan los servicios y mecanismos de seguridad. En cada fila se encuentran listados los servicios de seguridad comentados, en cada columna se ubican los mecanismos de seguridad, y mediante una marca (X) se indican los mecanismos que implementan algún servicio. En esta relación no se incluye el servicio de disponibilidad debido a que se implementa mediante técnicas muy específicas para una aplicación dada.

		MECANISMOS				
		Cifrados	Firma Digital	Control de Acceso	Integridad	Autenticación
SERVICIOS	Confidencialidad	X				
	Autenticación	X	X			X
	Integridad	X	X		X	
	Control de Acceso			X		
	No Repudio		X		X	

Tabla 1-1. Relación de los servicios y mecanismos de seguridad. La marca (X) indica que el mecanismo implementa el servicio.

Como puede observarse en la Tabla 1-1, a través de los mecanismos de cifrado y firma digital pueden implementarse la mayoría de los servicios de seguridad. El cifrado es el mecanismo que puede implementar un mayor número de servicios, por lo que se le tratará a mayor profundidad más adelante.

1.2.2.4 Valoración de bienes.

La valoración de bienes es la importancia relativa de la información que reside, procesa o se transmite entre los sistemas de cómputo de una organización. La información que es menos importante para la operación diaria de la organización, debería tener un valor menor.

Para determinar este valor, deben considerarse los costos monetarios relativos a los equipos, programas y gente involucrada, así como también las pérdidas que se generan como consecuencia de no contar con la información.

Aunque en principio todo se considera importante, es necesario catalogar y asignar prioridades a los bienes de una organización en los siguientes rubros [23]:

- **Hardware.**

Sin equipos no es posible procesar o tener acceso a la información, pero al mismo tiempo, los equipos son relativamente fáciles de reemplazar y tienen un costo financiero identificable. Es necesario tener una lista que identifique las características de los equipos y los servicios que proporcionan, así como también la forma en que se interconectan e interactúan unos con otros.

- **Software.**

El software es un componente del procesamiento de información que manipula los datos para cubrir las necesidades de los usuarios de información. Sin el software, los equipos de cómputo y comunicaciones no sirven para nada. Existe software de índole general que puede ser adquirido relativamente fácil y tiene un costo financiero identificable. Pero para el caso de software propietario, desarrollado dentro de la organización, el costo debe ser equivalente al que ha generado el desarrollo, pruebas y mantenimiento del mismo.

- **Datos.**

Los datos que almacenan los equipos de cómputo, pueden representar distinto valor a través del tiempo, algunos tendrán importancia histórica, otros se requieren para la operación diaria y algunos más tienen relevancia para las operaciones futuras, por lo que deberá asignársele el valor apropiado a cada conjunto de acuerdo al tipo de organización de que se trate.

Además, independientemente de cómo y dónde fueron capturados los datos, el proceso de captura es por sí mismo caro. Por lo que se deben considerar los dispositivos de captura utilizados y los costos asociados para la preservación de los datos (los equipos de respaldo, los medios utilizados, la administración y la verificación de los mismos).

- **Políticas y procedimientos.**

Las políticas y procedimientos conforman la manera en que una organización funciona, y son desarrolladas para disminuir los errores en la toma de decisiones a diferentes niveles. Sin políticas y procedimientos es prácticamente imposible

recuperarse de forma adecuada cuando se presentan problemas, debido a esto es importante identificarlas, catalogarlas y asignarle un valor en base a la importancia que la organización determine. El valor de las políticas y procedimientos estarán en relación al impacto que tienen en la operación diaria de la organización.

- **Gente.**

Las personas en una organización son el bien más preciado debido a que poseen el conocimiento para que la organización funcione. Los bienes descritos anteriormente, pierden valor si no son utilizados por personal calificado. El valor asignado a la gente de una organización debe basarse en el nivel de responsabilidad, los costos que ha invertido la organización en su entrenamiento y las pérdidas que pueden presentarse si no realiza adecuadamente sus deberes.

1.2.2.5 Vulnerabilidades.

Las vulnerabilidades, son una medida real de la implantación de la tecnología actual. Se derivan típicamente de una revisión de la seguridad informática y las prácticas generales aplicadas a los sistemas de cómputo [2, pp. 11-12]. Las vulnerabilidades son debilidades en la implantación de la tecnología que pueden ser explotadas por una entidad maliciosa para obtener mayor acceso del que tiene autorizado a los sistemas de cómputo [6]. En este sentido, todos los sistemas tienen vulnerabilidades que bajo ciertas circunstancias podrán ser explotadas.

Haciendo una referencia a lo bienes identificados anteriormente, se pueden presentar las siguientes vulnerabilidades en cada uno. Cabe aclarar que no es una lista exhaustiva y está asociada a eventos que suceden, cuyas medidas para mitigarlos son generalmente reactivas [23].

- **Hardware**: abuso de los empleados, capacidad insuficiente, dispositivos mal configurados y pérdida o robo de equipo.

- **Software**: "bugs", puertas traseras, virus informáticos[4], bombas lógicas[5], problemas de actualización y de licenciamiento.

- **Datos**: corrupción, pérdida o robo, divulgación accidental, modificaciones incorrectas, problemas de almacenamiento y respaldo.

- **Políticas y procedimientos**: mala implantación, inexistencia, inaccesibles y divulgación accidental.

---

[4] Piezas de software que se diseñan con el objetivo de que se repliquen automáticamente dentro del sistema y hacia otros sistemas de cómputo, utilizando medios diversos de distribución. El comportamiento de estos programas es similar al comportamiento de los virus biológicos, de ahí el nombre.

[5] Piezas de software, generalmente incrustadas en el código de aplicaciones, que ejecutan acciones dañinas a los sistemas, al cumplirse ciertas condiciones específicas.

- **Gente**: accidentes, enfermedades, descontento, hostilidad, mala utilización de la tecnología y falta de ética.

Para el caso de las vulnerabilidades exclusivamente de índole tecnológico, la solución común es aplicar actualizaciones. Pero no todas las vulnerabilidades pueden ser corregidas satisfactoriamente mediante este método, por lo que los administradores de sistemas deben no solo estar atentos a las vulnerabilidades y actualizaciones de la tecnología, sino también tienen que buscar opciones para mitigar el nivel de riesgo de los sistemas de cómputo mediante la utilización de otras técnicas (p. ej.: "Firewalls", listas de control de acceso en ruteadores o software adicional en los equipos de cómputo, aplicaciones de pruebas psicológicas al personal, entre otros). Es un error común entre los administradores de sistemas monitorear solo las actualizaciones de la tecnología y no así las vulnerabilidades que presentan los sistemas de cómputo [6].

1.2.2.6 Amenazas.

Las amenazas son una forma de cuantificar los intereses de la organización en cuanto a los tipos de ataques que pueden ocurrir. Esto permite a una organización concentrar sus esfuerzos y soluciones en seguridad hacia las diferentes fuentes de ataques identificados [2, p. 12].

En la Tabla 1-2 se muestra una clasificación de las amenazas. Como se podrá observar se dividen en naturales y humanos, de los cuales estos últimos son más difíciles de identificar y de prevenir.

NATURALES	**Tornados**		
	Huracanes		
	Erupción volcánica		
	Inundación		
	Terremoto		
HUMANAS	**Hostiles**	**Internos**	**Sofisticados**
			No sofisticados
		Externos	**Sofisticados**
			No sofisticados
	No Hostiles	**Internos**	**Sofisticados**
			No sofisticados
		Externos	**Sofisticados**
			No sofisticados

Tabla 1-2. Clasificación de las amenazas [2, pp. 14-17].

Las amenazas humanas las llevan a cabo sujetos conocidos como "intrusos", "atacantes" o "crackers[6]", los cuales pueden ser clasificados en los siguientes perfiles [25]:

- **Curioso**. Este tipo de intruso está interesado básicamente en conocer los de sistemas y datos que se tienen.

- **Malicioso**. Este intruso intentará modificar el comportamiento del sistema, dejarlo inservible o realizar cualquier otra acción que implique a la parte afectada invertir tiempo y dinero para recuperarse de los daños causados.

- **De alto perfil**. Este tipo de intruso intenta llamar la atención al ingresar de manera no autorizada a los sistemas, con el objetivo de obtener popularidad al demostrar sus "habilidades" para ingresar a los sistemas.

- **Competidor**. Este intruso está interesado en los datos que contiene el sistema, con el objeto de emplearlos para obtener beneficios para sí mismo.

- **Usuario**. Este tipo de intruso está interesado en la utilización de los recursos de cómputo del sistema, con el objeto de ejecutar programas de software que utilizará el mismo para su propio beneficio.

- **De paso**. Este intruso utilizará los sistemas como paso intermedio para atacar otros sistemas.

De manera general, se pueden identificar 4 tipos de amenazas que explotan las vulnerabilidades de los activos en el sistema; si partimos de que una comunicación normal entre dos entidades se realiza como se indica en la Figura 1-2:

Figura 1-2 Flujo normal de información en una comunicación entre dos partes.

- **Interrupción.**

Un activo del sistema se pierde, no está disponible o está inutilizable. En este caso, el emisor no puede establecer la comunicación. Esto se ilustra en la Figura 1-3. Este es el tipo de amenazas que se presentan con un intruso malicioso.

[6] Es importante saber diferenciar entre lo que es un "cracker" y un "hacker". Por lo general un cracker es un sujeto que intenta hacer daño a los sistemas, mientras que un hacker es un sujeto al cual le gusta conocer a profundidad la tecnología y con ello resolver problemas. Mayor información en http://www.tuxedo.org/~esr/faqs/hacker-howto.html.

Figura 1-3. Comunicación interrumpida.

- **Intercepción.**

Alguna parte no autorizada logra acceso a un activo del sistema. En este caso, el emisor establece comunicación, pero sin que las partes involucradas lo sepan, una tercera parte tiene acceso a la información transmitida. Esto se ilustra en la Figura 1-4. Este tipo de amenaza es implementada principalmente por intrusos curiosos, maliciosos, de alto perfil o competidores.

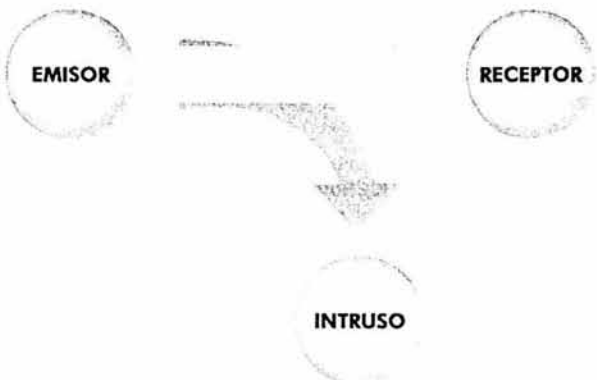


Figura 1-4. Comunicación interceptada.

- **Modificación.**

Una parte no autorizada logra acceso al activo del sistema y puede manipular ese activo. En este caso, una tercera parte no sólo tiene acceso a la información transmitida, sino que realiza modificaciones a la misma y la reenvía como si fuera el emisor original de dicha información. Esto se ilustra en la Figura 1-5. Este tipo de amenazas las implementan intrusos de tipo malicioso, de alto perfil, competidor, usuario y de paso.

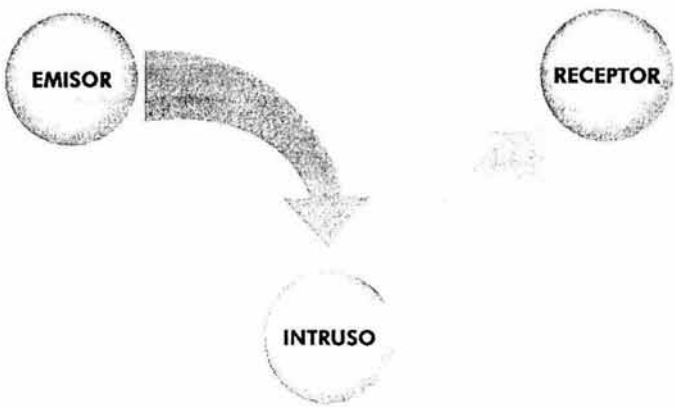


Figura 1-5. Comunicación modificada.

- **Fabricación.**

Una parte no autorizada puede fabricar objetos falsos en un sistema. En este caso, una tercera parte envía información haciéndose pasar por el emisor. Esto se ilustra en la Figura 1-6. Este tipo de amenazas las implementan los intrusos maliciosos, de alto perfil, competidores, usuarios y de paso.

Figura 1-6. Comunicación fabricada.

La relación entre tipos de intrusos y elaboración de tipos de amenazas se muestra en la Tabla 1-3.

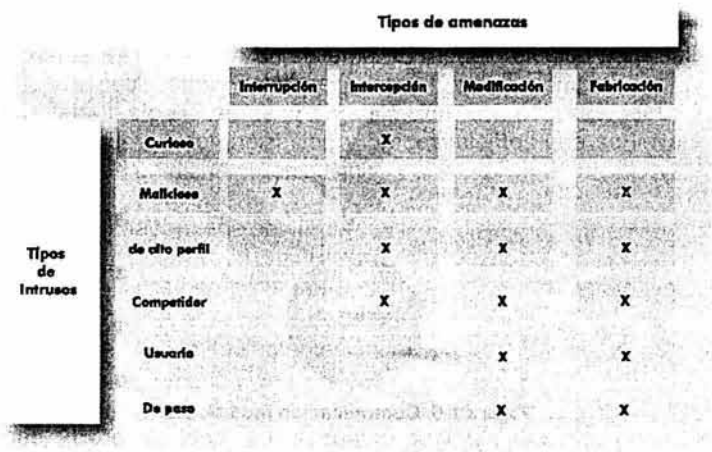


Tabla 1-3. Relación entre tipos de intrusos y amenazas implementadas.

1.2.3 Modelos.

El propósito de un modelo de seguridad es expresar de manera precisa los requerimientos de seguridad de un sistema. El modelo elegido para satisfacer los requerimientos especificados debe ser claro, fácil de comprender, posible de implantar y reflejar las políticas de la organización. Las políticas deben incorporar los requerimientos de una organización y procedimientos que se llevarán a cabo para lograr seguridad en dicha organización.

Un modelo de seguridad, puede servir para:

- probar las políticas de una organización de manera completa y consistente
- documentar dichas políticas
- ayudar a diseñar una implementación
- verificar una implementación

Los sistemas ya construidos, a los cuales se les debe incorporar seguridad, requieren la aplicación de un modelo de seguridad menos formal, ya que los modelos formales son orientados para uso en el diseño y especificación de los sistemas que serán construidos [2, p. 35].

En los sistemas que requieren un alto grado de seguridad, es esencial contar con especificaciones y verificaciones formales del modelo de seguridad. El objetivo de las especificaciones es describir el comportamiento propuesto del sistema en una manera clara que apoyen por sí mismos a los métodos de verificación. El propósito de la verificación de la seguridad es probar de manera inicial que las

especificaciones se ajustan al modelo de seguridad formal. Posteriormente, la implantación debe ser probada para verificar que cumple con las especificaciones [2, pp. 35-36].

Para desarrollar un sistema seguro, debe incluirse un modelo formal para la seguridad como parte de la definición del sistema. El propósito de este modelo es definir de manera precisa el comportamiento deseado de las partes relacionadas a la seguridad del sistema. Este comportamiento deseado es fuertemente influenciado por las amenazas previstas para el sistema y los datos que dicho sistema procesa. Hasta ahora, existen pocos modelos de seguridad desarrollados que hayan sido expuestos de manera amplia, debido principalmente a que no todos los modelos pueden ser aplicados de esta manera. Los principios que residen en los modelos de seguridad, son los siguientes [2, pp. 36-37]:

- **Identidad**. Identificar de manera única cada usuario, programa, objeto y recurso.
- **Responsabilidad**. Los usuarios deben ser responsables de sus acciones.
- **Monitoreo**. Contar con registros de las acciones de los usuarios.
- **Autorización**. Reglas para determinar las entidades que pueden tener accesos a los objetos.
- **Menor privilegio**. Restringir a los usuarios a un conjunto mínimo de recursos necesarios para llevar a cabo su trabajo.
- **Separación**. Separar las acciones de las entidades para que no interfieran o confabulen con las acciones de otras entidades.
- **Redundancia**. Mantener copias de los componentes de hardware, software y datos para asegurar la consistencia, exactitud y oportunidad de los resultados.

Los modelos más representativos se comentan a continuación.

1.2.3.1 Bell y LaPadula.

Durante la década de los años de 1970s, David Bell y Leonard LaPadula desarrollaron el primer modelo matemático de una política de seguridad multinivel. El modelo Bell-LaPadula fue clave para el desarrollo de estándares básicos de seguridad informática y fue fundamental para el desarrollo de otros modelos de seguridad.

El model Bell-LaPadula se concentra en el aspecto de la confidencialidad y señala las propiedades requeridas para prevenir la "fuga" de información. Este modelo asume el hecho de que existe una clasificación de seguridad parcialmente ordenada, y define básicamente dos propiedades que a continuación se describen.

*Sea **C(o)** la clasificación del objeto **o**, y **C(s)** el nivel de autorización del sujeto **s**.*

1. **Propiedad de seguridad simple (ss-property)**: Un sujeto ***s*** puede tener acceso de lectura a un objeto ***o*** sólo si ***C(o) <= C(s)***.

Esta primera propiedad se refiere a que ninguna persona o proceso puede recibir una pieza de información, si no tiene un nivel de autorización tan alto, por lo menos, como el nivel de clasificación de la información que va a recibir. Esto previene la lectura "hacia arriba".

2. **Propiedad estrella (*-property)**: Un sujeto ***s*** quien tiene acceso de lectura a un objeto ***o***, puede tener acceso de escritura a un objeto ***p*** sólo si ***C(o) <= C(p)***.

Esta segunda propiedad asume que la información obtenida de un objeto puede ser transferida a otro objeto solo si el nivel de clasificación del objeto destino es por lo menos tan alto como el del objeto origen. Esto previene la escritura "hacia abajo". En la Figura 1-7 se puede observar el funcionamiento de este modelo.

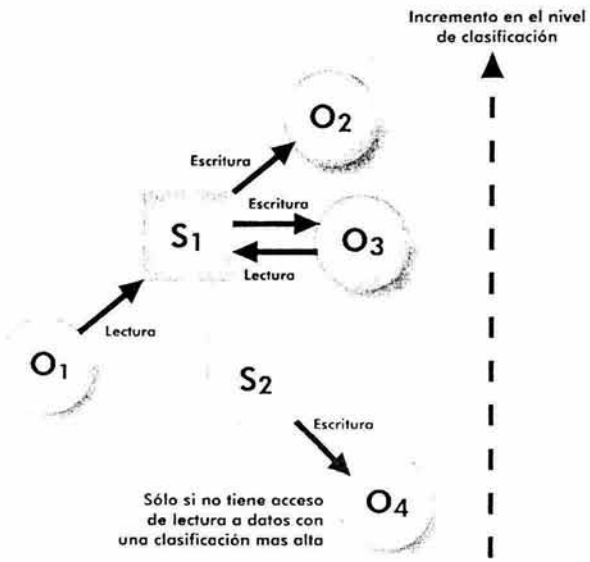


Figura 1-7 Propiedades del modelo Bell-LaPadula.

Este modelo, sin embargo, presenta las siguientes limitantes:

- Está orientado hacia sistemas con niveles de seguridad estáticos, ya que no considera la administración del control de acceso.
- Está restringido a la confidencialidad y no considera la integridad de la información.

- Es posible el uso de canales encubiertos[7].

1.2.3.2 Biba.

Este modelo aborda la limitante del modelo Bell-LaPadula, relacionada a la integridad. El modelo de Biba está basado en una estructura jerárquica de niveles de integridad, los cuales son asignados a sujetos y a objetos.

El nivel de integridad de un objeto es asignado en relación a que tan catastrófico sería si fuera modificado de manera inapropiada por un sujeto. A los sujetos en turno les es asignado un nivel de integridad en relación al propio usuario y al nivel de integridad mínimo requerido para llevar a cabo su trabajo (el principio del mínimo privilegio).

*Sea **I(o)** el nivel de integridad del objeto **o**, e **I(s)** el nivel de integridad del sujeto **s**.*

1. **Propiedad de integridad simple (si-property)**: Un sujeto ***s*** puede tener acceso de escritura a un objeto ***o*** sólo si ***I(o) <= I(s)***.

Esta primera propiedad se refiere a que un sujeto puede modificar un objeto sólo si la fiabilidad del sujeto es por lo menos tan alta como la confianza puesta en el objeto.

2. **Propiedad de integridad estrella (integrity-*-property)**: Un sujeto ***s*** que tiene acceso de lectura a un objeto ***o*** puede tener acceso de escritura a un objeto ***p*** sólo si ***I(p) <= I(o)***.

Esta segunda propiedad expresa esencialmente, que la información obtenida de un objeto no puede ser transferida a otro objeto con un nivel de integridad mayor. Esto lo que hace es prevenir la "corrupción" de la información de un alto nivel de integridad con la información de un menor nivel de integridad. En la Figura 1-8 se pueden observar el funcionamiento de este modelo.

[7] Un canal encubierto es un conducto de comunicación que no es controlado por los mecanismos de seguridad por el cual es posible obtener y transferir información en una manera que viola las políticas de seguridad del sistema. Ejemplos de canales encubiertos son el intercambio de mensajes mediante la presencia de archivos o alteración e interpretación de parámetros de desempeño de los sistemas de cómputo.

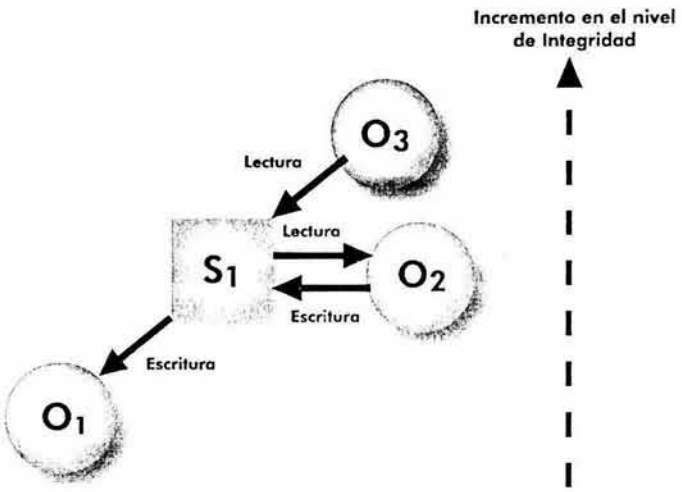


Figura 1-8. Propiedades del modelo de Biba.

1.2.3.3 Clark-Wilson.

Este modelo está enfocado a la integridad basado en las prácticas comerciales de procesamiento de datos. Fue desarrollado por David C. Clark y David R. Wilson, quienes identifican que en los ámbitos comerciales es importante impedir la fuga de información, pero lo es aún más impedir la modificación no autorizada de los datos.

El modelo Clark-Wilson establece dos tipos de consistencias:

- **Consistencia interna**, relativa a las propiedades de estado interno de un sistema, mismas que se encargan de asegurar la consistencia de los datos, y

- **Consistencia externa**, la cual es la relación que guarda la consistencia interna con respecto al mundo "real" y se consigue mediante la aplicación de auditorias.

Los mecanismos utilizados en este modelo para mantener la integridad son:

- **Transacciones bien formadas**, los datos sólo pueden ser manipulados a través de un conjunto específico de programas.

- **Separación de deberes**, los usuarios están restringidos a utilizar sólo un conjunto específico de programas para manipular los datos.

Como se puede observar los usuarios son restringidos en la forma en que manipulan los datos, y los elementos de datos son restringidos a que sólo ciertos procesos o programas pueden manipularlos. Además, ciertas operaciones críticas deben ocurrir en un orden específico, por lo que incluso, puede establecerse que distintos ordenamientos de las operaciones sean realizados por diferentes usuarios. En este modelo un elemento de datos no tiene un nivel de seguridad en particular, sino que está asociado a un conjunto de programas que pueden operar sobre dichos datos. A su vez los usuarios están permitidos sólo a operar ciertos programas. Por lo que, los sujetos y objetos son asociados con los programas, y los programas sirven como un nivel intermedio entre sujetos y objetos.

La formalización de este modelo divide los elementos de datos en restringidos (CDI, Constrained Data Items) y no restringidos (UDI, Uncostrained Data Items) e identifica dos clases de procedimientos: procedimientos de verificación de integridad (IVP, Integrity Verification Procedures) y procedimientos de transformación (TP, Transformation Procedures). Los procedimientos de verificación de integridad confirman que todos los elementos de datos restringidos se ajustan a una especificación de integridad. Los procedimientos de transformación corresponden al concepto de transacciones bien formadas y se encargan de cambiar el conjunto de elementos de datos restringidos de un estado válido a otro.

En relación al control de acceso, este modelo define las operaciones de acceso o procedimientos de transformación que pueden realizarse en cada elemento de dato (tipos de datos), así como las operaciones de acceso que pueden realizar los sujetos (roles).

Existen reglas de certificación (C), relacionadas al oficial de seguridad o custodia del sistema y reglas de aplicación (E), relacionadas al propio sistema. Las políticas de este modelo está definida por cinco reglas de certificación y cuatro reglas de aplicación, que a continuación se enuncian:

C1: Todos los procedimientos de verificación de integridad deben asegurar apropiadamente que todos los elementos de datos restringidos están en un estado válido al momento en que esté funcionando el procedimiento de verificación de integridad.

C2: La validez de todos los procedimientos de transformación debe estar certificada. Es decir, los procedimientos de transformación deben tomar un elemento de dato restringido y llevarlo a un estado final válido. Para cada procedimiento de transformación y cada conjunto de elementos de datos restringidos que el procedimiento manipula, el oficial de seguridad debe especificar una relación que defina dicha manipulación. Una relación de este tipo es expresada de la forma (Tpi, (CDIa, CDIb, ...)), donde la lista de los elementos de datos restringidos definen un conjunto particular de argumentos para los cuales el procedimiento de transformación ha sido certificado.

E1: El sistema debe mantener la lista de relaciones especificadas en la regla C2 y debe asegurar que cualquier manipulación a cualquier elemento de dato restringido es a través de un procedimiento de transformación y que este procedimiento esté manipulando al elemento de dato restringido de acuerdo a lo especificado en alguna relación.

E2: El sistema debe mantener una lista de relaciones expresadas de la forma (userID, TPi, (CDIa, CDIb, ...)), lo cual relaciona a un usuario y los objetos que el procedimiento de transformación puede hacer referencia de acuerdo a los permisos que tenga el usuario.

C3: La lista de relaciones en la regla E2 deben ser cerificadas para cumplir con el requerimiento de separación de funciones.

E3: El sistema debe autenticar la identidad de cada usuario que intente ejecutar un procedimiento de transformación.

Cabe señalar que para esta regla, el modelo Clark-Wilson menciona que podrían aplicarse otro tipo de restricciones como por ejemplo, tener disponible un procedimiento de transformación para ejecución solamente durante ciertas horas del día.

C4: Todos los procedimientos de transformación deben ser certificados para que añadan a un elemento de dato restringido toda la información necesaria para reconstruir la operación realizada (una bitácora).

C5: Cualquier procedimiento de transformación que toma como valor de entrada un elemento de dato no restringido, debe certificarse para que realice sólo transformaciones válidas, o en su defecto que no realice transformaciones, esto para cualquier valor posible del elemento de dato no restringido. La transformación deberá tomar la entrada del elemento de dato no restringido a un elemento de dato restringido, de lo contrario el elemento de dato no restringido será rechazado.

E4: Sólo el agente autorizado para certificar las entidades puede cambiar la lista de tales entidades asociadas con otras entidades, específicamente aquellas asociadas a un procedimiento de transformación. Un agente que puede certificar una entidad no necesariamente tiene derechos de ejecución con respecto a dicha entidad.

En la Figura 1-9 se muestra un diagrama del modelo Clark-Wilson, señalando el uso y relación de las reglas de integridad.

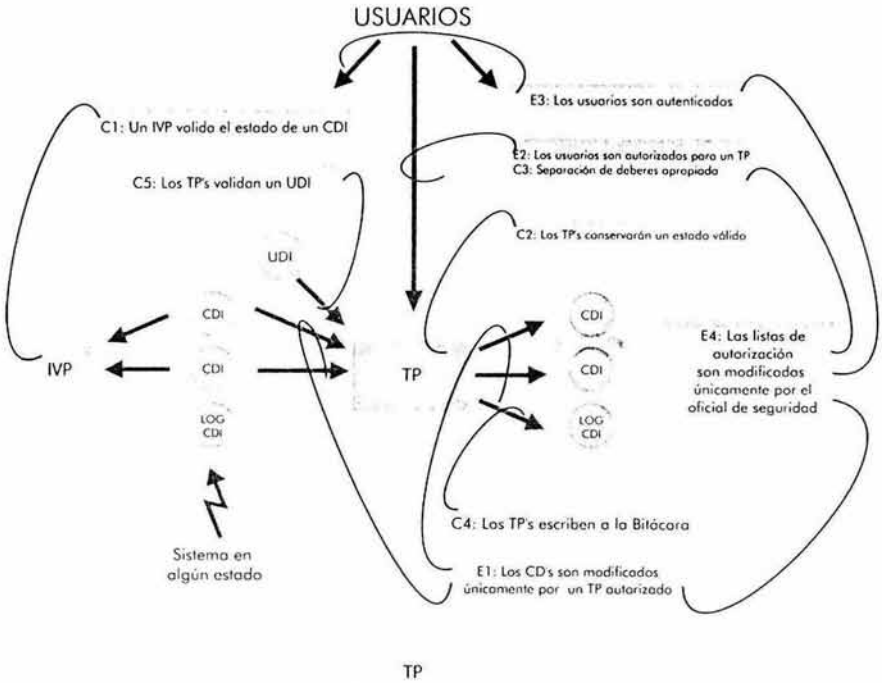


Figura 1-9 Reglas de integridad del modelo Clark-Wilson.

1.2.4 Estándares.

En un intento de alcanzar un alto nivel de seguridad informática coherente, varias organizaciones patrocinadas por diversos gobiernos han establecido sus propios estándares de seguridad informática. Los estándares son usados para determinar la clasificación de seguridad que se le atribuye a un producto de hardware o software. Los estándares identifican el criterio de seguridad que un producto debe cumplir para poder ser utilizado por los gobiernos y organizaciones privadas. Esta clasificación está basada en las características de seguridad implantadas y empleadas por un producto. Naturalmente, cada gobierno puede tener sus propias necesidades de seguridad y por lo tanto, sus propios estándares de seguridad [2, p. 291].

Haciendo una breve cronología de estos estándares, el Departamento de Defensa (DOD) de los Estados Unidos publicó, por primera vez, en 1985 los criterios de evaluación de sistemas de cómputo confiable, denominado Trusted Computer System Evaluation Criteria (TCSEC, también conocido como el Libro Naranja). El propósito de este documento fue proveer criterios de seguridad técnicos y metodologías de evaluación para el soporte de las políticas de seguridad en

sistemas de Procesamiento Automatizado de Datos (ADP, por sus siglas en inglés). Este documento no sólo fue usado en los Estados Unidos sino también en otros países.

En 1990, un grupo formado por cuerpos gubernamentales de Francia, Alemania, Países Bajos y el Reino Unido se reunieron para crear el primer borrador de un documento similar al TCSEC, llamado Information Technology Security Evaluation Criteria (ITSEC). El ITSEC se construyó sobre los conceptos del TCSEC pero con enfoques diferentes en algunos temas.

Al igual que el ITSEC, el Canadian Trusted Computer Product Evaluation Criteria (CTCPEC) tiene sus orígenes en el TCSEC. A diferencia del TCSEC, la premisa fundamental del CTCPEC es la separación de la funcionalidad y la seguridad. Sin embargo, esto no afectó la relación cercana entre el CTCPEC y el TCSEC. Un resultado directo de esta relación fue la actualización del TCSEC al Federal Criteria.

En febrero de 1991, seis años después de publicarse el TCSEC, fue desarrollado el Federal Criteria (FC). El FC fue construido sobre el TCSEC reconociendo e incorporando nuevas tecnologías como ambientes operativos que incluyen estaciones de trabajo con capacidades de múltiples ventanas. Estas actualizaciones fueron hechas en un intento para promover una armonía internacional en el desarrollo y uso de criterios de clasificación de seguridad. Debido a esto, se desarrollaron esfuerzos para la creación del Common Criteria (CC), el cual es un intento para combinar los criterios de seguridad existentes en un estándar unificado. Este proceso comenzó en 1993 y en 1996 se liberó la versión 1.0 y con la retroalimentación recibida se liberó en 1998 la versión 2.0 conocida como "Evaluation Criteria for Information Technology Security". Es importante notar que esto no significa que sea la versión final del CC, ya que está diseñado para crecer y evolucionar conforme la tecnología y otros requerimientos cambien [2, pp. 292-293, 301]. En la Figura 1-10 se puede observar la relación entre los diferentes estándares.

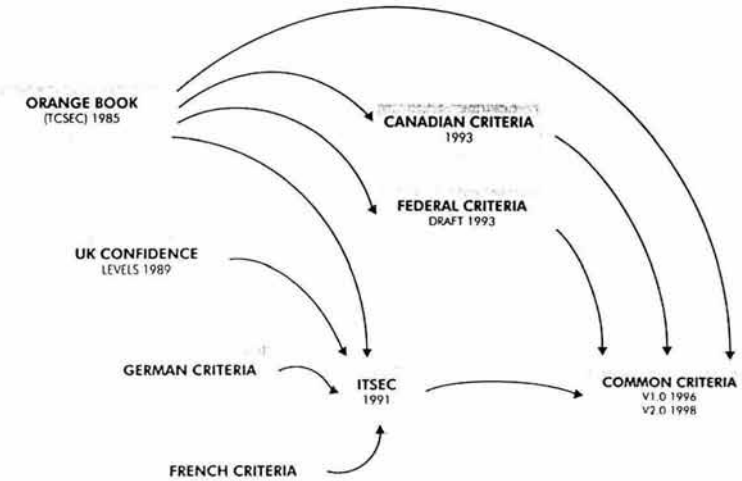


Figura 1-10. Evolución de los estándares de seguridad informática.

En diciembre del año 2000 la Organización Internacional para la Estandarización (International Organization for Standarization, ISO) publicó la primera versión de la norma ISO 17799, posteriormente liberada en septiembre de 2001 como ISO 17799:2000. Esta norma es un conjunto comprensivo de controles que incorpora las mejores prácticas en seguridad informática, y es esencialmente un estándar genérico de seguridad informática reconocido internacionalmente [13]. El ISO 17799 está basado en el estándar británico BS7799[8], el cual fue publicado en mayo de 1999, una versión que incluyó muchas adiciones y mejoras respecto a versiones anteriores [14]. El ISO 17799 comprende diez secciones principales:

1. Planeación de la continuidad del negocio.
2. Control de acceso a los sistemas.
3. Desarrollo y mantenimiento de los sistemas.
4. Seguridad física y ambiental.
5. Cumplimiento.
6. Seguridad en las personas.
7. Organización de la seguridad.
8. Administración de los equipos de cómputo y comunicaciones.
9. Control y clasificación de bienes.
10. Políticas de seguridad.

[8] Más información en http://www.c-cure.org/.

1.3 Criptografía.

La criptografía es la herramienta más utilizada para la implementación de servicios de seguridad, principalmente por el uso de cifrado y firma digital, los cuales, como se hacía referencia en la Tabla 1-1, implementan la mayoría de servicios de seguridad informática.

1.3.1 Antecedentes.

La palabra criptografía viene del griego *kryptos*, que significa oculto o secreto y *graphos* que significa escritura, y se refiere al conjunto de las técnicas que permiten proteger el secreto de una comunicación. La criptografía ha evolucionado y se ha utilizado de diversas maneras, por ejemplo, el lenguaje escrito por sí mismo fue usado como una forma de criptografía –en la antigua China sólo a los pertenecientes a clases altas se les permitía aprender a leer y escribir- aunque el primer uso documentado de la criptografía fue alrededor de 1900 A.C. en Egipto, donde un escriba usó jeroglíficos fuera de lugar en una inscripción. Otro ejemplo es una tableta de Mesopotamia que data del año 1500 A.C., la cual contiene una fórmula cifrada para preparar cerámica vidriada [9].

La ***criptografía*** es la disciplina relacionada al uso y desarrollo de técnicas para cifrar (o encriptar) y descifrar (o desencriptar) mensajes, por otro lado el ***criptoanálisis*** es el intento de romper una técnica criptográfica específica. La ***criptología*** es el campo de la ciencia que abarca el estudio tanto de la criptografía como del criptoanálisis.

El proceso de cifrado implica tomar un mensaje (comúnmente conocido como "texto plano" o "texto en claro") y modificarlo para ocultar el significado original para todos los observadores intermedios, excepto para los destinatarios elegidos. El descifrado es el proceso que toma el mensaje cifrado (conocido también como "texto cifrado") y recupera el mensaje original [2, p. 225]. El proceso de transformación del texto en claro a texto cifrado, involucra la utilización de un algoritmo y una llave[9], misma que es utilizada para hacer este proceso único. En la Figura 1-11 se puede observar en forma esquemática este proceso.

Figura 1-11. Procesos de cifrado y descifrado.

[9] La llave permite la ubicuidad del algoritmo utilizado para el proceso de cifrado/descifrado.

Los procesos de transformar el texto en claro a texto cifrado y este último volverlo a transformarlo al texto en claro original, requiere que un par de transformaciones se lleven a cabo. Estas transformaciones utilizan funciones matemáticas que incorporan una pieza adicional de datos, conocida como "llave", para realizar las transformaciones requeridas. La llave es mantenida en secreto, de tal manera que sólo las entidades autorizadas pueden descifrar el mensaje. Las transformaciones pueden ser representadas de la siguiente forma [2, p. 226]:

*Sea **C** el texto cifrado, **E** la función para cifrar, **K** la llave, **M** el mensaje en claro, y **D** la función para descifrar.*

$$C = E_{[K]}(M)$$

$$M = D_{[K]}(C)$$

Estas transformaciones se conocen como algoritmo criptográfico, y son funciones matemáticas usadas para cifrar y descifrar. Si la seguridad de un algoritmo está basada en mantener en secreto la forma en que un algoritmo funciona, se le conoce como algoritmo restringido. Los algoritmos restringidos son de interés histórico, y son inadecuados a los requerimientos actuales. La criptografía moderna utiliza, como se ha visto antes, una llave (denotada por ***K***) para que la seguridad del cifrado no dependa de mantener en secreto el algoritmo, con lo cual la seguridad se garantiza aún conociendo el funcionamiento del algoritmo utilizado para cifrar la información que se desea proteger [18].

Cabe señalar que la criptografía es una de las herramientas más utilizadas para implementar servicios de seguridad relativos a confidencialidad, autenticación, integridad y no repudio.

1.3.2 Tipos de cifrados.

Existen básicamente dos tipos de cifrados, los de sustitución y los de transposición. Lo importante de un cifrado es que no debe basarse en el secreto del algoritmo, sino en la seguridad del algoritmo y de la llave utilizada.

1.3.2.1 Cifrados de sustitución.

Los cifrados de sustitución están basados en el principio de reemplazar cada caracter con otro caracter distinto con el objetivo de esconder el significado del mensaje. La sustitución por sí misma no es nueva y tampoco es usada siempre para esconder el significado de un mensaje. Un ejemplo de sustitución bien conocida es el código Morse, el cual fue creado no para mantener en secreto un mensaje, sino para facilitar la transmisión de mensajes utilizando distintos medios. Otra sustitución es el alfabeto Braille, el cual es usado por aquellas personas con impedimentos para ver caracteres impresos [2, p. 226]. Algunos cifrados de

sustitución para ocultar el significado del mensaje, son: cifrado César y sus variaciones, cifrado Vigenere y One Time Pads (donde la llave es tan grande como el mensaje).

1.3.2.2 Cifrados de transposición.

Los cifrados de transposición se distinguen de los cifrados de sustitución en que no cambian los caracteres por otros, sino el orden en el cual aparecen en el mensaje [2, p. 233]. Existen diferentes formas de intercambiar el orden de los caracteres del mensaje con el objeto de hacer más difícil la obtención del mensaje original utilizando criptoanálisis.

1.3.3 Algoritmos.

En una definición formal, un algoritmo es un conjunto de pasos finitos y ordenados para resolver un problema. Un algoritmo puede ser fácilmente traducido a un programa de computadora para implementar la solución. En relación a la criptografía, los algoritmos para cifrar y descifrar pueden ser clasificados por una parte por el proceso que realizan en bloque o en flujo, y por otra por el tipo de llaves utilizadas para realizar el cifrado y descifrado conocidos como simétricos y asimétricos. En este punto se comentarán estos últimos por resultar de mayor relevancia al presente trabajo, así como las funciones hash y las implementaciones de estos algoritmos.

1.3.3.1 Algoritmos simétricos.

Un criptosistema simétrico es aquel que usa la misma llave tanto para el proceso de cifrar como para el proceso de descifrar[10]. Si dos individuos desean comunicarse, deben obtener un valor de datos que sea utilizado como llave. Este valor es mantenido en secreto por ambas partes para proteger el contenido de un mensaje que haya sido cifrado con dicha llave [2, p. 235]. También se le conoce como algoritmos de llave privada.

El proceso de los algoritmos de llave privada, puede ser expresado de la siguiente manera:

*Sea **C** el texto cifrado, **E** la función para cifrar, **K** la llave, **M** el mensaje en claro, y **D** la función para descifrar.*

$$C = E_{[K]}(M)$$

$$M = D_{[K]}(C)$$

[10] En realidad, pueden ser llaves diferentes, sólo que la llave de cifrado puede ser calculada a partir de la llave de descifrado y viceversa, y comúnmente se utiliza la misma llave [47, p. 4].

Como puede observarse, se utiliza la misma llave para cifrar y descifrar. Y el proceso de acuerdo de llave no es parte del algoritmo. Por lo que los criptosistemas simétricos tienen tres problemas principales [2, p. 235]:

1. Si un tercer individuo o parte obtiene la llave, los mensajes enviados entre las dos partes originales estarán comprometidos. Además, la tercer parte puede enviar mensajes cifrados a las dos partes originales.
2. La llave debe ser distribuida de una manera secreta.
3. Se requiere un gran número de llaves para permitir que diferentes parejas de individuos se comuniquen en forma secreta. El número de llaves requeridas es de hecho $(n^2-n)/2$.

La seguridad de un algoritmos simétrico radica en la llave; divulgar la llave involucra que cualquier individuo puede cifrar y descifrar mensajes [18].

Los algoritmos de este tipo utilizados de manera estándar son DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm) y AES (Advanced Encryption Standard), este último sustituye a DES. La ventaja de utilizar estos algoritmos es que son relativamente rápidos en su funcionamiento.

1.3.3.2 Algoritmos asimétricos.

Los criptosistemas asimétricos (también conocidos como algoritmos de llave pública[11]) se distinguen de los criptosistemas simétricos en que utilizan llaves diferentes (pero relacionadas) para los procesos de cifrar y descifrar. Los criptosistemas asimétricos se basan en el concepto de criptografía de llave pública, mismo que fue introducido por Whitfield Diffie y Martin Hellman en 1976 [2, p. 244].

El proceso de los algoritmos de llave pública, puede ser expresado de la siguiente manera:

Sea ***C*** *el texto cifrado,* ***E*** *la función para cifrar,* K_C *la llave para cifrar,* K_D *la llave para descifrar,* ***M*** *el mensaje en claro, y* ***D*** *la función para descifrar.*

$$C = E_{[KC]}(M)$$

$$M = D_{[KD]}(C)$$

Como puede observarse y ya se mencionó, se requieren dos llaves: una para cifrar y otra para descifrar. Este proceso no requiere acuerdo previo de la llave a utilizar, debido a que una de las llaves es pública y puede ser conocida ampliamente.

---

[11] Llamados así porque la llave utilizada para cifrar puede hacerse pública sin comprometer la seguridad del cifrado, ya que la llave de descifrado no puede ser calculada (al menos en una cantidad de tiempo razonable) a partir de la llave de cifrado [47, p. 4].

Algo importante de mencionar es que los algoritmos simétricos y asimétricos pueden ser combinados para obtener las ventajas de ambos en una comunicación cifrada. Dado que los requerimientos computacionales de los algoritmos simétricos son menores a los requeridos por los algoritmos asimétricos, pueden utilizarse estos últimos para el proceso de acuerdo de llave de un algoritmo simétrico y a partir de ahí seguir utilizándolo para mantener una comunicación cifrada más eficiente.

Existen dos modos de operación de los sistemas de llave pública. La llave públic puede ser usada para cifrar el texto en claro, o puede ser usada para descifar e texto cifrado. Un sistema que trabaja en sólo uno de estos modos es conocido como ***criptosistema irreversible de llave pública***. Uno que trabaja en ambas direcciones es conocido como ***criptosistema reversible de llave pública***. La razón por la que se puede utilizar la llave pública para descifrar es para autenticación del origen de los datos [2, p. 244].

Los algoritmos de llave pública más conocidos son RSA y ElGamal, mismos que se comentarán más adelante.

1.3.3.3 Funciones hash.

Las funciones hash tienen una entrada variable (texto claro), una salida fija (el hash propiamente) y no utilizan ninguna llave. Una propiedad importante de las funciones hash es que sirven para verificar la integridad de un mensaje o archivo, ya que si se cambia un solo bit del texto en claro, cambiará la salida. Estas funciones son unidireccionales y son fáciles de calcular en un sentido pero no en el otro, además de que no es posible que dos entradas generen la misma salida.

El proceso de las funciones hash, puede ser expresado de la siguiente manera:

*Sea **P** el mensaje de longitud variable, **H** una función de un solo sentido (one-way) y **h** un valor de longitud fija.*

h = H(P)

Algunos de los algoritmos que implementan funciones hash se encuentran MD2, MD4, MD5 y SHA.

Más adelante se comentarán la aplicación de estos algoritmos y funciones.

1.3.3.4 DES.

El algoritmo criptográfico DES (Data Encryption Standard) fue aprobado el 15 de julio de 1977, como el estándar federal de Estados Unidos para cifrar las comunicaciones de información no clasificada del gobierno de ese país. Aunque el

algoritmo es llamado "DES", de hecho es el nombre del estándar, el cual abarca las especificaciones para la implementación y operación [20].

El algoritmo DES está basado en un algoritmo simétrico de cifrado en bloque de 128-bits desarrollado en la década de los años 1960s por IBM, llamado Lucifer, el cual fue parte de un sistema criptográfico experimental. Lucifer es un cifrado de bloque iterativo, que utiliza rondas Feistel, las cuales consisten en cifrar bloques de datos un cierto número de veces, en cada ciclo se aplica la llave utilizada para cifrar a la mitad del bloque y se realizan operaciones XOR con la otra mitad del bloque [20].

DES fue diseñado para usar una llave de 64-bits[12] para cifrar y descifrar bloques de datos de 64-bits, usando un ciclo de permutaciones, intercambios y sustituciones. Para realizar el proceso de cifrar, se aplica una permutación inicial a un bloque de datos, después se realiza una función dependiente de la llave[13] y luego una permutación final. Las permutaciones primera y última toman el bloque de 64-bits y cambian la posición de cada bit de una forma predeterminada. La última permutación es inversa a la permutación inicial.

El algoritmo DES tiene los siguientes modos de operación:

- **ECB** (Electronic Code Book). En este modo, cada bloque de 64-bits es cifrado independientemente usando la misma llave. Este modo es útil sólo para mensajes cortos, ya que bloques de texto en claro idénticos, serán bloques de texto cifrado idénticos también.

- **CBC** (Cipher Block Chaining). En este modo, la entrada para el algoritmo de cifrado es el resultado de la operación XOR del bloque de 64-bits del texto en claro anterior. Este modo es útil cuando se cifran mensajes grandes, debido a que bloques de texto en claro idénticos resultarán en bloques cifrados diferentes.

- **CFB** (Cipher Feed Back). En este modo, la entrada es procesada cierto número de bits cada vez. El texto cifrado anterior es utilizado como entrada para el algoritmo de cifrado para producir una salida pseudo-aleatoria, la cual se ingresa en una operación XOR con el texto en claro para producir el siguiente bloque cifrado.

---

[12] Una llave DES consiste de 64 dígitos binarios de los cuales 56 bits son generados aleatoriamente y usados directamente por el algoritmo. Los otros 8 bits, los cuales no son usados por el algoritmo, son usados para detección de errores. Cada uno de estos 8 bits, es la paridad de cada byte de 8-bits de la llave [14].

[13] Una función toma la llave de 64-bits y a partir de ésta crea un conjunto de 16 bloques de 48-bits. Cada bloque es usado en cada una de las 16 rondas de la función de cifrado, lo cual altera el texto en claro de acuerdo con la especificación de una caja-S. Las definiciones de la caja-S fueron especificadas por el gobierno de los Estados Unidos, y han sido objeto de mucha controversia porque son cruciales en la fortaleza del algoritmo.

- **OFB** (Output Feed Back). Este modo es similar al CFB, sólo que se utiliza la salida anterior como entrada.

Los modos CBC, CFB y OFB usan un vector de inicialización (IV, Initialization Vector), el cual es utilizado como la primer entrada para el algoritmo de cifrado y es utilizado en una operación XOR con el primer bloque de 64-bits del texto en claro. Sin este vector el primer bloque de 64-bits de texto en claro resultaría similar al texto cifrado como en el modo ECB. En los modos CBC y OFB el vector, opcionalmente, puede ser diferente cada vez que el texto en claro es cifrado, mientras que en el modo CFB el vector, obligatoriamente, tiene que ser diferente, porque de lo contrario un criptoanalista puede obtener el bloque de texto en claro a partir del cifrado.

El algoritmo DES ha sido utilizado ampliamente durante muchos años, pero ya no es considerado adecuadamente seguro debido a que una llave de 56-bits puede ser obtenida mediante la utilización de fuerza bruta en un periodo de tiempo relativamente corto (dependiendo de la velocidad de procesamiento de las computadoras utilizadas). DES ha sido reemplazado por AES (Advanced Encryption Standard).

Una variante de DES que es conveniente es 3DES (triple DES), la cual se comenta a continuación.

- ***3DES (triple DES).***

Esta modificación utiliza tres procesos de cifrado y dos llaves. En el proceso de cifrado, se utiliza el algoritmo DES para cifrar-descifrar-cifrar, aplicando la primer llave al primer proceso, la segunda llave al segundo proceso y nuevamente la primer llave al tercer proceso. Para el proceso de descifrado, se aplica el algoritmos DES para descifrar-cifrar-descifrar, aplicando la primer llave al primer proceso, la segunda llave al segundo proceso y nuevamente la primer llave al tercer proceso. Este proceso está ilustrado en la Figura 1-12.

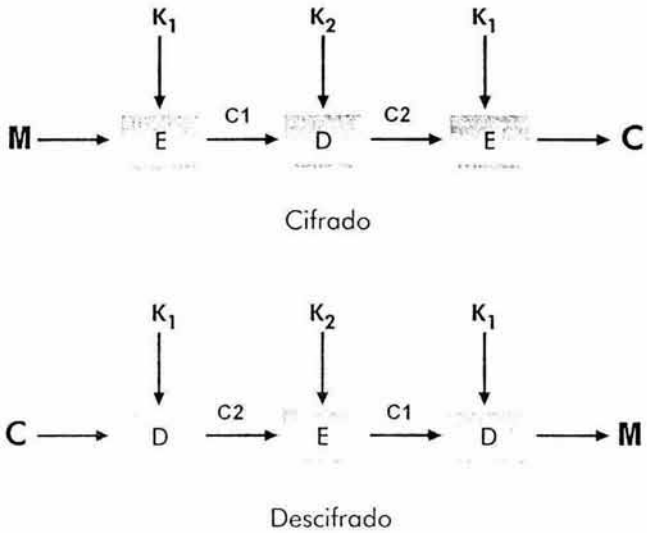


Figura 1-12. Esquema de cifrado/descifrado triple des con dos llaves.

1.3.3.5 AES.

AES (Advanced Encryption Standard) es el nuevo estándar para la protección de la información definido por los Estados Unidos para proteger cierto nivel de la información y las comunicaciones del gobierno federal de ese país. El proceso de selección de un algoritmo AES comenzó en 1997, y el nuevo estándar, así como la especificación del algoritmo, fue aprobado en noviembre de 2001.

El algoritmo AES está basado en el algoritmo simétrico Rijndael[14], el cual consiste de una red de transformación y sustitución lineal con 10, 12 o 14 rondas dependiendo del tamaño de la llave. Para cifrar un bloque de datos utilizando Rijndael, dicho bloque debe ser dividido en un arreglo de bytes, y cada operación de cifrado es realizada a nivel de bytes. La función de ronda de este algoritmo consiste de cuatro niveles. En el primer nivel, una caja-S 8x8 es aplicada a cada byte. El segundo y tercer niveles son niveles de intercambios lineales, en los cuales las filas del arreglo son corridas y las columnas son intercambiadas. En el cuarto nivel, se realizan cálculos XOR con bytes extraídos de la llave y bytes extraídos del arreglo. En la última ronda, no se realiza el intercambio de columnas [21].

El algoritmo Rijndael es un cifrado de bloque iterado, lo que significa que el bloque de entrada inicial y la llave llevan a cabo múltiples ciclos de transformación antes de producir la salida. Cada resultado de cifrado intermedio es llamado Estado.

[14] El algoritmo Rijndael fue diseñado por Joan Daemen y Vincent Rijmen.

Este algoritmo puede operar sobre un bloque de longitud variable usando llaves de longitud variable, y aunque Rijndael soporta la utilización de llaves de 128, 192 y 256 bits para cifrar bloques de datos de 128, 192 y 256 bits y sus nueve combinaciones posibles (de tamaño de llave y longitud de bloque), el algoritmo AES contiene sólo algunas de las capacidades totales del algoritmo Rijndael[15]. El algoritmo está escrito para que las longitudes tanto de la llave como del bloque puedan ser extendidos fácilmente en múltiplos de 32-bits, y el sistema está específicamente diseñado para una implementación eficiente en hardware o software sobre diversos procesadores. El diseño del algoritmo tiene una fuerte influencia del cifrado de bloque llamado SQUARE[16].

1.3.3.6 IDEA.

IDEA (International Data Encryption Algorithm) es un algoritmo de cifrado simétrico patentado por la empresa suiza Ascom, pero se permite el uso para fines no comerciales. Opera sobre bloques de 64-bits pero a diferencia de DES, el algoritmo IDEA utiliza llaves de 128-bits y es considerado por algunos superior a DES. Algo interesante de destacar es que el algoritmo IDEA no utiliza tablas predefinidas o cajas-S.

Este algoritmos utiliza 52 subllaves, cada una de una longitud de 16-bits que son utilizadas en cada una de las ocho rondas[17]. Cada bloque es dividido en cuatro subbloques de 16-bits cada uno, y se le aplican, en conjunto con las subllaves, diversas operaciones de adición de enteros módulo 2^{16}, multiplicación de enteros módulo $2^{16}+1$ y XOR. Entre cada ronda, el segundo y tercer subbloque son intercambiados.

Cabe señalar que las implementaciones en software son comparables en su velocidad de funcionamiento a las implementaciones DES, pero debido a que utiliza llaves más grandes, un ataque por fuerza bruta[18] a IDEA tomaría mucho más tiempo que DES.

Al igual que DES, el algoritmo IDEA tiene los siguientes modos de operación:

- **ECB** (Electronic Code Book).
- **CBC** (Cipher Block Chaining)
- **CFB** (Cipher Feedback Block)

---

[15] Por ejemplo, AES sólo soporta un tamaño de bloque de 128-bits.

[16] También diseñado por Daemen y Rijmen.

[17] Seis de estas subllaves son utilizadas en cada iteración, las otras dos con usadas en el siguiente bloque. Después que todas las subllaves son utilizadas, la llave es corrida 25 bits a la izquierda y dividida nuevamente en ocho subllaves.

[18] Un ataque por fuerza bruta consiste en obtener el texto en claro, a partir del texto cifrado conocido, probando todo el espacio de llaves posibles que pueden formarse. Es un proceso de prueba y error.

- **OFB** (Output Feedback Block)

1.3.3.7 Diffie-Hellman.

En 1976, Whitfield Diffie y Martin Hellman publicaron una solución conocida como *criptografía de llave pública*, con la cual dos individuos pueden generar y distribuir una llave secreta en un ambiente público o no controlado y por lo tanto inseguro. La seguridad de esta técnica radica en el problema de logaritmo discreto[19].

El protocolo tiene dos parámetros públicos ***p*** y ***g*** que pueden ser utilizados por cualquier usuario. El parámetro ***p*** es un número primo y el parámetro ***g*** (conocido como generador) es un entero menor que ***p***, con la siguiente propiedad: para cada número ***n*** entre 1 y ***p***-1 (inclusive), existe una potencia ***k*** de ***g*** tal que $n = g^k \bmod p$ [22].

Este intercambio de llaves es vulnerable al ataque de "hombre en medio"[20], lo cual se debe a que el protocolo Diffie-Hellman no considera la autenticación de los participantes y una posible solución es el uso de firmas digitales y otras variantes del protocolo como el Diffie-Hellman autenticado, también conocido como protocolo Station-To-Station (STS).

En años recientes, este protocolo ha sido utilizado como una técnica criptográfica más general, siendo el elemento común la derivación de un secreto compartido (la llave) a partir de una llave pública de una parte y una llave privada de otra. Los pares de llaves pueden ser generados nuevamente para cada ejecución del protocolo, como en el protocolo original Diffie-Hellman.

1.3.3.8 RSA.

RSA es un algoritmo de llave pública que implementa tanto cifrado como firma digital. Fue desarrollado en 1977 por Ronald Rivest, Adi Shamir y Leonard Adleman, y se basa en el problema de factorizar número grandes. RSA es un acrónimo formado por la primera letra de los apellidos de sus creadores.

Este algoritmo funciona de la siguiente forma, a partir de dos número primos grandes, ***p*** y ***q***, se calcula su producto $n = pq$. Luego, se selecciona un número ***e*** menor que ***n*** y que sea primo relativo a $(p-1)(q-1)$, es decir, que ***e*** y $(p-1)(q-1)$ no

---

[19] El problema de logaritmo discreto asume que es computacionalmente infactible calcular un valor secreto $k = g^{ab} \bmod p$, dados dos valores públicos $g^a \bmod p$ y $g^b \bmod p$ cuando el valor p es un número primo suficientemente grande.

[20] El ataque de hombre en medio (man-in-the-middle, en inglés) consiste en que un tercer participante (***C***) se coloca en medio de dos individuos (***A*** y ***B***) que realizan un intercambio de llaves, e intercepta la llave pública de ***A*** y en su lugar le envía a ***B*** la llave pública de ***C***. Hace esto mismo para el caso en el que ***B*** envía su llave pública, enviándole a ***A*** la llave pública de ***C***, con lo cual permite que ***A*** y ***B*** se comuniquen, pero ***C*** intercepta (y posiblemente modifica) todos los mensajes realizados entre ***A*** y ***B***, debido a que ***C*** funciona como un repetidor entre ***A*** y ***B***.

tengan factores comunes excepto 1. Después, se debe encontrar otro número ***d*** que ***(ed-1)*** sea divisible entre ***(p-1)(q-1)***. Los valores ***e*** y ***d*** son llamados exponentes públicos y privados respectivamente. La llave pública es el par ***(n, e)***; la llave privada es ***(n, d)***. Los factores ***p*** y ***q*** pueden ser destruidos o mantenidos con la llave secreta [22].

1.3.3.9 ElGamal.

ElGamal es un algoritmo de llave pública basado en el problema de logaritmo discreto y sirve tanto para cifrado como para firma digital. El algoritmo de cifrado es similar al protocolo de acuerdo de llave de Diffie-Hellman. Los parámetros consisten de un número primo ***p*** y un número entero ***g***, cuyas potencias módulo ***p*** generan un gran número de elementos.

ElGamal y RSA implementan una seguridad similar cuando se utilizan llaves de la misma longitud. La principal desventaja de ElGamal es el requerimiento de aleatoriedad y un lento funcionamiento (especialmente para firma digital). Otra desventaja potencial del algoritmos ElGamal es que expande el mensaje en un factor de dos durante el cifrado. Sin embargo, dicha expansión es insignificante si se utiliza ElGamal sólo para el intercambio de llaves secretas.

1.3.3.10 DSA y DSS.

DSA (Digital Signature Algorithm) es un algoritmo para firma digital desarrollado por el NIST y es una variación de los algoritmos de firma digital de Schnorr y ElGamal, por lo que DSA está basado en el problema de logaritmo discreto.

DSS (Digital Signature Standard) es parte del proyecto Capstone del gobierno de los Estados Unidos y es el estándar de autenticación digital del gobierno de ese país. En 1991, el NIST propuso que DSA fuera aceptado como DSS. En 1994 se formalizó que DSA fuera DSS.

DSA ha sido criticado por la industria de cómputo desde su anuncio. Las críticas están enfocadas a la carencia de intercambio de llaves, a que es muy reciente y se ha revisado poco, a que la verificación de firmas es muy lenta, a que la mayoría de los fabricantes de hardware y software han estandarizado a RSA y a que el proceso de selección fue secreto y arbitrario. Otras críticas fueron resueltas modificando la propuesta original.

1.3.3.11 MD2, MD4 y MD5.

MD2, MD4 y MD5 (Message-Digest) son algoritmos o funciones hash desarrollados por Rivest y que están enfocados como apoyo para la firma digital de mensajes grandes, donde se convierte dicho mensaje a una forma única más pequeña que es la que se firma.

Estos algoritmos toman un mensaje de longitud arbitraria y producen una salida de 128-bits. Aunque las estructuras de estos algoritmos son muy similares, el diseño de MD2 fue orientado para máquinas de 8-bits de procesamiento, mientras que MD4 y MD5 fueron desarrollados para máquinas de 32-bits de procesamiento.

1.3.3.12 SHA

SHA (Secure Hash Algorithm) fue desarrollado por NIST y es especificado en el Secure Hash Standard (SHS, FIPS 180). SHA-1 es una revisión a la versión original y fue publicado en 1994. SHA-1 toma como entrada un mensaje menor a 2^{64} bits de longitud y produce una salida de 160-bits (20 bytes) y aunque es más lento que MD5, su salida de mayor tamaño lo hace más resistente a ataques por fuerza bruta.

1.3.4 Esteganografía.

Un área similar a la criptografía en su propósito, pero diferente en su aplicación es la esteganografía. La esteganografía es el arte de ocultar información en una manera tal que previene su detección. La información no necesariamente es alterada de la forma en que el cifrado lo hace, sino que simplemente no es percibida por los individuos que no cuenten con los elementos o conocimientos para recuperar la información [2, p. 253].

Utilizando técnicas esteganográficas puede ocultarse información en una imagen digital, que a simple vista no presenta alteraciones, pero que al procesar a nivel de bits dicha imagen puede obtenerse la información oculta. La esteganografía es utilizada principalmente en imágenes digitales, pero también se usa en texto, sonido y archivos binarios.

Es de observar que la esteganografía por sí misma no es segura, ya que su seguridad radica en el desconocimiento de la forma en que se ocultó el mensaje (seguridad por oscuridad[21]). Pero si además el mensaje a ocultar se cifra antes de incorporarlo al medio que lo alojará, el archivo resultante es mucho más seguro [2, p. 255].

[21] "Seguridad por oscuridad" es un término empleado al hecho de confiar en que algo no se descubrirá, y sólo basa su seguridad en el hecho de mantener un secreto, por lo que, en el momento que se averigüe dicho secreto, la seguridad no existe más. La seguridad informática no debe establecerse bajo este concepto, sino que debe basarse en modelos y algoritmos verificables, que no importando que se conozca a fondo su funcionamiento, sigan siendo confiables y seguros.

1.4 Herramientas de seguridad.

Las herramientas de seguridad representan una parte muy importante de la práctica de la seguridad informática, ya que funcionan como elementos de apoyo en la implementación de esquemas de seguridad en sistemas de cómputo.

Cabe señalar que las herramientas comentadas más adelante están orientadas para su utilización en sistemas operativos Linux, ya que este sistema operativo será usado en el desarrollo del prototipo, además de que todas cuentas con licencias de uso libre que permiten obtenerlas en la mayoria de los casos con el código fuente incluido. Existen herramientas de seguridad (adicionales y complementarias) que pueden ser configuradas en sistemas Unix y Windows, entre otros.

1.4.1 Evolución.

La evolución de las herramientas de software en el área de la seguridad informática, responde al incremento en la sofisticación de los ataques a los sistemas de cómputo: desde los errores básicos de programación hasta los complejos ataques de negación de servicio distribuido.

En la Figura 1-13 se ilustra la evolución de los ataques. Se puede observar que por una parte la complejidad y sofisticación de los ataques se ha incrementado. Por otra parte, el conocimiento que requiere un atacante para llevar a cabo sus cometidos disminuye, en razón a que el crecimiento de las redes públicas de información ha permitido una distribución más amplia de herramientas de software que automatizan y replican métodos de ataques a la seguridad de los sistemas de cómputo, lo que permite a los atacantes tener mayores probabilidades de éxito en su propósito [15].

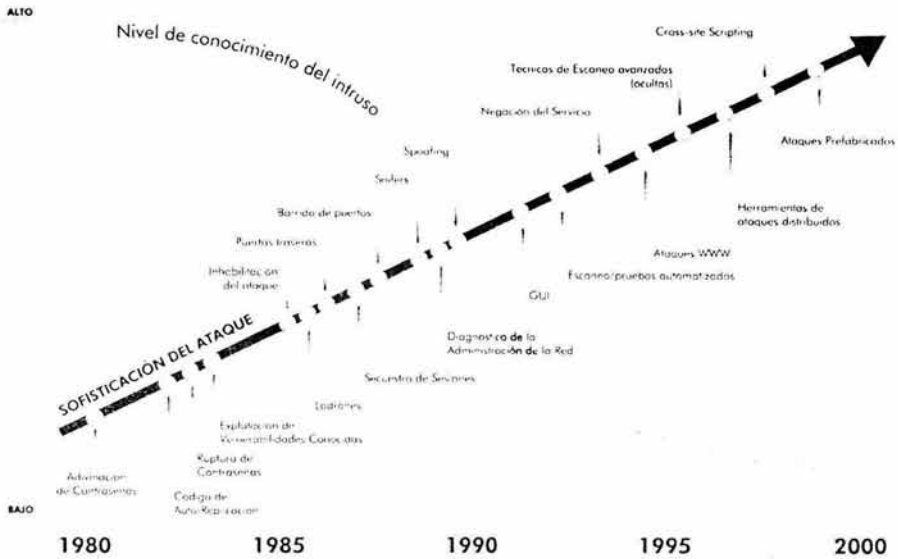


Figura 1-13. Evolución de los ataques informáticos.

1.4.2 Clasificación.

La clasificación de las herramientas de seguridad puede ser muy variada en función de los aspectos que se quieran resaltar. Así, desde el punto de vista de un atacante pueden clasificarse en reconocimiento, obtener acceso y cubrir pistas, desde el punto de vista del encargado de la seguridad de los sistemas, pueden dividirse en defensa y aseguramiento [16]. Sin embargo, puede asumirse una clasificación más descriptiva aunque no demasiado extensa, de la siguiente manera [17].

1.4.2.1 Monitoreo.

En este rubro se tienen herramientas cuya función principal está asociada a revisar el funcionamiento de las aplicaciones, el tráfico en la red o el comportamiento interno del sistema de cómputo.

Netcat. http://www.atstake.com/research/tools/index.html.
Esta utilería lee y escribe datos a través de conexiones de red usando los protocolos TCP o UDP. Está diseñada para se un servicio de red confiable que puede ser utilizado directamente, o manejado de manera relativamente sencilla, por otros programas o scripts. Al mismo tiempo, es una herramienta con muchas funcionalidades para la exploración y puesta a punto de aplicaciones en red. Puede crear cualquier tipo de conexión que se requiera y tiene diversas e interesantes capacidades interconstruidas.

Algunas de las funcionalidades de netcat son:

- Realizar conexiones de entrada y salida, TCP o UDP, desde o hacia cualquier puerto.
- Revisión DNS completa (directa ó inversa), con mensajes de advertencia apropiados.
- Capacidad de utilizar cualquier puerto origen.
- Capacidad para usar cualquier dirección de red origen configurada localmente.
- Capacidad interconstruida para realizar barrido de puertos[22] con sistema aleatorio.
- Capacidad interconstruida para eliminar la ruta origen.
- Capacidad para leer argumentos de línea de comando desde la entrada estándar.
- Modo de envío lento, una línea cada *N* segundos.
- Vaciado hexadecimal de los datos enviados y recibidos.
- Capacidad opcional para permitir a otro programa atender las conexiones establecidas.
- Contestador opcional con opciones de telnet.

Nmap. http://www.nmap.org/nmap/.
"Network MAPper" es una utilería para la exploración o auditoría de la seguridad en los servicios de red. Uno de los objetivos principales del proyecto Nmap es proporcionar a los administradores de sistemas una herramienta avanzada para la exploración de los sistemas conectados en red.

Nmap usa paquetes IP para determinar:

- Los servidores que están conectados a la red.
- Los servicios (puertos) que los equipos ofrecen.
- El sistema operativo (y versiones) que corren los equipos.
- El tipo de filtros de paquetes y firewalls[23] utilizados.
- Y muchas otras funcionalidades.

---

[22] Un barrido de puertos o "port scan" es una serie de intentos, realizados por una atacante, para descubrir que servicios de red tiene habilitados un equipo de cómputo y consecuentemente explotar las posibles vulnerabilidades que existan. Consiste básicamente en enviar un mensaje a la vez a cada puerto para identificar, a través de los mensajes de respuesta, el servicio proporcionado por el puerto probado.

[23] Un firewall o "cortafuegos" es un conjunto de programas que protegen el acceso a los recursos de una red a los usuarios de otras redes. Básicamente estos programas examinan los paquetes de red que se transmiten y en base a reglas de configuración, descarta o deja pasar dichos paquetes.

Ntop. http://www.ntop.org
Es una herramienta que revisa el tráfico de la red para mostrar el uso de la misma. Entre los protocolos que maneja se encuentran IP, IPX, DecNet, AppleTalk y NetBIOS.

Entre algunas de sus funcionalidades destacan:

- Ordenamiento del tráfico de la red de acuerdo a los diversos protocolos.
- Ordenamiento del tráfico de la red de acuerdo a diversos criterios.
- Mostrar estadísticas del tráfico de la red.
- Mostrar la distribución del tráfico IP entre los diversos protocolos.
- Analizar el tráfico IP y ordenarlo de acuerdo a las direcciones origen y destino.
- Mostrar una matriz del tráfico IP en una subred, con el objeto de determinar qué equipos se están comuicando.
- Reportes del uso del protocolo IP ordenado por tipo de protocolo.

scanlogd. http://www.openwall.com/scanlogd/
Esta es una herramienta de detección de barrido de puertos TCP, diseñada originalmente como apoyo para mostrar cómo es que un sistema de detección de intrusos (IDS, Intrusion Detection System[24]) tiene que manejar diversos ataques.

Logcheck. http://www.psionic.com/products/logsentry.html
Automáticamente realiza el monitoreo periódico de las bitácoras generadas por diversas aplicaciones de seguridad en los sistemas de cómputo. También proporciona reportes de violaciones de la seguridad o eventos inusuales.

Swatch. http://www.oit.ucsb.edu/~eta/swatch/
Esta herramienta está diseñada para monitorear la actividad de los sistemas de cómputo, es necesario ajustar adecuadamente el archivo de configuración para obtener el mayor beneficio en el reporte de eventos.

IPTraf. http://cebu.mozcom.com/riker/iptraf/
IPTraf es una utilería, no gráfica, para la obtención de estadísticas de la red. Los protocolos que reconoce IPTraf son IP, TCP, UDP, ICMP, IGMP, IGP, IGRP, OSPF, ARP, RARP. Algunas de sus funcionalidades incluyen:

- Un monitor de tráfico IP que muestra información sobre información en las banderas TCP, número de paquetes y bytes transmitidos, detalles ICMP, tipos de paquetes OSPF.
- Estadísticas, generales y detallas, por interfase de red. Con lo que se obtiene el número y tamaño de paquetes IP, TCP, UDP, ICMP y otros

[24] Un IDS, de manera general, es un conjunto de programas que examinan el tráfico de la red o las bitácoras donde están instalados con el objetivo de descubrir patrones de intentos de accesos no autorizados.

procesados por la interfase, además, errores de verificación de paquetes IP, actividad de la interfase.
- Un monitor de tráfico TCP y UDP que muestra, para puertos de uso común en aplicaciones, el número de paquetes recibidos y enviados.
- Un módulo para obtener estadísticas en la LAN para mostrar la actividad de los diferentes equipos conectados.
- Filtros que muestran sólo el tráfico que se desea revisar.
- Bitácoras.

1.4.2.2 Autenticación.

Aquí se encuentran herramientas asociadas a la revisión de las contraseñas de los sistemas de cómputo, generación de contraseñas fuertes o fortalecer el esquema de almacenamiento de las contraseñas.

John the Ripper. http://www.openwall.com/john/
Esta herramienta se utiliza para la obtención de contraseñas mediante ataques de diccionario y fuerza bruta. El propósito principal de esta herramienta es detectar contraseñas débiles[25].

Crack / Libcrack. http://www.users.dircon.co.uk/~crypto/
Crack es un utilería y Libcrack es un conjunto de librería que pueden incorporarse a otros programas, cuya funcionalidad es adivinar contraseñas en base a un archivo de contraseñas cifradas, y revisa la existencia de contraseñas débiles.

1.4.2.3 Filtrado de servicios.

En este punto se ubican herramientas asociadas al control de acceso a los servicios de red y aplicaciones que proporciona el sistema de cómputo en base a diferentes criterios, como la dirección IP, el tipo de servicio requerido, la hora de acceso, entre otros.

netfilter/iptables. http://www.netfilter.org/
El proyecto netfilte/iptables es el subsistema de firewall incluido en los kernels Linux 2.4.x y 2.5.x. Las principales funcionalidades son:

- Filtrado de paquetes (tanto en protocolos orientados a conexión como sin conexión).
- Realizar todos los diferentes tipos de traducción de dirección de red (NAT, Network Address Translation).
- Mutilación de paquetes.

---

[25] Una contraseña débil está compuesta por palabras de fácil memorización, palabras de diccionario o palabras relacionadas al poseedor de la contraseña, que pueden ser investigadas fácilmente.

tcp_wrappers. ftp://ftp.porcupine.org/pub/security/index.html
Esta es una herramienta utilizada para filtrar las conexiones remotas intentadas realizar a los diversos servicios de red configurados en un equipo de cómputo. Los filtros pueden ser configurados por rangos de direcciones IP o tipo de servicio (puerto) y sus combinaciones. Las conexiones que se intenten realizar y no cumplen los criterios establecidos son rechazadas y no se les permite establecerse.

Esta herramienta es indispensable como una primera línea de defensa de los sistemas de cómputo conectados en red.

1.4.2.4 Búsqueda de vulnerabilidades conocidas.

En esta clasificación se encuentran herramientas que permiten hacer una revisión completa a un sistema para detectar la presencia de vulnerabilidades conocidas y que podrían ser explotadas por un atacante.

Nessus. http://www.nessus.org
El proyecto Nessus está orientado a proveer un analizador remoto de vulnerabilidades de seguridad fácil de usar, actualizable, de gran alcance y de libre distribución.

Entre sus funcionalidades, esta herramienta intenta explotar las vulnerabilidades que puedan presentar los sistemas que revisa, con el objeto de informar sus hallazgos e incluso hacer recomendaciones para que se corrijan.

Whisker. http://www.wiretrip.net/rfp/p/doc.asp?id=21&iface=2
El proyecto whisker consiste en una biblioteca de programas en Perl llamada libwhisker y un programa que funciona como CGI llamado whisker. Libwhisker es tiene diversas funcionalidades relacionadas al protocolo HTTP, que incluye revisión y explotación de vulnerabilidades, y whisker es la interfase web que se utiliza para realizar el barrido de vulnerabilidades.

1.4.2.5 Multipropósito.

En esta sección se tienen herramientas que proporcionan utilidad diversa para detectar vulnerabilidades, ataques o intrusos, además de funcionalidades para fortalecer la seguridad de un sistema de cómputo.

Snort. http://www.snort.org
Snort es un sistema de detección de intrusos, capaz de realizar análisis de tráfico de red en tiempo real y registrar los paquetes en redes IP. Lleva a cabo análisis de los protocolos, búsqueda por contenido y puede ser utilizado para detectar ataques, como intentos para provocar desbordamientos de estructuras de programas en memoria, barrido de puertos, ataques a CGIs, intentos para identificar el sistema operativo utilizado y muchas otras funcionalidades.

dsniff. http://naughty.monkey.org/~dugsong/dsniff/
Esta es una colección de herramientas para realizar auditorías y pruebas de penetración a la red.

Conforman esta colección, en primer lugar, las utilerías dsniff, filesnarf, mailsnarf, msgsnarf, urlsnarf, y webspy. Cuyo funcionamiento consiste en que de manera pasiva, se realiza el monitoreo de una red en búsqueda de datos como contraseñas, correo electrónico o archivos.

En segundo lugar, las utilerías arpspoof, dnsspoof, y macof facilitan la intercepción de tráfico de red a nivel de la capa 2 del modelo OSI[26], el cual normalmente no está disponible a un atacante.

En tercer lugar las utilerías sshmitm y webmitm implementan ataques de "hombre-en-medio" contra sesiones redireccionadas de protocolos SSH[27] y HTTPS[28] y explota debilidades ligadas a una infraestructura de llave pública (PKI, Public Key Infraestructure[29]).

hping. http://www.hping.org/
Esta herramienta es un analizador y ensamblador de paquetes TCP/IP, cuya interfase es orientada a línea de comando. Soporta los protocolos TCP, UDP, ICMP y RAW-IP. Tiene la habilidad de enviar archivos a través de un canal encubierto y entre las funcionalidades más destacables se encuentran:

- Pruebas de firewalls.
- Barrido de puertos.
- Pruebas a la red utilizando diversos protocolos.
- Trazado de rutas avanzado para todos los protocolos soportados.
- Identificación del sistema operativo utilizado en los sistemas revisados.
- Auditorías locales al protocolo TCP/IP.

SARA. http://www-arc.com/sara/
SARA son las siglas para Security Auditor's Research Assistant y es una herramienta para realizar análisis de la seguridad en sistemas Unix. Esta herramienta soporta y cumple completamente el consenso y la especificación sobre las principales 20 amenazas en los sistemas, el cual fue desarrollado en conjunto por el FBI y SANS[30].

---

[26] El modelo OSI está compuesto por 7 capas, de las cuales la capa 2 corresponde a Enlace de Datos.
[27] Secure SHell.
[28] HyperText Transfer Protocol (HTTP) funcionando sobre Secure Socket Layer.
[29] Una PKI es un sistema de certificados digitales, autoridades certificadoras y otras autoridades de registro que verifican y autentican la validez de cada parte involucrada en una transacción en el ámbito de dicho sistema.
[30] http://www.sans.org/top20/.

Firewalk. http://www.packetfactory.net/Projects/Firewalk/
Esta herramienta utiliza una técnica conocida como "firewalking", la cual fue desarrollada por Mike D. Schiffman y David E. Goldsmith y consiste en utilizar trazado de rutas, generadas en base al tráfico de red, para analizar las respuestas de paquetes IP, con lo cual es posibles determinar los filtros y listas de control de acceso (ACL, Access Control Lists[31]) utilizados en los ruteadores y crear mapas de la red.

Firewalk utiliza la técnica descrita, para determinar las reglas de control de acceso utilizadas por un dispositivo de red.

HUNT. http://lin.fsid.cvut.cz/~kra/index.html#HUNT
El objetivo principal del proyecto HUNT es desarrollar una herramienta que explote vulnerabilidades bien conocidas del conjunto de protocolos TCP/IP.

Cheops. http://www.marko.net/cheops/
Esta es una interfase gráfica para monitorear la red. Está diseñada para unificar las utilerías de red y tiene como objetivo ser una herramienta para el administrador de sistemas que le permita localizar, tener acceso, diagnosticar y controlar los sistemas de cómputo conectados en red, todo a través de una sola interfase.

LIDS. http://www.lids.org
LIDS son las siglas para Linux Intrusion Detection System, la cual es una herramienta de administración que permite aumentar la seguridad mediante la implantación de un monitor de referencia[32] de control de acceso mandatorio (MAC, Mandatory Access Control[33]) a nivel del kernel.

Entre las funcionalidades de esta herramienta se encuentra:

- Protección de archivos: ningún usuario del sistema, incluyendo root, puede modifcar los archivos protegidos por LIDS. Inclusive, los archivos pueden ser ocultados.
- Protección de procesos: ningún usuario del sistema, incluyendo root, puede enviar señales a los procesos protegidos por LIDS. Inclusive, los procesos pueden ser ocultados.
- Control de acceso muy preciso, mediante ACLs.

---

[31] Una ACL es un conjunto de datos que indica al sistema de control de acceso, los permisos de acceso que un sujeto tiene sobre los objetos en un sistema de cómputo. Cada objeto tiene un atributo de seguridad único que identifica los sujetos que pueden tener acceso a éste objeto.
[32] Un monitor de referencia se refiere a un módulo independiente que asegura las decisiones en el control de acceso dentro de una base de cómputo confiable (TCB, Trusted Computing Base). Una TCB es la suma de todo el hardware y software requerido para tener un sistema de cómputo seguro.
[33] El control de acceso mandatorio es un modelo de política de seguridad donde el dueño del recurso no es quien asigna los permisos de acceso a dichos recursos, sino que son asignados directamente por el administrador.

- Capacidad extendida para controlar todo el sistema.
- Alertas de seguridad a nivel de kernel.
- Detector de barridos de puertos a nivel de kernel.

The Coroner's Toolkit. http://www.porcupine.org/forensics/tct.html
Esta es una colección de programas desarrollados por Dan Farmer y Wietse Venema para realizar análisis forense a un sistema Unix comprometido. Entre los programas se encuentran grave-robber que captura información, ils y mactime que despliegan patrones de archivos, unrm y lazarus que recuperan archivos borrados y findkey que recupera llaves criptográficas a partir de un proceso en ejecución o de archivos.

1.4.2.6 Integridad.

En este rubro se ubican herramientas para determinar si los archivos protegidos han sido alterados o el sistema ha sido modificado de forma no autorizada.

AIDE. http://www.cs.tut.fi/~rammer/aide.html
AIDE son las siglas para Advanced Intrusion Detection Environment, el cual utiliza una base de datos, creada a partir de las directivas de su archivo de configuración, para verificar la integridad de los archivos y sus atributos. Para esto puede utilizar las funciones hash md5, sha1, rmd160, tiger y haval entre otras. La base de datos debería contener información acerca de archivos importantes en el sistema que se supone no deberían ser modificados, como por ejemplo archivos binarios.

1.4.2.7 Confidencialidad.

Aquí se encuentran herramientas que permiten, mediante algoritmos de cifrado y diversos controles, mantener la confidencialidad de la información protegida.

OpenSSH. http://www.openssh.com/
OpenSSH es una versión de libre distribución del conjunto de protocolos SSH y es un conjunto de herramientas para incrementar la confidencialidad en las transferencias de datos en la red. Generalmente sustituyen a las herramientas telnet, rlogin, ftp y otras más que transmiten las contraseñas en claro a través de la red al momento de establecer una conexión. Por su parte, OpenSSH cifra todo el tráfico (incluyendo las contraseñas), con lo que se evitan ataques del tipo "secuestro de conexión", intercepción de paquetes y otros ataques a nivel de red. Además, OpenSSH tiene la capacidad de crear "túneles" seguros[34] y diversos métodos de autenticación.

Las herramientas incluidas en OpenSSH son ssh, el cual reemplaza utilerías como rlogin y telnet, scp que reemplaza rcp y sftp que reemplaza ftp. También incluye

[34] Un túnel seguro se refiere a crear un canal de comunicación cifrada para protocolos que originalmente transmiten en texto plano la información. Evitando con esto la intercepción de la información transmitida por protocolos inseguros.

sshd, que funciona como servidor y otras utilerías básicas como ssh-add, ssh-agent, ssh-keygen y sftp-server. OpenSSH soporta el protocolo SSH en sus versiones 1.3, 1.5 y 2.0.

GPG. http://www.gnupg.org/
GPG son la siglas para GNU Privacy Guard la cual es una herramienta que utiliza criptografía de llave pública para el almacenamiento de datos y establecimiento de comunicaciones seguras, ya que puede ser utilizada para cifrar datos o crear firmas digitales. GPG no utiliza el algoritmo patentado IDEA, por lo que es un reemplazo de PGP (Pret Good Privacy[35]) y puede ser utilizado sin restricciones.

GPG es una aplicación que cumple con el estándar de Internet descrito en RFC2440 (OpenPGP[36]).

Las herramientas de seguridad descritas y comentadas en cada una de las secciones anteriores, son las más reconocidas dentro de su ámbito de funcionamiento y permiten implementar esquemas de seguridad diversos a bajo costo, debido a que son distribuidas bajo licenciamiento de software libre. En los siguientes capítulos se seguirán comentando dichas herramientas en relación a la integración con otros programas para conformar soluciones de seguridad informática más completas.

[35] PGP fue creado por Phil Zimmermann, mayor informacón en http://www.philzimmermann.com/.
[36] OpenPGP es un estándar que usa criptografía de llave pública y criptografía simétrica para proveer servicios, a través de los cuales se pueden establecer comunicaciones electrónicas y almacenar datos en forma segura.

Capítulo 2. Administración segura de sistemas de cómputo.

Implantar y mantener la seguridad en los sistemas de cómputo es una tarea compleja, y el contexto en el cual se encuentran los sistemas de cómputo es incluso más importante que la tecnología a utilizar para proteger dichos sistemas. Un sistema se mantendrá seguro hasta que ciertos avances en las matemáticas o en la tecnología no ocurran. Es por esto que denominar a un sistema "seguro", no tiene sentido si no se ubica dentro de un contexto [9].

La función de los administradores de redes y sistemas, debe incorporar aspectos de seguridad informática cada vez en mayor medida. Desafortunadamente, el concepto de seguridad, desde la perspectiva de los administradores de redes y sistemas, es extraño. Muchos simplemente no piensan ni como un atacante, ni como un policía [1, p. 11].

Esta combinación de contexto y conceptos de seguridad que deben manejar los administradores de sistemas hace que implantar y mantener seguridad en los sistemas de cómputo sea una disciplina a la que debe prestarse la debida atención y debe estar basada en una estructura organizada para este fin y no en la actividad aislada de los integrantes de una organización.

En este capítulo se abordan temas relativos a la práctica de asegurar y mantener seguros los sistemas de cómputo, comentando aspectos que complican la administración de la seguridad y también algunas formas para simplificarla. Además, se detallan esquemas para la implantación de políticas, procesos y herramientas que permitirán el aseguramiento de los sistemas.

2.1 Aseguramiento de los sistemas de cómputo.

Una vez que un atacante ha obtenido acceso a una cuenta de usuario en un sistema de cómputo, existen una gran variedad de cosas que puede hacer. Entre otras, un usuario malicioso creará una puerta trasera en el sistema, por donde será más fácil reingresar posteriormente en el sistema. También, el intruso borrará cualquier evidencia de su actividad. Existen toda clase de utilerías de software para hacer estas labores fácilmente [1, p. 3].

Parte fundamental para tomar acciones adecuadas en el ámbito de la seguridad informática, es el conocimiento y valoración que deben tener las organizaciones acerca de los recursos informáticos que poseen, para que a partir de ahí se pueda estimar el nivel de riesgo en que se encuentran y consecuentemente se implanten las medidas que ayuden a mitigarlo, por lo que a continuación se profundiza en este tema.

2.1.1 Valoración y mitigación del riesgo.

De manera general, la valoración y mitigación del riesgo, son un conjunto de actividades primordiales que deben ser desarrolladas en las organizaciones para conocer el grado de riesgo en el que se encuentran. Los objetivos generales al desarrollar una valoración de riesgos son, primero, determinar la fortaleza de los sistemas de cómputo y comunicaciones y, segundo, realizar una decisión informada sobre cómo puede y debe ser mejorada la seguridad de los sistemas [2, p. 11].

De manera más precisa, la valoración y mitigación de riesgos, es el proceso de encontrar, evaluar y corregir el daño potencial asociado con una brecha de seguridad. Los objetivos particulares de una valoración de riesgos, son identificar las áreas de una computadora, sistema de información o red que son más susceptibles a comprometerse y determinar la protección más apropiada para esas áreas. Esto se logra al analizar los atributos del riesgo, los cuales son: ***valoración de bienes***, ***vulnerabilidades*** y ***amenazas***, mismo que se revisaron en las secciones 1.2.2.4 a 1.2.2.6 [2, p. 12].

2.1.1.1 Manejo del riesgo.

La relación que guardan los tres atributos del riesgo (valoración de bienes, vulnerabilidades y amenazas) puede observase en la Tabla 2-1. Como se puede notar, el riesgo total alto sólo se presenta cuando los tres atributos son altos, por lo que, al disminuir alguno de estos tres componentes, provoca que disminuya el riesgo total.

Valoración del Bien	Amenaza	Vulnerabilidad	Riesgo Total
Alto	Alto	Alto	Alto
Alto	Alto	Bajo	Bajo
Alto	Bajo	Alto	Bajo
Bajo	Alto	Alto	Bajo

Tabla 2-1. Relación entre atributos (bienes, amenazas, vulnerabilidades) y riesgo.

El riesgo puede ser expresado a partir de sus atributos, de la siguiente forma:

Sea R el nivel de riesgo, B la valoración de bienes, A las amenazas y V las vulnerabilidades.

$$R = B\ V\ A$$

A este nivel de riesgo inicial lo afectan las contramedidas que sean tomadas con el objeto de reducir dicho riesgo, por lo que se puede redefinir la fórmula de la siguiente manera:

Sea R_R el nivel de riesgo residual, R el nivel de riesgo inicial y C el nivel de reducción del riesgo en razón de las contramedidas implementadas.

$$R_R = R - C$$

Cabe señalar que a las amenazas se le pueden asignar cierta probabilidad de que sucedan y en este sentido, las vulnerabilidades representan la posibilidad de que sean explotadas por un atacante y ambas están en razón inversa a las previsiones o contramedidas que se implementen.

En la Figura 2-1 se muestra la interacción entre los principales componentes del riesgo (valoración de bienes, amenazas y vulnerabilidades) y la incorporación de contramedidas, con la consecuente disminución del nivel de riesgo.

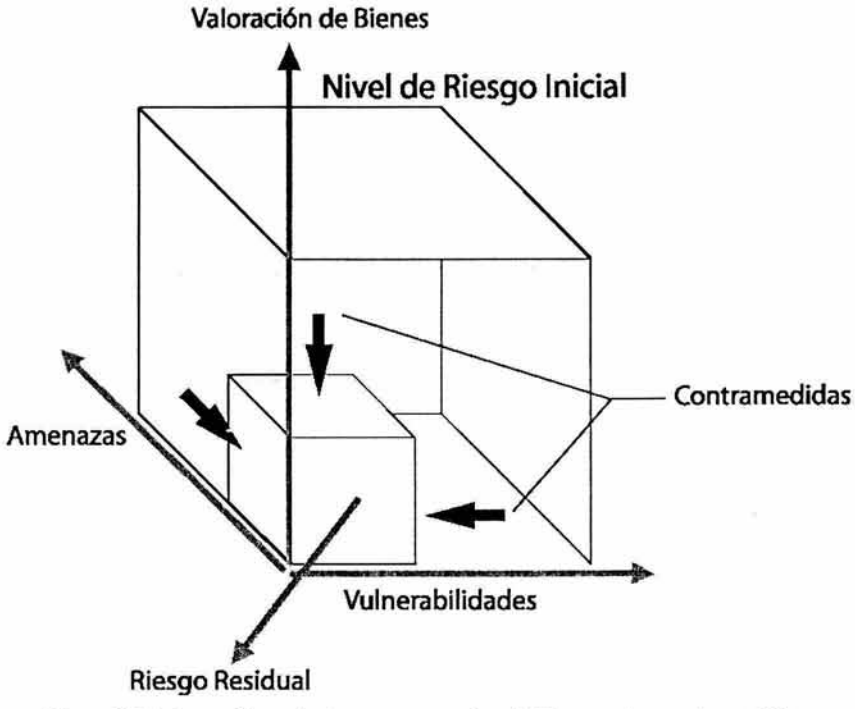


Figura 2-1. Interacción entre los componentes del riesgo y las contramedidas.

Existe un gran consenso entre los profesionales de la seguridad informática acerca de que el riesgo no puede eliminarse por completo. Aún cuando fuera posible, eliminarlo completamente implicaría un alto costo o simplemente no tendría sentido, dado que el costo de proteger sería mayor que el valor del bien protegido. Debido a esto el énfasis en este contexto cambia de la eliminación del riesgo a la administración del riesgo. Por lo que el proceso de valoración y mitigación del riesgo puede ser dividido en dos partes [24]:

- **Análisis de riesgos**, que involucra la identificación y cuantificación de los niveles de riesgo calculados a partir del valor de los bienes y los niveles de las amenazas y vulnerabilidades que afectan a dichos bienes.

- **Administración del riesgo**, que involucra la identificación, selección y adopción de contramedidas o protecciones justificadas por los riesgos en que se encuentran los bienes y la reducción de esos riesgos a niveles aceptables.

Considerando el ambiente en que se desenvuelve una organización y los recursos con que cuente, los responsables de la seguridad en la organización pueden optar

por implantar una o más de las siguientes estrategias de administración de riesgos [24]:

- **Mitigación del riesgo**. Reducir los riesgos mediante la aplicación de contramedidas seleccionadas. Este caso se aplica cuando se cuenta con una o varias soluciones que son viables para la organización de implantar y que permiten reducir en cierto grado el riesgo presentado.

- **Aceptación del riesgo**. Aceptar el riesgo residual o aún el nivel inicial de riesgo, si las contramedidas son más costosas que la valoración de los bienes. Se opta por esta estrategia cuando no es viable para la organización implantar las soluciones identificadas.

- **Transferencia del riesgo**. Transferir el riesgo a otra organización como por ejemplo una aseguradora o subcontratar los servicios de la administración de riesgos.

2.1.2 Administración de sistemas.

Los sistemas de cómputo son entidades dinámicas que se encuentran en constante adaptación tanto por los nuevos requerimientos de los usuarios, como también por las vulnerabilidades descubiertas que los afectan y por lo cual es necesario adecuarlos para corregir dichas vulnerabilidades.

Además de las modificaciones internas a los sistemas de cómputo, también se presentan las acciones de eliminación o adición de nuevos equipos a la infraestructura de cómputo y comunicaciones de las organizaciones, lo que conlleva contar con los procedimientos que permitan realizar estas labores de la mejor manera posible.

Por ejemplo, parecería trivial el desconectar un equipo, que ya no se requiere o que ha sido sustituido por otro de mejores capacidades, y proceder a "tirarlo a la basura". Pero antes de realizar esto, es muy importante determinar si en dicho equipo se tiene aún almacenada información valiosa para la organización, lo cual implicaría que se debería respaldar, borrar o destruir dicha información antes de tirar el equipo, así como la manera apropiada de realizar estas acciones.

Por otro lado, la adquisición e instalación de nuevos equipos implica la evaluación, instalación y configuración adecuada de los mismos, de acuerdo a procedimientos que incluyan consideraciones de seguridad informática.

Existen diversas prácticas asociadas para las diferentes etapas de la puesta en producción de un sistema de cómputo que es conveniente tener en cuenta con el propósito de que su implantación coadyuve a mejorar la seguridad total del sistema de cómputo de que se trate.

A manera de resumen, se listan en la Tabla 2-2 y se tratarán a mayor profundidad más adelante [3].

Etapa	Prácticas recomendadas
Planeación	1.- Desarrollar un plan de implantación de que incluya consideraciones de seguridad informática. 2.- Incluir requerimientos explícitos de seguridad al seleccionar el equipo.
Configuración	3.- Actualizar el sistema operativo y las aplicaciones. 4.- Habilitar solamente los servicios esenciales. 5.- Configurar los equipos para la autenticación de usuarios. 6.- Configurar el sistema operativo con controles adecuados de acceso a los objetos, dispositivos y archivos. 7.- Identificar y habilitar mecanismos de auditoría. 8.- Configurar los equipos para la realización de respaldos.
Mantenimiento	9.- Proteger los equipos de vulnerabilidades descubiertas, códigos maliciosos o software similar. 10.- Configurar los equipos para ser administrados remotamente. 11.- Proveer mecanismos y controles de acceso físico al equipo de cómputo.

Tabla 2-2. Prácticas recomendadas en la implantación de seguridad informática.

Las etapas mostradas en la Tabla 2-2 son iterativas y deben realizarse de manera constante con la finalidad de adecuar los sistemas a los cambios que normalmente ocurren debido a nuevos requerimientos en la funcionalidad de las aplicaciones o actualizaciones de software obligadas por fallas en la operación del mismo. En la Figura 2-2 se muestra un esquema de esta iteración.

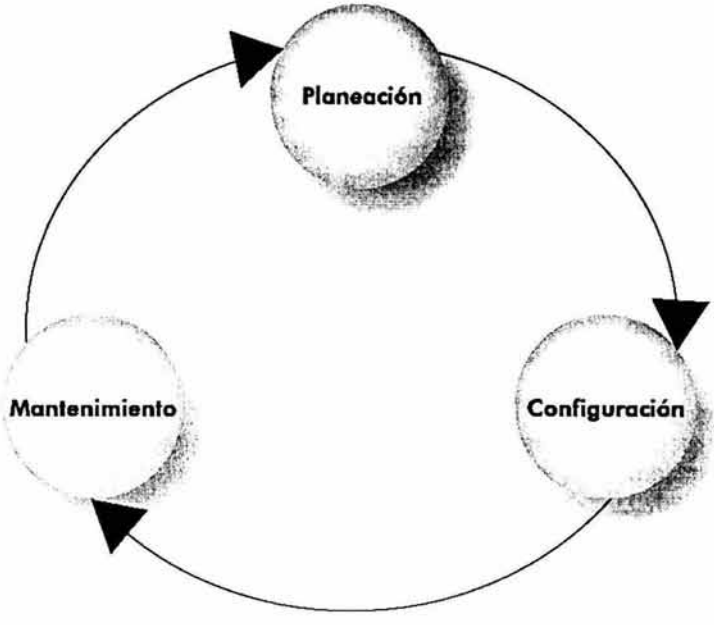


Figura 2-2. Etapas de la administración de sistemas de cómputo.

De las etapas mostradas, la planeación involucra diversos aspectos que deben tenerse en cuenta, ya que a partir de estos se desprenderán las otras etapas. A continuación se detallan algunos puntos a considerar para la planeación de la implantación de la seguridad informática en los sistemas de cómputo de una organización.

2.1.2.1 Propósito de cada equipo.

Antes que nada, es necesario saber la función que desempañará cada equipo y cómo se relaciona con los otros equipos instalados. Ya que en base a esto, se deberán establecer el nivel de esfuerzo que se invertirá en su protección.

Es importante comentar que las consideraciones relativas a la información que almacenan, procesan o transmiten los sistemas de cómputo, deben hacerse en conjunto con los diferentes usuarios de la información de que se trate, con el objetivo de poder realizar una valoración más real de dicha información.

Entre los puntos principales a identificar, están [3]:

- Información almacenada.
  Identificar el tipo de información que se almacenará en el equipo, organizarla de acuerdo al usuario o área de la organización a la que

pertenece, investigar la periodicidad y volumen de las modificaciones, y asignarle un nivel de clasificación e integridad.

- Información procesada.
  Dependiendo de la funcionalidad del equipo, tal vez no almacenará información directamente, pero seguramente procesará información que tenga cierto nivel de importancia para la organización, por lo que igualmente, dicha información debe ser identificada, organizada y se le debe asignar un nivel de clasificación e integridad.

- Requerimientos de seguridad para la información.
  En base a los niveles de clasificación e integridad y al valor asignado a la información identificada, se deben seleccionar los requerimientos de seguridad para dicha información.

- Servicios de red.
  De manera general, se deben identificar los servicios de red que proporcionará el equipo en cuestión y en base a esto determinar la criticidad, o importancia relativa, que tendrá el equipo.

- Requerimientos de seguridad para los servicios de red.
  De los servicios listados, se deben establecer los niveles mínimos de seguridad que deben tener los servicios identificados.

2.1.2.2 Usuarios o categorías de usuarios.

Como parte inicial para la implantación del control de acceso a los sistemas, se deben determinar e identificar a los usuarios que tendrán acceso al sistema y la manera en que se organizarán, ya que esto facilitará la administración de los mismos. Principalmente deberán identificarse dos conjuntos [3]:

- Roles (actividades autorizadas).
  En este conjunto, también conocido como grupo, se deben listar las actividades que realizarán los usuarios en el sistema, para luego agregar a cada uno los usuarios en función con las funciones que deben desempeñar.

- Listas de usuarios por cada rol.
  Esta es la lista de usuarios propiamente, los cuales deberán ser asignados a uno o más roles.

2.1.2.3 Privilegios de cada categoría de usuarios.

Como parte central del control de acceso al sistema, se deberán asignar los permisos correspondientes que cada usuario y conjunto de usuarios tendrán en el sistema, para lo cual es recomendable realizar lo siguiente [3]:

- Generar una matriz que muestre la relación usuario-privilegio.
  Esta matriz servirá como una referencia rápida para saber qué privilegios tiene cada usuario en todo momento. Algo importante en relación a esta matriz, es que se debe contar con los programas necesarios para que se pueda generar de manera automática en cualquier momento, con el objeto de poder validar que la información que tiene en funcionamiento el sistema, corresponde efectivamente con lo mostrado en la relación obtenida.

- Agrupar privilegios en grupos.
  Esta agrupación deberá definir los recursos del sistema que un usuario o grupo de usuarios pueden leer, escribir, cambiar, ejecutar, crear, borrar, instalar, remover, arrancar o detener.

2.1.2.4 Esquema de autenticación de los usuarios.

Como parte final del control de acceso debe implantarse la tecnología que llevará debidamente el control de acceso a los sistemas, para esto deberán observarse los siguientes puntos [3]:

- Contemplar aspectos tanto metodológicos como tecnológicos.
- Definir la utilización de esquemas usuario/contraseña, tokens o dispositivos biométricos.
- Eliminar a los usuarios configurados por default en el sistema.
- Deshabilitar el acceso a cuentas de usuario no interactivas (aquellas que no requieran acceso a una terminal para interactuar con el shell directamente).
- Crear grupos y asignar los usuarios a esos grupos (en lugar de asignar permisos directamente a usuarios).
- Especificar que las cuentas son personales e intransferibles.
- Establecer políticas de contraseñas (no permitir contraseñas débiles y establecer la renovación periódica de las mismas).
- Configurar reautenticación después de ciertos periodos de inactividad (por ejemplo, protectores de pantalla habilitados con contraseñas).
- Negar el acceso después de cierto número de intentos fallidos.

2.1.2.5 Obligatoriedad para el acceso apropiado a los recursos de información.

Eliminar los métodos alternativos de acceso a la información, ya sea mediante los propios permisos del sistema operativo o por mecanismos adicionales de seguridad como la utilización de cifrado para proteger la confidencialidad e integridad de la información. Para apoyar esta parte se deberán de definir e implantar los siguientes puntos [3]:

- Políticas para cada tipo de servicio informático proporcionado por el sistema de cómputo.

- Permisos aplicables en el sistema operativo.
- Permisos adaptables en la configuración de las aplicaciones.
- Cifrado de información sensible.

2.1.2.6 Procedimientos de respaldo y recuperación de información.

La generación y verificación de respaldos es una primera línea de defensa al momento en que se presentan problemas con algún sistema, por lo que se deben tener organizados y sustentados en los siguientes puntos [3]:

- Políticas de respaldo y recuperación.
  Se deben establecer políticas y procedimientos para la realización de respaldos y recuperaciones de información en los sistemas de cómputo, en las cuales deben establecerse claramente los involucrados, responsables y encargados de autorizar la realización de estas actividades a nivel de la organización, así como también indicar los tipos y medios de respaldos de que se dispone. Para los casos que lo requieran, deberán generarse políticas específicas.

  Debido al conocimiento y valoración de la información con que cuenta, el dueño o usuario de la información es el responsable de fijar los tiempos de realización de los respaldos, el tipo de respaldo a realizar cada vez (total, parcial o incremental) y el tiempo de retención que deberá conservarse el respaldo.

  Es conveniente definir un formato de acuerdo de respaldo entre los administradores de sistemas y los dueños de la información en el cual se debe establecer, para cada sistema de información: periodicidad, tipo y retención del respaldo, esquema de solicitud y autorización para la recuperación de información y consideraciones adicionales que se requieran.

- Herramientas utilizadas.
  El administrador del sistema deberá determinar las herramientas apropiadas, que garanticen que los respaldos se realizan de manera confiable.

  Es conveniente contar con herramientas que permitan la realización de respaldos sin intervención de operadores, que apoyen en la organización y control de los medios utilizados y que faciliten la restauración de la información.

- Uso de cifrado, para los casos que requieran protección adicional.
  La utilización de cifrado en los respaldos incrementa el tiempo de realización de los respaldos, pero definitivamente aumenta la confidencialidad e integridad de los mismos.

- Registrar y verificar los respaldos.
  Llevar un adecuado control de los medios utilizados para respaldar es primordial para poder realizar una adecuada restauración de un sistema de cómputo en caso de presentarse problemas. Además es muy conveniente la verificación que se haga de cada respaldo, con el objeto de no llevarse sorpresas desagradables al momento de recuperar la información.

2.1.2.7 Procedimiento para instalar el sistema operativo.

Con el objeto de garantizar homogeneidad en la instalación y configuración inicial de todos los sistemas de cómputo en una organización, es preciso contar con un procedimiento que, como mínimo [3]:

- Incluya todas las decisiones requeridas para la instalación.
- Describa todos los parámetros a utilizar.
- Indique todos los valores de parámetros de manera explícitos (aún cuando sean valores por omisión).
- Señale la realización de revisiones de actualizaciones o parches que deban ser aplicados.
- Mencione la manera en que deben ser configurados los distintos componentes.
- Refiera la instalación herramientas adicionales, en caso de requerirse.

2.1.2.8 Esquemas periódicos de revisión de integridad de archivos.

Es imprescindible contar con esquemas para verificar la integridad de los archivos del sistema, sobre todo aquellos que no deberían cambiar, como por ejemplo: archivos binarios, de configuración o catálogos [3].

En los esquemas, deberán incluirse como mínimo la lista de archivos o directorios a revisar, la periodicidad en que se realizarán las revisiones y los procedimientos a efectuar cuando se detectan alteraciones no autorizadas en los archivos.

2.1.2.9 Acciones para proteger la información contenida en hardware fuera de uso.

Generar procedimientos inherentes a las acciones que deben seguirse cuando se desincorpora un elemento informático de la organización. Entre los puntos principales, deben considerarse los siguiente [3]:

- Borrar y reformatear discos duros.
- Rescribir cintas.
- Eliminar contraseñas almacenadas en dispositivos.
- Destrucción física.

2.1.2.10 Revisión y actualización periódica de la documentación.

Periódicamente revisar y actualizar la documentación generada con el objetivo de mantenerla actualizada e incorporar aspectos relativos a [3]:

- Nuevas tecnologías.
- Nuevas amenazas y vulnerabilidades.
- Actualizaciones a la infraestructura tecnológica de la organización.
- Incorporación o eliminación de usuarios o unidades organizacionales.

2.1.2.11 Sistemas externos asociados.

Además de las previsiones que a nivel interno de los sistemas de cómputo se realicen, es necesario considerar los sistemas externos de apoyo, con el objeto de disminuir el nivel de riesgo de los sistemas. A continuación se comentan algunos puntos mínimos para garantizar que los servicios que proporciona un sistema de cómputo puedan mantenerse funcionando de manera apropiada:

- Lugar donde se ubicará el equipo.
  Deberá ser un lugar donde principalmente las amenazas naturales no produzcan un gran impacto.

- Conexiones físicas de la red.
  Tener una infraestructura de cableado estructurado certificado, esto evitará que se presenten problemas continuamente en las conexiones de red y dará certidumbre a la funcionalidad de los servicios de red.

- Sistemas de respaldo y protección de energía eléctrica.
  Contar con las provisiones necesarias en la infraestructura eléctrica (tales como tierra física, reguladores de voltaje, sistemas de respaldo de energía, entre otros), permitirán obtener un mejor desempeño de los equipos de cómputo y comunicaciones, y aumentará el tiempo promedio entre fallas de los componentes.

- Control de temperatura y humedad.
  Para un correcto funcionamiento de los equipos es necesario tener una ambiente de temperatura y humedad controladas, ya que la condensación al interior de los equipos puede ocasionarles daños severos. Existen equipos de precisión que realizan estas labores.

- Sensores y alarmas diversas.
  Con el objeto de proteger tanto a los equipos como personas que residen en el inmueble es necesario incorporar sistemas de detección y extinción de incendios, sistemas detectores de movimiento en lugares desatendidos, sismógrafos y alarma sísmica.

- Controles de acceso físico.
  Definir diversos perímetros de protección física a los equipos y contar con el control de acceso a cada uno de éstos mediante la utilización de sistemas de reconocimiento.

- Vigilancia del inmueble.
  Contar con el personal adecuado para realizar la vigilancia del inmueble en su conjunto, apoyado por un sistema de circuito cerrado de vigilancia que incluya cámaras de video, monitores y grabadoras.

- Calendarización de mantenimientos preventivos a los equipos.
  Parte fundamental para el buen funcionamiento de los equipos, debe ser la previsión de realizar mantenimientos preventivos a la infraestructura tecnológica con que cuenta la organización.

2.2 Administración remota de la seguridad de sistemas en red.

Colocar un sistema de cómputo en red, implica exponerlo a ataques y a un alto nivel de riesgo, por lo que requerirá mantenimiento para mitigar las vulnerabilidades del software instalado, además de las modificaciones que los nuevos requerimientos afecten en su funcionamiento y en los servicios que proporciona. Llevar una adecuada administración remota de los sistemas implica tener en cuenta más consideraciones y utilizar las herramientas y controles adecuados para que la administración no se vuelva un caos.

2.2.1 Consideraciones generales.

Para llevar a cabo la administración de un gran número de equipos remotos, es importante tener en cuenta los controles necesarios para que se pueda obtener acceso y poder llevar una adecuada administración del sistema de cómputo, no importando el lugar físico en donde se encuentre.

Debido a que los equipos estarán ubicados en lugares geográficamente distantes, se deben prever las fallas más comunes que se pueden presentar, y en base a esto, diseñar los procedimientos que se aplicarán cuando se presenten los problemas. Es imprescindible contar entre otras cosas, con los medios para llegar a la ubicación física del equipo remoto, la lista de personas a las que se puede contactar para solicitar apoyo y los planes de contingencia que se aplicarán para mantener los servicios funcionando.

A continuación se comentan los principales puntos para mantener la administración remota de los sistemas de una manera segura.

2.2.1.1 Esquema de conexión a la red.

La manera en que los equipos se conectarán a la red y la forma en que intercambiarán información es muy importante, ya que de no tomar las medidas adecuadas, un intruso podría obtener información no autorizada o incluso ingresar a un sistema de cómputo. Algunas de las medidas que se deben tomar son las siguientes [3]:

- Utilización de cifrado.
  Cifrar la comunicación entre los equipos, es método de conexión que no permite que la información transmitida entres máquinas, y sobre todo las contraseñas, sean obtenidas mediante la utilización de sniffers[37].

- Restricciones de acceso.
  Es conveniente incorporar elementos de software que validen que las conexiones se realicen sólo desde ciertos equipos autorizados.

- Esquemas de alta disponibilidad.
  Esta actividad involucra la identificación y evaluación de puntos únicos de fallas en la infraestructura de cómputo y comunicaciones. Una vez identificados dichos puntos se deben valorar con el objetivo de reforzar aquellos que lo requieran.

2.2.1.2 Servicios de red.

Los servicios de red incluyen una amplia gama de posibilidades para que un intruso obtenga acceso al sistema, por lo que deben identificarse, a nivel conceptual, las siguientes características de cada uno [3]:

- Tipo de servicio.
  Identificar el tipo de servicio, es decir, si el equipo funcionará como un servidor de correo, archivos, web, bases de datos o servidor de nombres, entre otros. En este rubro es importante recomendar que, de ser posible, se configure un servicio por equipo, con el objeto de evitar que a través de las vulnerabilidades que pudiera presentar algún servicio, se afecten otros que residen en el mismo equipo. De esta manera, el impacto que pudiera causar la falla de un equipo o servicio a nivel de la infraestructura tecnológica de la organización, se reduce al mínimo.

[37] Un sniffer es un programa que analiza el tráfico de datos en una red, que utilizado apropiadamente sirve para detectar problemas y cuellos de botella en la infraestructura de red. Desgraciadamente, también puede ser utilizado legítima o ilegítimamente para capturar los datos transmitidos en la red y obtener información de forma clandestina.

- Requerimientos de seguridad para cada tipo de servicio.
  Para cada tipo de servicio se debe identificar el esquema de conexión a la red, determinar si dicho servicio estará público o será de uso interno para la organización, establecer días y horas de funcionamiento.

  Como parte de la instalación de cada sistema, debe revisarse que sólo los servicios de red requeridos se encuentren habilitados y que estén configurados adecuadamente. Adicionalmente, se deberán implantar servicios de red que ayuden a una mejor administración de los equipos.

2.2.1.3 Software de los servicios de red.

En este punto se deberán revisar los detalles de la implantación de los servicios de red y las características particulares del software que será instalado en las computadoras [3]:

- Identificar opciones disponibles.
  Para cada tipo de servicio es conveniente revisar las diferentes opciones de implantación que existen. Algunas de las formas de realizar esto es consultar en algún buscador de información en Internet[38], e indagar en listas de correo o grupos de discusión especializados[39].

- Cumplimiento de los requerimientos de seguridad establecidos.
  Una vez identificado el software disponible, se debe discriminar aquel que no cumpla con los requerimientos mínimos de seguridad establecidos.

2.2.1.4 Niveles de privilegios y separación de tareas.

Es conveniente implementar separación de tareas para los distintos miembros del grupo de administración de los equipos y ajustar al mínimo los privilegios de cada uno. Todas las acciones por parte de los administradores de los equipos deben ser registradas en bitácoras, esto como apoyo en la realización de auditorías y ayuda a la identificación de errores cometidos [3].

2.2.1.5 VPNs para comunicarse entre servidores públicos e internos.

Es necesario implantar redes privadas virtuales (Virtual Private Networks[40]), con cifrado y autenticación, entre los equipos que están públicos a otras redes y

---

[38] Algunos pueden ser: www.google.com, www.yahoo.com, www.dmoz.org, www.freshmeat.net, o www.sourceforge.net.

[39] Un buen lugar para empezar puede ser http://www.faqs.org/faqs/ o http://usenet-addresses.mit.edu/.

[40] Una red privada virtual (Virtual Private Network) es una forma de utilizar una infraestructura pública de telecomunicaciones para proveer una forma de comunicación segura entre las oficinas o usuarios de una organización, lo que permite reducir costos.

aquellos que se encuentran detrás del firewall de la organización, con el objeto de mantener los equipos internos más seguros [3].

De esta manera, se podrá tener acceso a los equipos públicos de la organización desde los equipos internos de una forma más segura y sin exponer a los equipos internos.

2.2.1.6 Estrategias de detección de intrusos.

Se debe establecer un plan para la detección de intrusos o actividad sospechosa por parte de los usuarios de los sistemas [3].

Desde la instalación de un sistema de cómputo, debe realizarse la activación de bitácoras que registren información que sea útil al momento de realizar un análisis al sistema, identificando la información que será recolectada en cada computadora, como soporte a la seguridad.

Se deben implantar métodos que permitan revisar y generar reportes en relación a la información que generan las bitácoras. Además, en caso de que se especifique que las bitácoras se almacenarán en otro equipo, la transferencia de las mismas debe ser cifrada y autenticada.

2.2.1.7 Instalación de herramientas de seguridad.

La instalación de herramientas de seguridad adecuadas a los sistemas de cómputo es una necesidad básica que debe ser cubierta de una manera apropiada, ya que de lo contrario, en lugar de fortalecer la seguridad de los sistemas, una mala instalación de las herramientas puede terminar en exponer aún más el sistema o añadirle vulnerabilidades.

- Evaluación de herramientas.
  Realizar una investigación sobre las herramientas disponibles que pueden ser útiles para disminuir ciertos riesgos identificados en un sistema de cómputo y hacer una selección de aquellas que mejor se adaptan para solucionar los problemas presentados, en base a una evaluación objetiva.

- Plan de instalación.
  Antes de proceder a la instalación de las herramientas, se deben tener muy claros los pasos a seguir para llevar a cabo la instalación de las herramientas seleccionadas, con el objetivo de que no afecten negativamente al sistema de cómputo y a los servicios que proporciona.

- Revisión de la integridad.
  Un punto importante que debe incluirse en el plan de instalación, es revisar la integridad del origen de las herramientas. Esto comúnmente se realiza aplicando una función hash a las herramientas a instalar y compararlo con

los que proporcionan los encargados de distribuirlo, por lo que también habrá que asegurarse de alguna manera que el sitio desde el cual se obtiene el software es confiable.

- Configuración adecuada.
  Una herramienta de software no sirve de mucho si no se configura apropiadamente, por esto es imprescindible que se revise a profundidad la documentación asociada a las herramientas a instalar y se comprenda adecuadamente cada opción de configuración para obtener el máximo beneficio de las nuevas instalaciones de software y no se generen efectos negativos. (si no se realiza esto, puede convertirse en una herramienta que funcione a favor del atacante).

2.2.2 Puntos a considerar para la aplicación de actualizaciones.

Las vulnerabilidades son debilidades en el software que puede ser explotado por una entidad maliciosa para obtener mayor acceso del que tiene autorizado en una computadora. No todas las vulnerabilidades tienen actualizaciones de software relacionadas; por lo que los administradores de sistemas deben no solo estar atentos a las vulnerabilidades y actualizaciones de software, sino también mitigar aquellas vulnerabilidades que no tienen actualizaciones de software relacionadas, mediante otros métodos (p. ej. Firewalls, listas de control de acceso en ruteadores o software adicional en los equipos de cómputo). Es un error común entre los administradores de sistemas monitorear solo las actualizaciones de software y no las vulnerabilidades [6].

Puntos a considerar para mantener actualizados el sistema operativo y las aplicaciones [3]:

2.2.2.1 Desarrollar y mantener una lista de fuentes de información.

Documentar los sitios y fuentes de información relacionados a problemas de seguridad y actualizaciones del software que se encuentra instalado en los servidores, incluyendo [6]:

- Sitios web de los proveedores del software.
- Sitios web de organizaciones relacionadas a la seguridad informática.
- Listas de correo.
- Grupos de noticias.
- Bases de datos de vulnerabilidades.
- Herramientas de notificación.

La lista de sitios deberá estar relacionada al software utilizado en los sistemas de cómputo de la organización.

2.2.2.2 Establecer un procedimiento para monitorear las fuentes de información.

Este procedimiento deberá contener los siguientes puntos :

- Periodicidad para revisar sitios, correo o grupos de noticias.
- Flujo de información en caso de detectarse vulnerabilidades.
- Aplicación de soluciones o actualizaciones.

2.2.2.3 Evaluación de la información referente a las vulnerabilidades o actualizaciones para determinar su aplicación.

Respecto a esta parte, deberá considerarse que no todas las actualizaciones o vulnerabilidades aplican a todos los sistemas. Además, establecer una relación costo-beneficio de la incorporación de las actualizaciones o vulnerabilidades. También se debe considerar que de no instalarse las correcciones, se estará expuesto a una falla conocida públicamente.

2.2.2.4 Planear la instalación de actualizaciones aplicables.

La planeación es importante debido a que se algunas actualizaciones pueden introducir nuevas vulnerabilidades, o las actualizaciones pueden causar problemas a algún otro software instalado. Debido a esto, se importante analizar y comprender los efectos de la actualización al sistema.

Como regla general, siempre se debe respaldar el sistema antes de aplicar las actualizaciones, y para el caso de una gran cantidad de máquinas, se debe considerar la utilización de herramientas que automaticen el proceso de actualización. Además, la instalación de las actualizaciones debe realizarse utilizando un plan documentado del proceso que incluya todas las consideraciones al respecto.

2.2.3 Pruebas de seguridad.

La razón principal para realizar pruebas de seguridad a los sistemas es identificar las vulnerabilidades potenciales y consecuentemente repararlas. Típicamente, las vulnerabilidades son usadas repetidamente por atacantes para explotar las debilidades de la infraestructura tecnológica de las organizaciones que aún no las han corregido. La realización de pruebas de seguridad es una actividad fundamental que debe orientarse a lograr un ambiente operativo seguro y a la vez cumplir los requerimientos de seguridad de la organización [5].

Existen diversos tipos de pruebas de seguridad e Independientemente del tipo de la prueba, el personal que defina y realice pruebas de seguridad a los sistemas deberá contar con un significativo grado de conocimiento sobre redes, sistemas operativos y seguridad informática, incluyendo experiencia en las áreas de

seguridad en redes, firewalls, sistemas de detección de intrusos, seguridad en sistemas operativos, programación y protocolos de redes [5].

Tipos de pruebas de seguridad:

2.2.3.1 Mapeo de la red.

Esta revisión involucra la utilización de un escáner de puertos, como por ejemplo la herramienta Nmap, para identificar todos los sistemas de cómputo activos en una red de la organización, los servicios de red que tienen funcionando dichos sistemas y las aplicaciones específicas que ofrecen tales servicios [5].

El funcionamiento de un escáner de puertos consiste en tomar una dirección o un rango de direcciones IP y para cada dirección, de manera sistemática, realizar un barrido de puertos con el objetivo de identificar el servicio de red que funciona en dicho puerto y la aplicación que está proporcionando dicho servicio. El resultado del mapeo de la red es una lista comprensible de todos los sistemas y servicios que funcionan en el espacio de las direcciones revisadas por el escáner de puertos.

Una gran limitación de los escáner de puertos es que, aunque identifica los sistemas de cómputo, los servicios de red, las aplicaciones y sistemas operativos activos, éstos no identifican las vulnerabilidades. A partir de la información obtenida por el escáner de puertos, se requerirá intervención humana para detectar e identificar las vulnerabilidades en los sistemas de cómputo [5].

Algunos objetivos que se persiguen al realizar un mapeo de la red, son los siguientes [5]:

- Verificar que no existan equipos no autorizados conectados a la red de la organización.
- Identificar servicios de red vulnerables.
- Identificar desviaciones o alteraciones en los servicios implantados, en relación a las políticas de seguridad de la organización

2.2.3.2 Búsqueda de vulnerabilidades.

Este tipo de búsquedas se realizan mediante un escáner de vulnerabilidades, la cual es una herramienta que lleva el concepto del escáner de puertos al siguiente nivel, ya que identifica no sólo los sistemas de cómputo y puertos abiertos sino también las vulnerabilidades que presentan, y la mayoría de los escáners de vulnerabilidades también proveen información para corregir las vulnerabilidades descubiertas [5]. Una herramienta de este tipo es Nessus.

Una limitación significativa es que el escáner de vulnerabilidades requiere una actualización constante de la base de datos de vulnerabilidades para que pueda reconocer las vulnerabilidades más recientes.

Los escáners de vulnerabilidades proveen generalmente los siguientes servicios [5]:

- Identificar los sistemas de cómputo activos.
- Identificar los servicios de red activos y vulnerables.
- Identificar los sistemas operativos y las aplicaciones.
- Identificar las vulnerabilidades asociadas al sistema operativo y las aplicaciones.
- Probar el cumplimiento de las políticas de seguridad y uso de los sistemas.

La revisión de las vulnerabilidades mediante un escáner, permite validar que los sistemas operativos y las aplicaciones de los sistemas de cómputo se encuentran actualizados tanto en las versiones de software como en la aplicación de parches de seguridad [5].

2.2.3.3 Pruebas de penetración.

Las pruebas de penetración son revisiones a la seguridad en las cuales los evaluadores tratan de violar la seguridad de un sistema de cómputo basados en el conocimiento del diseño y los detalles de la implantación del sistema. El propósito de las pruebas de penetración es identificar los métodos para obtener acceso a un sistema mediante la utilización de técnicas y herramientas desarrolladas [5].

Debido a que las pruebas de penetración están diseñadas para simular ataques y utilizar técnicas y herramientas que pueden estar restringidas por la ley o por políticas de la organización, es necesario obtener permiso escrito de la organización antes de llevar a cabo dichas pruebas. Entre los puntos que de manera explícita debe incluir la solicitud de permiso, se encuentran: direcciones o rango de direcciones IP a ser probadas, cualquier sistema de cómputo que no deba ser considerado para las pruebas, una lista de técnicas y herramientas que se emplearán, horarios en los que las pruebas serán realizadas, las direcciones IP de los equipos desde donde se realizarán los ataques, datos para contactar tanto a los que ejecutarán las pruebas como a los encargados de los sistemas, la forma en que se manejará la información que obtengan quienes llevan a cabo las pruebas de penetración [5].

Las pruebas de penetración, generalmente consisten de cuatro fases (planeación, descubrimiento, ataque y descubrimiento adicional) que se ilustran en la Figura 2-3.

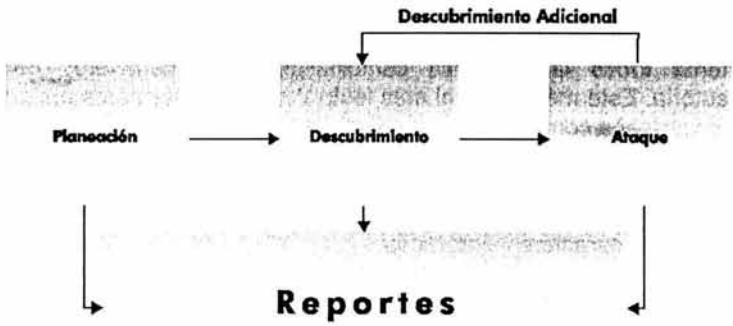


Figura 2-3. Pasos de la metodología para las pruebas de penetración.

2.2.3.4 Pruebas y evaluación de la seguridad.

Las pruebas y evaluación de la seguridad consisten en la revisión o análisis de las medidas de protección que han sido implementadas en los sistemas de información que se encuentran funcionando de manera productiva. Los objetivos de las pruebas y evaluación de la seguridad son principalmente [5]:

- Descubrir desperfectos en el diseño, implantación y operación de los sistemas de cómputo que podrían permitir violaciones a las políticas de seguridad.

- Determinar los mecanismos de seguridad apropiados para garantizar el cumplimiento de las políticas de seguridad.

- Evaluar el grado de consistencia entre la documentación del sistema y su implantación.

2.2.3.5 Identificación de contraseñas débiles.

Los programas como John the Ripper y Crack son utilizados para identificar contraseñas débiles en los sistemas de cómputo, este proceso parte del hecho de que las contraseñas se almacenan en realidad en los sistemas como un valor hash, por lo que un atacante intentará obtener dicho valor para intentar adivinar la contraseña correspondiente.

Existen básicamente tres tipos de ataques para la identificación de contraseña débiles una vez que se conoce el valor hash de la contraseña [5]:

- Ataque de diccionario.
  Utiliza todas las palabras que se encuentran en un diccionario o archivo de texto. Este método es uno de los más rápidos para obtener contraseñas.

- Ataque por fuerza bruta.
  Intenta todas las posibles combinaciones de caracteres de manera aleatoria. Este método es el más lento y puede tomar hasta meses obtener la contraseña correcta.

- Ataque híbrido.
  Este método utiliza un diccionario, pero sustituye algunos caracteres por números o caracteres simbólicos.

Debido a esto, se deben tener políticas y procedimientos para la generación de contraseñas y su renovación periódica.

2.2.3.6 Revisión de bitácoras.

La revisión de bitácoras puede ser utilizada para validar que los sistemas están operando de acuerdo a las políticas establecidas. Para llevar a cabo esta tarea se pueden utilizar diversas bitácoras de los sistemas para identificar desviaciones de la aplicación de las políticas de seguridad de la organización [5].

Aunque no es considerada tradicionalmente una actividad de "pruebas", la revisión y análisis de las bitácoras de los sistemas de cómputo y comunicaciones, pueden proporcionar un panorama amplio y dinámico de la actividad en los equipos en operación que pueden ser comparadas contra las políticas de seguridad [5].

Entre las herramientas que pueden utilizarse para llevar a cabo la revisión y análisis de las bitácoras, se encuentra Snort.

2.2.3.7 Revisión de la integridad de archivos.

Esta revisión es llevada a cabo por herramientas evaluadoras de la integridad de los archivos, las cuales apoyan al administrador de sistemas para reconocer cambios realizados a los archivos, en particular cambios no autorizados [5].

Como herramienta de apoyo para este punto, puede utilizarse AIDE (Advanced Intrusion Detection Environment).

2.2.3.8 Detección de virus.

Existen básicamente dos tipos de programas de detección y eliminación de virus disponibles: aquellos que se instalan en la infraestructura de red y aquellos que se instalan en los equipos de cómputo del usuario final. El primero de estos, es instalado normalmente en los servidores de correo o en conjunción con firewalls para impedir el paso de virus informáticos al interior de la organización. El segundo tipo detecta código malicioso en los equipos de usuario final en diversos

medios y formatos, como discos flexibles, discos duros, diversos tipos de archivos y correo electrónico, entre otros [5].

Sin importar el tipo de programa de software que se utilice, este no ofrecerá una protección completa a menos de que se tenga actualizada la base de datos de virus conocidos.

2.2.3.9 Revisión de modems no autorizados.

Se deben establecer procedimientos para uso de modems. Un modem representa una forma alterna de conexión a un equipo que puede llegar a ser más complicada para vigilar y controlar, por lo que es imprescindible contar con una relación de cuántos modems se tienen y de que tipo, dónde están instalados y cómo están configurados.

Además, es conveniente revisar periódicamente la existencia de modems no autorizados, mediante la realización de barridos en las líneas telefónicas de la organización.

Capítulo 3. Contexto y determinación del problema.

Hasta ahora se han comentado los conceptos relacionados a los ataques informáticos y algunas formas genéricas para proteger los sistemas de cómputo, pero son muchas y muy diversas las causas de los crímenes ó delitos informáticos, las cuales involucran violaciones a la seguridad de los sistemas de cómputo, uso no autorizado de los recursos de cómputo y comunicaciones de una organización e infracciones realizadas mediante la tecnología, afectando la confidencialidad, integridad y disponibilidad de la información involucrada.

Como se ha visto hasta ahora en los capítulos anteriores, tener una adecuada protección de los sistemas puede ser bastante compleja y se le debe poner mucha atención para asegurar que los sistemas informáticos se encuentran debidamente protegidos. Para lograr tales objetivos, es muy valioso conocer las causas que originan los delitos informáticos, los motivos que hacen que las personas los cometan y que han ocasionado el aumento de tales ataques.

En este capítulo se comentarán tres aspectos, que por sus características dan una idea de por qué el crimen informático afecta tanto. En primer lugar, las organizaciones y la sociedad en su conjunto dependen cada vez más de los sistemas de cómputo, tanto para el procesamiento de información compleja, como para apoyar sus labores cotidianas. Como segundo punto, se encuentra el incremento de ataques informáticos que involucran diversas situaciones. Y en tercer lugar, la complejidad cada vez mayor de los sistemas de cómputo y la falta de organización y control por parte de los responsables.

3.1 Mayor dependencia de las aplicaciones computacionales.

En los años recientes el incremento en la utilización de la tecnología se ha hecho evidente, es cotidiano e involucra a la mayoría de las personas, lo que trae consigo grandes beneficios a la humanidad en diversos aspectos. Sin embargo, tal reestructuración de la sociedad también presenta nuevas problemáticas que deben ser atendidas y resueltas.

Aunque es difícil la obtención de cifras y que éstas reflejen la realidad[41], existen algunos estudios, que se presentan en esta sección, que proporcionan una idea aproximada de la actividad del crimen informático y el impacto financiero que produce.

3.1.1 Evolución del procesamiento de la información.

La implantación de ideas revolucionarias, la estructuración del conocimiento y las transformaciones sociales han provocado grandes avances que se reflejan en la

[41] En parte porque la base instalada de sistemas informáticos es muy grande y también porque los responsables de la administración de los sistemas son renuentes a proporcionar información real.

forma de vida de las personas. Históricamente la humanidad ha presenciado cambios que han revolucionado a la humanidad y que pueden ser claramente identificados, por ejemplo: la imprenta, la revolución industrial, las computadoras e Internet.

El procesamiento de información no ha sido la excepción, y se ha transformado de una actividad manual, lenta y propensa a errores, para convertirse en un proceso altamente automatizado, rápido y confiable. Esta transformación se debe principalmente a la convergencia de diversas áreas del conocimiento entre las que destacan las tecnologías de información, especialmente las computadoras y redes de datos, las cuales han evolucionado de una manera muy característica.

El procesamiento de datos en equipos de cómputo, empezó de forma centralizada en las organizaciones, los equipos constaban de mainframes[42] y eran utilizados principalmente para realizar procesamiento para las áreas de finanzas y contabilidad. Paulatinamente el uso de las computadoras se extendió hacia departamentos diversos como ingeniería, producción y comercialización [26].

Con la introducción de las computadoras personales (PC) al ámbito de los negocios, el procesamiento de datos se diversificó prácticamente a todas las áreas de las organizaciones, así como también a los hogares y escuelas.

Otro aspecto importante que incide en el incremento del procesamiento de información es la incorporación de redes de datos, que interconectan los equipos y permiten, entre otras cosas, mayor rapidez para el intercambio de información entre dichos equipos [26].

En México, se reporta que el 9.5% de los hogares mexicanos cuenta con una computadora [36], este dato se obtiene por primera vez en el censo del año 2000, por lo que no es posible contar con un comparativo. En Estados Unidos se tienen cifras mucho más abundantes al respecto: en agosto de 2000, el 51% de los hogares contaban por lo menos con una computadora, comparado contra 42% en diciembre de 1998, además de que el 41.5% del total de los hogares estadounidenses contaban con acceso a Internet en el 2000 comparado contra 26.2% en 1998 [37].

Por otra parte, el procesamiento de información de los sistemas de cómputo, sufrió modificaciones debido a cambios en la manera en que los programas de aplicación funcionan y manipulan la información. Este cambio en los esquemas para el procesamiento de la información está muy relacionado a la evolución de la arquitectura cliente-servidor, la cual ha permito que las aplicaciones puedan atender a un mayor número de usuarios.

---

[42] Equipos de procesamiento centralizado.

3.1.2 Información en los equipos de cómputo.

La evolución del procesamiento de la información es resultado del aumento en el volumen de la información procesada en equipos de cómputo y la manera en que se ofrecen soluciones tecnológicas a las organizaciones para resolver sus necesidades en este rubro.

Cada vez más información es procesada y almacenada en equipos de cómputo, y cada vez se transmite más información por redes públicas, lo que ocasiona incremento en la dependencia que las organizaciones tienen en los sistemas de cómputo y comunicaciones para el procesamiento y consulta de información.
Diversas estimaciones calculan que el total de información en páginas y sistemas web en Internet, ocupaba más de 7,500 TB de almacenamiento en el año 2000, y el correo electrónico había generado más de 20,000 TB en ese mismo año (aunque por supuesto no todos los correos se almacenan) [38].

Lo anterior proporciona una idea de que en mayor medida se "confía" y se "depende" en sistemas informáticos para la realización de las labores de las organizaciones, pero también esta "confianza/dependencia" afecta directamente a las personas, en particular en aspectos tan diversos como pagos electrónicos, manejo de cuentas bancarias o expedientes médicos, por mencionar sólo algunos.

La evolución de las arquitecturas de procesamiento de datos, ofrece muchas ventajas, pero a la vez esto implica la necesidad de administrar una mayor cantidad de servicios de información y en un alto porcentaje de los casos, esto significa que se tienen que administrar una mayor cantidad de equipos de cómputo y comunicaciones.

Tan solo como referencia, puede observarse el dramático crecimiento de equipos conectados a Internet. En 1993, ISC reportaba 29,670,000 servidores, para 2002 esta cifra se encuentra en los 162,128,493 servidores. Esto representa un crecimiento de más del 446% en menos de 10 años [28].

Estas cifras hacen evidente el hecho de que probabilísticamente, se incrementa la posibilidad de ocurrencia de ataques informático y que se presenten vulnerabilidades en el software que se ejecuta en los equipos de cómputo, toda vez que existe un mayor número de equipos y software siendo utilizados y el número de usuarios de servicios informáticos aumenta día con día.

3.2 Incremento de la posibilidad de ataques informáticos.

El uso de sistemas de cómputo en las organizaciones, posibilita los ataques informáticos debido a que hay más gente utilizando dichos sistemas y el valor de la información que se procesa es alto. Asimismo, el incremento de usuarios de la Internet favorece el crecimiento de dicha posibilidad de ataques, ya que existe una mayor difusión de información que le permite a cualquier usuario estructurar un

ataque a los sistemas de cómputo de las organizaciones de manera relativamente sencilla.

Los problemas relacionados a la seguridad de los sistemas de cómputo se complican debido principalmente a cuatro puntos [2, p. 3]:

1. Las aplicaciones de cómputo están compuestas de una gran cantidad de software, e históricamente está probado que es prácticamente imposible realizar implantaciones libres de errores.

2. La seguridad comúnmente no es considerada en el diseño o implantación de los sistemas de cómputo sino que es "añadida" durante el transcurso o posterior al proyecto, lo que resulta en soluciones inadecuadas.

3. La seguridad es costosa y por esta razón es dejada de lado o pendiente para un futuro que no siempre llega.

4. Muy frecuentemente el problema está asociado con la gente que usa los sistemas de cómputo y no con la tecnología.

Debido a esto, es de observarse que el número de vulnerabilidades descubiertas en los programas de software aumenta cada año. ICAT Metabase del NIST (http://icat.nist.gov/), reporta el número de vulnerabilidades descubiertas en los años recientes. Esta relación se muestra en la Tabla 3-1.

Año	Número de vulnerabilidades
1999	862
2000	990
2001	1,506
2002	1,307
2003	1,007

Tabla 3-1. Número de vulnerabilidades descubiertas por año.

Cabe señalar que al revisar el detalle de las vulnerabilidades reportadas, es de observar que la mayoría (más del 50%) corresponden a aplicaciones. También es sobresaliente que la mayoría corresponden a errores en la validación de las entradas (errores de condición de límites y desbordamientos de buffers). Por otra parte, el total de las vulnerabilidades registradas por ICAT Metabase, la mayoría pueden ser explotadas remotamente, según se muestra la Tabla 3-2.

Característica de las vulnerabilidades	2003	2002	2001	2000	1999
Ataque remoto	755 (75%)	1052 (80%)	1055 (70%)	685 (69%)	446 (52%)
Ataque local	252 (25%)	275 (21%)	525 (35%)	414 (42%)	316 (37%)
Accesos logrados	3 (0%)	12 (1%)	25 (2%)	34 (3%)	33 (4%)

Tabla 3-2. Estadísticas de las vulnerabilidades registradas por ICAT Metabase.

Adicionalmente, según estadísticas del CERT [29], el número de incidentes reportados[43] de 1988 a 2002 han sido 173,728. Y es en los años de 2000 a 2002, en que el número se ha disparado de manera sorprendente como se muestra en la Tabla 3-3.

Año	Número de incidentes reportados
1999	9,859
2000	21,756
2001	52,658
2002	82,094
2003	137,529

Tabla 3-3. Estadísticas del número de incidentes reportados al CERT.

3.2.1 El valor de la información.

Como se comentó en el Capítulo 1, la valoración de bienes es una parte crucial para el análisis de riesgos, uno de los bienes más valiosos para cualquier organización es la información con que cuenta, ya que ha invertido diversos recursos para generarla y mantenerla. Aunque esto es así, la mayoría de las veces, puede ser muy complicado asignarle un valor exacto. Por lo que es recomendable involucrar a todos los usuarios directos e indirectos de la información para tener una estimación más real.

No obstante, se puede observar que el valor de la información que se procesa y transmite entre los sistemas de cómputo de las organizaciones ha aumentado, en razón de las aplicaciones informáticas que utilizan dicha información, el propio volumen de datos acumulado (el cual refleja por sí mismo un costo de captura) y el valor estimado de la información por el uso que puede tener a futuro.

Resulta evidente que la información procesada y transmitida entre los equipos de cómputo de las diversas organizaciones a nivel mundial, resulta más atractiva para ser utilizada por delincuentes informáticos alentados por fines muy diversos, por lo que existen los incentivos para realizar ataques informáticos a los sistemas de cómputo de las organizaciones.

Existen estudios que estiman que para el año 2004, los montos por concepto de comercio electrónico alcanzarán los 6.8 billones de dólares a nivel mundial, en transacciones en línea, según Forrester Research, Inc [39]. De acuerdo a este estudio, los montos acumulados anualmente por concepto de comercio electrónico en línea a nivel mundial por año, tanto en transacciones "empresa-empresa" como "empresa-cliente final", puede observarse en la Tabla 3-4.

[43] Un incidente puede involucrar uno, cientos o miles de sitios afectados. Asimismo, ciertos incidentes pueden implicar actividad continua por largos periodos de tiempo.

Año	Monto (miles de millones)
2000	US$ 657.00
2001	US$ 1,233.60
2002	US$ 2,231.20
2003	US$ 3,979.70
2004	US$ 6,780.80

Tabla 3-4. Proyecciones de montos acumulados por concepto de comercio electrónico.

De lo cual se obtiene que durante el año 2004 se tendrá un incremento de 8.6% en ingresos adicionales por este concepto, el estudio contiene información adicional que resulta interesante de comentar. Para el 2004, Estado Unidos tendrá el 47% del comercio en línea a nivel mundial, mientras que Japón contará con 13%, y Alemania el 5.7%. Los porcentajes por región se muestran en la Tabla 3-5.

Región	Porcentaje de comercio electrónico (2004)
América del Norte	50.9 %
Asia / Pacífico	24.3 %
Europa	22.6 %
América Latina	1.2 %

Tabla 3-5. Proyección de participación en el comercio electrónico por región.

Estas cifras proporcionan una idea de la cantidad de dinero que representa la información almacenada y transmitida que albergarán los sistemas de cómputo de las organizaciones, de las necesidades de proteger adecuadamente la información y del reto que se les presenta a los administradores de sistemas para llevar a cabo la administración de equipos y sistemas de cómputo.

En relación con ataques informáticos las cifras son alarmantes, de acuerdo a los resultados del *2002 CSI/FBI Computer Crime and Security Survey*[44][4], el monto acumulado en pérdidas económicas, relacionadas al crimen informático, en el periodo de 1997 al 2002 es aproximadamente de US$1,459,755,245.00 (casi mil quinientos millones de dólares), esta cifra aumenta de US$100,119,555.00 (poco más de cien millones de dólares) en 1997 a U$455,848,000.00 (casi quinientos millones de dólares) en el 2001, lo que representa un incremento de más de 355% en este periodo. Además, es de considerarse que el 80% de los participantes en la encuesta tenían conocimiento de pérdidas financieras, pero sólo el 44% pudo cuantificarlas.

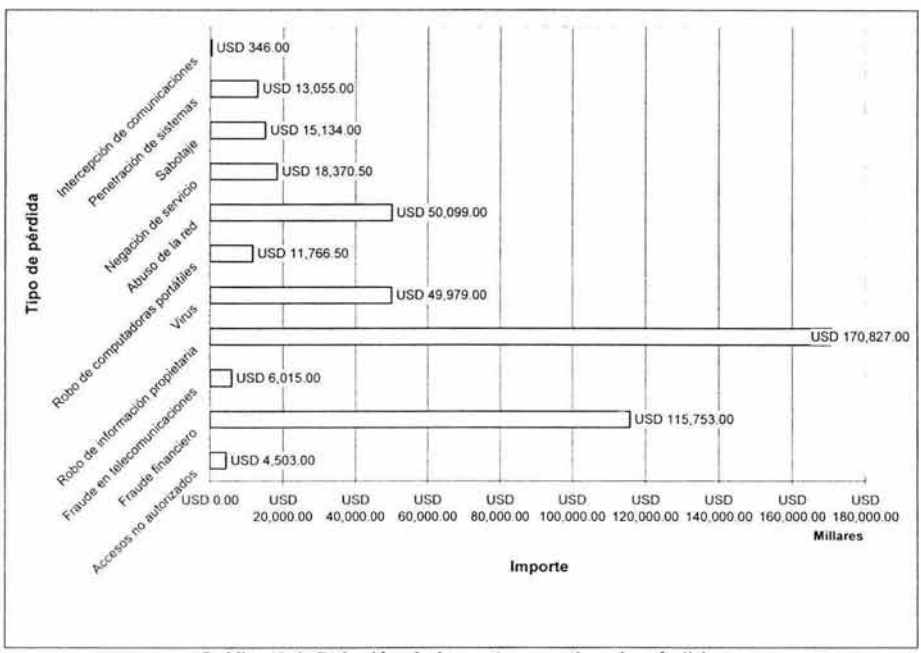
[44] Esta encuesta es llevada a cabo por el Computer Security Institute (CSI) con la participación del San Francisco's Federal Bureau of Investigation's (FBI) Computer Intrusión Squad, y está basada en las respuestas de 503 organizaciones facultadas en seguridad informática en Estados Unidos, que incluye a corporacion agencias del gobierno de ese país, universidades e instituciones médicas y financieras.

El desglose de los montos en pérdidas financieras acumuladas anualmente, como resultado obtenido de la aplicación del Computer Crime and Security Survey en los años de 1997 a 2002, puede observase en la Tabla 3-6.

Año	Total anual de pérdidas financieras
1997	US$100,119,555.00
1998	US$136,822,000.00
1999	US$123,799,000.00
2000	US$265,337,990.00
2001	US$377,828,700.00
2002	US$455,848,000.00

Tabla 3-6. Relación de pérdidas financieras por año reportados por CSI.

Con el fin de proporcionar mayor detalle de la situación, el monto correspondiente al 2002, se divide por tipo de pérdida y se muestra en la Gráfica 3.1.



Gráfica 3.1. Relación de importes por tipo de pérdida.

3.2.2 Disponibilidad de información sobre cómo realizar ataques a sistemas informáticos.

Otro aspecto a considerar es que cada vez existe más información que tiene a su disposición un delincuente informático para realizar ataques. Dicha información se

encuentra en libros, revistas, publicaciones electrónicas, boletines emitidos por organizaciones especializadas en el área de seguridad, grupos de noticias y listas de correo en Internet.

Aunque el objetivo de publicar esta información es mejorar el diseño de la seguridad de los sistemas, también es utilizada de manera ventajosa por los delincuentes informáticos, quienes se aprovechan de los descuidos, desinformación o lenta reacción que pueden tener los administradores de sistemas en las organizaciones [27].

Aunado a esto, y para hacer aún más grave el problema es que no sólo está la información sino que además existen los programas que implementan la funcionalidad de los ataques, lo cual facilita en gran medida que la persona que está dispuesta a explotar vulnerabilidades en los sistemas de cómputo, pueda hacerlo sin mayor complicación.

En este punto es conveniente comentar algunas de las características de los ataques informáticos que posibilitan que los atacantes tengan éxito y un mayor alcance de sus objetivos. A diferencia del mundo real, las amenazas en el mundo digital implican consideraciones que dificultan prevenir de manera adecuada los ataques, las principales son las siguientes [9]:

- **Automatización.**

Las computadoras son magníficas para automatizar procesos y son también aplicadas para automatizar ataques. El atacante sólo tiene que indicar al programa que se vuelva a ejecutar las veces que se requiera, sin que tenga que intervenir el atacante de manera directa.

Inclusive, un ataque puede ser programado de tal forma, que posibilite que varios equipos actúen en forma coordinada para incrementar su impacto.

- **Acción a distancia.**

La Internet no tiene fronteras ni conoce las divisiones políticas y cualquiera que se conecte a esta red, tendrá eventualmente la posibilidad de obtener acceso a cualquier sistema de cómputo que funcione en esta red de redes.

Esto, además de que proporciona un mayor número de posibles atacantes, conlleva complicaciones para ejecutar acciones legales en contra de dichos atacantes, en virtud de que se requiere una acción coordinada entre los responsables de la justicia en diversos países, lo cual no es sencillo.

- **Técnica de propagación.**

Programar un ataque, puede en algunos casos resultar muy complicado, e incluso puede requerir de un gran esfuerzo humano y financiero, pero una vez que se ha hecho, difundirlo es tan simple como copiar el programa o ponerlo a disposición en una página web.

Esto posibilita que aún personas que no tienen el conocimiento para entender los detalles del funcionamiento del ataque, puedan llevarlo a cabo. La Internet es un excelente medio para difundir información sobre cómo realizar ataques.

Adicional a todo esto, existe la evidente complejidad que representa para los administradores de sistemas, tener que administrar un gran número de equipos, lo cual se comenta con más detalle a continuación.

3.3 Dificultades presentadas al aumentar el número de equipos de cómputo.

Como ya se ha mencionado en los capítulos anteriores, la administración de los sistemas de cómputo es complicada, y en particular la administración de la seguridad relacionada a los mismos. Pero cuando esto es llevado a un gran número de equipos y sistemas, la situación en verdad se vuelve alarmantemente compleja.

Esta complejidad se debe a que al aumentar el número de equipos, se requiere destinar mayores recursos materiales, financieros y humanos para mantener en condiciones adecuadas cada equipo, con el objetivo de que cuando se presenten vulnerabilidades, éstas puedan ser detectadas y corregidas de manera oportuna, para con ello disminuir el riesgo de ataques informáticos a la infraestructura de cómputo y comunicaciones de la organización.

Este aumento de equipos y sistemas de cómputo es propiciado por las mismas organizaciones, ya que cada vez es más común la incorporación de sistemas de cómputo para la realización de tareas sustantivas en la organización.

3.3.1 Mayor número de servicios de información en las organizaciones.

Las organizaciones cada vez en mayor medida, requieren contar con una diversidad de aplicaciones informáticas que les permitan cumplir en tiempo y forma con todas sus actividades y cubrir sus necesidades de información. Algunas de las aplicaciones principales son: bases de datos, servidores web, correo electrónico y aplicaciones propietarias, entre otras.

Esta explosión en el uso del cómputo al interior de las organizaciones, se debe principalmente a la incorporación de tecnologías y protocolos utilizados en Internet, por lo que durante la segunda mitad de la década de los 90s surgen las Intranets, con lo que una gran cantidad de empleados comparten información de la organización para la cual laboran.

Las tecnologías de información han contribuido de manera directa al aumento de la productividad en las organizaciones en los años recientes, afirman los

economistas. Las compañías están obteniendo ganancias en productividad al mejorar sus procesos de negocio (por ejemplo, al eliminar las barreras entre diferentes áreas de la organización). Entre las principales tecnologías que contribuyen a la productividad se encuentran herramientas de software para colaboración, computadoras personales, incremento en el ancho de banda de la redes de comunicación, dispositivos de cómputo móvil y dispositivos inalámbricos [40].

Así mismo, las tecnologías de información están mejorando el servicio a los clientes en aspectos tales como diversificación de opciones, tiempo de respuesta, calidad y personalización de productos y servicios. De hecho, muchas compañías están considerando el comercio electrónico para incrementar su productividad y diversos expertos en la materia, predicen que Internet será un gran detonador de la productividad [40].

Sin embargo, aunque existen mucho beneficios en la incorporación de la tecnología a las organizaciones, también esto provoca la generación de más problemas, ya que entre mayor es el número de aplicaciones que requiere una organización, es mayor el número de sistemas de cómputo requeridos para el procesamiento de la información, con el consecuente aumento de la probabilidad del riesgo en la ejecución de ataques informáticos o uso indebido de los equipos de cómputo y comunicaciones por parte de los usuarios.

Es de notar, que al interior de las organizaciones se pueden observar los cambios, ya que en un principio las aplicaciones eran hechas a la medida de las necesidades del procesamiento de información requerido y dichas aplicaciones sólo eran utilizadas por unas cuantas personas de ciertas áreas en particular de la organización; en la actualidad, las aplicaciones informáticas llegan (y afectan) prácticamente a todas y cada una de las personas que integran las organizaciones.

Esto impacta a los responsables de informática en las organizaciones, ya que desde el punto de vista de la administración de sistemas, administrar un solo sistema o unos pocos, permite que se le pueda dedicar la atención adecuada a cada sistema, pero al aumentar el número de equipos y sistemas, surgen las dificultades para otorgar a cada uno la atención que requiere por parte de los administradores.

Adicionalmente, el incremento en el número de sistemas afecta de manera sensible la forma en que dichos sistemas deben ser administrados, ya que se requieren utilizar nuevas formas que permitan llevar a cabo una adecuada administración. Para estos casos, es necesario implantar las metodologías y herramientas que permitan escalar la calidad de la administración no importando de cuantos equipos se trate.

3.3.2 Aumento de los equipos de cómputo en las organizaciones.

De igual manera, diversas circunstancias han concurrido para facilitar a las organizaciones contar con más y mejores equipos de cómputo y software de aplicación. Algunas de estas circunstancias son: las ventajas que proporciona el contar con equipos de cómputo para el procesamiento de la información, la baja en los precios del hardware, el avance en el desarrollo de sistemas y la facilidad de uso para el usuario final.

Dependiendo de las actividades sustantivas de las organizaciones, variará el tipo y número de sistemas de cómputo que utilicen, así como también el número de equipos para uso de los empleados y las políticas establecidas de uso adecuado de las redes de comunicaciones y sistemas informáticos.

Según cifras del año 2001, las exportaciones de equipo de cómputo y productos electrónicos del estado de California hacia México, totalizaron 6,800 millones de dólares. Esto proporciona una idea del número creciente de equipos que ingresan al país.

Sin embargo, aunque esto representa ventajas para las organizaciones y usuarios de equipos de cómputo, para los administradores de sistemas se complica llevar una adecuada administración de los servicios informáticos, ya que se tienen que organizar una mayor cantidad de equipos y/o servicios, lo cual implicará invariablemente reducir el nivel de atención a cada sistema y darle un trato más general a cada uno, con lo que, en muchas ocasiones, son descuidados aspectos como monitoreo, actualización y control de software y aplicación de parches. Esto posibilita el aumento de vulnerabilidades, por lo que tiende a aumentar el riesgo de ataques informáticos a la infraestructura de cómputo de la organización.

Esta problemática se acentúa más aún cuando no se tiene definido e implementado un adecuado proceso de control de cambios que garantice que los cambios realizados a sistemas productivos no afectará de manera sensible la disponibilidad de servicios informáticos.

3.3.3 Aumento de la complejidad de los sistemas.

Aunado a lo anterior, y debido a la evolución de la tecnología, tanto en hardware como en software se puede observar un incremento colosal en las funcionalidades de los sistemas de cómputo.

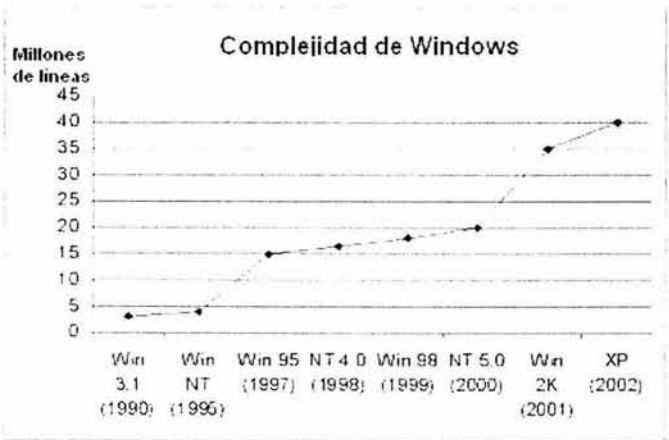
Respecto al desarrollo de software, se tienen hasta ahora 5 décadas en que la norma ha sido el software defectuoso y la alta calidad, la excepción. Desgraciadamente, la calidad del software actualmente no es mejor de lo que era hace varias décadas, en algunos casos es incluso peor. Algo importante de destacar a este respecto, es el hecho de que para generar software de calidad es necesario contar con buenos procesos de desarrollo de software. Pero más importante aún, es reconocer que el desarrollo de software es fundamentalmente

una tarea técnica, por lo que solamente se obtiene software de calidad cuando se tienen buenos desarrolladores (independientemente de que se tengan o no buenos administradores de desarrollo de software). Lo contrario de esto no es factible, ya que malos desarrolladores no generarán software de calidad, aún cuando se tengan a los mejores administradores de desarrollo de software [43].

Para el caso de las aplicaciones informáticas, la complejidad se refleja en el número de líneas de código que tiene cada aplicación, así como también las interrelaciones que guarda con otras aplicaciones igualmente complejas. Muchas veces una aplicación se construye sobre la integración de diversas aplicaciones, por lo que complejidad de la aplicación resultante es la suma de las complejidades de las aplicaciones que la conforman. La complejidad inherente del software en general, puede ser controlada solamente a través de un diseño cuidadoso del software [42].

Cuando se incrementa la complejidad de los sistemas, invariablemente aumenta el factor de riesgo debido a que aumenta la probabilidad de que se presenten vulnerabilidades en los sistemas de cómputo.

Como ejemplo de esto, se puede mencionar el sistema operativo Windows, el cual ha crecido de 3 millones de líneas de código en 1990, hasta 40 millones en 2002 [41], en la Gráfica 3.2 se puede observar este crecimiento.



Gráfica 3.2. Evolución del número de líneas de código del sistema operativo Windows.

El crecimiento de las aplicaciones significa que cada vez los sistemas informáticos cuentan con características más avanzadas, pero esto implica que las organizaciones requieren contar con especialistas que administren adecuadamente cada parte de dichos sistemas. Esto representa un reto importante, ya que se debe conseguir que dichos especialistas interactúen de

manera adecuada y de forma oportuna, y a la vez que dispongan de las herramientas apropiadas que los apoyen en sus labores.

Si bien es cierto que las funcionalidades de los sistemas de cómputo son muy convenientes para los usuarios de información, al mismo tiempo este incremento en las funcionalidades, involucra de forma directa un aumento en la complejidad de los sistemas de cómputo, por lo que se han desarrollado diversas metodologías que tratan de mantener control dentro de la complejidad, pero aún no se tiene resuelto. La complejidad del software es, desde hace 40 años, uno de los problemas más importantes a resolver dentro de la ciencia de la computación [42].

3.4 Determinación del problema.

Como se ha mostrado en este capítulo, el incremento en la utilización de sistemas de cómputo para el procesamiento de la información en las organizaciones, ha propiciado una mayor utilización y dependencia de los equipos de cómputo.

Por una parte, tanto las implantaciones de los sistemas de información como de los equipos de cómputo tienen fallas inherentes que históricamente han probado ser recurrentes, conocidas comúnmente como "bugs", mismas que si no son corregidas a tiempo, exponen a los sistemas de cómputo a ataques informáticos. Asimismo, los alicientes para la realización de ataques informáticos se han incrementado debido a que en los sistemas de cómputo se almacena y procesa información que puede ser utilizada para beneficio personal de los atacantes.

La realidad actual de los sistemas de cómputo es que, cada vez en mayor medida, las organizaciones los requieren para llevar a cabo el procesamiento y almacenamiento de información diversa. También la complejidad de los sistemas tanto en hardware como en software ha aumentado, lo que en la mayoría de los casos aumenta la probabilidad de que se presenten fallas en la configuración y funcionamiento de las aplicaciones, las cuales serán perceptibles para los administradores, hasta que sea demasiado tarde o por lo menos, cuando ya haya causado problemas.

Además, los sistemas de cómputo no pueden operar de forma aislada, necesariamente tienen que compartir información e interactuar con otros sistemas dentro o fuera de la organización, por lo que es preciso que los equipos de cómputo y las aplicaciones informáticas que funcionan en éstos se encuentren interconectados en red. Esto hace ineludible que la administración de los sistemas de cómputo debe ser realizada, la mayoría de las veces, de manera remota y se hace patente la situación de que los administradores deben tener acceso remoto de manera segura a los sistemas, y a la vez evitar accesos no autorizados a éstos.

Desgraciadamente, los administradores de sistemas, en la mayoría de los casos, no crean una conciencia de los riesgos que pueden presentarse cuando se administran los sistemas de cómputo de manera remota, así como también no

toman en cuenta las consideraciones que deben tenerse al administrar los sistemas remotamente y por lo mismo no seleccionan adecuadamente las herramientas que deben utilizar para llevar a cabo sus funciones. Inclusive, a nivel organizacional, en muchas ocasiones no se cuenta con políticas o estándares relacionados a la selección y uso adecuado de herramientas de apoyo a la administración remota.

Adicionalmente, las herramientas disponibles como apoyo de la administración de sistemas, en muchas ocasiones solucionan sólo parte del problema en forma aislada y es necesario integrarlas para obtener mayores beneficios de tales herramientas. Desafortunadamente los administradores de sistemas, en muchas ocasiones por falta de tiempo y otras por desconocimiento, no investigan a detalle las características de las herramientas que utilizan y las posibilidades adicionales de utilización e integración con otras herramientas.

Añadiendo a esto que las características comentadas de los ataques informáticos (automatización, acción a distancia y técnicas de propagación) representan un factor a favor para los atacantes, el problema se torna muy serio. Por lo que todos estos elementos se conjugan para producir un panorama no muy atractivo para los responsables de la administración de los sistemas de cómputo quienes a su vez se encuentran con una constante necesidad de aplicar actualizaciones de software para mitigar las vulnerabilidades que se descubren cotidianamente, lo cual debe realizarse de una manera apropiada para evitar que en lugar de corregir un problema, se genere otro.

Por todo lo anterior, se hace evidente que se requieren metodologías y herramientas que permitan tener un control más preciso y detallado del estado que guardan los sistemas de cómputo y sus configuraciones; así como también es imprescindible contar con los elementos que permitan la protección de las configuraciones de los equipos y aseguren llevar a cabo una adecuada administración remota de los sistemas.

Capítulo 4. Alternativas de solución.

Para enfrentar la problemática expuesta, existen diversos esquemas de solución disponibles, los cuales se describirán en este capítulo. Pero antes de abordarlos, es conveniente identificar los elementos que deben ser tomados en cuenta para elegir la mejor opción.

4.1 Principales elementos.

Los elementos a considerar, están orientados al tipo de servicio informático que se pretende proporcionar, las políticas y procedimientos que se desean implantar y a la tecnología que debe ser utilizada para alcanzar los objetivos de seguridad informática deseados.

En cada parte de estos elementos deben tomarse cada una de las decisiones con un enfoque orientado a la seguridad informática. Lo más conveniente es precisamente avanzar en este orden, definiendo primero el tipo de servicio que se desea proporcionar, las políticas y procedimientos que soportarán el funcionamiento de dicho servicios, es decir, las reglas bajo las que operará y por último, realizar las investigaciones que sean necesarias para al final, seleccionar la mejor tecnología que permita la implantación del servicio requerido. La interrelación de estos elementos se muestra en la Figura 4-1.

Figura 4-1. Interrelación de elementos de servicios, políticas, procedimientos y tecnología.

Cada uno de estas partes deben ser consideradas con mucha atención para lograr una adecuada implantación de herramientas de seguridad informática cuando se administran de manera remota sistemas de cómputo conectados en red.

4.1.1 Servicios.

En primer término es necesario definir los servicios informáticos que serán proporcionados, los cuales son solicitados en base a parámetros establecidos, ya sea por requerimientos estrictos de la organización o como conveniencia para facilitar procesos administrativos y de operación. Es importante señalar que es

necesario investigar las opciones que se tengan disponibles a este respecto, para que desde un inicio se tenga un esquema de servicio confiable.

Como ya se ha mencionado, es de suma importancia identificar y elegir adecuadamente los servicios informáticos que se pretenden proporcionar en cada equipo de cómputo. En primer lugar, debe identificarse el tipo de funcionalidad que proporciona y si se trata de un servicio que utilizará solamente el administrador del sistema o si lo utilizarán todos los usuarios de la organización o sólo una parte de éstos. En segundo lugar, es necesario que se conozca el tipo de protocolo de comunicación que utiliza y el tipo de información que estará transmitiéndose por la red de datos. Además de todo esto, es muy importante que el personal encargado de administrar dichos servicios cuente con el suficiente conocimiento en el funcionamiento de cada uno de los servicios instalados en los equipos para estar en condiciones de configurarlos de manera óptima.

A partir de que se hayan seleccionado los servicios informáticos se procederá a proteger todos y cada uno. Particularmente y en relación a los servicios requeridos en los equipos para administrar de manera remota los sistemas de cómputo, se pueden enunciar los siguientes: sesión remota (acceso al shell del sistema operativo), autenticación, control de acceso, auditoría (bitácoras), monitoreo genérico del estado del sistema, transferencia de archivos y correo electrónico o servicio de mensajería electrónica.

Como parte básica para el adecuado funcionamiento de los sistemas de cómputo se encuentra la realización de una configuración apropiada y la selección y habilitación de sólo aquellos servicios requeridos y no más. Esto es primordial para tener una mayor posibilidad de éxito en el logro de los objetivos de seguridad informática trazados. El impacto principal de estas dos prácticas, es que disminuyen considerablemente el nivel de riesgo del sistema en su conjunto, ya que con esto se tienen menos elementos involucrados a monitorear.

Es imprescindible que los administradores de sistemas se documenten adecuadamente de las funcionalidades que ofrece cada servicio y cómo afecta a los elementos de la configuración de seguridad, inclusive es conveniente, en caso de no tener suficiente conocimiento, solicitar la opinión de otros administradores de sistemas con mayor experiencia, respecto a la conveniencia de utilizar o no algún servicio o, de ser posible, sustituirlo por alguna alternativa que proporcione mejores funcionalidades para el esquema de seguridad establecido.

4.1.2 Políticas, estándares y procedimientos.

Las soluciones de seguridad informática deben estar orientadas a garantizar que las políticas y procedimientos establecidos por la organización se cumplan, por este motivo es muy importante seleccionar las herramientas adecuadas que faciliten el control de la aplicación de las políticas y la verificación de la ejecución de los procedimientos.

Esto es notorio cuando se trata de administrar sistemas de cómputo de manera remota, ya que la interconexión de equipos en las organizaciones es cada vez más evidente y de práctica común, lo cual permite que la información almacenada y trasmitida se encuentre más expuesta a ataques o accesos no autorizados. Este escenario obliga a que las medidas que deben tomarse para salvaguardar la información, deben adaptarse a las necesidades y condiciones específicas de cada organización.

Los administradores de sistemas necesitan contar con elementos para realizar su trabajo de manera más segura, apegados a las políticas y procedimientos establecidos en la organización, ya que muchos de los sistemas informáticos han sido diseñados sin tomar en cuenta aspectos de seguridad informática. Antes se restringía físicamente el acceso a los equipos, ahora se puede tener acceso prácticamente desde cualquier parte del mundo, ya que independientemente de las restricciones físicas que puedan tener los equipos o el lugar donde residen, son las restricciones lógicas y de configuración de las aplicaciones, las que efectivamente delimitan el acceso a la información.

Ningún bien puede ser protegido al 100% de robos, espionaje o daño accidental. Esto es especialmente cierto para los bienes informáticos. Si un intruso está suficientemente determinado, es paciente y cuenta con las habilidades necesarias, ningún sistema le será impenetrable y ninguna solución de seguridad será efectiva durante mucho tiempo [33].

Tradicionalmente, la mayor atención de la seguridad informática se ha centrado en procedimientos tácticos preventivos o reactivos, y se ha puesto menos énfasis a examinar propiamente las políticas de seguridad y mantener su vigencia [33].

Aún las tecnologías más avanzadas y confiables pueden ser socavadas por decisiones erróneas en las políticas o por prácticas operacionales incorrectas. Nuevamente el elemento humano en la seguridad informática es comúnmente la parte más débil del proceso, y por lo tanto debería ponerse mayor énfasis en la seguridad al momento de diseñar políticas y procedimientos [33].

Algunos de los puntos importantes a tomar en cuenta para esta parte son: la definición de las políticas de manera adecuada para la organización en que serán implantadas, la educación que debe brindarse a los usuarios en dichas políticas, las revisiones constantes para verificar la adecuada implantación y la realización de correcciones [33].

En este punto es importante diferenciar, como parte del establecimiento de un lenguaje común, las diferencias entre una política, un estándar y un procedimiento. En principio, una política es normalmente un documento que delinea los requerimientos o reglas específicas que deben ser cumplidas. Por otro lado, un estándar es habitualmente una colección de requerimientos específicos a un sistema que deben ser cumplidos por todos los involucrados en el uso de dicho sistema. Y por otra parte, un procedimiento es una colección de "sugerencias"

para realizar las actividades cotidianas en algún sistema específico, no necesariamente tienen que cumplirse pero es recomendada fuertemente su utilización. Como resultado de esto, para que una política sea realmente efectiva, debe hacer referencia a los estándares y procedimientos establecidos dentro de una organización [30].

Para el establecimiento de políticas adecuadas, más que definir reglas genéricas, es necesario evaluar el entorno de la organización para lo cual es conveniente contestarse algunas preguntas, como primer paso ¿Qué datos necesitan protección?, ¿Por qué esos datos necesitan protección? ¿Cuáles son los impactos negativos si ocurre una brecha de seguridad o si existe pérdida o destrucción de información?, para esto en particular debe ser considerado principalmente la valoración del impacto financiero. Adicionalmente, y como un reporte final, se deben tomar en cuenta el porcentaje de ocurrencia anual. Como segundo paso, habrá que contestar adicionalmente ¿Está justificada financieramente la política? ¿Está claramente definidas las responsabilidades que tiene cada persona? ¿Será obligatoria? ¿Puede ser implantada? ¿Existe experiencia en la organización para implantarla y mantenerla? [44].

4.1.3 Tecnología.

Este punto debe considerarse al final, una vez completados los dos anteriores, ya que el grado de avance de la tecnología es importante, en muchas ocasiones existen soluciones teóricas viables, pero que no pueden ser llevadas a cabo debido a las limitantes en el desarrollo de la tecnología requerida para implantarlas. También existen soluciones que tecnológicamente son posibles, pero que pueden ser mal implementadas, debido al desconocimiento o a una incorrecta aplicación de la tecnología utilizada.

La aplicación adecuada de la tecnología es un factor muy importante que debe ser considerado ampliamente cuando se busca implantar soluciones de seguridad informática, ya que la solución deberá estar orientada al tipo de servicios que requiere la organización de que se trate.

Es importante seleccionar las soluciones tecnológicas después de haber realizado una revisión cuidadosa del estado de la tecnología y haber considerado diversas variables que incluyen, entre otras, el costo total de propiedad, el retorno de la inversión y el desarrollo y respaldo que, en el futuro, tendrá dicha solución tecnológica.

Cabe señalar que la utilización de la tecnología deberá estar orientada por las políticas establecidas por la organización que la utilizará. La adopción de tecnología en una organización debe ser un proceso balanceado entre adecuar la tecnología a las políticas establecidas y ajustar la organización a la tecnología.

Debido a esto, es de suma importancia evaluar adecuadamente el estado de la tecnología, en virtud del tipo de servicio informático a implantar y en relación con la organización a la cual se aplicará.

4.2 Soluciones diversas.

Para ubicar adecuadamente las soluciones que pueden utilizarse, es necesario identificar las principales actividades involucradas en la administración remota de sistemas de cómputo, mismas que deberán ser aseguradas. En esta sección, se describirá cada una de las principales actividades que se desarrollan para administrar remotamente sistemas de manera segura, así como también las soluciones disponibles para cada una.

Se comentará cada una de estas soluciones cuyo éxito depende en gran medida de una adecuada implantación e integración con otras herramientas. Cabe señalar que estas soluciones por sí solas no representan un frente a todos los problemas de seguridad en una organización y deben siempre utilizarse como piezas claves que deben servir de apoyo a las políticas y procedimientos establecidos en la organización.

Es importante mencionar que las soluciones expuestas en este capítulo están orientadas al sistema operativo Linux y son desarrollos basados en código abierto de libre obtención aunque la mayoría podrán aplicarse a otros sistemas operativos. Será necesario discutir los beneficios que aporta cada herramienta, así como las ventajas y desventajas con respecto a otras herramientas. Sobre todo será importante tener en cuenta los conceptos, más que las herramientas en particular.

4.2.1 Organización por funciones.

Como se ha mencionado, uno de los primeros pasos para implantar seguridad a la administración remota de sistemas, es identificar los diversos rubros que dicha actividad abarca, para esto, se pueden identificar de manera general los siguientes: conexión remota, autenticación, control de acceso, integridad de archivos, control de actualización de software y configuraciones, análisis de bitácoras e identificación de vulnerabilidades. A continuación se detalla cada uno y se comentan las herramientas que pueden ser utilizadas para llevar a cabo estas tareas de manera segura.

4.2.1.1 Conexión remota.

Una actividad básica para administrar remotamente sistemas de cómputo, es poder conectarse desde algún punto de la red a los equipos que se desea administrar. Cuando se inició el desarrollo de los protocolos de comunicación TCP/IP, el método inicial de conexión remota a los equipos de cómputo fue telnet, el cual permite ejecutar comandos en el equipo remoto mediante una terminal.

Aunque telnet es útil, es una forma insegura de conectarse debido a que el protocolo utilizado en dicho servicio transmite en claro la contraseña por la red, de esta manera puede ser obtenida dicha contraseña mediante "sniffers", lo que facilita a los intrusos el ingreso a los sistemas. Sin embargo, existen diversas alternativas que pueden ser utilizadas para este propósito, cada una de las cuales tiene ventajas y desventajas respecto a las otras.

4.2.1.1.1 Secure Shell (SSH).

Una de las alternativas más populares es ***secure shell*** (SSH), ya que permite conectarse a equipos, ejecutar comandos y copiar archivos entre una máquina y otra utilizando autenticación fuerte y comunicaciones seguras aún y cuando la comunicación se realice mediante canales inseguros.

Una de las razones de que esta alternativa sea tan popular, se debe a su relativa facilidad de instalación, configuración y uso, así como también a que muchos sistemas operativos lo incluyen como parte de su distribución.

De manera general, todo lo que se requiere para tener operando este servicio es tener un servidor y un cliente de ***secure shell*** funcionando. ***OpenSSH*** es una excelente alternativa para ambientes Linux y Unix, tanto para servidor como cliente. Para ambientes Windows, un cliente muy popular es ***PuTTY***[45].

Cabe señalar que ***secure shell*** sólo cifra la comunicación específica que se realiza, por lo que la sobrecarga al ancho de banda de la red por la que transitan los paquetes es mínima.

4.2.1.1.2 Virtual Private Networks (VPN).

Otra alternativa para asegurar las conexiones remotas entre los equipos, pero que relativamente es más compleja, es construir redes privadas virtuales (VPN, por sus siglas en inglés).

Cuando se implanta una VPN, todo el tráfico que fluye en dicha red está cifrado, por lo que existe una pequeña penalización en el ancho de banda disponible. Aunque lo importante es que funciona de manera transparente para cualquier aplicación, formando un "túnel" entre los diversos equipos configurados para operar en la VPN, por lo que no requiere que se recompilen programas de aplicación.

La herramienta ***FreeS/WAN*** es una muy buena opción para crear VPNs entre servidores Linux, ya que implementa IPSec e IKE para crear túneles seguros en redes inseguras. Utiliza criptografía para autenticar y cifrar las conexiones, aunque

[45] http://www.chiark.greenend.org.uk/~sgtatham/putty/.

todo esto como ya se mencionó anteriormente causa un pequeño incremento tanto en el procesamiento de los equipos como en el uso de ancho de banda de la red, por lo que hay que valorar adecuadamente cuando es conveniente utilizar una VPN y cuando no lo es.

4.2.1.2 Autenticación.

La autenticación es un proceso básico de los sistemas multiprocesamiento para permitir o denegar el acceso a un equipo de cómputo y es un punto muy importante que debe considerarse. Generalmente, en los sistemas tipo Unix, la autenticación se realiza, de manera local en cada equipo, a través de un archivo llamado /etc/passwd.

En el /etc/passwd se almacena la información de los usuarios del sistema, incluyendo la contraseña (cifrada) con la que se identifica cada usuario, por lo mismo dicho archivo debe ser visible para todos los usuarios del sistema. Uno de los problemas de este esquema es que la contraseña del "superusuario"[46] puede ser obtenida por cualquier usuario del sistema mediante intentos de adivinarla por fuerza bruta. Debido a esto se incorporó /etc/shadow, el cual es un archivo visible sólo para el "superusuario", por lo que es importante habilitarlo. Adicionalmente también es importante utilizar un algoritmo de cifrado más avanzado, como md5, en lugar de utilizar el default (crypt).

4.2.1.2.1 Pluggable Authentication Modules (PAM).

Por lo general, la mayoría de las aplicaciones están configuradas para utilizar /etc/passwd y /etc/shadow como método de autenticación, pero existen otros esquemas que se han desarrollado para facilitar la administración de la autenticación cuando se incorporan una gran cantidad de equipos. Estos esquemas son NIS, NIS+ y LDAP entre otros, aunque para implementarlos de manera directa se requiere programar dentro de cada uno de los servicios de red, la funcionalidad que les permita utilizar alguno de estos esquemas para realizar autenticación.

Sin embargo, para disminuir la complejidad se desarrollaron esquemas de autenticación flexibles que pueden resultar convenientes de utilizar, un ejemplo de esto es ***Pluggable Authentication Modules*** (PAM[47]), el cual permite configurar un sólo esquema de autenticación intermedio dentro de cada una de los servicios de red y aplicaciones de programa, pero que puede ser ajustado mediante archivos de configuración sin tener que reprogramar y recompilar las aplicaciones que requieren algún tipo de autenticación.

---

[46] Se le conoce como "superusuario" a aquel usuario que no tiene restricciones para realizar modificaciones al sistema, por lo que comúnmente es el objetivo primordial de los atacantes. Para el caso de sistemas tipo Unix este usuario es conocido como "root", en sistemas Windows NT/2000 el usuario es "Administrator" o "Administrador".

[47] http://www.kernel.org/pub/linux/libs/pam/.

Algo que es muy importante de tener en cuenta a este respecto, es la necesidad de realizar revisiones periódicas para verificar que los sistemas no tienen usuarios con contraseñas débiles. Para realizar estas actividades pueden resultar muy útiles herramientas como ***John The Ripper***, ***Crack/Libcrack*** y ***pam_cracklib***.

4.2.1.2.2 Kerberos.

Esta herramienta implementa un protocolo de autenticación en red. Está diseñado para proporcionar autenticación fuerte para aplicaciones cliente-servidor utilizando criptografía de llave pública. Existen versiones comerciales así como también de libre distribución.

Kerberos resuelve el problema de realizar la autenticación sin necesidad de transmitir una contraseña por la red. Para realizar esto, ***kerberos*** utiliza el modelo de distribución de llaves desarrollado por Needham y Schroeder. Una llave es utilizada para cifrar y descifrar mensajes cortos, precisamente las llaves proporcionan la base de la autenticación en ***kerberos***.

Desgraciadamente, ***kerberos*** es relativamente difícil de configurar y administrar, por lo que bajo ciertas condiciones o circunstancias específicas puede resultar no ser una buena opción para el administrador de sistemas promedio.

4.2.1.2.3 LDAP.

LDAP (Lightweight Directory Access Protocol) es una alternativa al uso de /etc/passwd y /etc/shadow. La ventaja que LDAP tiene sobre la utilización de archivos de contraseñas, es que toda la información acerca de los usuarios y grupos puede ser mantenida en un servidor administrado centralmente, por lo que la información del usuario no tiene que ser necesariamente replicada.

La manera más sencilla para integrar en los sistemas autenticación con LDAP es utilizar PAM, por lo que cada servicio que requiere autenticación, puede ser configurado mediante los archivos de configuración de PAM para utilizar diversos métodos de autenticación. Por lo que es posible, utilizando los archivos de configuración de PAM, tener claramente una lista de los requerimientos particulares de autenticación que un usuario debe cumplir para obtener acceso a un recurso.

4.2.1.3 Control de acceso.

El control de acceso es básico al interior de los sistemas multiusuario y existen diversas opciones al respecto. Los sistemas tipo Unix tienen un esquema por default rwx/ogw (protección de lectura, escritura y ejecución para el dueño, grupo y demás usuarios).

Aunque ese esquema es funcional, tiene ciertas debilidades y limitantes, por lo que se han desarrollado diversas alternativas que pueden ser configuradas de manera adicional en los sistemas tipo Unix. Una de estas alternativas o mejoras son las listas de control de acceso (ACL, acces control lists), las cuales permiten organizar a más detalle la asignación de permisos.

Otra alternativa importante a tener en cuenta, particularmente para sistemas Linux es ***LIDS***, el cual limita las atribuciones que tiene "root" como "superusuario" y protege al sistema en caso de que algún atacante logre obtener acceso a este usuario, con lo que se logra limitar en gran medida el alcance de algún ataque o intrusión que pudiera presentarse.

4.2.1.4 Integridad de archivos.

En este rubro, ya se han comentado anteriormente los diversos aspectos que deben tenerse en cuenta para evitar que se realicen modificaciones no autorizadas a los archivos que residen en un equipo de cómputo. Para llevar a cabo estas actividades, se pueden utilizar herramientas adicionales como ***AIDE***.

Algunos puntos que vale la pena tener en cuenta, es la realización de un inventario inicial de los archivos que deben monitorearse de manera constante, además de definir adecuadamente los mecanismos de alerta que serán utilizados.

Por lo general se utilizan implementaciones del algoritmo MD5 para verificar que los archivos seleccionados se mantengan sin alteraciones no autorizadas.

4.2.1.5 Control de la actualización de software y configuraciones.

Conforme los sistemas operativos y las aplicaciones incorporan más funcionalidades y se vuelven más complejos, se hace más evidente la necesidad de contar con esquemas de control de versiones y empaquetamiento de programas que faciliten la estandarización y actualización del software que reside en un equipo.

La mayoría de los sistemas operativos cuentan con su propio sistema de empaquetamiento de software, con el cual se puede instalar, actualizar, revisar y quitar elementos particulares de software.

Una de las ventajas de los sistemas de empaquetamiento de libre distribución como ***RPM***, ***apt*** y ***OpenPKG*** entre otros, y a diferencia de los comerciales, es que están ampliamente documentos y soportados por una gran cantidad de usuarios, además de que permiten realizar ajusten más específicos dado que se cuenta con el código fuente.

4.2.1.6 Análisis de bitácoras.

La revisión constante de las bitácoras que generan las diversas aplicaciones que funcionan sobre un sistema operativo, es una función muy importante para detectar errores en el momento en que suceden, identificar intentos de accesos no autorizados o determinar la causa de algún problema.

En relación a la utilidad que proporciona la revisión de bitácoras, se pueden organizar las herramientas que apoyan esta actividad en herramientas de análisis preventivo y herramientas de análisis forense (reactivo). A continuación se comentará cada una.

4.2.1.6.1 Análisis preventivo.

En este rubro se tienen herramientas que obtienen información de las bitácoras, prácticamente de manera instantánea y toman acciones de manera inmediata al detectar alguna actividad anómala.

Las herramientas para este tipo de monitoreo que pueden mencionarse son ***Logcheck***, ***Swatch***, ***scanlogd***, ***IPTraf*** y ***Snort***.

4.2.1.6.2 Análisis forense.

Cuando se ha presentado un incidente, independientemente del alcance o impacto que haya tenido, es necesario realizar un estudio para conocer las causas que permitieron que se llevara a cabo dicho ataque.

Para poder obtener la información que permita determinar las causas del incidente se deben utilizar herramientas especializadas para dichas funciones, algunas que se pueden mencionar son ***The Coroner's Toolkit***, ***AIDE*** y ***dsniff***.

4.2.1.7 Identificación de vulnerabilidades.

Otro aspecto que no se debe perder de vista es la ejecución periódica de programas que revisen la presencia de vulnerabilidades en los sistemas administrados.

Existen diversas herramientas de libre distribución que pueden mencionarse a este respecto, entre las que destacan ***Nesus***, ***nmap***, ***dsniff***, ***whisker***, ***SARA*** y ***HUNT***.

Cabe señalar que estas herramientas deben actualizarse con el objeto de que puedan detectar las nuevas vulnerabilidades descubiertas. Adicionalmente al uso de estas herramientas es necesario que los administradores se suscriban a listas de discusión donde se obtenga información sobre nuevas vulnerabilidades de las aplicaciones con que cuenta la organización para la cual labora.

4.2.2 Protección de la red.

Adicional a lo ya comentado es conveniente contar con herramientas de protección de la red a la cual se encuentran conectados los equipos y que permitirán una mayor tranquilidad para los administradores, pero entendiendo que las herramientas por sí mismas no proporcionan una protección completa, sino el conocimiento y habilidades de los administradores es lo que a fin de cuentas permitirá tener sistemas confiables en su operación.

Muchos de los protocolos utilizados en una red no proporcionan seguridad alguna, por lo que las aplicaciones que utilizan dichos protocolos envían y reciben información sensitiva en forma de texto en claro, el cual puede ser capturado por las personas incorrectas. Inclusive muchas aplicaciones son desarrolladas considerando que serán utilizadas por personas "honestas" en las que se puede confiar, lo que desgraciadamente no sucede.

Con el objeto de reforzar la seguridad en las redes de datos de las organizaciones, han surgido diversas soluciones a problemas específicos a lo largo de la evolución de la seguridad informática, dentro los cuales destacan "firewalls", detectores de intrusos y de manera más general, herramientas de monitoreo.

4.2.2.1 Firewalls.

Los "firewalls" son dispositivos de protección colocados como divisores de las fronteras entre una red interna de equipos, que la organización necesita proteger, y una red externa sobre la cual la organización no tiene control [31]. El propósito principal del "firewall" es controlar el acceso a la red interna desde la red externa, y viceversa. Otro uso que tienen los "firewalls" es que pueden servir para controlar los accesos entre subredes dentro de la red interna [31].

Actualmente, un "firewall" puede funcionar inclusive como una frontera entre la red de datos y el sistema operativo de los equipos de cómputo. Cabe señalar que la seguridad de un "firewall" radica en que se haya configurado adecuadamente. Los "firewalls" pueden agruparse en tres categorías: filtrado de paquetes, filtrado de aplicaciones e híbridos [31].

Desgraciadamente algunas organizaciones consideran que por el mero hecho de colocar un "firewall" solucionarán sus problemas de seguridad informática, lo cual es totalmente una mala concepción del problema, ya que en este caso se asume que los atacantes se encuentran fuera de la organización, lo cual deja de lado el hecho de que dentro de la organización también se encuentran personas que pueden tener diversos intereses para llevar a cabo ataques informáticos a la infraestructura de cómputo y comunicaciones.

Existen diversas formas para implantar un "firewall", ya sea de libre distribución o comercial. Entre las herramientas más convenientes de utilizar para filtrar servicios

y funcionar como "firewall" incorporado al sistema operativo Linux, se encuentra ***iptables***.

4.2.2.2 IDS.

IDS son las siglas para "Intrusion Detection Systems" (Sistemas de Detección de Intrusiones), lo cual se refiere a las técnicas y las herramientas de seguridad informática utilizadas para detectar actividad inapropiada, incorrecta o anómala en un equipo de cómputo o una red [32].

Los IDS identifican patrones específicos en el tráfico de la red o bitácoras del sistema operativo de los equipos de cómputo. Los IDS pueden ser subdivididos entre los que operan directamente en un equipo de cómputo para detectar actividad maliciosa en dicho equipo (conocido como IDS de host), y los IDS que operan sobre el flujo de datos de una red (comúnmente llamado IDS de red) [32].

El término "intrusión" es utilizado para referirse a los ataques que se realizan desde las redes externas a la organización, mientras que "mal uso de los recursos" es utilizado para referirse a los ataques que se realizan desde las redes internas de la organización. Sin embargo, la mayoría de la gente no hace estas distinciones [32].

Algunas herramientas que pueden utilizarse para llevar a cabo funciones de detección de intrusos se encuentran principalmente ***snort***, ***scanlogd***, ***logcheck*** y ***LIDS***. Estas herramientas son relativamente fáciles de obtener, instalar, configurar y proporcionan grandes beneficios para detectar y proteger los sistemas en contra de actividades sospechosas o no permitidas.

4.2.2.3 Herramientas de monitoreo.

Las herramientas de monitoreo son muchas y muy variadas, y en general permiten a los administradores de sistemas contar con información precisa del estado de los sistemas de cómputo en una red en cualquier instante de tiempo.

Existen herramientas especializadas en monitorear diferentes aspectos de un sistema de cómputo, desde las que interpretan la información obtenida directamente del hardware, hasta las que verifican el funcionamiento de las aplicaciones.

Estas herramientas pueden interactuar con otros servicios para que en conjunto, identifiquen problemas, disparen alarmas y notifiquen a los encargados por diversas vías (correo electrónico, mensaje instantáneo, pager, teléfono o algún otro medio).

En relación a la seguridad informática, las herramientas de monitoreo permiten verificar la disponibilidad de los sistemas de cómputo y, mediante el análisis de la

información recolectada en cierto periodo de tiempo, obtener patrones de comportamiento de dichos sistemas que permitan incluso predecir fallas o contratiempos que se presentarán en el futuro.

La selección de herramientas de monitoreo deberá realizarse en base al número y tipo de los sistemas de cómputo que se vayan a monitorear, a las necesidades de monitoreo que se requieran y a los recursos que pueda invertir la organización de que se trate. En este rubro existe una gran cantidad de herramientas de libre distribución que proporcionan diversas funcionalidades y características, pero como ya se ha comentando anteriormente, es muy importante dedicar suficiente tiempo para conocer a fondo la herramienta que se quiera utilizar antes de ponerla a funcionar en ambientes productivos.

Existen herramientas de este tipo que son de libre distribución, entre las que se pueden mencionar ***Cheops***, ***Netsaint***, ***mrtg***, ***lm_sensor***, ***sar***, ***Netcat***, ***IPTraf*** y ***Nagios***,

4.3 Metodologías.

Una parte fundamental en la implantación de soluciones de seguridad informática es la metodología[48] que se utilizará para implantar dicha solución, ya que llega a ser más importante la metodología que se emplea que las herramientas a utilizar, por lo que es necesario dedicarle suficiente atención y cuidado al seleccionar o desarrollar alguna.

Es decir, una metodología adecuada nos permitirá que inclusive utilizando herramientas modestas en sus funcionalidades, puedan llegar a componer una solución robusta, no siendo así lo contrario, ya que si la metodología de implantación no es apropiada, aún y cuando se utilicen herramientas con muchas funcionalidades no se podrá garantizar una solución sólida.

Desgraciadamente, hasta hace poco tiempo, toda la información de seguridad que se encontraba en Internet con respecto a metodologías era generalmente, o insustancial o secreta. Frases como "utilizamos una metodología única, desarrollada internamente y herramientas de auditoria..." eran un comunes. Sin embargo, esto no significa que muchos proveedores de soluciones de seguridad no tengan herramientas propietarias [34].

A continuación se comentará, de manera general, las principales características de las metodologías existentes en la actualidad, que dicho sea de paso, son más fáciles de tener acceso.

---

[48] Una metodología es la aplicación coherente de un método. Método es un conjunto de operaciones ordenadas con que se pretende obtener un resultado. Y, coherencia es la conexión de unas cosas con otras.

4.3.1 Aspectos comunes.

Prácticamente, la mayor parte de las metodologías de análisis de la seguridad en las organizaciones, abarcan los siguientes puntos [34], algunos de los cuales ya se han comentado anteriormente:

- Inventario de la red.
- Escaneo de puertos.
- Identificación de servicios.
- Identificación de sistemas de cómputo.
- Identificación y verificación de vulnerabilidades.
- Pruebas a las aplicaciones.
- Pruebas a los dispositivos de red.
- Pruebas a sistemas de seguridad.
- Pruebas a "firewalls"
- Pruebas a sistemas de detección de intrusos.
- Pruebas a contraseñas de usuarios.
- Pruebas de negación de servicios (DoS).

Lo que describen las metodologías para cada uno de estos puntos, es relativo a los datos que deben recolectarse para contar con elementos que permitan realizar una evaluación y también hacen énfasis en la necesidad de recolectar dichos datos y realizar pruebas adecuadas a la infraestructura de cómputo y comunicaciones, para determinar el nivel de seguridad con que cuenta la organización a la que se aplique dicha metodología.

Generalmente lo que se logra con esto es tener una "foto fija" de la seguridad con que se cuenta, esto es debido a que una prueba de seguridad a una organización no es más que la visibilidad de un sistema en un momento concreto en el tiempo. En este periodo de tiempo, las vulnerabilidades conocidas, las amenazas conocidas, y las configuraciones del sistema conocidas no han cambiado para ese instante y por eso se le debe considerar una "foto fija" [34].

Por lo que a este punto se refiere, no hay muchas diferencias entre una metodología y otra, pero lo que sí es importante de tomar en cuenta es que se tengan los elementos suficientes para realizar una evaluación adecuada de la seguridad para el tipo y tamaño de organización de que se trate.

4.3.2 Diversos enfoques.

Aunque en el punto anterior se puede observar que existen muchas coincidencias entre la mayoría de las metodologías existentes, asimismo concurren muchas diferencias en cuanto a la forma de llevar a cabo cada una de las acciones a desarrollar, las herramientas a utilizar, la manera en que se recolectarán y organizarán los datos, el análisis que debe hacerse sobre dichos datos y la forma en que se deberá presentar la información obtenida.

En este punto es donde se debe mayor atención para estar en concordancia a las necesidades y posibilidades de la organización a la que se aplicará la evaluación de seguridad, ya que dependiendo de las herramientas requeridas y el nivel de preparación de los recursos humanos requeridos para llevarla a cabo, el factor costo será el principal afectado.

4.3.3 Amplias y complejas.

Dependiendo de los alcances que tenga cada metodología, éstas pueden variar en amplitud y complejidad. Algunas pueden ser muy estrictas en su aplicación, mientras que otras pueden ser más flexibles.

En virtud de esto habrá metodologías que puedan servir para una organización pequeña y no para una de mayor tamaño, o dependiendo de los objetivos que se tengan al realizar una evaluación de seguridad, puede resultar conveniente utilizar una u otra.

4.4 Problema existente: seguridad integrada.

Uno de los retos en el área de la seguridad informática es asegurar que las personas correctas tengan acceso a los recursos de cómputo y la información correctos, y a la vez que no se permita el acceso a estos recursos a las personas incorrectas [35].

La administración remota de sistemas es práctica común en cualquier organización que cuente con equipos de cómputo y redes, por lo que es muy importante la necesidad de contar con herramientas que puedan, de manera integral, apoyar a los administradores de sistemas en sus labores cotidianas, a la vez que se robustece la seguridad en dichas actividades.

Las soluciones actuales de seguridad por lo general consisten en múltiples productos o herramientas, los cuales deben ser adquiridos, instalados, administrados y actualizados de manera separada. Ante este escenario los administradores de sistemas se enfrentan a diversos problemas de interoperabilidad entre productos y a una lenta respuesta ante ataques o vulnerabilidades presentadas en los diferentes productos de software utilizados en la infraestructura de cómputo y comunicaciones de la organización de que se trate.

De aquí resulta conveniente contar con una solución integrada que permita obtener los máximos beneficios de cada una de las herramientas utilizadas y al mismo tiempo poder reaccionar de manera expedita ante cualquier ataque o vulnerabilidad en el software que pueda presentarse, manteniendo en todo momento el control de las configuraciones de los equipos.

Cabe señalar que este tipo de integraciones por lo general, lo realizan los mismos administradores de sistemas y están comúnmente orientados a resolver alguna situación en específico y no como una herramienta configurable para diversos tipos de equipos y configuraciones.

Aunque de inicio pudiera pensarse en este tipo de soluciones como algo que añade complejidad y pudiera resultar en un desastre, la realidad no necesariamente es así, ya que la parte compleja se realiza una sola vez y si se tiene el debido cuidado en el análisis, diseño y documentación de la integración de las herramientas de seguridad, en verdad se obtienen grandes beneficios en los aspectos de administración, confiabilidad y escalabilidad de la solución.

Las soluciones, metodologías y herramientas comentadas en este capítulo pueden ser utilizadas en conjunto para lograr mayores beneficios orientados a mejorar la administración remota de la seguridad de los sistemas y facilitar las labores de los administradores de sistemas. Adicionalmente, cabe resaltar el hecho de que al utilizar herramientas de libre distribución, se tiene el control total de la implantación de dichas herramientas y pueden personalizarse tanto como se requiera, permitiendo la integración e interoperabilidad de unas con otras.

Capítulo 5. Propuesta de solución.

Como se ha podido observar hasta ahora, es necesaria la integración de herramientas en una solución que proporcione un esquema de seguridad integrada para la administración de sistemas remotos. En este capítulo se propone una solución cuyo objetivo es precisamente la integración de herramientas que permitan, principalmente la automatización de tareas en los sistemas de cómputo y que a la vez sea lo suficientemente escalable para ser implantada en un gran número de equipos.

En primer lugar se comenta el esquema general y modelado de la solución propuesta, para luego explicar el análisis y diseño de la solución, incluyendo las consideraciones bajo las que se toman las decisiones y se construye dicha solución.

5.1 Esquema general.

Para el caso en particular del presente trabajo, la solución propuesta se enfoca a la integración de herramientas de monitoreo de libre distribución y de fácil obtención, que puedan ser utilizadas de manera conjunta para apoyar las labores de mantenimiento cotidianas de los administradores de sistemas, relativas a la administración remota de los sistemas de cómputo, no importando el tipo de organización para la cual prestan servicios.

Uno de los principales objetivos al utilizar herramientas de libre distribución es permitir la autonomía en el uso de dichas herramientas, ya que dichas herramientas no dependen del pago de licencias ni tampoco están limitadas al uso dentro de alguna organización en particular. Adicionalmente, cabe señalar que con esto no se limita, en ningún modo, la utilidad que pueden proporcionar a las tareas de la administración remota de sistemas, ya que las funcionalidades que proporcionan las herramientas disponibles en muchos casos están a la altura de las herramientas comerciales.

5.1.1 Situación actual.

Generalmente la administración de sistemas es una actividad que, incluso actualmente, lleva de manera manual la aplicación de cambios y no se tiene demasiada automatización de las tareas, esto obedece principalmente a que en los inicios de la administración de sistemas el número de equipos a administrar eran unos cuantos y por lo general eran muy diferentes unos de otros, por lo que resultaba más fácil para los administradores la aplicación manual de cambios.

En este tipo de escenarios, los administradores de sistemas se enfocan por lo general a realizar trabajo rutinario y a atender de manera reactiva los problemas y fallas que presentan los equipos a su cargo, por lo general no dedican suficiente

tiempo a realizar análisis y planeación de la infraestructura hasta que es inevitable hacerlo.

Aunque es de señalar que automatizar y proporcionar cierta autonomía a los sistemas de cómputo es una necesidad muy importante en las áreas de administración de sistemas de las organizaciones en la actualidad, debido principalmente a que el número de equipos de procesamiento de datos que se requieren administrar ha aumentado increíblemente en los años recientes y en la mayoría de los casos existe una tendencia a homogenizar el tipo de sistemas de cómputo utilizados, por lo que básicamente existen dos alternativas para solucionar esta problemática, las cuales son: contratar más administradores de sistemas o automatizar en lo posible la administración de los sistemas de cómputo.

Adicionalmente, se encuentra la necesidad de monitorear los equipos de cómputo con el objeto de tener datos que permitan realizar un adecuado análisis que a fin de cuentas, posibilite la toma de acciones en las direcciones más convenientes. Estas por sí solas son actividades que pueden resultar muy pesadas y propensas a errores si son realizadas exclusivamente por los administradores de sistemas.

Como resultado de lo anterior, resulta muchísimo más conveniente invertir recursos en automatizar la administración de sistemas, ya que además de ser una solución efectiva a largo plazo, permite disminuir el trabajo rutinario a los administradores de sistemas y enfocar sus esfuerzos al análisis, organización, optimización y planeación de la infraestructura de cómputo de la organización.

5.1.2 Modelo de solución.

La solución propuesta en este trabajo, como ya se ha mencionado, se enfoca a la integración de herramientas que permitan automatizar las actividades de los administradores, relativas a la administración remota de sistemas, principalmente en los aspectos de revisión de la integridad de archivos, identificación de vulnerabilidades de los sistemas, actualización de software, revisión de bitácoras y autonomía de los sistemas en la realización de tareas.

Esta integración deberá estar orientada a proporcionar esquemas de monitoreo de los servicios informáticos que proporcionan los sistemas de cómputo, sustentados en las políticas y procedimientos definidos por los responsables de la operación de dichos servicios. Esto deberá realizarse de esta manera, en virtud de que existen soluciones por separado que funcionan muy bien para problemas específicos, pero es necesaria la integración adecuada de dichas soluciones para que funcionen en conjunto y de manera orquestada.

5.1.2.1 Administración integrada de la seguridad.

Para lograr el cometido planteado, es necesario involucrar diversos conceptos en la solución. Estos conceptos son relativos, como ya se ha mencionado, a la definición de servicios, políticas, procedimientos y tecnología.

La integración de estos conceptos deberá realizarse en un ambiente organizado mediante una metodología que deberá estar compuesta, al menos, por las fases de evaluación, planeación, implantación y monitoreo. A este esquema se le denominará "Administración Integrada de la Seguridad", y la relación que guardan todos los elementos se ilustra en la Figura 5-1.

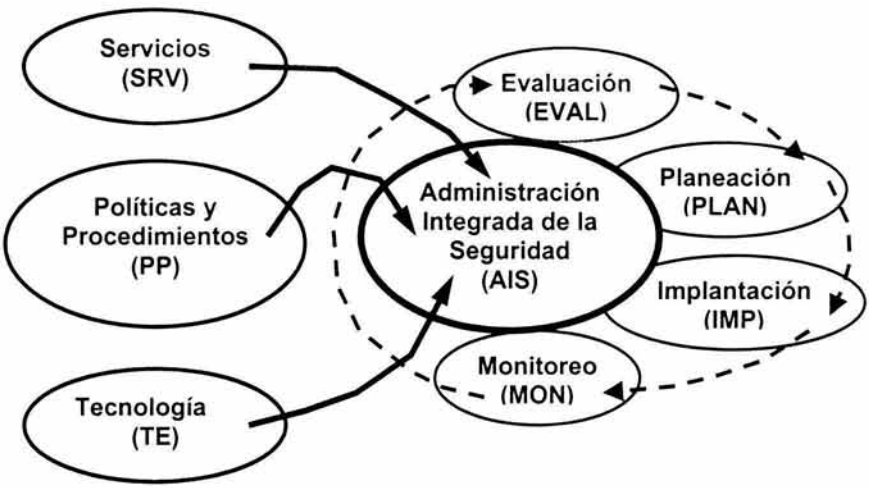


Figura 5-1. Relación de los conceptos involucrados en la administración integrada de la seguridad.

Se puede observar en la Figura 5-1, que los servicios, políticas, procedimientos y tecnología, confluyen en fases cíclicas y repetitivas de evaluación, planeación, implantación y monitoreo, con el fin de obtener un esquema inicial del tipo de servicios a utilizar, las políticas y procedimientos a implantar y la tecnología requerida para soportar el funcionamiento de la operación de los sistemas seleccionados.

A continuación, se describe cada una de las fases mencionadas. Cabe señalar que en cada fase se generará diferente documentación, por lo que es importante que dichos documentos guarden un formato homogéneo y que sean producidos en un formato de documento que pueda fácilmente ser consultado en cualquier tipo de equipo, por lo que ***DocBook***[49] es una buena opción.

[49] http://www.docbook.org/.

5.1.2.1.1 Evaluación.

En esta fase, se debe formalizar la opción propuesta y definir los elementos que serán evaluados, así mismo, se deben revisar las opciones adicionales que se encuentren disponibles.

Una vez que se tengan las diversas opciones y que se hayan definido los aspectos de las soluciones a evaluar, se deberán llevar a cabo las pruebas necesarias para obtener elementos que faciliten la toma de decisiones.

El producto final de esta fase, será un reporte que contenga los siguientes elementos:

- Identificador del documento.
- Título.
- Resumen.
- Nombre y cargo de los evaluadores.
- Descripción de las actividades realizadas en la evaluación por cada evaluador.
- Contexto de las necesidades a resolver.
- Descripción general de todas las opciones evaluadas.
- Descripción de los elementos evaluados.
- Descripción de cada una de las opciones evaluadas.
- Resultados de cada una de las opciones evaluadas.
- Conclusiones de la evaluación.
- Referencias.

5.1.2.1.2 Planeación.

Una vez completada la fase de evaluación, se deberá llevar a cabo la planeación para implementar la solución seleccionada en la infraestructura de cómputo y comunicaciones con que cuente la organización. En esta fase deberán considerarse los aspectos relativos a los requerimientos de la solución, tanto para instalarla como para mantenerla funcionando.

Como producto de esta fase, deberá generarse un documento que contenga los siguientes puntos:

- Identificación del documento.
- Título.
- Resumen.
- Justificación de la selección de la solución a implantar.
- Descripción de los elementos considerados para la implantación.
- Aspectos considerados para la instalación.
- Aspectos considerados para el mantenimiento.

- Políticas y procedimientos recomendadas.
- Áreas de la organización que deberán involucrarse.
- Actividades que deberá realizar cada área de la organización.
- Plan recomendado para la implantación de la solución.
- Elementos que deben verificarse para asegurar el funcionamiento de la solución.
- Elementos de la solución que deben monitorearse.
- Referencias.

5.1.2.1.3 Implantación.

En esta fase, debe ejecutarse el plan definido en la fase anterior, para lo cual serán realizados los pasos definidos en dicho plan y tomados en cuenta los elementos a verificar para asegurar el funcionamiento de la solución.

El producto final de esta etapa, será la generación de una bitácora en la que se registren todos y cada uno de los pasos llevados a cabo, así como también los comentarios necesarios acerca de los diversos eventos que pudieron presentarse durante la instalación y operación de la solución. Esta bitácora deberá estar vigente y actualizada durante todo el tiempo de vida de la solución.

5.1.2.1.4 Monitoreo.

Una vez puesta en operación la solución, deberán realizarse los monitoreos que sean necesarios para asegurar la disponibilidad y desempeño del servicio implantado. En esta fase deben ser considerados los elementos propuestos durante la fase de planeación.

El producto final generado en esta fase, serán los registros del propio monitoreo en los cuales se almacenen los valores correspondientes a los diversos elementos monitoreados. Los datos recopilados deberán servir para la generación de estadísticas del comportamiento del servicio, así como también para realizar análisis de mejora en el desempeño.

5.1.2.1.5 Repetición del ciclo.

El ciclo descrito deberá repetirse para cada servicio que se requiera implantar o sustituir. Esta repetición debe llevarse a cabo una vez que se identifique la necesidad de incorporar un nuevo servicio al esquema de operación de la organización, así como también cuando se detecte que un servicio ya no tiene el desempeño requerido y es necesario realizar optimizaciones en la configuración para mejorar el funcionamiento de dicho servicio.

Como parte de la repetición del ciclo, los datos obtenidos de la fase de monitoreo, deben ser incorporados a la nueva fase de evaluación.

5.1.2.2 Etapas necesarias para la implantación.

El modelo propuesto hasta el momento, aplica sólo para la etapa de definición de servicios y no considera la instalación de dichos servicios en múltiples servidores remotos, por lo que deberán incorporarse algunos aspectos de índole práctico con el objeto de poder realizar la implantación de dicho modelo de una manera más cómoda. A grandes rasgos, se trata de una extensión del modelo, con el objeto de cubrir una mayor cantidad de equipos.

En primer término y como parte de la extensión del modelo se define que, con el fin de mantener íntegra la implantación de la solución propuesta, las políticas deben especificarse en un solo equipo central, desde el cual todos los demás equipos tomen dichas políticas y las incorporen en su funcionamiento, con lo que se logrará homogenizar y asegurar la aplicación de las políticas definidas a todos y cada uno de los equipos administrados, a la vez que se mantiene consistencia en la aplicación de las políticas.

Por otra parte, la actualización de software se realizará de manera centralizada a todos los demás equipos utilizando la infraestructura de comunicaciones digitales con que cuente la organización. Adicional a la replicación de las políticas y la distribución del software, deberán considerarse los elementos necesarios para realizar el monitoreo de los diversos parámetros definidos y que sean importantes para la adecuada operación del sistema.

Combinando estos elementos, se ilustra el modelo propuesto de acuerdo a la Figura 5-2. Como se puede observar, en primer lugar se realiza una definición de los servicios, políticas, procedimientos y tecnología utilizada, se llevan a cabo evaluaciones de cada uno de los aspectos, se definen planes, se ejecutan y monitorean para revisar que todo este funcionando correctamente. En segundo lugar y una vez que se comprueba que todo está operando de manera adecuada, se lleva a cabo la liberación de los ajustes realizados y se pone a disposición para que se repliquen a todos los equipos involucrados, los cuales aplicarán de manera local las nuevas definiciones y las adaptarán de acuerdo a la configuración local que cada equipo tenga.

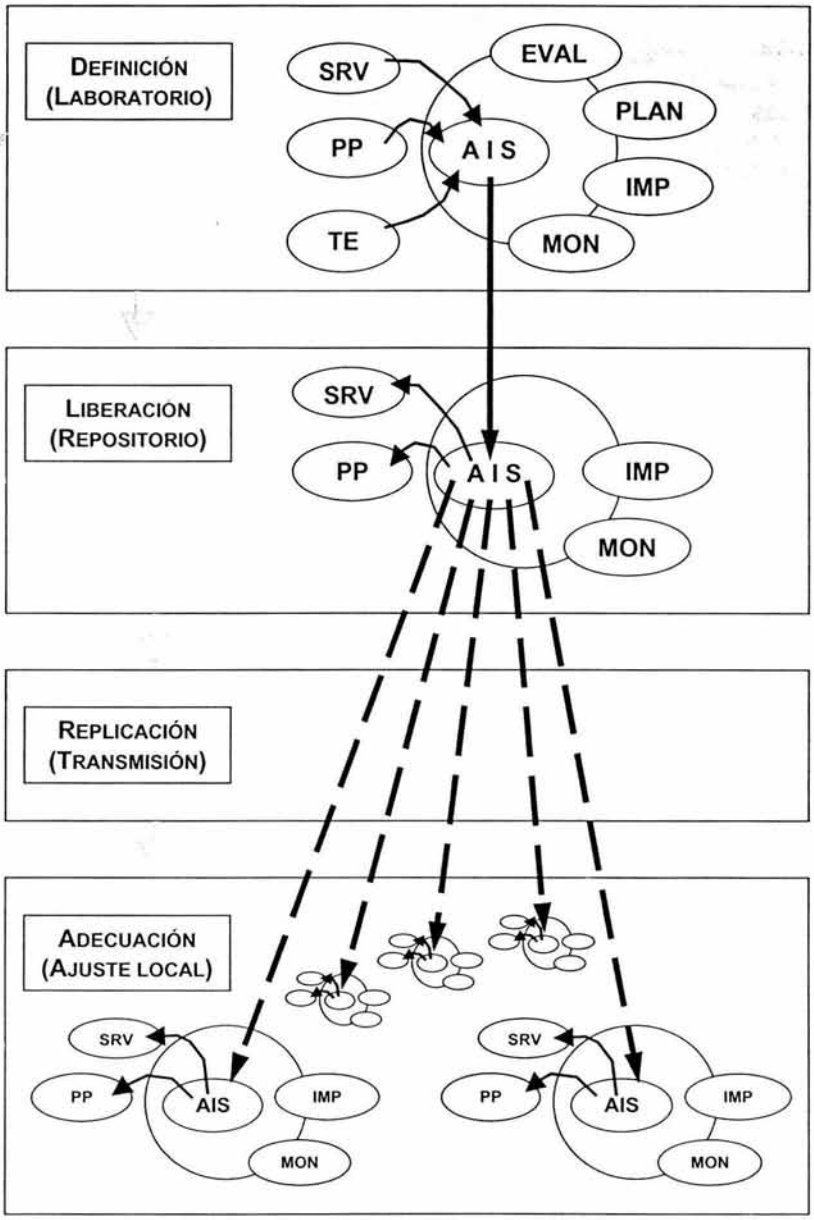


Figura 5-2. Modelo general de la solución propuesta.

A continuación se describe cada etapa a mayor detalle.

5.1.2.2.1 Definición.

Esta etapa debe ser entendida como un "laboratorio" donde se realizarán las pruebas en los ciclos necesarios para llegar a un punto donde los servicios a liberar se encuentren lo suficientemente probados y afinados. Así como también deben definirse los aspectos correspondientes a las políticas y procedimientos que regulan los servicios a implantar. Además, en esta etapa, se deben completar las pruebas necesarias a la tecnología que se pretenda incorporar a la infraestructura existente.

Los detalles de esta etapa, fueron descritas en el punto 5.1.2.1. Cabe mencionar que esta es la etapa inicial del modelo general de la solución propuesta, mostrado en la Figura 5-2 , el cual es recurrente en su conjunto.

5.1.2.2.2 Liberación.

Una vez completada la etapa de definición se procederá a la liberación de los servicios informáticos y las políticas y procedimientos evaluados, para que funcionen en la infraestructura tecnológica de la organización.

Con el objeto de mantener coherencia en la implantación de los servicios, la liberación deberá llevarse a cabo sobre un equipo que servirá como repositorio[50] para que los demás equipos productivos obtengan los servicios liberados según les corresponda, ya que los servicios liberados no necesariamente aplicarán para todos los equipos de la organización.

El equipo que funciona como repositorio es también un equipo productivo, por lo que los servicios liberados pueden llegar a afectarle directamente. Cabe señalar que para este caso, la actualización del repositorio está controlada por la etapa de definición, es decir el equipo repositorio es únicamente receptor de actualizaciones.

Los equipos productivos restantes solicitan al repositorio únicamente los servicios liberados que les corresponden, según la plataforma tecnológica a la que pertenezca cada uno. Para este caso, el repositorio no envía las actualizaciones, sino que son los propios equipos productivos los que "solicitan" al repositorio las actualizaciones y son los responsables de auto-mantenerse actualizados.

Como puede observarse en la Figura 5-2, para la etapa de liberación se eliminan algunas fases y se invierte el flujo de transferencia de información relativa a la Administración Integrada de la Seguridad (AIS). En esta etapa, la AIS ya no se modifica (dado que ya fue definida) y sólo se aplican las fases correspondientes a

[50] Cabe señalar que para que la solución sea lo suficientemente escalable, pueden utilizarse más de un equipo para funcionar como repositorio, aunque genéricamente se les identifique simplemente como "repositorio".

la implantación y monitoreo de los servicios y de las políticas y procedimientos, de acuerdo a lo que se haya definido en la etapa de definición.

En esta etapa se incorporan los cambios relacionados de manera particular al repositorio, en los aspectos de servicios, políticas y procedimientos, así como también se genera información que retroalimenta la etapa de definición, los cuales serán utilizados para el siguiente ciclo.

5.1.2.2.3 Replicación.

La etapa de replicación consiste básicamente en la transmisión de las actualizaciones de una manera segura entre el repositorio y los servidores productivos, utilizando la infraestructura de comunicaciones digitales con que cuente la organización. Como ya se mencionó anteriormente, los servidores productivos son los responsables de solicitar las actualizaciones al repositorio, por lo que el esquema de autenticación utilizado deberá ser capaz de atender a todos y cada uno de los equipos involucrados en la implantación de la solución.

La sincronización de los relojes de los servidores es otro aspecto que debe tomarse muy en cuenta para evitar que en determinado momento los servidores productivos saturen al repositorio. Esto se logra mediante la incorporación de esperas aleatorias en los programas responsables de obtener actualizaciones del repositorio.

5.1.2.2.4 Ajustes a nivel local.

Uno de los aspectos más delicados en el modelo de la solución propuesto es la aplicación de los cambios a nivel local en los servidores productivos, es decir, incorporar los servicios, políticas y procedimientos en el funcionamiento propio de los equipos a los cuales están destinados.

Para llevar a cabo esta etapa, es necesario haber completado de manera satisfactoria las etapas anteriores y contar con mecanismos que limiten, dentro de lo posible, la generación de errores. Algunos de estos mecanismos (que deben ser verificados antes de proceder con la aplicación de cambios), son la verificación de la integridad de los archivos transmitidos mediante algoritmos hash, verificación de variables de ambiente del sistema operativo y verificación de dependencias con otros programas.

Al completarse la actualización, deben llevarse a cabo las pruebas de verificación necesarias para garantizar la operación correcta de los cambios realizados. Adicionalmente se debe generar información que sirva para retroalimentar a la etapa de definición.

5.2 Análisis y diseño.

Es necesario identificar, acotar y delimitar adecuadamente el problema para poder estar en posibilidad de lograr los objetivos trazados. Parte del análisis que se realizará en esta parte obedece a que existen diversos factores prácticos y de operación que deben ser considerados para tener mayores posibilidades de éxito al momento de la implantación de la solución propuesta.

Así mismo, será necesario estructurar el modelo propuesto en un esquema que pueda ser programado y llevado a la práctica, identificando para esto las estructuras de datos que servirán como soporte, así como también los procesos que deberán funcionar como infraestructura de la solución.

5.2.1 Delimitar factores importantes.

Es primordial identificar los elementos críticos a monitorear en cada sistema y definir las políticas que deben cumplirse en cada tipo de equipo y sistema operativo, es decir, organizar los recursos con que se cuentan y los cuales se desean administrar.

Adicionalmente, es necesario no perder de vista y prever los elementos que como parte de la operación cotidiana de los sistemas de cómputo, podrían afectar o ser aprovechados para el mejor desempeño de la solución.

5.2.1.1 Consideraciones para la operación de la solución.

Algo importante de considerar es que para cumplir con el objetivo de que la solución propuesta sea escalable, se requiere que dicha solución opere de manera desatendida en los equipos que se desea monitorear, pero a la vez como ya se definió, deben mantenerse y actualizarse de manera centralizada las políticas que se requieren cumplir en cada sistema, para asegurar congruencia en la aplicación de dichas políticas.

Para esto, los equipos que deben ser administrados contactarán a intervalos definidos al servidor donde se almacena las políticas que deben implementar de manera local. Adicionalmente se debe considerar la generación de reportes periódicos sobre el estado que guardan los equipos tanto en su funcionamiento como en su configuración. También habrá que considerar que el equipo dedicado a servir las políticas debe ser capaz de soportar las peticiones de todos los demás equipos, o en su caso, la solución deberá permitir que existan diversos servidores de políticas.

Por otra parte, la actualización de software deberá ser también a partir de un servidor de actualizaciones central, del cual los equipos tomarán únicamente las actualizaciones que le correspondan de acuerdo a su configuración local. Además, será necesario un esquema de generación de reportes relativos a la actividad

normal de los equipos así como también la emisión de advertencias y alarmas para condiciones de actividad sospechosa.

La solución deberá estar definida y construida sobre conceptos de seguridad informática y la implementación de la misma deberá será bajo revisiones y verificaciones estrictas del cumplimiento de dichas definiciones.

5.2.2 Estructura de datos.

Se requiere contar con un mínimo de estructuras de datos para poder organizar la información recolectada y poder llevar a cabo análisis de dicha información. Es conveniente que la implantación de las estructuras de datos sea generada a través de un manejador de bases de datos independiente al lenguaje de programación utilizado, con el objeto de dividir en módulos la programación y mantener separación de funciones en la solución, con lo cual se tendrá flexibilidad en la implantación.

5.2.2.1 Tablas principales.

En esta sección se identificarán las tablas más representativas utilizadas en la solución propuesta y que dan una idea clara de la implantación. Existen tablas adicionales que sirven como apoyo a las principales, y aunque son importantes, sólo sirven para organizar de mejor manera la información por lo que conservan un papel secundario en la conformación de la solución.

A continuación se comenta cada una de las tablas principales de la solución propuesta.

5.2.2.1.1 Equipo.

Se podría decir que esta tabla es la principal de la solución, ya que ocupa un papel protagónico dentro del esquema de la solución. Esto es debido a que la solución está sustentada en el hecho de que la organización de los equipos determina en mayor grado las labores administrativas y de seguridad informática requeridas en diversas situaciones.

Los campos considerados para esta tabla, se describen a continuación en la Tabla 1-1.

CAMPO	*DESCRIPCIÓN*
Id_equipo	Identificador del equipo.
Id_ubicación	Identificador de la ubicación del equipo (referencia a tabla ubicación).
Id_soporte_técnico	Identificador del contacto para soporte técnico (referencia a tabla soporte técnico).
Nombre	Nombre del equipo (asignado por el administrador).
Descripción	Descripción del equipo.

Marca	Marca del fabricante del equipo.
Modelo	Modelo del equipo.
Procesador_cantidad	Cantidad de procesadores que tiene configurados el equipo.
Procesador_marca	Marca del procesador.
Procesador_modelo	Modelo del procesador.
Procesador_velocidad	Velocidad d operación del procesador.
Memoria	Cantidad de emoria configurada.
Número_serie	Número de serie asignado.
Número_inventario	Número de inventario asignado.
Notas_informativas	Notas para intercambio de información entre administradores. Este campo debe ser revisado antes de tomar alguna acción en relación con el equipo.
Sistema_operativo	Sistema operativo instalado en el equipo.
Clasificación	Tipo de servidor. Este es un campo configurable de acuerdo a las políticas de administración definidas en la organización.
Última_verifiación	Fecha y hora de la verificación más reciente llevada a cabo.
Último_cambio	Fecha y hora del cambio más reciente realizado.
Anomalía	En caso de presentarse algún problema en la verificación, se debe reportar en este campo.
Actualizaciones_pendientes.	Lista de actualizaciones de software pendientes por aplicarse.
Vulnerable.	Bandera para indicar vulnerabilidades detectadas en el equipo. Este campo deberá permanecer nulo, excepto cuando se presenten vulnerabilidades, las cuales deben ser descritas en este campo.
Actividad_sospechosa	Bandera para indicar actividades sospechosas detectadas en el equipo. Este campo deberá permanecer nulo, excepto cuando se presenten actividades sospechosas, las cuales deben ser descritas en este campo.
Registrar_fecha	Fecha y hora de la afectación al registro actual.
Estatus	Campo configurable de acuerdo a las políticas de administración internas definidas en la organización. Valores posibles: activo, inactivo, en línea, fuera de línea, y combinaciones diversas.
Notas	Notas generales relativas al equipo.

Tabla 5-1. Descripción de los campos de la tabla *equipo*.

5.2.2.1.2 Ubicación.

Esta tabla sirve como referencia para tener la localización precisa de los equipos que se administran y de esta manera poder ubicarlos de una manera fácil y rápida. Un equipo sólo podrá estar en una sola ubicación en un momento dado.

Los campos considerados para esta tabla, se describen a continuación en la Tabla 5-2.

CAMPO	*DESCRIPCIÓN*
Id_ubicación	Identificador de la ubicación.
Nombre	Nombre asignado a la ubicación.
Descripción general	Breve descripción de la ubicación.
Calle	Nombre de la calle.
Colonia	Nombre de la colonia.
Codigo_postal	Código postal.
Ciudad	Nombre de la ciudad.
Estado	Nombre del estado.
País	Nombre del país.
Ubicación específica	Identificador del edificio, número de piso, nombre del área y referencias adicionales.
Estatus	Campo configurable de acuerdo a las políticas internas definidas en la organización. Valores posibles: activo, inactivo, y combinaciones diversas.
Notas	Notas generales relativas a la ubicación, en este campo se puede indicar entre qué calles se encuentra o las indicaciones para llegar.

Tabla 5-2. Descripción de los campos de la tabla *ubicación*.

5.2.2.1.3 Administrador.

Cada equipo estará bajo la responsabilidad de uno o más administradores, de los cuales se deberán tener los datos más relevantes para notificarles la detección de eventos críticos o la necesidad de atención a algún equipo. Adicionalmente esta es una tabla autorreferenciada para indicar la organización jerárquica entre personas.

Los campos considerados para esta tabla, se describen a continuación en la Tabla 5-3.

CAMPO	*DESCRIPCIÓN*
Id_administrador	Identificador del administrador.
Id_ubicación	Identificador de la ubicación del administrador.
Título	Corresponde al título que tiene el administrador. Valores posibles: Lic., Ing., etc.
Apellido_paterno	Apellido paterno de la persona.
Apellido_materno	Apellido materno de la persona.
Nombre	Nombre de la persona.
Puesto	Nombre del puesto que ocupa la persona dentro de la organización.
Area_especifica	Nombre del área a la que pertenece.
Adscripción	Nombre de la adscripción a la que pertenece.
Organización	Nombre de la organización.
Jefe_inmediato	Nombre del jefe inmediato. (Esta tabla es autoreferenciada, por

	lo que se pueden estructurar las relaciones jefe-subordinado entre las diversas personas, con el objeto de definir escalaciones de problemas o notificaciones).
Asistente	Nombre del asistente, secretaria o subordinado al que puede contactarse en caso de no encontrarse disponible.
Estatus	Campo configurable de acuerdo a las políticas internas definidas en la organización. Valores posibles: activo, inactivo, y combinaciones diversas.
Notas	Notas generales relativas al administrador.

Tabla 5-3. Descripción de los campos de la tabla *administrador*.

5.2.2.1.4 Servicio.

Un equipo podrá ofrecer diversos servicios informáticos, dependiendo la manera en que se hayan asignado las tareas a cada equipo, por lo que es conveniente tener un relación que permita identificar el número y tipo de servicios ofrecidos en cada uno de los equipos.

Los campos considerados para esta tabla, se describen a continuación en la Tabla 5-4.

CAMPO	*DESCRIPCIÓN*
Id_servicio	Identificador del sevicio.
Id_administrador	Identificador del administrador específico encargado del servicio.
Id_equipo	Identificador del equipo en el cual funciona el servicio.
Id_ubicación	Identificador de la ubición fisica.
Id_soporte_técnico	Identificador del soporte técnico asignado a dicho servicio (en caso de existir).
Nombre	Nombre del servicio. Se sugiere utilizar un esquema <nombre_equipo:servicio:puerto>.
Descripción	Descripción general del servicio.
Puerto	Puerto configurado en el cual funciona el servicio.
Protocolo	Protocolo utilizado por el servicio.
Programa	Nombre del programa ejecutable que arranca el servicio. Se recomienda incluir en este campo las indicaciones para iniciar y detener el serivicio.
Versión	Versión del programa que arranca el servicio.
Registrar_fecha	Fecha y hora de la última modificación del registro.
Estatus	Campo configurable de acuerdo a las políticas de administración internas definidas en la organización. Valores posibles: activo, inactivo, en línea, fuera de línea, y combinaciones diversas.
Notas	Notas generales relativas al servicio.

Tabla 5-4. Descripción de los campos de la tabla *servicio*.

5.2.2.1.5 *Política.*

La definición de políticas, como ya se mencionó, es un elemento primordial en la puesta en operación de una solución de seguridad en cualquier ambiente de cómputo. Para este caso, las políticas estarán almacenadas en una tabla y cada una se asignará, dependiendo del tipo de política, al equipo, sistema operativo o servicio de que se trate.

Los campos considerados para esta tabla, se describen a continuación en la Tabla 5-5.

CAMPO	*DESCRIPCIÓN*
Id_politica	Identificador de la política definida.
Nombre	Nombre asignado a la política.
Tipo	Clasificación de la política.
Plataforma	Nombres de las plataformas de cómputo a las que aplica la política.
Descripción	Descripción general de la política.
Definición	Definición y alcances de la política.
Registrar_fecha	Fecha y hora de la última actualización del registro.
Estatus	Campo configurable de acuerdo a las políticas internas definidas en la organización. Valores posibles: activa, inactiva, y combinaciones diversas.
Notas	Notas generales relativas a la política.

Tabla 5-5. Descripción de los campos de la tabla *política*.

5.2.2.1.6 *Bitácora.*

El sistema de monitoreo integrado en los equipos, registrará en esta tabla las actividades críticas o de alerta que sean consideradas importantes de registrar.

Los campos considerados para esta tabla, se describen a continuación en la Tabla 5-6.

CAMPO	*DESCRIPCIÓN*
Id_bitácora	Identificador del registro de bitácora.
Id_equipo	Identificador del equipo.
Id_ubicación	Identificador de la ubicación del equipo.
Id_soporte_técnico	Identificador del soporte técnico del equipo.
Id_servicio	Identificador del servicio que generó el registro de bitácora.
Tipo_evento	Clasificación de eventos. Valores posibles: emergencia, crítico, error, advertencia, informativo.
Mensaje	Mensaje emitido directamente por la aplicación que generó el registro en la bitácora.
Registrar_fecha	Fecha y hora de la última actualización del registro.
Estatus	Campo configurable de acuerdo a las políticas internas definidas en la organización. Valores posibles: activo, inactivo, sin_revisar, revisado, corregido, entre otros.

Notas	Notas generales relativas a registro de la bitácora. Puede ser utilizado para documentar la forma en que se resolvió el problema.

Tabla 5-6. Descripción de los campos de la tabla *bitácora*.

5.2.2.1.7 Correlación.

Esta tabla permitirá obtener un mapa de la relación que guardan todos los elementos involucrados y que permiten asegurar la disponibilidad de un servicio informático, por lo que resulta muy útil para detectar problemas y emitir una solución rápidamente. Inclusive, puede apoyar en la prevención de problemas antes de aplicar algún cambio.

El motivo de esta tabla es debido a que el funcionamiento de un servicio depende no solamente del equipo en el cual funciona, sino también de todos los elementos involucrados en la infraestructura de cómputo y comunicaciones de la organización.

Los campos considerados para esta tabla, se describen a continuación en la Tabla 5-7.

CAMPO	*DESCRIPCIÓN*
Id_correlación	Identificador de la correlación.
Id_equipo	Identificador del equipo.
Id_ubicación	Identificador de la ubicación del equipo.
Id_soporte_técnico	Identificador del soporte técnico.
Id_servicio	Identificador del servicio.
Id_política	Identificador de la política.
Id_inactividad	Identificador de inactividad.
Id_vulnerabilidad	Identificador de vulnerabilidades.
Nombre	Nombre de la correlación.
Tipo	Clasificación de la vulnerabilidad.
Resumen	Esquema general de la correlacion realizada.
Descripción	Descrición general de la correlación. Se deben incluir todos los equipos involucrados en el funcionamiento de un servicio en particular.
Registrar_fecha	Fecha y hora de la última actualización del registro.
Estatus	Campo configurable de acuerdo a las políticas internas definidas en la organización. Valores posibles: activo, inactivo, entre otros.
Notas	Notas generales relativas a la correlación.

Tabla 5-7. Descripción de los campos de la tabla *correlación*.

5.2.2.2 Diagrama entidad-relación.

En la Figura 5-3, se muestra el diagrama entidad-relación sugerido para ser utilizado en la solución propuesta.

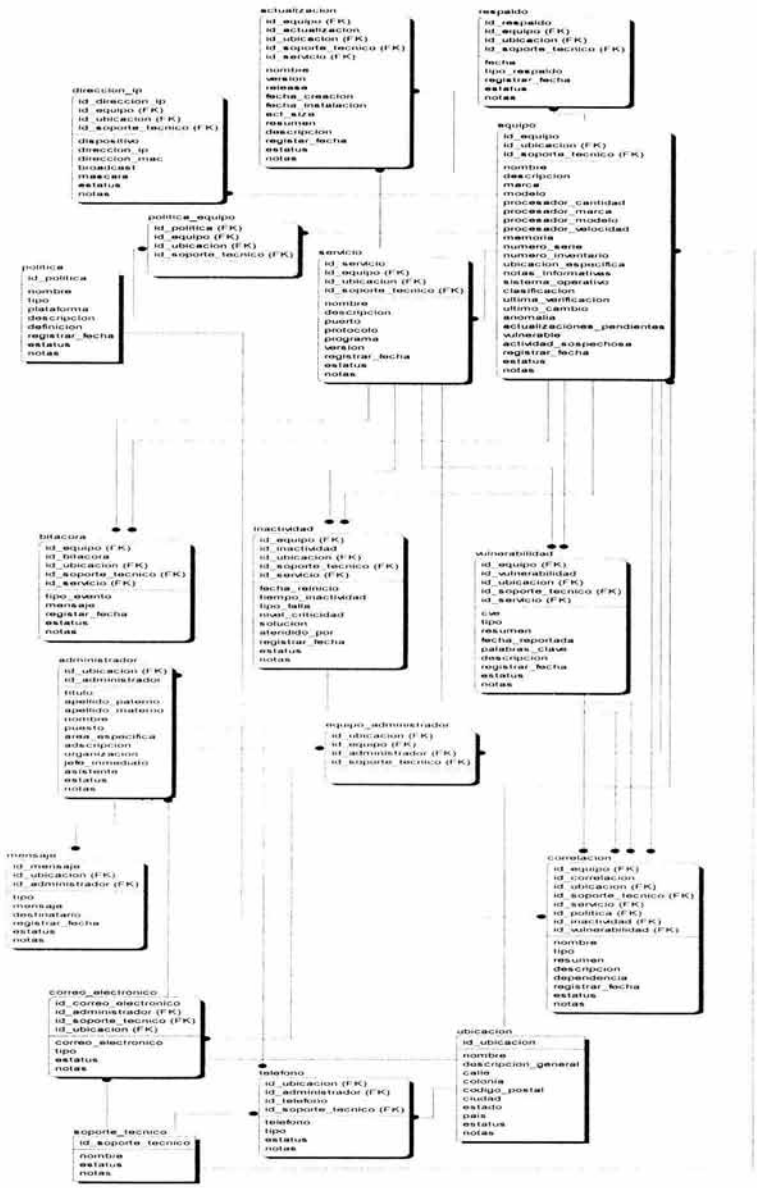


Figura 5-3. Diagrama entidad relación.

A partir de este modelo entidad-relación se derivan los principales módulos de programa que deben ser considerados en la solución, y que deben estar operando en concordancia para alimentar a la base de datos. En el **ANEXO A** se lista el código SQL rquerido para la creación de la base de datos en un manejador de bases de datos relacional.

5.2.3 Procesos necesarios.

En virtud de los requerimientos expuestos, será necesario desarrollar e integrar herramientas con las que se puedan obtener los datos necesarios con los cuales se puedan realizar las decisiones precisas para poder ejecutar las acciones adecuadas.

Cabe mencionar que las herramientas enunciadas más adelante en esta sección, fueron descritas en el Capítulo 1.

5.2.3.1 Módulos de programa de la solución.

Como ya se ha comentado, el modelo entidad-relación planteado proporciona los elementos mínimos para definir los módulos requeridos que llevarán a cabo la carga de información a la base de datos.

En primer término, cabe señalar que en virtud de que no es posible obtener de manera automática muchas de las actualizaciones a la información de la base de datos deberán realizarse manualmente, por lo que será necesario que los propios administradores realicen la actualización de información. En este caso se encuentran las siguientes tablas:

- Administrador
- Ubicación
- Soporte técnico
- Política
- Correlación.
- Teléfono
- Correo_electrónico

En segundo lugar, es de mencionar que la otra gran parte de la información será posible recolectarla de manera automática, mediante procesos que son ejecutados tanto en los servidores a nivel local, así como también en equipos centralizados. En este caso, los módulos pueden organizarse como a continuación se enlista.

5.2.3.1.1 Obtención de datos.

Este módulo es el encargado de analizar las características de los equipos y obtener los datos necesarios para ingresar información a las siguientes tablas:

- Equipo
- Dirección_IP
- Servicio

El proceso que realiza este módulo, puede ser ejecutado desde un equipo central que se conecta a los otros equipos o de manera independiente en cada equipo a través de un agente de software. Idealmente, es conveniente contar con los esquemas ya que ambos se complementan. El caso del equipo central es más fácil de mantener, pero el esquema de agentes de software puede resultar en un esquema más sofisticado, que permita la obtención de información más específica.

En la mayoría de los equipos es factible obtener los valores de las condiciones de temperatura y operación de diversos circuitos, con lo cual es posible determinar problemas e inclusive, prevenirlos. La información se obtiene a través de utilerías del sistema operativo o mediante programas específicos que tengan acceso a las estructuras de datos del kernel del sistema operativo. Por ejemplo, para obtener la información relativa al CPU con que cuenta el equipo, basta con hacer un script que adquiera dichos datos del archivo /proc/cpuinfo.

En la Figura 5-4 se puede observar el esquema de operación de este módulo. El cuadro "Recolector" representa a un programa encargado de obtener los datos y almacenarlos de manera local, así como también pasar dichos datos a otro programa llamado "Transmisor" que se encargará de enviarlos al equipo central para su almacenamiento y consulta.

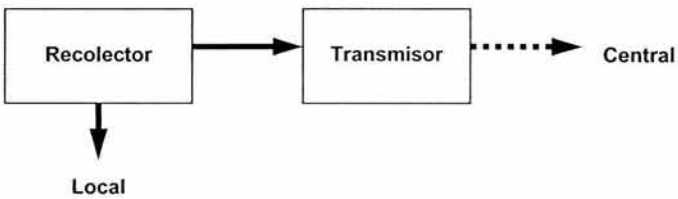


Figura 5-4. Esquema de operación del módulo de obtención de datos.

5.2.3.1.2 Monitor de bitácoras.

La información generada por los diversos servicios que funcionan en los diversos equipos de cómputo, y que es grabada en bitácoras dentro del sistema, es recolectada, organizada y analizada para emitir reportes de eventos que serán almacenados en la tabla:

- Bitácora

El módulo monitor de bitácoras funciona a través de un agente de software instalado en todos y cada uno de los equipos administrados. Este agente se encarga de realizar el monitoreo e informar en todo momento al equipo central el estado que guarda cada sistema.

Es común que en los sistemas de multiprocesamiento de datos, coexista una gran diversidad de servicios informáticos, cada uno de los cuales genera su propia bitácora de eventos a través de la cual es posible conocer el estado y comportamiento de cada uno de dichos servicios informáticos.

Este módulo reporta diversas condiciones de los equipos incluyendo la integridad de archivos (analizando las bitácoras del programa ***AIDE***), las condiciones de procesamiento, memoria y disco (mediante utilerías como ***sar*** y ***vmstat***), los mensajes que emite tanto el sistema (utilizando el subsistema ***syslog***) como los diversos servicios configurados en cada equipo (a través de la información reportada por ***nmap***), intentos de explotación de vulnerabilidades (empleando ***nessus***) y detección de intrusos (revisando bitácoras de ***snort***).

Para apoyar en las labores del análisis de cada una de las bitácoras de los programas de aplicación, existen herramientas de libre distribución que realizan estas tareas. Entre las herramientas más populares se encuentran ***Logwatch***, así como también ***Logcheck***.

En la figura Figura 5-5 se puede observar el funcionamiento de este módulo. Cada subsistema tiene un monitor de bitácora asignado, el cual almacena localmente sus datos, a la vez que un procesos "Analizador" correlaciona y obtiene información útil que en caso de requerirse se envía al equipo central mediante el proceso "Transmisor".

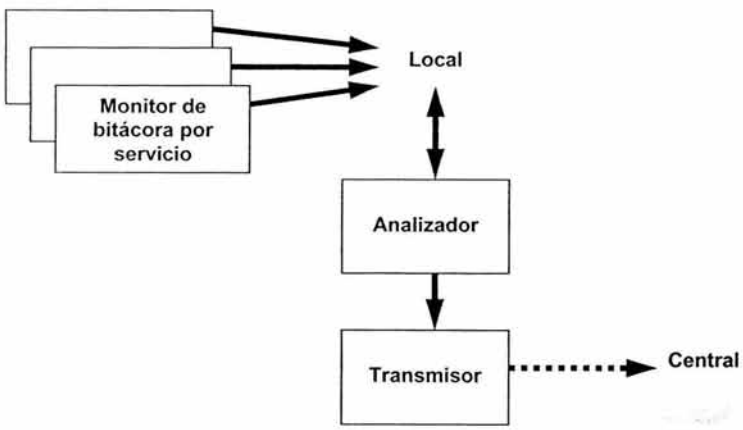


Figura 5-5. Esquema de operación del módulo monitor de bitácoras.

5.2.3.1.3 Monitor de disponibilidad.

El monitoreo de la disponibilidad de los servicios es una tarea que debe realizarse de manera centralizada y con apoyo de las correlaciones definidas para los diversos servicios y equipos.

La revisión de la disponibilidad será realizada a intervalos de tiempo definidos y será llevada a cabo mediante programas que analicen las distintas condiciones que deben cumplirse para que un servicio dado pueda ser considerado como disponible. La información obtenida de este módulo servirá para ingresar datos en la tabla:

- Inactividad

Cabe señalar que los tipos de alerta y el nivel de criticidad de cada uno, deberán estar definidos de acuerdo a los requerimientos de la organización.

Con el objeto de que la determinación del estatus sea más acertada se emplearán una combinación de herramientas para obtener esta información, algunas de las utilerías principales serán ***ping***, ***nmap***, ***expect*** y clientes de aplicación programados específicamente para verificar alguna condición en particular.

En la Figura 5-6 se puede observar el esquema de operación de este módulo. Existe un monitor por cada tipo de servicio, el cual se conecta a intervalos regulares de tiempo al servicio remoto para verificar su funcionamiento. Una vez obtenido el estatus, se almacena en el repositorio central.

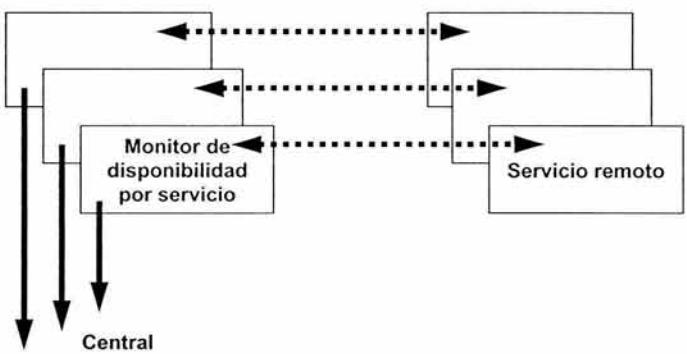


Figura 5-6. Esquema de operación del módulo monitor de disponibilidad.

5.2.3.1.4 Analizador de vulnerabilidades.

Este módulo será el encargado de ejecutar periódicamente el análisis de vulnerabilidades en los distintos equipos administrados. Esta revisión deberá llevarse a cabo de manera central y desde distintos equipos, con el objeto de tomar en cuenta distintas condiciones para la evaluación.

La tabla a la que afecta este módulo es:

- Vulnerabilidad

En esta tabla se registrarán las vulnerabilidades descubiertas y a partir de esto, se generarán las alarmas y notificaciones pertinentes.

Para obtener la información requerida, este módulo emplea las herramientas ***nmap*** y ***nessus***, así como también scripts programados para llevar a cabo actualizaciones a las definiciones de las vulnerabilidades descubiertas o reportadas.

En la Figura 5-7 se puede observar el esquema de operación de este módulo. El analizador de vulnerabilidades se conecta a cada uno de los servicios remotos configurados y el resultado del análisis es almacenado a nivel central.

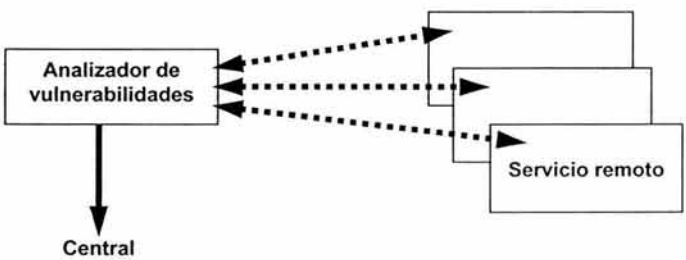


Figura 5-7. Esquema de operación del módulo analizador de vulnerabilidades.

5.2.3.1.5 Administrador de actualizaciones de software.

Uno de los módulos más complejos es este, ya que será el encargado de asegurar que los equipos se encuentran funcionando con las versiones de software correctas. Este módulo afecta la tabla:

- Actualización

El funcionamiento de este módulo es una combinación de programas a nivel central que se encargan de diversas tareas, entre las principales, servir como un

repositorio de actualizaciones de software disponibles para instalarse en los equipos administrados, generar reportes del estatus de la instalación de dichas versiones de software en cada uno de los equipos, determinar aquellos equipos en los que haga falta instalar actualizaciones de software, y en su caso, emitir alertas y notificaciones en consecuencia.

La parte más importante de este módulo se encuentra distribuida en forma de agentes de software en cada uno de los equipos administrados, los cuales "revisan" y "obtienen" las actualizaciones más recientes que se encuentren disponibles en el repositorio central, ya que el esquema de actualización de software está basado en un esquema en el cual cada equipo es responsable mantenerse actualizado, mediante la instalación de las actualizaciones que le correspondan. Con esto se logra mayor escalabilidad de la solución, ya que gran parte de la carga de actualización se delega del servidor central a los equipos distribuidos.

Una parte fundamental en la administración de actualizaciones es llevar un adecuado control de las versiones en la distribución de software, y por lo tanto contar con un sistema de control de versiones que permita identificar y sobre todo asegurar que una versión de software corresponde a los cambios que se supone fueron realizados a los programas que la componen.

En este rubro, uno de los programas más utilizados es ***CVS*** (Concurrent Versioning System), el cual es ampliamente utilizado por la comunidad de desarrolladores de software libre y ha probado su estabilidad y madurez como sistema administrador de versiones.

Por otra parte, la distribución y transferencia de archivos son actividades que demandan gran confiabilidad, a la vez de que se requiere la automatización de tales actividades para poder estar en posibilidades de brindar un servicio adecuado en tiempo y forma.

Para realizar la distribución de archivos existen diversas herramientas especializadas como ***CVSup***, ***rsync*** e incluso ***scp*** en combinación con otros programas. Para la automatización de estas actividades se requiere orquestar estas herramientas con otras como ***cron***, ***at***, ***sh***, y ***expect***.

Como parte de la distribución del software, surge adicionalmente el problema de mantener control en la instalación, actualización y desinstalación de los programas que corresponden a un cierto sistema. Este problema se deriva principalmente a que los programas de aplicación comúnmente se componen por diversos archivos ejecutables, librerías estáticas o dinámicas y archivos de configuración, entre otros, y tener control de todos y cada uno se vuelve una tarea compleja.

Para resolver esto, se han desarrollado diversos sistemas que "empaquetan" el software de aplicación y llevan el control de todos los archivos instalados correspondientes a cada paquete. Existen tanto soluciones comerciales como

también de libre distribución, correspondientes a este último rubro, se encuentran sistemas como ***RPM***, ***dpkg*** y ***OpenPKG***, principalmente.

En la Figura 5-8 se puede observar el funcionamiento de este módulo. Como se puede ver, el módulo se encuentra dividido en dos partes, por un lado se encuentra lo correspondiente a la generación, empaquetamiento y puesta a disposición de la actualización de software a nivel central. Por otro lado se encuentran los procesos que se llevan a cabo en cada uno de los equipos remotos, los cuales constan de la verificación, obtención, instalación y notificación de las nuevas versiones de software. Al final de cada una de las etapas se registra la información relevante a nivel central.

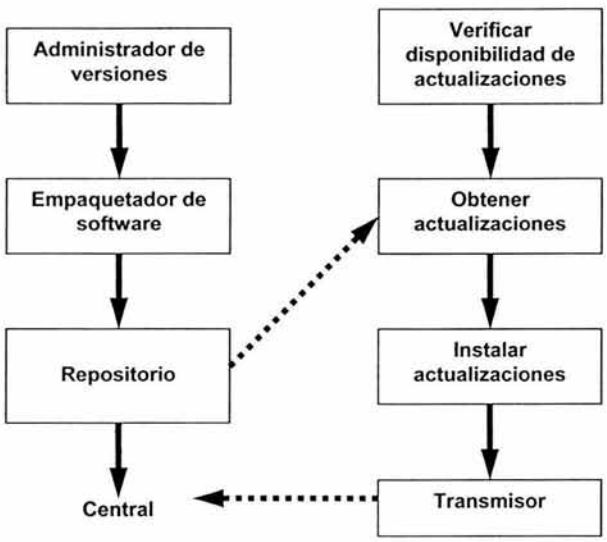


Figura 5-8. Esquema de operación del módulo administrador de actualizaciones de software.

5.2.3.2 Procesos de soporte.

Como parte básica de la solución se requiere contar con proceso de soporte que permitan la coordinación de tareas y control de las operaciones que como parte de la solución se desarrollen. Para esto, será necesario contar con un mínimo grupo de procesos que proporcionen principalmente, resolución de nombres de dominio (DNS), autenticación centralizada, sincronización de relojes de todos y cada uno de los equipos administrados, monitoreo y administración remota de los equipos, así como también la automatización de diversas tareas.

A continuación se describe cada uno de estos rubros.

5.2.3.2.1 Resolución de nombres de dominio.

Otro punto importante a cuidar es contar con un sistema de resolución de nombres apropiado y configurado adecuadamente, el cual permita la identificación de los servidores por medio de nombres en lugar de IPs, de una manera confiable, lo que involucra poner mucha atención en la configuración de este servicio.

Para esto existen diversos programas que implementan el protocolo DNS, entre los más populares se encuentran ***BIND***, ***NSD*** y ***djbdns***. Para los cuales se deben tomar en cuenta las recomendaciones de seguridad aplicables. En este rubro, BIND sigue siendo el más popular como servidor DNS.

5.2.3.2.2 Autenticación.

Preferentemente se debe contar con un sistema de autenticación que sea manejado a partir de un repositorio central, que permita asegurar una correcta autenticación en todos y cada uno de los equipos que estén funcionando dentro del esquema propuesto.

En este rubro, resulta muy útil desarrollar un esquema de autenticación basado en LDAP y que permita la difusión de cambios desde un repositorio central a los equipos mediante replicaciones (para los casos en que la red no se encuentre disponible todo el tiempo). Una implementación muy popular de LDAP es ***OpenLDAP***, el cual es de libre distribución.

5.2.3.2.3 Sincronización de tiempo.

La única manera de realizar correlaciones de eventos en diferentes equipos de manera confiable, es que todos los servidores administrados e integrados en una solución de monitoreo, sincronicen sus relojes con un servidor de tiempo. Para implantar esto existe el protocolo NTP, con el cual se pueden mantener sincronizados los relojes de los equipos con servidores de tiempo ubicados en diversas partes alrededor del mundo.

Los servidores de tiempo obtienen la hora de computadoras conectadas a relojes atómicos de alta precisión y el protocolo NTP contempla los retrasos que pueden presentarse en la transmisión de la información por la red, por lo que la hora asignada a los servidores prácticamente es exacta.

5.2.3.2.4 Monitoreo y administración remota.

Asegurar que un sistema remoto se comporte de cierta manera es una tarea compleja que involucra el manejo de una gran cantidad de variables dentro del sistema, así como validar muchas y muy diversas circunstancias.

Hasta ahora, pocas han sido los desarrollos que aborden de manera adecuada esta problemática. Una de las herramientas que ha tenido éxito en este tipo de tareas es ***cfengine***. Esta herramienta incorpora conceptos de inteligencia artificial para implementar un agente autónomo que en base a las políticas de administración definidas, se asegura que el sistema esté cumpliendo con dichas políticas.

Por otra parte, la conexión remota para realizar ajustes o revisar el estado de cada servidor es una tarea rutinaria que deben realizar los administradores de sistemas, ya sea de manera manual o automatizada, para asegurarse que los equipos están funcionando correctamente.

Una herramienta para llevar a cabo estas actividades de manera más confiable es Secure Shell (SSH), la cual es una forma segura para conectarse a servidores remotos, ya que implementa algoritmos de criptografía de llave pública para cifrar la información que se transmite entre los equipos enlazados, con lo cual se garantiza la confidencialidad de la información transmitida.

5.2.3.2.5 Automatización de tareas.

Para la automatización de tareas en los equipos de cómputo, los administradores de sistemas comúnmente utilizan el conjunto de utilerías del sistema operativo, apoyados en programación en ***shell***, ***Perl***, ***Python*** o algún otro lenguaje intérprete de comandos.

Adicional a estas herramientas, se encuentra también ***Expect***, el cual es un lenguaje intérprete de comandos que permite la automatización de tareas para procesos que forzosamente son interactivos, como por ejemplo automatizar una sesión remota. Por lo que este programa es una herramienta verdaderamente útil para la automatización de tareas de administración de sistemas.

5.2.3.3 Interfase para el usuario.

La interfase que se propone para la implantación del presente proyecto, se constituye a partir de la organización de la información de tal forma que resulte en una navegación sencilla para el usuario, a la vez que proporcione el acceso rápido a la información requerida.

Para lograr lo comentado en el párrafo anterior, es necesario, como primera consideración, que se trabaje en todo momento en una pantalla genérica organizada de acuerdo a como se ilustra en la Tabla 5-8.

<Menú principal> **Inicio** **Servicios** **Políticas** **Equipos** **Catálogos**	
<Barra informativa> **Fecha-Hora** **Sección de la aplicación** **Alarmas** **Alertas** **Notificaciones** **Mensajes urgentes** **Mensajes nuevos** **Mensajes totales**	
<Menú ajustable al contexto>	***<Área de despliegue de información>***

Tabla 5-8. Esquema de navegación de la interfase de usuario.

5.2.3.3.1 Menú principal

Este menú se presenta como un elemento de navegación a base de pestañas, que permiten desplazarse rápidamente a las diferentes secciones principales de la aplicación.

En la Tabla 5-9 se muestra la lista de menús y submenús requeridos en la aplicación.

Menú principal	Inicio	Alarmas	Listado de alarmas
		Alertas	Listado de alertas
		Notificaciones	Listado de notificaciones
	Servicios	Activos	Listado de todos los servicios activos
		Puertos	Listado de todos los servicios organizados por puerto de red utilizado
		Protocolo	Listado de todos los servicios organizados por protocolo de red

			utilizado
		Programa	Listado de todos los servicios organizados por nombre de programa utilizado
		Ubicación	Listado de todos los servicios organizados por ubicación específica
		Estatus	Listado de todos los servicios organizados por estatus
	Políticas	Activas	Listado de todas las políticas activas
		Tipo	Listado de todas las políticas organizadas por tipo
		Plataforma	Listado de todas las políticas organizadas por plataforma
		Estatus	Listado de todas las políticas organizadas por estatus
	Equipos	Activos	Listado de todos los equipos activos
		Marca	Listado de todos los equipos organizados por marca
		Sistema operativo	Listado de todos los equipos organizados por sistema operativo
		Clasificación	Listado de todos los equipos organizados por clasificación
		Procesador	Listado de todos los equipos organizados por procesador
		Ubicación	Listado de todos los equipos organizados por ubicación específica
		Soporte tecnico	Listado de todos los equipos organizados por soporte técnico
		Estatus	Listado de todos los equipos organizados por estatus
	Catálogos	Listado de tablas	Acciones a realizar sobre las tablas: - Examinar - Seleccionar - Insertar - Propiedades - Vaciar - Eliminar

Tabla 5-9. Lista de menús y submenús de la aplicación

5.2.3.3.2 Barra informativa

Esta barra deberá estar disponible desde cualquier sección de la aplicación y deberá mantener la información actualizada para notificar sobre los siguientes aspectos:

- Fecha-Hora
- Identificación de la sección de la aplicación en la que se encuentra el usuario.
- Número de alarmas
- Número de alertas
- Número de notificaciones
- Número de mensajes urgentes
- Número de mensajes nuevos
- Número de mensajes totales

De esta manera, el usuario de la aplicación contará con información que le permitirá conocer de manera rápida el estatus de los equipos y servicios monitoreados.

5.2.3.3.3 Pantalla inicial.

La pantalla inicial proporcionará, en el área de despliegue de información, un resumen con la información suficiente del estado que guardan los sistemas monitoreados y que permita tomar decisiones inmediatamente, pero sin saturar al usuario con demasiada información. La información considerada para conformar esta pantalla estará constituida por los siguientes elementos:

- Alarmas en equipos.

En este grupo se encuentran aquellos equipos que presenten situaciones graves, ya sea debido a que tengan alguna anomalía o porque tengan instalado software que presenta vulnerabilidades de seguridad.

- Alertas en equipos.

En este grupo se encuentran aquellos equipos que presenten situaciones menos graves, como la detección de actividad sospechosa o que tenga pendiente la instalación de actualizaciones que no sean críticas.

- Notificaciones.

En este grupo se encuentran los equipos a los cuales algún administrador colocó un mensaje para otro administrador, con el objeto de que antes de aplicar cambios a los equipos de este grupo se verifiquen dichos mensajes.

5.2.3.3.4 Consideraciones generales.

Adicionalmente, como se ha mencionado, deberán integrarse opciones para que el usuario pueda por sí mismo, organizar la información por diversos criterios y por supuesto que pueda tener acceso a la información a detalle que requiera.

Los elementos básicos para organizar la información, de acuerdo al esquema entidad-relación planteado anteriormente, son los relacionados a servicios, políticas y equipos.

Otro aspecto que es necesario considerar y que servirá para mantener la información que se requiera introducir de manera manual, deberá ser una sección destinada al mantenimiento de catálogos de información.

Con el análisis realizado hasta ahora, se tienen suficientes elementos para pasar a la etapa de construcción del prototipo de la solución propuesta.

Como se ha podido observar a lo largo de este capítulo, la propuesta de solución al problema de la integración de la seguridad, resulta relativamente fácil de implantar debido a que está basado en herramientas de libre distribución, además de los conceptos sobre los cuales reposa son ampliamente utilizados.

Capítulo 6. Desarrollo de la propuesta.

En base a los elementos expuestos hasta ahora, en este capítulo se documenta el desarrollo de la propuesta, que incluye los elementos para construir la solución así como, las consideraciones necesarias y la logistica a llevar a cabo para la implantación de la solución.

Como parte de la implantación de la solución propuesta, se incluye un procedimiento de instalación y operación así como, las consideraciones para el mantenimiento de la misma.

6.1 Construcción.

La construcción de la solución se lleva a cabo mediante la integración de herramientas programadas mediante el concepto de software libre, también conocido como Open Source[51].

Cabe mencionar que bajo este concepto se ha desarrollado software muy diverso, desde compiladores, librerías y sistemas operativos, hasta herramientas de aplicación muy avanzadas como interfases gráficas, sistemas manejadores de bases de datos y asistentes personales.

Para el caso de la construcción del prototipo de la solución propuesta en esta tesis, se emplean lenguajes de programación con los cuales se construyen diversos módulos de programas. Por otra parte, se realizan las adecuaciones necesarias a aplicaciones que actualmente existen, las cuales se han seleccionado en virtud de que proporcionan una adecuada funcionalidad para la solución propuesta.

6.1.1 Requisitos previos.

Los lenguajes utilizados en el desarrollo del presente proyecto son Perl y PHP principalmente. En Perl se desarrollan los programas recolectores de información, así como también los programas para carga de información a la base de datos. El lenguaje PHP es utilizado para construir la interfase de usuario, a la cual se tendrá acceso a través de un ambiente web. El punto de interacción entre los diversos programas que conforman la solución será a través de la información almacenada en la base de datos.

Adicionalmente, para el adecuado funcionamiento de la solución, se requiere que previamente se tengan los siguientes programas instalados:

- Apache (http://httpd.apache.org/), servidor web.

[51] Para más información relacionada con el desarrollo del concepto Open Source, dirigirse al sitio http://www.opensource.org/.

El proyecto Apache HTTP Server, es un esfuerzo para desarrollar y mantener un servidor HTTP basado en software libre. Cabe señalar que de acuerdo a las estadísticas de netcraft[52], Apache es el servidor de web más utilizado en Internet, ya que más del 63% de los sitios en Internet funcionan con este software.

- OpenSSL (http://www.openssl.org/), herramientas criptográficas.

El proyecto OpenSSL es un esfuerzo colaborativo basado en software libre para desarrollar un conjunto de herramientas robustas, competitivas comercialmente y con toda la funcionalidad de las especificaciones de los protocolos Secure Socket Layer (SSL v2/v3) y Transport Layer Security (TLS v1), así como también la integración de un biblioteca criptográfica de propósito general.

- Perl (http://www.perl.com/), lenguaje de programación.

El lenguaje de programación Perl (Practical Extraction and Report Language), es un esfuerzo basado en software libre iniciado por Larry Wall para construir un lenguaje optimizado para examinar archivos de texto arbitrario, extraer información de dichos archivos y elaborar reportes basados en la información obtenida. También es un buen lenguaje para apoyar con muchas tareas de administración de sistemas. Este lenguaje fue pensado para ser práctico, es decir, fácil de usar, eficiente y completo en primer término, no tanto para ser bonito, ligero, elegante o mínimo.

- PHP (http://www.php.net/), lenguaje de programación.

PHP, es acrónimo recursivo de "PHP: Hypertext Preprocessor", el cual es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP.

- PostgreSQL (http://www.postgresql.org/), manejador de bases de datos.

Este es un desarrollo de software libre que implementa un manejador de bases de datos relacional (DBMS) con un amplio soporte del lenguaje de consulta estructurado (SQL). PostgreSQL es uno de los manejadores de bases de datos más avanzado y es utilizado en diversos sitios para soportar una gran cantidad de aplicaciones aplicaciones.

- CVS (http://www.cvshome.org/), sistema de control de versiones.

Este proyecto desarrolla, bajo el esquema de software libre, un sistema de control de versiones de software concurrentes (Concurrent Versions System). CVS es el controlador de versiones de software más popular en Internet.

- OpenSSH (http://www.openssh.org/), conexión remota segura.

---

[52] http://news.netcraft.com/.

Este software es una versión libre de la suite de protocolos secure shell (SSH). OpenSSH cifra todo el tráfico (incluyendo contraseñas) para eliminar de manera efectiva ataques a nivel de red.

- Mindterm (http://www.appgate.com/mindterm/), conexión remota segura.

Mindterm es un cliente que implementa los protocolos SSH1 y SSH2 y está escrito en lenguaje Java puro. Puede ser utilizado como aplicación de software independiente o como applet en operación conjunta con un navegador de web para tener acceso a servidores mediante secure shell. La operación mediante applet, permite que la conexión pueda realizarse sin la necesidad de contar en el sistema operativo con un cliente SSH, de lo cual se aprovecha la solución propuesta para ser totalmente independiente de la plataforma en que funciona.

- Cfengine (http://www.cfengine.org/)

Configuration engine o cfengine, es un proyecto basado en software libre para desarrollar un agente autónomo y un lenguaje de alto nivel para definición de políticas para construir un sistema experto que administre y configure grandes redes de computadoras.

- RPM (http://www.rpm.org/)

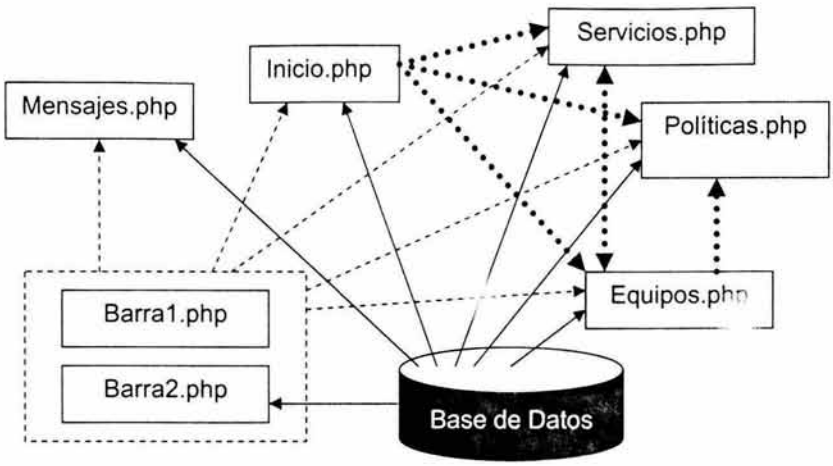
Este es un proyecto desarrollado bajo esquema de software libre para desarrollar un sistema de administración de paquetes de software. RPM es un acrónimo recursivo de RPM Package Manager (RPM), el cual es capaz de instalar, desinstalar, verificar, consultar y actualizar paquetes de software en diversos sistemas operativos.

6.1.2 Programas desarrollados.

Para la puesta en funcionamiento de la consola de administración remota de seguridad en web, fue necesario desarrollar los programas que en esta sección se describen. Los programas desarrollados son los siguientes:

- `Inicio.php`
- `Barra1.php`
- `Barra2.php`
- `Mensajes.php`
- `Servicios.php`
- `Politicas.php`
- `Equipos.php`

En la Figura 6-1 se muestra de manera esquemática la interrelación de estos programas entre sí y con la base de datos.



Simbología:

Transferencia de datos ——→
Integración de programas - - - - - →
Navegación entre programas • • • • • →

Figura 6-1. Relación de flujos de datos e interacción entre programas.

6.1.2.1 Funcionamiento de la interfase.

Como ya se ha comentado, la interfase para el usuario fue desarrollada para operar en ambiente web, utilizando cualquier navegador estándar, esto con el objeto de que la aplicación pueda funcionar prácticamente en cualquier ambiente de trabajo y que el acceso a la misma sea fácil para el usuario.

Una vez que el usuario de la aplicación ha solicitado el URL de la aplicación, se le requerirá que ingrese el nombre de usuario y la contraseña asignados para autenticarse. Una vez cumplido este requisito, se tendrá acceso a la pantalla inicial.

Entre las principales características de la interfase de usuario del sistema para la administración remota de seguridad, destacan las siguientes:

- La navegación a través de las distintas funcionalidades, se realiza a través de pestañas.
- En todo momento existe una barra de estado que indica información relevante:
  - Fecha y hora de la última consulta realizada,
  - Sección de la aplicación en la que se encuentra el usuario,

- Número de alarmas,
- Número de alertas,
- Número de notificaciones,
- Número de mensajes urgentes,
- Número de mensajes nuevos,
- Total de mensajes.

- En todo momento existe un menú contextual relacionado a la funcionalidad elegida por el usuario.
- Vista de los resultados de la funcionalidad elegida por el usuario.

A manera de ejemplo, se muestra en la Figura 6-2 la pantalla inicial de la aplicación en la cual se indican los distintos elementos de navegación.

Como puede observarse, en la pantalla inicial se presentan ligas hacia los equipos que presentan alarmas, alertas o notificaciones, lo cual permite que de manera inmediata se pueda obtener mayor información en relación con el problema reportado.

Es importante mencionar que se trata de una aplicación dinámica, en la que en todo momento es posible moverse entre las pestañas de navegación, así como también hacer uso de las ligas que ofrece la barra de estado, con lo cual se tiene acceso directo a las alarmas, alertas, notificaciones y mensajes que se generan.

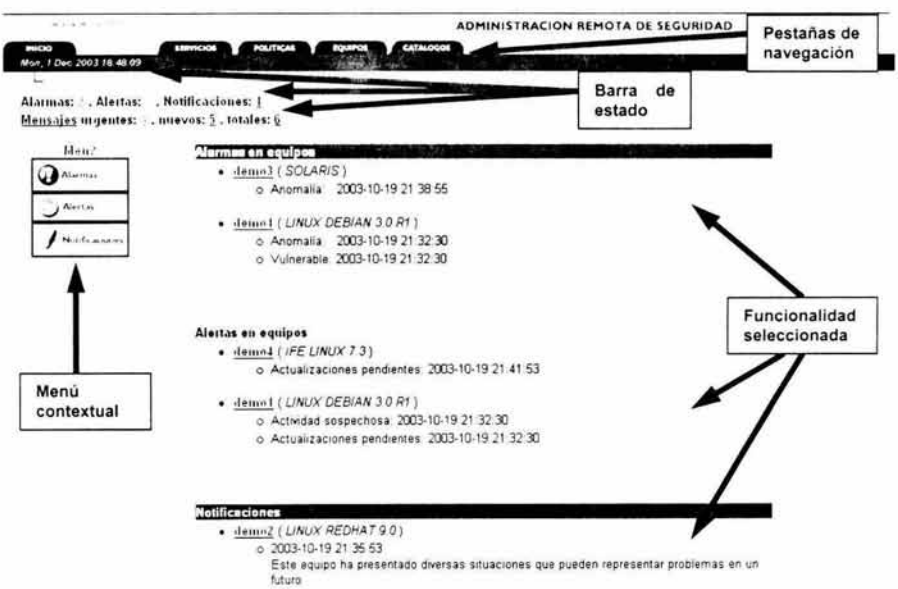


Figura 6-2. Pantalla inicial y descripción de la interfase.

Dependiendo de la funcionalidad seleccionada, será la información que se despliegue, pero en todos los casos se mantiene la misma organización de la interfase de usuario, así como también los elementos de navegación que permiten establecer vínculos entre los diversos módulos de la aplicación.

6.1.2.2 Inicio.php

Este es el módulo inicial y permite visualizar, en forma de resumen, los equipos que presentan alarmas, alertas o notificaciones. De esta manera, el usuario tiene un panorama muy claro de la situación que guardan los servicios informáticos relacionados en la base de datos. En la Figura 6-3 se puede observar la pantalla generada por este módulo.

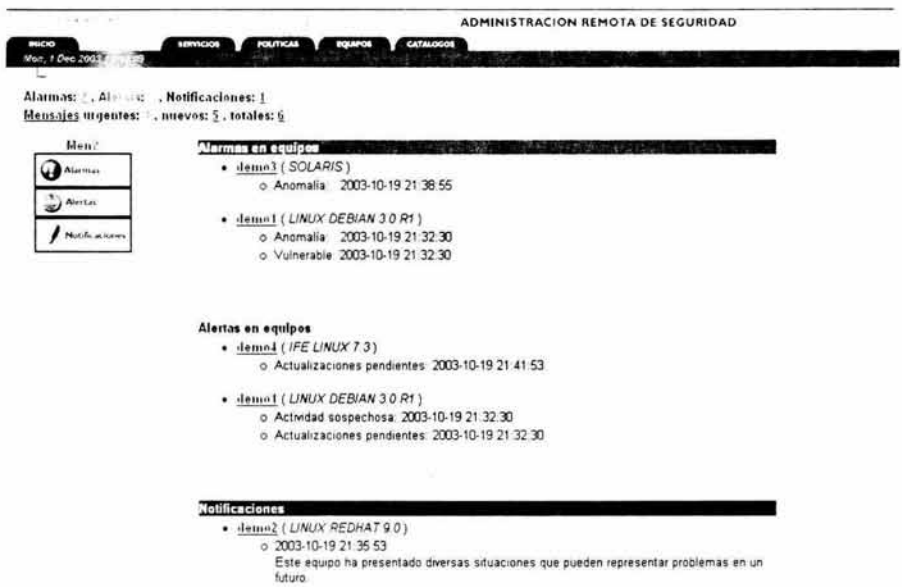


Figura 6-3. Pantalla de inicio.

La información que utiliza este módulo se obtiene de acuerdo a la relación mostrada en la Tabla 6-1.

Información desplegada	Obtenida de	
	Tabla	*Campo*
Alarmas	Equipo	Anomalia Vulnerable
Alertas	Equipo	Actividad_sospechosa Actualizaciones_pendientes
Notificaciones	Equipo	Notas_informativas

Tabla 6-1. Relación del origen de datos para el módulo `inicio.php`.

A continuación, en la Tabla 6-2 se muestra el código del programa.

```
<html>
<head>
<title>inicio</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
</head>

<? include './barra1.php' ?>

      <td width="99%" height="19"><font face=Arial size=2><b><img
src="images/inicio.jpg" width="80" height="18">

<? include './barra2.php' ?>

<table cellspacing=0 cellpadding=0 width="100%" border=0
summary="Tabla principal">
  <tbody>
  <tr>
    <td valign=top width=148>
      <div align="center"><font face="Arial, Helvetica, sans-serif"
size="2" color="#003399">
<b>Menú</b>
</font><br>
        <a href="#alarmas"><img src="images/menu_alarmas.png"
width="99" height="32"></a><br>
        <a href="#alertas"><img src="images/menu_alertas.png"
width="99" height="27"></a><br>
        <a href="#notificaciones"><img
src="images/menu_notificaciones.png" width="99" height="33"></a></div>
    </td>
    <td valign=top width="614">
      <table cellspacing=0 cellpadding=0 width="100%" border=1
            summary="Tabla de alarmas">
        <tbody>
        <tr>
          <td bgcolor=#ff0000><font face=Arial size=2><b><font
color="#FFFFFF" id="alarmas">
Alarmas en equipos
</font></b>
            </font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
```

```
<?
$ARS_query = "select * from equipo where anomalia is not null or
vulnerable is not null order by anomalia desc, vulnerable desc";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
   echo "<ul>";
   while ($ARS_arr = pg_fetch_array($ARS_result))
   {
      echo "<li>";
      echo "<b><a href=\"equipos.php?ver=$ARS_arr[id_equipo]\">";
      echo $ARS_arr["nombre"];
      echo "</a></b>";
      echo " ( <i>";
      echo $ARS_arr["sistema_operativo"];
      echo "</i> ) ";
      echo "<ul>";
      $ARS_anomalia = explode (".",$ARS_arr["anomalia"]);
      $ARS_vulnerable = explode (".",$ARS_arr["vulnerable"]);
      if ( $ARS_anomalia[0] != "" )
         echo "<li>Anomal&iacute;a:   $ARS_anomalia[0]";
      if ( $ARS_vulnerable[0] != "" )
         echo "<li>Vulnerable: $ARS_vulnerable[0]";
      echo "</ul>";
      echo "<br>";
   }
   echo "</ul>";
?>
</font></td>
        </tr>
        <tr>
          <td height="11">  </td>
        </tr>
        </tbody>
      </table>
      <table cellspacing=0 cellpadding=0 width="100%" border=1
            summary="Tabla de alertas">
        <tbody>
        <tr bgcolor="#FF9900">
          <td><font face=Arial size=2><b id="alertas">Alertas en
equipos</b> </font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
$ARS_query = "select * from equipo where actividad_sospechosa is not
null or actualizaciones_pendientes is not null order by
actividad_sospechosa desc, actualizaciones_pendientes desc";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
   echo "<ul>";
   while ($ARS_arr = pg_fetch_array($ARS_result))
```

```
   {
      echo "<li>";
      echo "<b><a href=\"equipos.php?ver=$ARS_arr[id_equipo]\">";
      echo $ARS_arr["nombre"];
      echo "</a></b>";
      echo " ( <i>";
      echo $ARS_arr["sistema_operativo"];
      echo "</i> ) ";
      echo "<ul>";
      $ARS_actsosp = explode (".",$ARS_arr["actividad_sospechosa"]);
      $ARS_actpend = explode
(".",$ARS_arr["actualizaciones_pendientes"]);
      if ( $ARS_actsosp[0] != "" )
         echo "<li>Actividad sospechosa: $ARS_actsosp[0]";
      if ( $ARS_actpend[0] != "" )
         echo "<li>Actualizaciones pendientes: $ARS_actpend[0]";
      echo "</ul>";
      echo "<br>";
   }
   echo "</ul>";
?>
</font></td>
        </tr>
        <tr>
          <td height="5">  </td>
        </tr>
        </tbody>
      </table>
      <table cellspacing=0 cellpadding=0 width="100%" border=1
           summary="Tabla de notificaciones" height="47">
        <tbody>
        <tr bgcolor="#003399">
          <td><font face=Arial
                 size=2><b><font color="#FFFFFF"
id="notificaciones">Notificaciones</font></b> </font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
$ARS_query = "select * from equipo where notas_informativas != '' order
by registrar_fecha";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
   echo "<ul>";
   while ($ARS_arr = pg_fetch_array($ARS_result))
   {
      echo "<li>";
      echo "<b><a href=\"equipos.php?ver=$ARS_arr[id_equipo]\">";
      echo $ARS_arr["nombre"];
      echo "</a></b>";
      echo " ( <i>";
      echo $ARS_arr["sistema_operativo"];
      echo "</i> ) ";
      echo "<ul>";
```

```
        $ARS_regfecha = explode (".", $ARS_arr["registrar_fecha"]);
        echo "<li>";
        echo $ARS_regfecha[0];
        echo "<br>";
        echo $ARS_arr["notas_informativas"];
        echo "</ul>";
        echo "<br>";
    }
    echo "</ul>";
?>
</font></td>
          </tr>
          <tr>
            <td>  </td>
          </tr>
          </tbody>
        </table>
      </td>
    </tr>
    <td width="148"> 
</table>

<? include './dbclose.php' ?>
</body>
</html>
```

Tabla 6-2. Código fuente del programa `inicio.php`.

6.1.2.3 Barra1.php y barra2.php

Estos módulos se encargan de generar una barra de estado de la aplicación, la cual despliega el número de equipos que presentan alarmas, alertas o notificaciones. También se indican los mensajes que han sido enviados al usuario, clasificados en urgentes, nuevos y totales. Estos módulos se encuentran incorporados en los otros programas.

La información que utilizan estos módulos se obtiene de acuerdo a la relación mostrada en la Tabla 6-3.

Información desplegada	Obtenida de	
	Tabla	***Campo***
Alarmas	Equipo	Anomalia Vulnerable
Alertas	Equipo	Actividad_sospechosa Actualizaciones_pendientes
Notificaciones	Equipo	Notas_informativas
Mensajes urgentes	Mensaje	Tipo
Mensajes nuevos	Mensaje	Estatus
Mensajes totales	Mensaje	Destinatario

Tabla 6-3. Relación del origen de datos para los módulos `barra2.php`.

A continuación, en la Tabla 6-4 y en la Tabla 6-5 se muestra el código fuente de estos programas.

```
<? include './dbconn.php' ?>

<body bgcolor="#FFFFFF" background="images/back.jpg">
<div id="Layer2" style="position:absolute; left:-1px; top:-1px;
width:1431px; height:37px; z-index:2">
  <table width="102%" border="0" cellpadding="0" cellspacing="0"
height="23" align="left">
    <tr bgcolor="#003366">
      <td height="19" colspan="2">
        <div align="left"><font face="Arial" size="1"
color="#FFFFFF"><b> </b></font></div>
      </td>
      <td height="19" colspan="2"><font face=Arial size=3><b><i><font
size="1" color="#FFFFFF">
<?
$ARS_func=date ("r");
$ARS_date=explode("-", $ARS_func);
echo $ARS_date[0];
?>
</font></i></b></font><font face="Arial" size="1" color="#FFFFFF"><b>
        </b></font></td>
    </tr>
    <tr bgcolor="#CCCCCC">
      <td colspan="2" height="38" rowspan="2"><font face=Arial
size=2><b> </b></font></td>
```

Tabla 6-4. Código fuente del programa `barra1.php`.

```
        </b></font></td>
      <td width="0%" height="38" rowspan="2"> </td>
    </tr>
    <tr bgcolor="#CCCCCC">
      <td width="99%" height="19"><font face=Arial size=2><b>Alarmas:
<a
      href="inicio.php">
<font color=#ff0000>
<?
$ARS_query = "select count(*) from equipo where anomalia is not null or
vulnerable is not null";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
$ARS_arr = pg_fetch_array($ARS_result);
echo $ARS_arr[0];
?>
</font></a>
        , Alertas: <a href="inicio.php">
<font color="#BB7700">
<?
```

```
$ARS_query = "select count(*) from equipo where actividad_sospechosa is
not null or actualizaciones_pendientes is not null";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
$ARS_arr = pg_fetch_array($ARS_result);
echo $ARS_arr[0];
?>
</font></a>
        , Notificaciones: <a
      href="inicio.php">
<font color="#003399">
<?
$ARS_query = "select count(*) from equipo where notas_informativas !=
''";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
$ARS_arr = pg_fetch_array($ARS_result);
echo $ARS_arr[0];
?>
</font></a>
        </b></font></td>
    </tr>
    <tr bgcolor="#CCCCCC">
      <td width="1%">
        <div align="left"><font face=Arial size=2><b><font
color="#003399"> </font></b></font></div>
      </td>
      <td width="0%"> </td>
      <td width="99%"><font face=Arial size=2><b><a
href="mensajes.php">Mensajes</a> urgentes: <a
      href="mensajes.php#mensajes_urgentes">
<font color=#ff0000>
<?
$ARS_UID=getenv("REMOTE_USER");
$ARS_query = "select count(*) from mensaje where tipo='URGENTE' and
destinatario like '%$ARS_UID%'";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
$ARS_arr = pg_fetch_array($ARS_result);
echo $ARS_arr[0];
?>
</font></a>
        , nuevos:<font color="#003399"> <a
href="mensajes.php#mensajes_nuevos">
<?
$ARS_query = "select count(*) from mensaje where estatus='SIN_LEER' and
destinatario like '%$ARS_UID%'";
$ARS_result = pg_query($ARS_query);
```

```
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
$ARS_arr = pg_fetch_array($ARS_result);
echo $ARS_arr[0];
?>
</a></font>
        , totales:<font color="#003399"> <a
href="mensajes.php#mensajes_totales">
<?
$ARS_query = "select count(*) from mensaje where destinatario like
'%$ARS_UID%'";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
$ARS_arr = pg_fetch_array($ARS_result);
echo $ARS_arr[0];
?>
</a></font>

</b></font></td>
      <td width="0%"> </td>
    </tr>
  </table>
</div>
<p><br>
</p>
<br>
<br>
<br>
```

Tabla 6-5. Código fuente del programa `barra2.php`.

6.1.2.4 Mensajes.php

Este módulo despliega la lista de mensajes enviados al usuario activo de la aplicación. Los mensajes son clasificados en urgentes, nuevos y totales. Este módulo permite la interacción de los diferentes administradores de sistemas encargados de los equipos, con el objeto de coordinar actividades e intercambiar ideas de manera asíncrona. En la Figura 6-4 se puede observar la pantalla generada por este módulo.

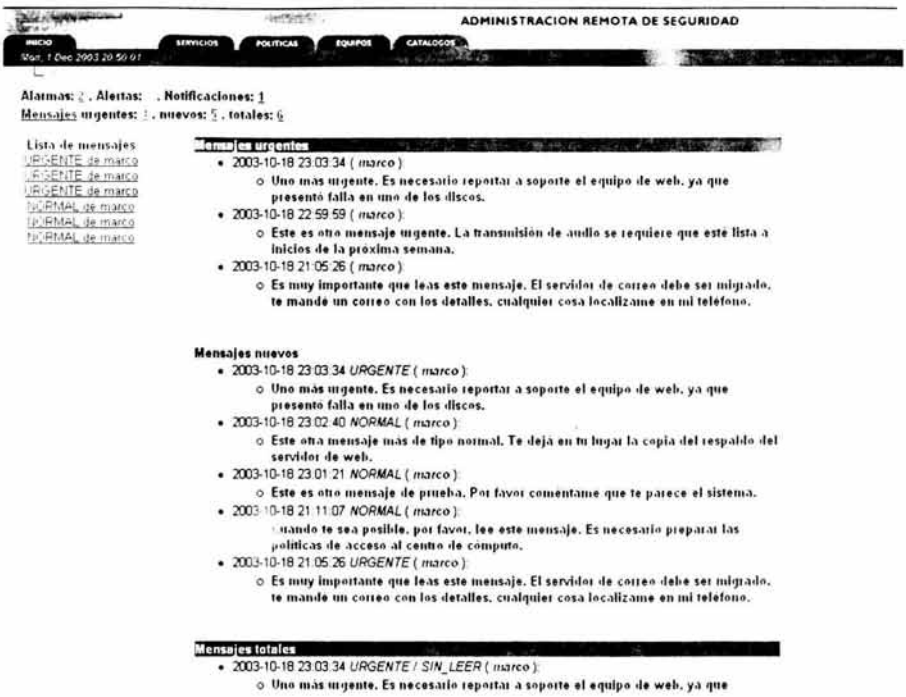


Figura 6-4. Pantalla de mensajes.

La información que utiliza este módulo se obtiene de acuerdo a la relación mostrada en la Tabla 6-6.

Información desplegada	Obtenida de	
	Tabla	***Campo***
Mensajes urgentes	Mensaje	Tipo
Mensajes nuevos	Mensaje	Estatus
Mensajes totales	Mensaje	Destinatario

Tabla 6-6. Relación del origen de datos para el módulo `mensajes.php`.

A continuación, en la Tabla 6-7 se muestra el código fuente de este programa.

```
<html>
<head>
<title>inicio</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
</head>

<? include './barra1.php' ?>
```

```
      <td width="99%" height="19"><font face=Arial size=2><b><img
src="images/inicio.jpg" width="80" height="18">

<? include './barra2.php' ?>

<table cellspacing=0 cellpadding=0 width="100%" border=0
summary="Tabla principal">
  <tbody>
  <tr>
    <td valign=top width=148>
      <div align="center"><font face="Arial, Helvetica, sans-serif"
size="2" color="#003399">
<b>Lista de mensajes</b>
<br>
<?
$ARS_UID=getenv("REMOTE_USER");
$ARS_query = "select * from mensaje where destinatario like
'%$ARS_UID%' order by tipo desc, registrar_fecha";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
while ($ARS_arr = pg_fetch_array($ARS_result))
{
   echo "<a href=\"mensajes.php#$ARS_arr[id_mensaje]\">";
   echo $ARS_arr["tipo"];
   echo " de ";
   echo $ARS_arr["id_administrador"];
   echo "</a>";
   echo "<br>";
}
?>
</font>
        <br>
    </td>
    <td valign=top width="614">
      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="Tabla de mensajes urgentes">
        <tbody>
        <tr>
          <td bgcolor=#ff0000><font face=Arial size=2><b><font
color="#FFFFFF" id="mensajes_urgentes">
Mensajes urgentes
</font></b>
            </font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
$ARS_query = "select * from mensaje where tipo='URGENTE' and
destinatario like '%$ARS_UID%' order by registrar_fecha desc";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
```

```
echo "<ul>";
while ($ARS_arr = pg_fetch_array($ARS_result))
{
   $ARS_fecha = explode (".", $ARS_arr["registrar_fecha"]);
   echo "<li id=\"$ARS_arr[id_mensaje]\">";
   echo "$ARS_fecha[0]";
   echo " ( <b><i>";
   echo $ARS_arr["id_administrador"];
   echo "</b></i> ): ";
   echo "<br>";
   echo "<ul>";
   echo "<li><b>";
   echo $ARS_arr["mensaje"];
   echo "</b></ul>";
}
echo "</ul>";
?>
</font></td>
        </tr>
        <tr>
          <td height="11">  </td>
        </tr>
        </tbody>
      </table>
      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="Tabla de mensajes nuevos">
        <tbody>
        <tr bgcolor="#FF9900">
          <td><font face=Arial size=2><b id="mensajes_nuevos">Mensajes
nuevos</b> </font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
$ARS_query = "select * from mensaje where estatus='SIN_LEER' and
destinatario like '%$ARS_UID%' order by registrar_fecha desc";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
echo "<ul>";
while ($ARS_arr = pg_fetch_array($ARS_result))
{
   $ARS_fecha = explode (".", $ARS_arr["registrar_fecha"]);
   echo "<li id=\"$ARS_arr[id_mensaje]\">";
   echo "$ARS_fecha[0] <i>";
   echo $ARS_arr["tipo"];
   echo "</i> ( <b><i>";
   echo $ARS_arr["id_administrador"];
   echo "</b></i> ): ";
   echo "<br>";
   echo "<ul>";
   echo "<li><b>";
   echo $ARS_arr["mensaje"];
   echo "</b></ul>";
}
```

```
echo "</ul>";
?>
</font></td>
        </tr>
        <tr>
          <td height="5">  </td>
        </tr>
        </tbody>
      </table>
      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="Tabla de mensajes totales" height="47">
        <tbody>
        <tr bgcolor="#003399">
          <td><font face=Arial
                  size=2><b><font color="#FFFFFF"
id="mensajes_totales">Mensajes totales</font></b> </font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
$ARS_query = "select * from mensaje where destinatario like
'%$ARS_UID%' order by registrar_fecha desc";
$ARS_result = pg_query($ARS_query);
if (!$ARS_result) {
    echo "Error al procesar: $ARS_query.\n";
    exit;
}
echo "<ul>";
while ($ARS_arr = pg_fetch_array($ARS_result))
{
   $ARS_fecha = explode (".", $ARS_arr["registrar_fecha"]);
   echo "<li id=\"$ARS_arr[id_mensaje]\">";
   echo "$ARS_fecha[0] <i>";
   echo $ARS_arr["tipo"];
   echo "</i> / <i>";
   echo $ARS_arr["estatus"];
   echo "</i> ( <b><i>";
   echo $ARS_arr["id_administrador"];
   echo "</b></i> ): ";
   echo "<br>";
   echo "<ul>";
   echo "<li><b>";
   echo $ARS_arr["mensaje"];
   echo "</b></ul>";
}
echo "</ul>";
?>
</font></td>
        </tr>
        <tr>
          <td>  </td>
        </tr>
        </tbody>
      </table>
    </td>
  </tr>
  <td width="148"> 
```

```
</table>

<? include './dbclose.php' ?>
</body>
</html>
```

Tabla 6-7.Código fuente del programa `mensajes.php`.

6.1.2.5 Servicios.php

Este módulo despliega la lista de servicios dados de alta en la base de datos, los cuales pueden ser organizados mediante diversos criterios según lo requiera el usuario. Además, se puede tener acceso a la información detallada de cada servicio y se relaciona al equipo en el cual funciona. En la Figura 6-5 se puede observar la pantalla generada por este módulo.

Figura 6-5. Pantalla de servicios.

La información que utiliza este módulo se obtiene de acuerdo a la relación mostrada en la Tabla 6-8.

Información desplegada	Obtenida de	
	Tabla	***Campo***
Listado de servicios	Servicio	Estatus Puerto Protocolo Programa Ubicación

Tabla 6-8. Relación del origen de datos para el módulo `servicios.php`.

A continuación, en la Tabla 6-9 se muestra el código fuente de este programa.

```
<html>
<head>
<title>Equipos</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
</head>

<? include './barra1.php' ?>

      <td width="99%" height="19"><font face=Arial size=2><b><img
src="images/1servicios.jpg" width="105" height="18">

<? include './barra2.php' ?>

<table cellspacing=0 cellpadding=0 width="100%" border=0
summary="Tabla principal">
  <tbody>
  <tr>
    <td valign=top width=155>
      <div align="center"><font face="Arial, Helvetica, sans-serif"
size="2" color="#003399"><b>Menú</b>
<br>
<?
//Generar menú de opciones

//Obtener parámetro enviado
$ARS_QUERY=getenv("QUERY_STRING");
$ARS_PARAM=explode("=", $ARS_QUERY);
?>
<br>
<a href="servicios.php">ACTIVOS</a><br>
<a href="servicios.php?puerto">PUERTO</a><br>
<a href="servicios.php?protocolo">PROTOCOLO</a><br>
<a href="servicios.php?programa">PROGRAMA</a><br>
<a href="servicios.php?id_ubicacion">UBICACION</a><br>
<a href="servicios.php?estatus">ESTATUS</a><br>
</font>
    </td>
    <td valign=top width="649">

      <table cellspacing=0 cellpadding=0 width="100%" border=0
           summary="Tabla de politicas">
        <tbody>
        <tr bgcolor="#004182">
          <td><font face="Arial" size="2" color="#FFFFFF">
<?
switch ($ARS_PARAM[0])
{
   case "":
// Inicia case ""
?>
      Listado de servicios activos
      </font></td></tr><tr><td><font face=Arial size=2>
<?
```

```
      $ARS_query = "select * from servicio where estatus='ACTIVO'";
      $ARS_result = pg_query($ARS_query);
      if (!$ARS_result) {
          echo "Error al procesar: $ARS_query.\n";
          exit;
      }
      echo "<ul>";
      while ($ARS_arr = pg_fetch_array($ARS_result))
      {
         echo "<li>";
         echo "<b><a
href=\"servicios.php?ver=$ARS_arr[id_servicio]\">";
         echo $ARS_arr["nombre"];
         echo "</a></b>";
         echo " ( equipo <i>";
         echo $ARS_arr["id_equipo"];
         echo "</i>, puerto <i>";
         echo $ARS_arr["puerto"];
         echo "</i>, protocolo <i>";
         echo $ARS_arr["protocolo"];
         echo "</i>, programa <i>";
         echo $ARS_arr["programa"];
         echo "</i>, versi&oacute;n <i>";
         echo $ARS_arr["version"];
         echo "</i> ) ";
         echo "<br>";
      }
      echo "</ul>";
?>
      </font></td></tr><tr><td
height="11"> </td></tr></tbody></table>
<?
      break;
// Termina case ""
   case "ver":
// Inicia case "ver"
?>
      Servicio:
<?
      $ARS_query = "select * from servicio where
id_servicio='$ARS_PARAM[1]'";
      $ARS_result = pg_query($ARS_query);
      if (!$ARS_result) {
          echo "Error al procesar: $ARS_query.\n";
          exit;
      }
      $ARS_arr = pg_fetch_array($ARS_result);
      echo "<b>";
      echo $ARS_arr["nombre"];
      echo "</b>";
      echo " ( id ";
      echo $ARS_arr["id_servicio"];
      echo ", ";
      echo $ARS_arr["estatus"];
      echo ", ";
      echo $ARS_arr["registrar_fecha"];
      echo " ) ";
```

```
?>
      </font></td></tr><tr><td><font face=Arial size=2>
<?
      echo "<ul>";
      echo "<li><i><b>Descripci&oacute;n:</b></i><br> ";
      echo $ARS_arr["descripcion"];
      echo "<li><i><b>Puerto:</b></i><br> ";
      echo $ARS_arr["puerto"];
      echo "<li><i><b>Protocolo:</b></i><br> ";
      echo $ARS_arr["protocolo"];
      echo "<li><i><b>Programa:</b></i><br> ";
      echo $ARS_arr["programa"];
      echo "<li><i><b>Versi&oacute;n:</b></i><br> ";
      echo $ARS_arr["version"];
      echo "<li><i><b>Notas:</b></i><br>";
      echo $ARS_arr["notas"];
      echo "</ul>";
?>
      </font></td></tr><tr><td
height=\"11\"> </td></tr></tbody></table>
      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="datos">
        <tbody>
        <tr bgcolor="#005BB7">
          <td><font face="Arial" size="2" color="#FFFFFF">
      Equipo
</font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
      echo "<ul>";
      echo "<li><i><b>Equipo:</b></i><br> ";
      echo "<b><a href=\"equipos.php?ver=$ARS_arr[id_equipo]\">";
      echo $ARS_arr["id_equipo"];
      echo "</a></b>";
      echo "</ul>";
?>
</font></td>
        </tr>
        <tr>
          <td height="5">  </td>
        </tr>
        </tbody>
      </table>
      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="estatus" height="47">
        <tbody>
        <tr bgcolor="#0981D9">
          <td><font face=Arial size=2>
      Soporte t&eacute;cnico
</font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
      echo "<ul>";
```

```
      echo "<li><i><b>Soporte t&eacute;cnico:</b></i><br> ";
      print $ARS_arr["id_soporte_tecnico"];
      echo "</ul>";
?>
</font></td>
        </tr>
        <tr>
          <td>  </td>
        </tr>
        </tbody>
      </table>
<?
      break;
// Termina case "ver"
      case "puerto":
      case "protocolo":
      case "programa":
      case "id_ubicacion":
      case "estatus":
// Inicia case "menu"
      echo "Listado de servicios por $ARS_PARAM[0]";
      //echo "</font></td></tr><tr><td><font face=Arial size=\"2\">";
         $ARS_query = "select distinct $ARS_PARAM[0] from servicio
order by $ARS_PARAM[0]";
         $ARS_result = pg_query($ARS_query);
         if (!$ARS_result) {
             echo "Error al procesar: $ARS_query.\n";
             exit;
         }
         while ($ARS_arr = pg_fetch_array($ARS_result))
         {
            echo "</font></td></tr><tr><td><font face=Arial
size=\"2\">";
            echo "<b>$ARS_arr[0]</b>";
            echo "</font></td></tr><tr><td><font face=Arial size=2>";
            $ARS_query2 = "select * from servicio where
$ARS_PARAM[0]='$ARS_arr[0]'";
            $ARS_result2 = pg_query($ARS_query2);
            if (!$ARS_result2) {
                echo "Error al procesar: $ARS_query2.\n";
                exit;
            }
            echo "<ul>";
            while ($ARS_arr2 = pg_fetch_array($ARS_result2))
            {
               echo "<li>";
               echo "<b><a
href=\"servicios.php?ver=$ARS_arr2[id_servicio]\">";
               echo $ARS_arr2["nombre"];
               echo "</a></b>";
               echo " ( equipo <i>";
               echo $ARS_arr2["id_equipo"];
               echo "</i>, puerto <i>";
               echo $ARS_arr2["puerto"];
               echo "</i>, protocolo <i>";
               echo $ARS_arr2["protocolo"];
               echo "</i>, programa <i>";
```

```
                echo $ARS_arr2["programa"];
                echo "</i>, versi&oacute;n <i>";
                echo $ARS_arr2["version"];
                echo "</i> ) ";
                echo "<br>";
            }
            echo "</ul>";
            echo "</font></td></tr><tr><td
height=\"11\"> </td></tr>";
         }
            //echo "</font></td></tr><tr><td
height=\"11\"> </td></tr></tbody></table>";
            echo "</tbody></table>";
      break;
// Termina case "menu"
   default:
      echo "Listado de servicios por $ARS_PARAM[0]";
      echo "</font></td></tr><tr><td><font face=Arial size=\"2\">";
}
?>

    </td>
  </tr>
    <td width="155"> 
  </table>
</body>
</html>
```

Tabla 6-9. Código fuente del programa `servicios.php`.

6.1.2.6 Políticas.php

Este módulo despliega la lista de políticas que se encuentran dadas de alta en la base de datos, las cuales pueden ser organizadas mediante diversos criterios según lo requiera el usuario. Además, se puede tener acceso a la información detallada de cada política, incluyendo la descripción y definición de cada una. En la Figura 6-6 se puede observar la pantalla generada por este módulo.

Figura 6-6. Pantalla de politicas.

La información que utiliza este módulo se obtiene de acuerdo a la relación mostrada en la Tabla 6-10.

Información desplegada	Obtenida de	
	Tabla	***Campo***
Listado de políticas	Politica	Estatus Tipo Plataforma

Tabla 6-10. Relación del origen de datos para el módulo `politicas.php`.

A continuación, en la Tabla 6-11 se muestra el código fuente de este programa.

```
<html>
<head>
<title>inicio</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
</head>

<? include './barra1.php' ?>

      <td width="99%" height="19"><font face=Arial size=2><b><img
src="images/1politicas.jpg" width="105" height="18">

<? include './barra2.php' ?>

<table cellspacing=0 cellpadding=0 width="100%" border=0
summary="Tabla principal">
  <tbody>
  <tr>
    <td valign=top width=155>
```

```
      <div align="center"><font face="Arial, Helvetica, sans-serif"
size="2" color="#003399"><b>Menú</b>
<br>
<?
//Generar menú de opciones

//Obtener parámetro enviado
$ARS_QUERY=getenv("QUERY_STRING");
$ARS_PARAM=explode("=", $ARS_QUERY);
?>
<br>
<a href="politicas.php">ACTIVAS</a><br>
<a href="politicas.php?tipo">TIPO</a><br>
<a href="politicas.php?plataforma">PLATAFORMA</a><br>
<a href="politicas.php?estatus">ESTATUS</a><br>
</font>
    </td>
    <td valign=top width="649">

      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="Tabla de politicas">
        <tbody>
        <tr bgcolor="#004182">
          <td><font face="Arial" size="2" color="#FFFFFF">
<?
switch ($ARS_PARAM[0])
{
   case "":
// Inicia case ""
?>
      Listado de pol&iacute;ticas activas
      </font></td></tr><tr><td><font face=Arial size=2>
<?
      $ARS_query = "select * from politica where estatus='ACTIVA'";
      $ARS_result = pg_query($ARS_query);
      if (!$ARS_result) {
          echo "Error al procesar: $ARS_query.\n";
          exit;
      }
      echo "<ul>";
      while ($ARS_arr = pg_fetch_array($ARS_result))
      {
         echo "<li>";
         echo "<b><a
href=\"politicas.php?ver=$ARS_arr[id_politica]\">";
         echo $ARS_arr["nombre"];
         echo "</a></b>";
         echo " ( tipo <i>";
         echo $ARS_arr["tipo"];
         echo "</i>, plataforma <i>";
         echo $ARS_arr["plataforma"];
         echo "</i> ) ";
         echo "<br>";
      }
      echo "</ul>";
?>
      </font></td></tr><tr><td
```

```
height="11"> </td></tr></tbody></table>
<?
      break;
// Termina case ""
   case "ver":
// Inicia case "ver"
?>
      Pol&iacute;tica:
<?
      $ARS_query = "select * from politica where
id_politica='$ARS_PARAM[1]'";
      $ARS_result = pg_query($ARS_query);
      if (!$ARS_result) {
          echo "Error al procesar: $ARS_query.\n";
          exit;
      }
      $ARS_arr = pg_fetch_array($ARS_result);
      echo "<b>";
      echo $ARS_arr["nombre"];
      echo "</b>";
      echo " ( id ";
      echo $ARS_arr["id_politica"];
      echo ", ";
      echo $ARS_arr["estatus"];
      echo ", ";
      echo $ARS_arr["registrar_fecha"];
      echo " ) ";
?>
      </font></td></tr><tr><td><font face=Arial size=2>
<?
      echo "<ul>";
      echo "<li><i><b>Resumen:</b></i><br> ";
      echo $ARS_arr["resumen"];
      echo "<li><i><b>Plataforma:</b></i><br> ";
      echo $ARS_arr["plataforma"];
      echo "<li><i><b>Notas:</b></i><br>";
      echo $ARS_arr["notas"];
      echo "</ul>";
?>
      </font></td></tr><tr><td
height=\"11\"> </td></tr></tbody></table>
      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="Tabla de alertas">
        <tbody>
        <tr bgcolor="#005BB7">
          <td><font face="Arial" size="2" color="#FFFFFF">
      Descripci&oacute;n
</font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
      echo $ARS_arr["descripcion"];
?>
</font></td>
        </tr>
        <tr>
```

```
          <td height="5">  </td>
        </tr>
        </tbody>
      </table>
      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="Tabla de notificaciones" height="47">
        <tbody>
        <tr bgcolor="#0981D9">
          <td><font face=Arial size=2>
      Definici&oacute;n
</font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<form action="politicas.php" method="get">
<textarea name="definicion" rows="24" cols="80">
<?
      print $ARS_arr["definicion"];
?>
</textarea>
</form>
</font></td>
        </tr>
        <tr>
          <td>  </td>
        </tr>
        </tbody>
      </table>
<?
      break;
// Termina case "ver"
      case "tipo":
      case "plataforma":
      case "estatus":
// Inicia case "tipo|plataforma|estatus"
      echo "Listado de pol&iacute;ticas por $ARS_PARAM[0]";
      //echo "</font></td></tr><tr><td><font face=Arial size=\"2\">";
         $ARS_query = "select distinct $ARS_PARAM[0] from politica
order by $ARS_PARAM[0]";
         $ARS_result = pg_query($ARS_query);
         if (!$ARS_result) {
             echo "Error al procesar: $ARS_query.\n";
             exit;
         }
         while ($ARS_arr = pg_fetch_array($ARS_result))
         {
            echo "</font></td></tr><tr><td><font face=Arial
size=\"2\">";
            echo "<b>$ARS_arr[0]</b>";
            echo "</font></td></tr><tr><td><font face=Arial size=2>";
            $ARS_query2 = "select * from politica where
$ARS_PARAM[0]='$ARS_arr[0]'";
            $ARS_result2 = pg_query($ARS_query2);
            if (!$ARS_result2) {
                echo "Error al procesar: $ARS_query2.\n";
                exit;
            }
```

```
            echo "<ul>";
            while ($ARS_arr2 = pg_fetch_array($ARS_result2))
            {
               echo "<li>";
               echo "<b><a
href=\"politicas.php?ver=$ARS_arr2[id_politica]\">";
               echo $ARS_arr2["nombre"];
               echo "</a></b>";
               echo " ( tipo <i>";
               echo $ARS_arr2["tipo"];
               echo "</i>, plataforma <i>";
               echo $ARS_arr2["plataforma"];
               echo "</i> ) ";
               echo "<br>";
            }
            echo "</ul>";
            echo "</font></td></tr><tr><td
height=\"11\"> </td></tr>";
         }
            //echo "</font></td></tr><tr><td
height=\"11\"> </td></tr></tbody></table>";
            echo "</tbody></table>";
      break;
// Termina case "tipo|plataforma|estatus"
   default:
      echo "Listado de pol&iacute;ticas por $ARS_PARAM[0]";
      echo "</font></td></tr><tr><td><font face=Arial size=\"2\">";
}
?>

    </td>
  </tr>
    <td width="155"> 
  </table>
</body>
</html>
```

Tabla 6-11. Código fuente del programa `politicas.php`.

6.1.2.7 Equipos.php

Este módulo despliega la lista de los equipos dados de alta en la base de datos, los cuales pueden ser organizados mediante diversos criterios según lo requiera el usuario. Además, se puede tener acceso a la información detallada de cada equipo y se muestran las relaciones de servicios y políticas asociadas a cada equipo. La operación de este módulo es fundamental dentro de la solución propuesta, ya que a partir de los equipos se relaciona información diversa, como por ejemplo la ubicación física, los administradores, el soporte técnico y mensajes inherentes a cada equipo. En la Figura 6-7 se puede observar la pantalla generada por este módulo.

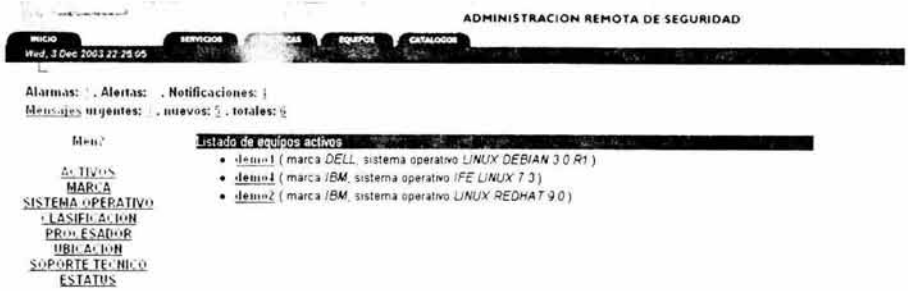


Figura 6-7. Pantalla de equipos.

La información que utiliza este módulo se obtiene de acuerdo a la relación mostrada en la Tabla 6-12.

Información desplegada	Obtenida de	
	Tabla	***Campo***
Listado de equipos	Equipo	Estatus Marca Sistema_operativo Clasificacion Procesador Ubicación Soporte_tecnico

Tabla 6-12. Relación del origen de datos para el módulo `equipos.php`.

A continuación, en la Tabla 6-13 se muestra el código fuente de este programa.

```
<html>
<head>
<title>Equipos</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-
1">
</head>

<? include './barra1.php' ?>

      <td width="99%" height="19"><font face=Arial size=2><b><img
src="images/lequipos.jpg" width="105" height="18">

<? include './barra2.php' ?>

<table cellspacing=0 cellpadding=0 width="100%" border=0
```

```
summary="Tabla principal">
  <tbody>
  <tr>
    <td valign=top width=155>
      <div align="center"><font face="Arial, Helvetica, sans-serif"
size="2" color="#003399"><b>Menú</b>
<br>
<?
//Generar menú de opciones

//Obtener parámetro enviado
$ARS_QUERY=getenv("QUERY_STRING");
$ARS_PARAM=explode("=", $ARS_QUERY);
?>
<br>
<a href="equipos.php">ACTIVOS</a><br>
<a href="equipos.php?marca">MARCA</a><br>
<a href="equipos.php?sistema_operativo">SISTEMA OPERATIVO</a><br>
<a href="equipos.php?clasificacion">CLASIFICACION</a><br>
<a href="equipos.php?procesador_marca">PROCESADOR</a><br>
<a href="equipos.php?id_ubicacion">UBICACION</a><br>
<a href="equipos.php?id_soporte_tecnico">SOPORTE TECNICO</a><br>
<a href="equipos.php?estatus">ESTATUS</a><br>
</font>
    </td>
    <td valign=top width="649">

      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="Tabla de politicas">
        <tbody>
        <tr bgcolor="#004182">
          <td><font face="Arial" size="2" color="#FFFFFF">
<?
switch ($ARS_PARAM[0])
{
   case "":
// Inicia case ""
?>
      Listado de equipos activos
      </font></td></tr><tr><td><font face=Arial size=2>
<?
      $ARS_query = "select * from equipo where estatus='ACTIVO'";
      $ARS_result = pg_query($ARS_query);
      if (!$ARS_result) {
          echo "Error al procesar: $ARS_query.\n";
          exit;
      }
      echo "<ul>";
      while ($ARS_arr = pg_fetch_array($ARS_result))
      {
         echo "<li>";
         echo "<b><a href=\"equipos.php?ver=$ARS_arr[id_equipo]\">";
         echo $ARS_arr["nombre"];
         echo "</a></b>";
         echo " ( marca <i>";
         echo $ARS_arr["marca"];
         echo "</i>, sistema operativo <i>";
```

```
         echo $ARS_arr["sistema_operativo"];
         echo "</i> ) ";
         echo "<br>";
      }
      echo "</ul>";
?>
      </font></td></tr><tr><td
height="11"> </td></tr></tbody></table>
<?
      break;
// Termina case ""
   case "ver":
// Inicia case "ver"
?>
      Equipo:
<?
      $ARS_query = "select * from equipo where
id_equipo='$ARS_PARAM[1]'";
      $ARS_result = pg_query($ARS_query);
      if (!$ARS_result) {
          echo "Error al procesar: $ARS_query.\n";
          exit;
      }
      $ARS_arr = pg_fetch_array($ARS_result);
      echo "<b>";
      echo $ARS_arr["nombre"];
      echo "</b>";
      echo " ( id ";
      echo $ARS_arr["id_equipo"];
      echo ", ";
      echo $ARS_arr["estatus"];
      echo ", ";
      echo $ARS_arr["registrar_fecha"];
      echo " ) ";
?>
      </font></td></tr><tr><td><font face=Arial size=2>
<?
      echo "<ul>";
      echo "<li><i><b>Sistema operativo:</b></i><br> ";
      echo $ARS_arr["sistema_operativo"];
      echo "<li><i><b>Procesador:</b></i><br> ";
      echo $ARS_arr["procesador_cantidad"];
      echo " ";
      echo $ARS_arr["procesador_marca"];
      echo " ";
      echo $ARS_arr["procesador_modelo"];
      echo ", 
      echo $ARS_arr["procesador_velocidad"];
      echo "<li><i><b>Memoria:</b></i><br>";
      echo $ARS_arr["memoria"];
      echo "<li><i><b>Clasificaci&oacute;n:</b></i><br> ";
      echo $ARS_arr["clasificacion"];
      echo "<li><i><b>Descripci&oacute;n:</b></i><br> ";
      echo $ARS_arr["descripcion"];
      echo "<li><i><b>Servicios activos:</b></i><br> ";
      $ARS_query2 = "select * from servicio where
id_equipo='$ARS_arr[id_equipo]' and estatus='ACTIVO'";
```

```
      $ARS_result2 = pg_query($ARS_query2);
      if (!$ARS_result2) {
          echo "Error al procesar: $ARS_query2.\n";
          exit;
      }
         echo "<ul>";
      while ($ARS_arr2 = pg_fetch_array($ARS_result2))
      {
         echo "<li><b><a
href=\"servicios.php?ver=$ARS_arr2[id_servicio]\">";
         echo $ARS_arr2["nombre"];
         echo "</a></b>";
         echo " ( puerto <i>";
         echo $ARS_arr2["puerto"];
         echo "</i>, protocolo <i>";
         echo $ARS_arr2["protocolo"];
         echo "</i> ) ";
         echo "<br>";
      }
         echo "</ul>";
//       echo "</ul>";

      echo "<li><i><b>Pol&iacute;ticas asociadas:</b></i><br> ";
      $ARS_plat = explode(" ", $ARS_arr[sistema_operativo]);
      //$ARS_plat_nwords=count($ARS_plat);
      //echo $ARS_plat_nwords;
      //echo ",";
      $ARS_plat_query[0]="TODAS";
      for ($i = 0; $i < 2; $i++)
      {
         $ARS_plat_query=array_merge($ARS_plat_query,$ARS_plat[$i]);
      }
      $ARS_opts=implode("|",$ARS_plat_query);

      $ARS_query3 = "select * from politica where estatus='ACTIVA' and
plataforma similar to '%($ARS_opts)%'";
      $ARS_result3 = pg_query($ARS_query3);
      if (!$ARS_result3)
          echo "Error al    cesar: $ARS_query3.\n";
          exit;
      }
         echo "<ul>";
      while ($ARS_arr3 = pg_fetch_array($ARS_result3))
      {
         echo "<li><b><a
href=\"politicas.php?ver=$ARS_arr3[id_politica]\">";
         echo $ARS_arr3["nombre"];
         echo "</a></b>";
         echo " ( tipo <i>";
         echo $ARS_arr3["tipo"];
         echo "</i>, plataforma <i>";
         echo $ARS_arr3["plataforma"];
         echo "</i> ) ";
         echo "<br>";
      }
         echo "</ul>";
      echo "<li><i><b>Direcci&oacute;n IP:</b></i><br> ";
```

```
      $ARS_query4 = "select host(direccion_ip),netmask(direccion_ip),*
from direccion_ip where id_equipo='$ARS_arr[id_equipo]' and
estatus='ACTIVA'";
      $ARS_result4 = pg_query($ARS_query4);
      if (!$ARS_result4) {
          echo "Error al procesar: $ARS_query4.\n";
          exit;
      }
         echo "<ul>";
      while ($ARS_arr4 = pg_fetch_array($ARS_result4))
      {
         echo "<li><b><a
href=\"mindterm.php?conectarse=$ARS_arr[nombre]\">";
         echo $ARS_arr4[0];
         echo "</a></b>";
         echo " (";
         echo "dispositivo <i>";
         echo $ARS_arr4["dispositivo"];
         echo "</i>, estatus <i>";
         echo $ARS_arr4["flags"];
         echo "</i>, m&aacute;scara <i>";
         echo $ARS_arr4[1];
         echo "</i>) ";
         echo "<br>";
      }
         echo "</ul>";

      echo "<li><i><b>Notas:</b></i><br>";
      echo $ARS_arr["notas"];
      echo "</ul>";
?>
      </font></td></tr><tr><td
height=\"11\"> </td></tr></tbody></table>
      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="datos">
        <tbody>
        <tr bgcolor="#005BB7">
          <td><font face="Arial" size="2" color="#FFFFFF">
      Datos adicionales
</font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
      echo "<ul>";
      echo "<li><i><b>N&uacute;mero de serie:</b></i><br> ";
      echo $ARS_arr["numero_serie"];
      echo "<li><i><b>N&uacute;mero de inventario:</b></i><br> ";
      echo $ARS_arr["numero_inventario"];
      echo "<li><i><b>Ubicaci&oacute;n:</b></i> ";
      $ARS_query2 = "select * from ubicacion where
id_ubicacion='$ARS_arr[id_ubicacion]'";
      $ARS_result2 = pg_query($ARS_query2);
      if (!$ARS_result2) {
          echo "Error al procesar: $ARS_query2.\n";
          exit;
      }
```

```
      $ARS_arr2 = pg_fetch_array($ARS_result2);
         echo $ARS_arr2["nombre"];
         echo "<ul>";
         echo "<li><i><b>Descripci&oacute;n:</b></i><br> ";
         echo $ARS_arr2["descripcion"];
         echo "<li><i><b>Calle:</b></i> ";
         echo $ARS_arr2["calle"];
         echo "<li><i><b>Colonia:</b></i> ";
         echo $ARS_arr2["colonia"];
         echo "<li><i><b>C&oacute;digo Postal:</b></i> ";
         echo $ARS_arr2["codigo_postal"];
         echo "<li><i><b>Ciudad:</b></i> ";
         echo $ARS_arr2["ciudad"];
         echo "<li><i><b>Estado:</b></i> ";
         echo $ARS_arr2["estado"];
         echo "<li><i><b>Pa&iacute;s:</b></i> ";
         echo $ARS_arr2["pais"];
         echo "</ul>";
      echo "<li><i><b>Ubicaci&oacute;n espec&iacute;fica:</b></i><br>
";
      echo $ARS_arr["ubicacion_especifica"];
      echo "</ul>";
?>
</font></td>
        </tr>
        <tr>
          <td height="5">  </td>
        </tr>
        </tbody>
      </table>
      <table cellspacing=0 cellpadding=0 width="100%" border=0
            summary="estatus" height="47">
        <tbody>
        <tr bgcolor="#0981D9">
          <td><font face=Arial size=2>
      Estatus
</font></td>
        </tr>
        <tr>
          <td><font face=Arial size=2>
<?
      echo "<ul>";
      echo "<li><i><b>&Uacute;ltima verificaci&oacute;n:</b></i><br> ";
      print $ARS_arr["ultima_verificacion"];
      echo "<li><i><b>&Uacute;ltimo cambio:</b></i><br> ";
      print $ARS_arr["ultimo_cambio"];
      echo "<li><i><b>Anomal&iacute;a m&aacute;s reciente:</b></i><br>
";
      print $ARS_arr["anomalia"];
      echo "<li><i><b>Actualizaciones pendientes desde:</b></i><br> ";
      print $ARS_arr["actualizaciones_pendientes"];
      echo "<li><i><b>Vulnerable desde:</b></i><br> ";
      print $ARS_arr["vulnerable"];
      echo "<li><i><b>Actividad sospechosa m&aacute;s
reciente:</b></i><br> ";
      print $ARS_arr["actividad_sospechosa"];
      echo "<li><i><b>Notas informativas:</b></i><br> ";
```

```
      print $ARS_arr["notas_informativas"];
      echo "</ul>";
?>
</font></td>
        </tr>
        <tr>
          <td>  </td>
        </tr>
        </tbody>
      </table>
<?
      break;
// Termina case "ver"
      case "marca":
      case "sistema_operativo":
      case "clasificacion":
      case "procesador_marca":
      case "id_ubicacion":
      case "id_soporte_tecnico":
      case "estatus":
// Inicia case "menu"
      echo "Listado de equipos por $ARS_PARAM[0]";
      //echo "</font></td></tr><tr><td><font face=Arial size=\"2\">";
         $ARS_query = "select distinct $ARS_PARAM[0] from equipo order
by $ARS_PARAM[0]";
         $ARS_result = pg_query($ARS_query);
         if (!$ARS_result) {
             echo "Error al procesar: $ARS_query.\n";
             exit;
         }
         while ($ARS_arr = pg_fetch_array($ARS_result))
         {
            echo "</font></td></tr><tr><td><font face=Arial
size=\"2\">";
            echo "<b>$ARS_arr[0]</b>";
            echo "</font></td></tr><tr><td><font face=Arial size=2>";
            $ARS_query2 = "select * from equipo where
$ARS_PARAM[0]='$ARS_arr[0]'";
            $ARS_result2 = pg_query($ARS_query2);
            if (!$ARS_result2) {
                echo "Error al procesar: $ARS_query2.\n";
                exit;
            }
            echo "<ul>";
            while ($ARS_arr2 = pg_fetch_array($ARS_result2))
            {
               echo "<li>";
               echo "<b><a
href=\"equipo.php?ver=$ARS_arr2[id_equipo]\">";
               echo $ARS_arr2["nombre"];
               echo "</a></b>";
               echo " ( marca <i>";
               echo $ARS_arr2["marca"];
               echo "</i>, sistema operativo <i>";
               echo $ARS_arr2["sistema_operativo"];
               echo "</i> ) ";
               echo "<br>";
```

```
            }
            echo "</ul>";
            echo "</font></td></tr><tr><td
height=\"11\"> </td></tr>";
         }
            //echo "</font></td></tr><tr><td
height=\"11\"> </td></tr></tbody></table>";
            echo "</tbody></table>";
      break;
// Termina case "menu"
   default:
      echo "Listado de equipos por $ARS_PARAM[0]";
      echo "</font></td></tr><tr><td><font face=Arial size=\"2\">";
}
?>

    </td>
  </tr>
    <td width="155"> 
  </table>
</body>
</html>
```

Tabla 6-13. Código fuente del programa `equipos.php`.

6.1.2.8 Conexión remota.

Dentro del módulo **`equipos.php`**, existe la funcionalidad para conectarse remotamente al equipo seleccionado, mediante una aplicación tipo "applet" que dentro del mismo navegador permite utilizar el protocolo SSH en sus versiones 1 y 2. Esta funcionalidad se proporciona para que de manera inmediata se tomen las acciones conducentes en caso de que el sistema monitoreado presente fallas.

Para tener acceso a esta aplicación, es necesario seleccionar el equipo al cual se desea conectar y elegir la dirección IP por la cual se realizará la conexión. Una vez hecho esto, arrancará la aplicación que se muestra en la Figura 6-8.

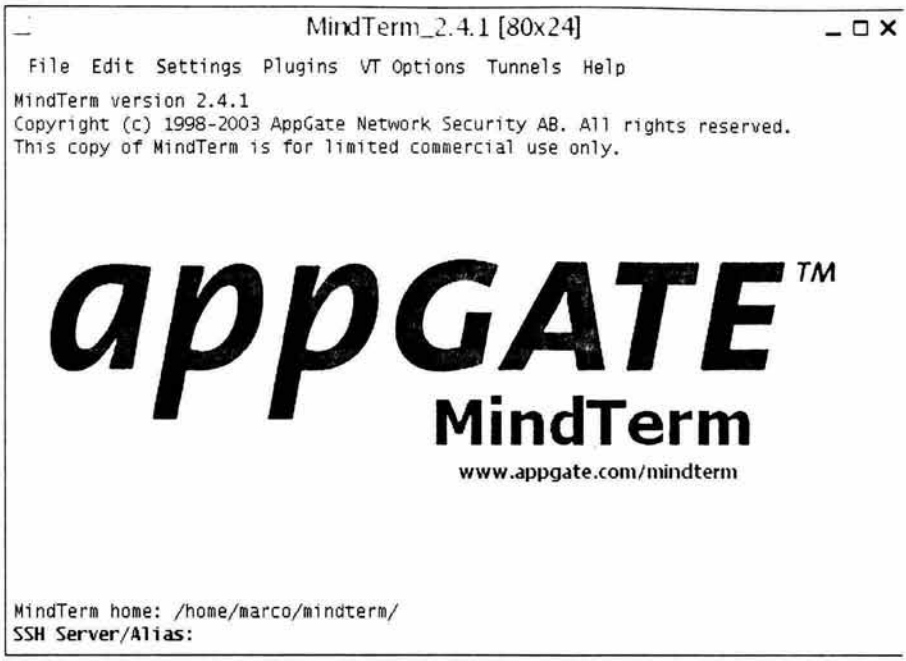


Figura 6-8. Aplicación para conexión remota.

Una vez arrancada, dicha aplicación solicita que se introduzca el nombre del equipo al que se desea conectar, así como también el usuario y contraseña necesarios para autenticarse en el equipo.

Cabe señalar que en los casos en que el equipo al que se desea conectar presente fallas irrecuperables, no podrá ser posible llevar a cabo la conexión remota y se tendrán que utilizar otros medios para llegar al equipo.

6.1.3 Esquema de funcionamiento.

La funcionalidad de la solución está sustentada en las relaciones que se establecen en la estructura de la base de datos descrita en el capítulo anterior, aunado a facilitar al usuario la navegación dentro de la aplicación.

6.1.3.1 Recolección y carga de datos.

La recolección y carga de datos se realiza mediante programas que funcionan de manera permanente en cada uno de los equipos, incluyendo el servidor que centraliza la información en la base de datos.

Uno de los principales programas para realizar el monitoreo y recolección de información en los equipos es cfengine, el cual funciona de manera proactiva apegándose a las políticas definidas en los archivos de configuración de la propia herramienta. De esta manera se logra que cada uno de los equipos tenga un comportamiento previsible y estable. En la Tabla 6-14 se puede observar un ejemplo básico de la definición de políticas para cfengine. En este ejemplo, y haciendo una descripción general, se puede comentar lo siguiente:

- Las líneas que inician con el símbolo #, representan comentarios.
- Se define el dominio de la red a la que pertenecen los equipos monitoreados.
- Se especifica el equipo que sirve correo electrónico.
- Se especifica el equipo que sirve la hora exacta.
- Se especifica la cuenta de correo del administrador encargado de los equipos.
- Se define el tamaño máximo de los archivos a editar.
- Se especifica que se verifiquen las políticas dos veces cada, la primera entre los minutos 0 y 5, la segunda entre los minutos 30 y 35 de cada hora.
- Se restringe la lista de usuarios permitidos para verificar políticas.
- Se especifica la secuenda de pasos que deberán ejecutarse para la aplicación de las políticas.
- Luego se definen cada uno de los pasos requeridos para la aplicación y verificación de las políticas. Cada una de las sentencias indicadas puede involucrar varias operaciones a realizar.

```
#############################################################
#
# cfagent.conf
#
#############################################################

control:

 # We probably need to define these so that cfexecd can mail the
 # output to a remote location, e.g. so that an MTA will
 # accept the routing of the mail outside the laptop.
 # Probably most users will not run local mail on their laptops
 # so we assume that problems get mailed to an external MTA

 domain     = ( domain.com )
 smtpserver = ( server.domain.com )
 timeserver = ( time.nist.gov )
 sysadm     = ( user@domain.com )
 editfilesize = ( 102400 )
 schedule   = ( Min00_05 Min30_35 )
 access = ( user root )

 actionsequence = ( shellcommands files editfiles tidy processes )

#########################################################

files:
```

```
 /var/cfengine/inputs owner=root mode=0600 action=fixall recurse=inf

 #
 # Some basic intrusion detection
 #

 Hr03::

 /usr mode=o-w checksum=md5 r=inf

#############################################################

shellcommands:

 #
 # Synchronize clocks
 #

 "/usr/sbin/ntpdate -u -s $(timeserver)"
# "/usr/bin/rdate -s $(timeserver)"
# "/bin/date"

#############################################################

tidy:

   /tmp pattern=* age=3
#   /etc pattern=route.conf age=0    # Corrupts SuSe's dhcpcd
#   /var/lib/dhcpcd/ pattern=* age=0 #   "  "

 Hr04::

   /home pattern=*~   age=5 r=inf
   /home pattern=core age=0 r=inf

#############################################################

 #
 # Security - you can never trust installation programs
 #

editfiles:

 { /etc/services

 HashCommentLinesContaining "login"
 HashCommentLinesContaining "talk"
 HashCommentLinesContaining "telnet"
 HashCommentLinesContaining "finger"
 HashCommentLinesContaining "time"
 AppendIfNoSuchLine "cfengine    5308/tcp                                #
CFengine"
 AppendIfNoSuchLine "cfengine    5308/udp                                #
CFengine"
 }

#############################################################
```

```
processes:

 "cfenvd"  restart "/usr/local/sbin/cfenvd"
 "cfexecd$"  restart "/usr/local/sbin/cfexecd"

 server::
    "inetd"  signal=hup
    "cfservd" restart "/usr/local/sbin/cfservd"
```

Tabla 6-14. Ejemplo básico de la definición de políticas para cfengine.

Cabe señalar que adicional a la aplicación cfengine, se utilizan otros programas desarrollados para cumplir con la función de obtener información, transferirla al equipo donde se encuentra la base de datos y cargar dicha información.

Por otra parte, la información recolectada por dichos programas se carga en la base de dato. En Tabla 6-15 se lista la relación de tablas y campos afectados de la base de datos, para los cuales la información se obtiene y carga de manera automática, mediante programas.

TABLA	*CAMPOS*
Equipo	Procesador_cantidad Procesador_marca Procesador_modelo Procesador_velocidad Memoria Sistema_operativo Ultima_verificacion Ultimo_cambio Anomalia Actualizaciones_pendientes Vulnerable Actividad_sospechosa
Direccion_ip	Dispositivo Direccion_ip Direccion_mac Broadcast Máscara
Servicio	Puerto Protocolo Programa Versión
Bitácora	Tipo_evento Mensaje
Inactividad	Fecha_reinicio Tiempo_inactividad

	Tipo_falla Nivel_criticidad
Vulnerabilidad	Cve Tipo Resumen Fecha_reportada Palabras_clave Descripción

Tabla 6-15. Relación de tablas y campos que son actualizados automáticamente.

6.2 Implantación.

En esta sección se describen los elementos a considerar para la implantación de la solución propuesta en este trabajo. De antemano, se asume que se tiene instalado y configurado el software mencionado en el punto 6.1.1 de este capítulo, así como también los programas propios de la aplicación desarrollada.

6.2.1 Consideraciones y logística.

Se debe definir un servidor central en el cual se debe instalar el software necesario para el funcionamiento de la aplicación propuesta, el cual deberá tener como mínimo las siguientes características:

- Procesador: Intel Pentium II a 300 MHz
- Memoria: 256 MB
- Espacio disponible en disco duro: 40 GB
- Tarjeta de red: ethernet 10/100 Mbits/seg
- (Opcional) Dispositivo de respaldo: CD-RW, DVD+RW o unidad de cinta.

En este servidor se deberá crear la base de datos que utilizará la aplicación yl deberá contar, además del software ya mencionado en este capítulo, con un servicio de correo electrónico que puedan utilizar los diversos programas involucrados en la recolección de información.

Una vez configurado el servidor central, se deberá proceder a la instalación del sistema cfengine en todos y cada uno de los equipos que se requiera dar de alta en el sistema.

Posterior a la instalación del sistema cfengine en los equipos a monitorear, se deben dar de alta dichos equipos en la base de datos empleando para esto la propia aplicación. Adicional a la información básica de los equipos, se debe dar de alta de forma manual la información relacionada a los administradores, soporte técnico y ubicación física de los equipos.

A partir de que se haya realizado lo anterior, los programas empezarán a realizar cargas automáticas a la base de datos, con lo que se mantendrá actualizada la información de los equipos monitoreados.

Conviene mencionar que el esquema de actualización de los equipos, empleado por cfengine, es tipo "pull" en lugar del tradicional esquema "push", es decir, las actualizaciones se llevan a cabo por solicitud de los clientes de acuerdo a su configuración, por lo cada cliente es responsable de mantenerse actualizado, en lugar de que desde el servidor se estén lanzando las actualizaciones.

Este esquema aporta diversas ventajas relacionadas con la conectividad de los equipos, ya que en ocasiones pueden estar desconectados, por lo que llevar el control centralmente de todas las actualizaciones resulta demasiado complicado, no así el caso de que cada equipo lleve su propio control y tome de las actualizaciones lo que le haga falta.

Un aspecto que es muy importante de considerar, es la creación de un laboratorio de prueba antes de instalar de manera productiva la solución propuesta, ya que debido a que se encuentran involucradas diversas herramientas de software, conviene familiarizarse y entender el funcionamiento de cada una de dichas herramientas y posteriormente entender la manera en que operan en conjunto. Esto facilitará la solución de los problemas que puedan presentarse una vez que se realice el despliegue de la solución a un ambiente productivo.

Para configurar y modificar la funcionalidad de las herramientas es necesario familiarizarse con la documentación de cada una. Además, antes de realizar algún cambio, es necesario entender perfectamente el impacto que tendrá en la solución de una manera integral. No es conveniente llevar a cabo cambios sin saber exactamente las repercusiones que tendrá.

6.2.2 Procedimientos de instalación y operación.

Cada una de las herramientas requeridas cuenta con suficiente información acerca de la instalación y configuración. Además existen listas de correo y grupos de discusión en cada una, por lo que es importante visitar los URLs indicados en el punto 6.1.1 para obtener mayor información de cada una de las herramientas mencionadas.

En relación a los programas desarrollados en este trabajo, lo único que se requiere es instalar los programas PHP en el directorio del servidor de web que se desee. Cabe señalar que para que los programas funciones, se debe tener configurado el servidor de web para el procesamiento de archivos con extensión php.

Por lo demás, la operación de la aplicación es bastante sencilla ya que está basada en web, como se comentó anteriormente. Entre las funciones básicas se

encuentran el envío de mensajes entre los diferentes administradores de los equipos, el reporte de anomalías en los equipos o servicios, la consulta de las políticas asociadas a cada equipo y el mantenimiento de catálogos.

Cabe destacar que desde la misma aplicación web es posible ejecutar un applet para conectarse remotamente, mediante una terminal que utiliza secure shell, a los equipos que se requiera. Este applet funciona asociado a la dirección IP de cada uno de los equipos. Para hacer uso de esta funcionalidad, y con el objeto de que dicho applet no se conecta arbitrariamente a cualquier equipo de la red, es necesario dar de alta en el servidor central cada equipo al que se requiera conectar remotamente en el archivo /etc/hosts, de esta manera se tiene control de los equipos a los que puede conectarse el applet.

6.2.2.1 Ejemplo del procedimiento de instalación.

Para el caso específico de instalar la aplicación en un equipo con sistema operativo Linux Red Hat 9, se deben ejecutar las siguientes instrucciones.

1. Verificar que se tengan instalados los siguientes paquetes:
   - Apache
     Verifica que esté instalado el paquete nombrado *httpd*, versión 2.0 o superior, utilizando el comando:
     # rpm –qi httpd
   - OpenSSL
     Verifica que esté instalado el paquete nombrado *openssl*, versión 0.9.7a o superior, utilizando el comando:
     # rpm –qi openssl
   - Perl
     Verifica que esté instalado el paquete nombrado *perl*, versión 5.8 o superior, utilizando el comando:
     # rpm –qi perl
   - PHP
     Verifica que esté instalado el paquete nombrado *php*, versión 4 o superior, utilizando el comando:
     # rpm –qi php
   - Postgresql
     Verifica que esté instalado el paquete nombrado *postgresql*, versión 7.3 o superior, utilizando el comando:
     # rpm –qi postgresql
   - CVS
     Verifica que esté instalado el paquete nombrado *cvs*, versión 1.11 o superior, utilizando el comando:
     # rpm –qi cvs
   - OpenSSH
     Verifica que esté instalado el paquete nombrado *openssh*, versión 3.6 o superior, utilizando el comando:

 # rpm –qi openssh

- RPM
  Verifica que esté instalado el paquete nombrado *rpm*, versión 4.2 o superior, utilizando el comando:
  # rpm –qi rpm

2. En caso de que alguno de los programas mencionados no se encuentre instalado, puede obtenerse el RPM de dicho programa en el sitio RPMFind (http://www.rpmfind.net/), haciendo la búsqueda por el nombre del programa y seleccionando el paquete que corresponda para la versión del sistema operativo que utilizamos.

3. Por otra parte, se deben obtener e instalar los programas cfengine y mindterm, de acuerdo a los requisitos comentados en el punto 6.1.1 de este capítulo.

4. Cabe reiterar la conveniencia de familiarizarse a fondo con cada uno de los programas y herramientas comentadas para que se facilite la instalación y configuración de cada una. En cada una de las herramientas existe documentación relativa a la instalación, configuración y uso de cada una.

5. Crear la base de datos.
   # su – postgres
   $ createdb consola

6. Crear usuario de la base de datos y asignar permisos.
   $ createuser dbuser
   $ psql consola
   =# grant all on consola to dbuser;
   =# \q

7. Ejecutar script de creación de base de datos con el usuario creado. Este script está listado en el apéndice de este documento.
   $ psql consola dbuser < script_db.sql > salida.log 2>&1

8. Verificar que la base de datos se haya creado correctamente, revisar el contenido del archivo salida.log e ingresar a la base de datos.
   $ more salida.log
   $ psql consola dbuser
   =# \dt
   =# \q

9. Se deben crear los siguientes directorios:
   # mkdir /var/www/html/consola
   # mkdir /var/www/html/consola/interfase

10. Los programas PHP desarrollados en este trabajo, deben depositarse en la siguiente ruta:
/var/www/html/consola/interfase

11. Verificar que el archivo `/var/www/html/consola/interfase/dbconn.php` tenga la configuración de acuerdo a la instalación realizada.

12. Verificar que se tenga acceso a la aplicación desde la consola de la máquina en la cual se instaló, utilizando un navegador web y el siguiente URL:
http://localhost/consola/interfase

Una vez concluidos los pasos anteriores, se tendrá acceso a la aplicación y podrá utilizarse de acuerdo a las funcionalidades comentadas anteriormente.

6.2.3 Consideraciones para el mantenimiento.

Parte fundamental de cualquier sistema, es el mantenimiento que se le debe brindar para mantener la adecuada funcionalidad y confiabilidad, por lo que es conveniente que de manera periódica se revise con los usuarios de la aplicación los requerimientos de funcionalidad o adecuación de cada uno de los módulos que conforman la aplicación, así como también que de manera periódica se lleven a cabo actualizaciones del software que forma parte de la solución.

En relación con el funcionamiento propio de la aplicación, se recomienda revisar de manera periódica el crecimiento de la tabla *bitácora*, ya que de acuerdo al funcionamiento de la herramienta, puede crecer rápidamente, afectando el espacio utilizado en disco duro. Otras tablas que también conviene revisar, aunque con lapsos más espaciados son *actualización*, *vulnerabilidad* y *mensaje*.

Asimismo, es conveniente que el servidor central también sea dado de alta en el propio sistema con el fin de que también se encuentre monitoreado por la solución propuesta y puedan identificarse problemas en dicho equipo.

Como parte del presente trabajo, se entrega el código fuente de los programas desarrollados, el cual servirá de base para llevar a cabo el mantenimiento de la aplicación. Por otra parte, a través de los URLs proporcionados del software incorporado, se pueden obtener las actualizaciones de cada aplicación.

Capítulo 7. Evaluación, resultados y conclusiones.

Como respaldo a las conclusiones de este trabajo de tesis, se incluyen en este capítulo la evaluación y resultados obtenidos en diversos aspectos de la solución prouesta.

Adicionalmente, y como parte de las conclusiones se indica el trabajo que falta por desarrollar para consolidar la solución propuesta como una herramienta más robusta y con mejores características.

7.1 Evaluación.

Como parte de las conclusiones del trabajo desarrollado, se evalúan diversos factores relacionados con las características de la aplicación desarrollada. Esta evaluación, permite identificar el impacto real que tiene la herramienta en un ambiente productivo.

Los comparativos de la herramienta se realizan contra el trabajo cotidiano y manual que lleva a cabo un administrador de sistemas, o el encargado de la seguridad informática de cualquier organización en la actualidad, a quienes está orientada la herramienta y se consideran "*usuarios*" de la aplicación.

7.1.1 Factores a evaluar.

A continuación se presentan los principales factores considerados en la evaluación, que mediante su valoración y análisis, permitieron llegar a una conclusión más precisa acerca del alcance que puede tener la implantación de la herramienta, producto de este trabajo.

7.1.1.1 Funcionalidad.

Aquí se engloban las características propias de la herramienta, identificando las cualidades que ofrece y alineando las particularidades de la misma a las funciones que desempeñan los adminsitradores de sistemas y seguridad informática en cualquier organización.

7.1.1.2 Utilidad.

Este rubro se orienta a identificar cuáles son las necesidades que satisface la herramienta, independientemente de sus características, es decir, qué aspectos de la herramienta resultan ventajosos para los usuarios.

7.1.1.3 Efectividad.

En este punto, se concentra uno de los factores más complejos de evaluar, ya que se debe determinar el grado de cumplimiento de la herramienta para cubrir las necesidades de los usuarios de la aplicación.

7.1.1.4 Seguridad.

Un aspecto fundamental, es tener la certeza que la utilización de la herramienta no representará un riesgo adicional en materia de seguridad informática para los sistemas monitoreados.

7.1.1.5 Facilidad de uso.

La apariencia y facilidad de uso, representan un punto básico para que los usuarios se sientan familiarizados con la herramienta y la utilicen con frecuencia, de lo contrario, será un factor adverso cuando se quiera implantar de manera productiva.

7.1.2 Resultados.

En esta sección se comentan los resultados de cada uno de los factores evaluados.

7.1.2.1 Funcionalidad.

En este aspecto se han tomado en cuenta las siguientes características:

- La herramienta permite mostrar las condiciones de seguridad informática en una red de equipos de cómputo y comunicaciones monitoreados.
- La navegación por la aplicación es fluida y el usuario tiene la información que necesita a su alcance en todo momento.
- En caso de ser requerido, el administrador de los equipos puede conectarse de manera segura mediante una sesión remota.
- Se puede tener acceso al detalle de las características de un equipo en particular así como, a las condiciones de seguridad específicas en que se encuentra.
- Se pueden generar reportes ordenados por diversos criterios.

7.1.2.2 Utilidad.

En este rubro se tiene lo siguiente:

- Los administradores de los equipos pueden tener una vista global de las condiciones de seguridad informática que guardan los sistemas monitoreados en todo momento.

- Los administradores cuentan con un esquema documentado y actualizado de la situación en que se encuentran sus equipos, lo que facilita priorizar las tareas que se necesitan realizar.
- El monitoreo puede realizarse prácticamente desde cualquier parte, con el único requisito de contar con conexión a red y navegador web.
- Es posible la interacción entre administradores que comparten responsabilidades en algunos equipos, lo cual aumenta la comunicación y permite la coordinación asíncrona de esfuerzos para la toma de decisiones o conclusión de tareas específicas.
- Se pueden tener diversas vistas de la información presentada, lo que permite hacer relaciones y facilita la comprensión de dicha información.
- No solo se presenta información de los equipos sino también de los servicios proporcionados, lo que permite una visión más completa de los sistemas monitoreados.

7.1.2.3 Efectividad.

Como resultado de la evaluación, encontramos que la herramienta cubre un amplio rango de las características deseables para la administración remota de la seguridad, entre los aspectos específicos de este rubro destacan los siguientes:

- La herramienta permite obtener información de manera inmediata del estado que guardan los equipos monitoreados, con lo cual es posible tomar acciones inmediatas, lo que redunda en una mayor disponibilidad y confiabilidad de los servicios informáticos de la organización.
- Los administradores de sistemas mantienen documentado, de manera electrónica, el estado que guarda cada equipo.
- La herramienta permite una mejor coordinación entre los integrantes de grupos de administración.
- Al tener la información relativa a los equipos disponible en la herramienta, se evitan búsquedas manuales en papel u otros medios de información, con lo cual se ahorra tiempo valioso al momento en que se presenta algún incidente.
- A través de la herramienta se pueden consultar las políticas que tienen configurados cada uno de los equipos, lo cual permite conocer de manera integral el funcionamiento de los sistemas así como, facilitar la adición, modificación o eliminación de políticas.
- La herramienta integra diversas características ya comentadas que resultan muy útiles y que agilizan las labores a los administradores de sistemas
- Se tiene la ventaja de contar con el código fuente de la aplicación, lo que permite adecuar la herramienta a necesidades muy particulares o que se puedan cubrir requerimientos futuros.

7.1.2.4 Seguridad.

En relación a este rubro, tanto en el diseño como en la construcción e implantación de la herramienta se tomaron las siguientes consideraciones:

- La aplicación está desarrollada en un ambiente web, utilizando Secure Socket Layer (SSL).
- La transferencia de información a la base de datos se realiza mediante una conexión a nivel local del equipo en el cual residen el servidor web y la base de datos, además de que dicha conexión es autenticada con usuario y contraseña y el acceso se encuentra restringido sólo al mismo equipo.
- Se tienen implementados esquemas de autenticación en la aplicación, la base de datos y los diversos programas involucrados.
- La conexión remota a los equipos se realiza a través de Secure SHell (SSH).
- La información que de manera directa utiliza la aplicación se encuentra almacenada en un solo equipo.

Adicionalmente, la aplicación integra herramientas que permiten el monitoreo de sistemas en aspectos de integridad de archivos, identificación de vulnerabilidades, actualización de software y revisión de bitácoras.

Con esto, se logra tener una solución para llevar a cabo la administración remota de la seguridad en los servidores conectados en red, lo que proporciona las siguientes ventajas:

- Uniformidad en la aplicación de políticas en los sistemas.
- Estandarización en el uso de herramientas.
- Control de la actualización de software en los sistemas.
- Automatización de procedimientos de administración.
- Monitoreo local y análisis centralizado de la información recolectada.
- Administración escalable a un gran número de sistemas.

7.1.2.5 Facilidad de uso.

En cuanto a este punto se tomaron en cuenta los siguientes aspectos:

- La interfase del usuario está basada en web.
- La navegación es consistente y se realiza mediante pestañas y menús.
- Se presentan interrelaciones entre los diferentes elementos para facilitar la navegación.
- Se tienen accesos directos a la información más utilizada.
- El uso de colores permite una mejor comprensión de la información.

7.1.2.6 Ventajas.

A partir del análisis realizado en los puntos anteriores se obtuvieron las siguientes ventajas:

- La herramienta posibilita integrar información diversa en un solo punto, lo que permite conocer la situación que guardan todos los equipos monitoreados.
- Los usuarios tienen facilidad para obtener la información que necesitan para tomar decisiones y solucionar problemas en tiempo y forma.
- Toda la operación de la herramienta se realiza sobre esquemas de acceso y autenticación seguros.
- La aplicación fue desarrollada en lenguajes de programación de amplia utilización y se proporciona el código fuente para realizar ajustes a la medida.
- La implantación de la solución se realiza mediante la utilización de programas de software libre, lo que elimina la dependencia de software propietario.
- El acceso a la aplicación es mediante interfase web, por lo que se puede utilizar sobre cualquier plataforma de cómputo y desde cualquier lugar.

7.1.2.7 Desventajas.

No obstante las ventajas comentadas, es necesario mencionar que también se cuenta con las siguientes desventajas.

- Falta realizar desarrollo y mejoramiento de módulos que le permitan a la aplicación proporcionar más y mejores funcionalidades.
- Se requiere mantenimiento manual para diversos catálogos, por lo cual es necesario el desarrollo de interfases para conectar a otros sistemas de los cuales se obtenga la información.
- Es necesario capacitar a los usuarios en aspectos metodológicos de administración de sistemas y seguridad e implantación de las herramientas utilizadas.
- Se necesita involucrar a los desarrolladores de software internos de la organización para que se encarguen del mantenimiento de la solución, tanto para corrección de problemas como para desarrollar nuevas funcionalidades.

7.2 Conclusiones.

El uso de computadoras en diferentes actividades al interior de las organizaciones se ha incrementado a niveles sin precedentes, por lo cual la administración de sistemas en general y la seguridad informática de manera particular ha cobrado una gran importancia para que las organizaciones puedan proporcionar servicios confiables, que generen ahorros en tiempo y costos.

En fechas recientes el tema de seguridad informática ha tomado mayor importancia, porque cada vez se depende más de los sistemas de cómputo para

llevar a cabo tareas críticas relacionadas íntimamente al negocio propio de cada organización.

Los sistemas de cómputo se vuelven más complejos en relación a que integran más elementos de equipo y programas, a su vez, cada sistema informático consta de más líneas de código para soportar las funcionalidades requeridas, lo que ocasiona que la probabilidad de que se presenten fallas se incremente, requiriéndose la implantación de controles y revisiones más exhaustivos en las liberaciones de sistemas y monitoreo de los mismos.

Adicionalmente, la expansión de las redes de cómputo han permitido que los equipos de procesamiento de datos puedan ubicarse físicamente en cualquier parte del planeta, por lo que los retos para la administración de la seguridad de dichos equipos también se amplían, requiriéndose el desarrollo de nuevas formas para organizar y adaptar la manera en que se lleva a cabo la operación cotidiana de los sistemas de cómputo.

Por lo anterior y para soportar de manera adecuada la demanda de servicios informáticos de alta disponibilidad e integridad que demandan las organizaciones, es necesario conjuntar herramientas de apoyo para que los administradores de sistemas puedan identificar problemas y tomar decisiones proactivas y reactivas de manera rápida y efectiva.

Como propuesta de solución a la problemática expuesta, en este trabajo se han presentado los elementos que manifiestan la necesidad de incorporar mejores esquemas de seguridad en las organizaciones así como, la inclusión de una metodología y una herramienta para afrontar los retos en materia de la administración remota de la seguridad de los sistemas informáticos.

Al evaluar los diversos factores analizados es posible apreciar la obtención de resultados favorables en el uso de la herramienta desarrollada, por lo que, en razón de los motivos expuestos resulta que la aplicación de una herramienta como la presentada, permitirá la identificación y solución de problemas de una manera más ágil y rápida en los equipos y sistemas de una organización.

La aplicación de una herramienta como la que presenta este trabajo, resulta útil en diversos ambientes de cómputo, desde redes pequeñas hasta ambientes empresariales que involucren una gran cantidad de equipo de cómputo y comunicaciones.

El uso de software libre como base en el desarrollo de la solución propuesta, proporciona las siguientes ventajas:

1. Independencia a las organizaciones para la implantación de las herramientas que integran la solución;
2. Libertad para llevar a cabo ajustes que permitan cubrir las necesidades particulares;

3. Estabilidad de la solución y optimización de los recursos con que cuenta la organización, ya que se pueden ajustar cada uno de los componentes de manera precisa;
4. Automatización de tareas de administración de sistemas y seguridad informática; y
5. Escalabilidad e independencia de plataformas de cómputo, sistemas operativos, manejadores de bases de datos o requerimientos especiales.

Entre las ventajas del uso del software libre respecto al software propietario, se puede comentar que:

1. Se obtienen ahorros en el costo total de propiedad de la herramienta al compararse con soluciones similares basadas en software propietario, ya que estas últimas por lo general tienen un precio de venta y una vez adquirido el producto deben cubrirse los costos de actualización y soporte técnico de la herramienta de manera permanente, lo que genera grandes desembolsos de dinero a las organizaciones; y
2. Al no contar con el código fuente de la herramienta propietaria se crea una gran dependencia de la organización con el fabricante de la herramienta en aspectos de la solución de problemas y la incorporación de nuevas características específicas para la organización.

De acuerdo a la descripción y evaluación llevada a cabo a través de todo el trabajo, se concluye la conveniencia de incorporar metodología y herramientas basadas en software libre que permitan unificar la administración remota de la seguridad informática en una organización.

7.2.1 Trabajo por desarrollar.

Los conceptos y el desarrollo de la herramienta que conforman este proyecto, pretenden mostrar las posibilidades que existen para lograr la implantación de una administración integral de la seguridad de sistemas de cómputo. A partir de este trabajo pueden desarrollarse más y mejores programas que integren aspectos no cubiertos en este proyecto.

La herramienta se presenta como un prototipo, sobre el cual se debe seguir desarrollando para cubrir todas las funcionalidades que una herramienta de este tipo requiere, además de incorporar características deseables en aspectos de navegación y obtención de información desde el punto de vista del usuario.

Algunos de los proyectos que se sugieren desarrollar a partir de éste, son los siguientes:

- Desarrollar módulos adicionales que permitan mejorar la colaboración entre los diversos administradores.

- Mejorar la presentación de información proporcionada mediante vistas configurables.
- Integrar otros módulos de programas que alimenten con más información a la base de datos.
- Mejorar el mantenimiento de catálogos.
- Incorporar un módulo de correlación de eventos y seguimiento a problemas.
- Desarrollar interfases con otros sistemas que permitan disminuir el mantenimiento manual de catálogos.
- Integrar un módulo mediante el cual se obtengan reportes del uso de la aplicación, indicando las acciones realizadas por los usuarios.

REFERENCIAS

[1] Mann, Scott. Mitchell, Ellen L. *Linux System Security*. USA. Prentice Hall PTR. 2000.

[2] Fisch, Eric A. White, Gregory B. *Secure Computers and Networks*. USA. CRC Press LLC. 2000.

[3] Allen, Julia. Kossakowski, Klaus-Peter. Ford, Gary. Konda, Suresh. Simmel, Derek. *Securing Network Servers*. Carnegie Mellon University, Software Engineering Institute. USA. 2000. http://www.sei.cmu.edu/pub/documents/sims/pdf/sim010.pdf.

[4] Power, Richard. *2002 CSI/FBI Computer Crime and Security Survey*. Computer Security Issues & Trends. Computer Security Institute. USA. 2002. Vol. VIII, No. 1. http://www.gocsi.com/press/20020407.html.

[5] Wack, John. Tracy, Miles. *Guideline on Network Security Testing*. National Institute of Standards and Technology. USA. 2001. http://csrc.nist.gov/publications/drafts/security-testing.pdf.

[6] Mell, Peter. Tracy, Miles C. *Procedures for Handling Security Patches*. National Institute of Standards and Technology. USA. 2002. http://csrc.nist.gov/publications/drafts/draft800-40.pdf.

[7] Jones Telecommunications and Multimedia Encyclopedia. *Computers: History and Development*. Jones Digital Century, Inc. 1999. http://www.digitalcentury.com/encyclo/update/comp_hd.html.

[8] Howe, Denis. *Free On-Line Dictionary Of Computing*. UK. 2002. http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?query=system+administration.

[9] Schneier, Bruce. *Secrets & Lies: Digital Security In A Networked World*. USA. Wiley Computer Publishing. 2000.

[10] Huggins, James S. *First Computer Bug*. 2002. http://www.jamesshuggins.com/h/tek1/first_computer_bug.htm.

[11] Darmohray, Tina. *Job Descriptions For System Administrators*. SAGE, The System Administrators Guild. 2002. http://www.usenix.org/sage/publications/jobdesc.html.

[12] Miller, Hall. *SAGE – Code Of Ethics*. SAGE, The System Administrators Guild. 2002. http://www.usenix.org/sage/publications/code_of_ethics.html.

[13] ISO 17799 Information Security Group. *The ISO 17799 Directory*. UK. 2002. http://www.iso-17799.com/.

[14] Security Risk Associates. *ISO 17799 Standard: ISO17799 Compliance & Positioning*. 2001. http://www.securityauditor.net/iso17799/.

[15] Schneier, Bruce. *Fixing Network Security by Hacking the Business Climate*. USA. 2002. http://www.counterpane.com/presentation4.pdf.

[16] Shaheen, Firas. *Tools, Tools, and TOOLS!!* SANS Institute. USA. 2001. http://rr.sans.org/tools/tools.php.

[17] The CERT Coordination Center (CERT/CC). *List of Security Tools*. Carnegie Mellon University, Software Engineering Institute. USA. 2001. http://www.cert.org/tech_tips/security_tools.html.

[18] Schneier, Bruce. *Applied Cryptography, 2nd Edition*. USA. John Wiley & Sons, Inc. 1996.

[19] McNamara, Joel. *The Complete, Unofficial TEMPEST Information Page*. 2002. http://www.eskimo.com/~joelm/tempest.html.

[20] CES Communications Ltd. *DES – the Data Encryption Standard*. 2002. http://www.cescomm.co.nz/encryption/des.html.

[21] CES Communications Ltd. *AES- the Advanced Encryption Standard*. 2002. http://www.cescomm.co.nz/encryption/aes.html.

[22] RSA Security Inc. *Cryptography FAQ*. 2002. http://www.rsasecurity.com/rsalabs/faq/sections.html.

[23] Litzau, David. *Risk Management: A Foundation for Information Security*. 2001. http://rr.sans.org/audit/risk_manage.php.

[24] Yazar, Zeki. *A Qualitative Risk Analysis and Management Tool – CRAMM*. 2002. http://rr.sans.org/audit/CRAMM.php.

[25] Fenzi, Kevin. Wreski, Dave. *Linux Security HOWTO*. 2002. http://www.tldp.org/HOWTO/Security-HOWTO/index.html.

[26] Melymuka, Kathleen. *The Evolution of IT Leader*. Computerworld. 2002. http://www.computerworld.com/printthis/2002/0,4814,74679,00.html.

[27] Moore, Andrew P. Ellison, Robert J. Linger, Richard C. *Attack Modeling for Information Security and Survivability*. Carnegie Mellon University, Software Engineering Institute. USA. 2001. http://www.sei.cmu.edu/publications/documents/01.reports/01tn001.html.

[28] Internet Software Consortium. *Internet Domain Survey*. 2002 (julio). http://www.isc.org/ds/WWW-200207/index.html.

[29] CERT Coordination Center. *CERT/CC Statistics 1988-2002*. Carnegie Mellon University, Software Engineering Institute. 2002. http://www.cert.org/stats/.

[30] The SANS Institute. *Security Policy Project*. 2002-2003. http://www.sans.org/resources/policies/.

[31] The Federal Computer Incident Response Center. *Firewalls and Guards*. 2001. http://www.fedcirc.gov/docs/firewalls.html.

[32] The SANS Institute. *Intrusion Detection FAQ*. 2002. http://www.sans.org/resources/idfaq/.

[33] Control Data. *Why Security Policies Fail*. 1999. http://downloads.securityfocus.com/library/Why_Security_Policies_Fail.pdf.

[34] Herzog, Pete.. *OSSTMM - Open Source Security Testing Methodology Manual*. Institute for Security and Open Methodologies 2002. http://www.isecom.org/projects/osstmm.htm.

[35] Symantec. *Integrated Security: Creating the Secure Enterprise*. 2002. http://www.fedcirc.gov/docs/Integrated_Security_WP.pdf.

[36] Instituto Nacional de Estadísticas, Geografía e Informática. *XII Censo General de Población y Vivienda, 2000*. http://www.inegi.gob.mx/difusion/espanol/fpobla.html.

[37] US Census Bureau. *Home Computers and Internet Use in the United Status: August 2000*. http://www.census.gov/Press-Release/www/2001/cb01-147.html.

[38] School of Information Management and Systems. *How Much Information? Project*. University of California in Berkeley. 2000. http://www.sims.berkeley.edu/research/projects/how-much-info/internet.html.

[39] Forrester Research, Inc. *Forrester Report On eBusiness*. 2001. http://www.sims.berkeley.edu/research/projects/how-much-info/internet.html.

[40] Marianne, McGee. *It's Oficial: IT Adds Up*. Information Week (04/17/00) No. 782. 2002.

[41] McGraw, Gary. *Internet Security: Issues and Trends*. Cigital. 2002. http://www.cigital.com/presentations/witi/sld001.htm

[42] Cochran, Shannon. *The Rising Costs of Software Complexity*. Dr. Dobb's Journal. Abril 2001. http://www.ddj.com/documents/s=868/ddj0104n/0104n.htm.

[43] Whittaker, James A. Voas, Jeffrey M. *50 years of Software: Key Principles for Quality*. Software Quality Management Magazine. Vol. 3, No. 1. Enero 2003. http://www.sqmmagazine.com/issues/2003-01/50years.html.

[44] Jarmon, David. *A Preparation Guide to Information Security Policies*. System Administration and Network Security. 2002. http://www.sans.org/rr/policy/prep_guide.php.

ANEXO A. Código SQL para la creación de la base de datos.

```
CREATE TABLE actualizacion (
       id_actualizacion     VARCHAR(20) NOT NULL,
       id_equipo            VARCHAR(20) NOT NULL,
       id_ubicacion         VARCHAR(20) NOT NULL,
       id_soporte_tecnico   VARCHAR(20) NOT NULL,
       id_servicio          VARCHAR(20) NOT NULL,
       nombre               VARCHAR(40) NOT NULL,
       version              VARCHAR(20) NOT NULL,
       release              VARCHAR(20) NOT NULL,
       fecha_creacion       TIMESTAMP WITH TIME ZONE NOT NULL,
       fecha_instalacion    TIMESTAMP WITH TIME ZONE NOT NULL,
       act_size             VARCHAR(20) NOT NULL,
       resumen              VARCHAR(90) NOT NULL,
       descripcion          TEXT NOT NULL,
       registar_fecha       TIMESTAMP WITH TIME ZONE NOT NULL,
       estatus              VARCHAR(20) NOT NULL,
       notas                TEXT NULL
);


ALTER TABLE actualizacion
       ADD PRIMARY KEY (id_actualizacion, id_equipo, id_ubicacion,
              id_soporte_tecnico, id_servicio);


CREATE TABLE administrador (
       id_ubicacion         VARCHAR(20) NOT NULL,
       id_administrador     VARCHAR(20) NOT NULL,
       titulo               VARCHAR(20) NULL,
       apellido_paterno     VARCHAR(40) NULL,
       apellido_materno     VARCHAR(40) NULL,
       nombre               VARCHAR(40) NULL,
       puesto               VARCHAR(90) NULL,
       area_especifica      VARCHAR(90) NULL,
       adscripcion          VARCHAR(90) NULL,
       organizacion         VARCHAR(90) NULL,
       jefe_inmediato       VARCHAR(90) NULL,
       asistente            VARCHAR(90) NULL,
       registar_fecha       TIMESTAMP WITH TIME ZONE NOT NULL,
       estatus              VARCHAR(20) NOT NULL,
       notas                TEXT NULL
);


ALTER TABLE administrador
       ADD PRIMARY KEY (id_ubicacion, id_administrador);


CREATE TABLE bitacora (
```

```
        id_bitacora             VARCHAR(20) NOT NULL,
        id_equipo               VARCHAR(20) NOT NULL,
        id_ubicacion            VARCHAR(20) NOT NULL,
        id_soporte_tecnico      VARCHAR(20) NOT NULL,
        id_servicio             VARCHAR(20) NOT NULL,
        tipo_evento             VARCHAR(40) NULL,
        mensaje                 TEXT NULL,
        registar_fecha          TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus                 VARCHAR(20) NOT NULL,
        notas                   TEXT NULL
);


ALTER TABLE bitacora
        ADD PRIMARY KEY (id_bitacora, id_equipo, id_ubicacion,
                id_soporte_tecnico, id_servicio);


CREATE TABLE correlacion (
        id_correlacion          VARCHAR(20) NOT NULL,
        id_equipo               VARCHAR(20) NOT NULL,
        id_ubicacion            VARCHAR(20) NOT NULL,
        id_soporte_tecnico      VARCHAR(20) NOT NULL,
        id_servicio             VARCHAR(20) NOT NULL,
        id_politica             VARCHAR(20) NOT NULL,
        id_inactividad          VARCHAR(20) NOT NULL,
        id_vulnerabilidad       VARCHAR(20) NOT NULL,
        nombre                  VARCHAR(40) NOT NULL,
        tipo                    VARCHAR(20) NOT NULL,
        resumen                 VARCHAR(90) NOT NULL,
        descripcion             TEXT NOT NULL,
        dependencia             VARCHAR(20) NULL,
        registrar_fecha         TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus                 VARCHAR(20) NOT NULL,
        notas                   TEXT NULL
);


ALTER TABLE correlacion
        ADD PRIMARY KEY (id_correlacion, id_equipo, id_ubicacion,
                id_soporte_tecnico, id_servicio, id_politica,
                id_inactividad, id_vulnerabilidad);


CREATE TABLE correo_electronico (
        id_correo_electronico VARCHAR(20) NOT NULL,
        id_administrador        VARCHAR(20) NOT NULL,
        id_soporte_tecnico      VARCHAR(20) NOT NULL,
        id_ubicacion            VARCHAR(20) NOT NULL,
        correo_electronico      VARCHAR(90) NULL,
        tipo                    VARCHAR(20) NULL,
        registrar_fecha         TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus                 VARCHAR(20) NOT NULL,
        notas                   TEXT NULL
);
```

```
ALTER TABLE correo_electronico
       ADD PRIMARY KEY (id_correo_electronico, id_administrador,
              id_soporte_tecnico, id_ubicacion);


CREATE TABLE direccion_ip (
       id_direccion_ip      VARCHAR(20) NOT NULL,
       id_equipo            VARCHAR(20) NOT NULL,
       id_ubicacion         VARCHAR(20) NOT NULL,
       id_soporte_tecnico   VARCHAR(20) NOT NULL,
       dispositivo          VARCHAR(20) NOT NULL,
       direccion_ip         INET NOT NULL,
       direccion_mac        MACADDR NOT NULL,
       flags                VARCHAR(40) NOT NULL,
       registrar_fecha      TIMESTAMP WITH TIME ZONE NOT NULL,
       estatus              VARCHAR(20) NOT NULL,
       notas                TEXT NULL
);


ALTER TABLE direccion_ip
       ADD PRIMARY KEY (id_direccion_ip, id_equipo, id_ubicacion,
              id_soporte_tecnico);


CREATE TABLE equipo (
       id_equipo            VARCHAR(20) NOT NULL,
       id_ubicacion         VARCHAR(20) NOT NULL,
       id_soporte_tecnico   VARCHAR(20) NOT NULL,
       nombre               VARCHAR(40) NOT NULL,
       descripcion          VARCHAR(90) NOT NULL,
       marca                VARCHAR(40) NULL,
       modelo               VARCHAR(40) NULL,
       procesador_cantidad  VARCHAR(20) NULL,
       procesador_marca     VARCHAR(40) NULL,
       procesador_modelo    VARCHAR(40) NULL,
       procesador_velocidad VARCHAR(20) NULL,
       memoria              VARCHAR(20) NULL,
       numero_serie         VARCHAR(40) NULL,
       numero_inventario    VARCHAR(40) NULL,
       notas_informativas   TEXT NULL,
       ubicacion_especifica VARCHAR(90) NULL,
       sistema_operativo    VARCHAR(40) NOT NULL,
       clasificacion        VARCHAR(40) NULL,
       ultima_verificacion  TIMESTAMP WITH TIME ZONE NULL,
       ultimo_cambio        TIMESTAMP WITH TIME ZONE NULL,
       anomalia             TIMESTAMP WITH TIME ZONE NULL,
       actualizaciones_pendientes TIMESTAMP WITH TIME ZONE NULL,
       vulnerable           TIMESTAMP WITH TIME ZONE NULL,
       actividad_sospechosa TIMESTAMP WITH TIME ZONE NULL,
       registrar_fecha      TIMESTAMP WITH TIME ZONE NOT NULL,
       estatus              VARCHAR(20) NOT NULL,
       notas                TEXT NULL
);
```

```
ALTER TABLE equipo
       ADD PRIMARY KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico);


CREATE TABLE equipo_administrador (
       id_equipo             VARCHAR(20) NOT NULL,
       id_ubicacion          VARCHAR(20) NOT NULL,
       id_administrador      VARCHAR(20) NOT NULL,
       id_soporte_tecnico    VARCHAR(20) NOT NULL
);


ALTER TABLE equipo_administrador
       ADD PRIMARY KEY (id_equipo, id_ubicacion, id_administrador,
              id_soporte_tecnico);


CREATE TABLE inactividad (
       id_inactividad        VARCHAR(20) NOT NULL,
       id_equipo             VARCHAR(20) NOT NULL,
       id_ubicacion          VARCHAR(20) NOT NULL,
       id_soporte_tecnico    VARCHAR(20) NOT NULL,
       id_servicio           VARCHAR(20) NOT NULL,
       fecha_reinicio        TIMESTAMP WITH TIME ZONE NOT NULL,
       tiempo_inactividad    INTERVAL NOT NULL,
       tipo_falla            VARCHAR(40) NOT NULL,
       nivel_criticidad      VARCHAR(20) NOT NULL,
       solucion              TEXT NOT NULL,
       atendido_por          VARCHAR(20) NOT NULL,
       registrar_fecha       TIMESTAMP WITH TIME ZONE NOT NULL,
       estatus               VARCHAR(20) NOT NULL,
       notas                 TEXT NULL
);


ALTER TABLE inactividad
       ADD PRIMARY KEY (id_inactividad, id_equipo, id_ubicacion,
              id_soporte_tecnico, id_servicio);


CREATE TABLE mensaje (
       id_mensaje            VARCHAR(20) NOT NULL,
       id_ubicacion          VARCHAR(20) NOT NULL,
       id_administrador      VARCHAR(20) NOT NULL,
       tipo                  VARCHAR(40) NOT NULL,
       mensaje               TEXT NOT NULL,
       destinatario          VARCHAR(20) NOT NULL,
       registrar_fecha       TIMESTAMP WITH TIME ZONE NOT NULL,
       estatus               VARCHAR(20) NOT NULL,
       notas                 TEXT NULL
);


ALTER TABLE mensaje
```

```
        ADD PRIMARY KEY (id_mensaje, id_ubicacion, id_administrador);


CREATE TABLE politica (
        id_politica             VARCHAR(20) NOT NULL,
        nombre                  VARCHAR(40) NOT NULL,
        tipo                    VARCHAR(20) NOT NULL,
        plataforma              VARCHAR(40) NOT NULL,
        resumen                 VARCHAR(90) NOT NULL,
        descripcion             TEXT NOT NULL,
        definicion              TEXT NOT NULL,
        registrar_fecha         TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus                 VARCHAR(20) NOT NULL,
        notas                   TEXT NULL
);


ALTER TABLE politica
        ADD PRIMARY KEY (id_politica);


CREATE TABLE politica_equipo (
        id_politica             VARCHAR(20) NOT NULL,
        id_equipo               VARCHAR(20) NOT NULL,
        id_ubicacion            VARCHAR(20) NOT NULL,
        id_soporte_tecnico      VARCHAR(20) NOT NULL
);


ALTER TABLE politica_equipo
        ADD PRIMARY KEY (id_politica, id_equipo, id_ubicacion,
                id_soporte_tecnico);


CREATE TABLE respaldo (
        id_respaldo             VARCHAR(20) NOT NULL,
        id_equipo               VARCHAR(20) NOT NULL,
        id_ubicacion            VARCHAR(20) NOT NULL,
        id_soporte_tecnico      VARCHAR(20) NOT NULL,
        fecha                   TIMESTAMP WITH TIME ZONE NOT NULL,
        tipo_respaldo           VARCHAR(40) NOT NULL,
        registrar_fecha         TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus                 VARCHAR(20) NOT NULL,
        notas                   TEXT NULL
);


ALTER TABLE respaldo
        ADD PRIMARY KEY (id_respaldo, id_equipo, id_ubicacion,
                id_soporte_tecnico);


CREATE TABLE servicio (
        id_servicio             VARCHAR(20) NOT NULL,
        id_equipo               VARCHAR(20) NOT NULL,
        id_ubicacion            VARCHAR(20) NOT NULL,
```

```
        id_soporte_tecnico    VARCHAR(20) NOT NULL,
        nombre                VARCHAR(40) NOT NULL,
        descripcion           VARCHAR(90) NOT NULL,
        puerto                VARCHAR(20) NOT NULL,
        protocolo             VARCHAR(40) NOT NULL,
        programa              VARCHAR(40) NOT NULL,
        version               VARCHAR(90) NOT NULL,
        registrar_fecha       TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus               VARCHAR(20) NOT NULL,
        notas                 TEXT NULL
);


ALTER TABLE servicio
        ADD PRIMARY KEY (id_servicio, id_equipo, id_ubicacion,
                id_soporte_tecnico);


CREATE TABLE soporte_tecnico (
        id_soporte_tecnico    VARCHAR(20) NOT NULL,
        nombre                VARCHAR(90) NOT NULL,
        registrar_fecha       TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus               VARCHAR(20) NOT NULL,
        notas                 TEXT NULL
);


ALTER TABLE soporte_tecnico
        ADD PRIMARY KEY (id_soporte_tecnico);


CREATE TABLE telefono (
        id_ubicacion          VARCHAR(20) NOT NULL,
        id_administrador      VARCHAR(20) NOT NULL,
        id_telefono           VARCHAR(20) NOT NULL,
        id_soporte_tecnico    VARCHAR(20) NOT NULL,
        telefono              VARCHAR(20) NOT NULL,
        tipo                  VARCHAR(20) NOT NULL,
        registrar_fecha       TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus               VARCHAR(20) NOT NULL,
        notas                 TEXT NULL
);


ALTER TABLE telefono
        ADD PRIMARY KEY (id_ubicacion, id_administrador,
                id_telefono, id_soporte_tecnico);


CREATE TABLE ubicacion (
        id_ubicacion          VARCHAR(20) NOT NULL,
        nombre                VARCHAR(90) NOT NULL,
        descripcion_general   VARCHAR(90) NOT NULL,
        calle                 VARCHAR(90) NOT NULL,
        colonia               VARCHAR(90) NOT NULL,
        codigo_postal         VARCHAR(20) NOT NULL,
```

```
        ciudad                  VARCHAR(90) NOT NULL,
        estado                  VARCHAR(90) NOT NULL,
        pais                    VARCHAR(90) NOT NULL,
        registrar_fecha         TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus                 VARCHAR(20) NOT NULL,
        notas                   TEXT NULL
);


ALTER TABLE ubicacion
        ADD PRIMARY KEY (id_ubicacion);


CREATE TABLE vulnerabilidad (
        id_vulnerabilidad       VARCHAR(20) NOT NULL,
        id_equipo               VARCHAR(20) NOT NULL,
        id_ubicacion            VARCHAR(20) NOT NULL,
        id_soporte_tecnico      VARCHAR(20) NOT NULL,
        id_servicio             VARCHAR(20) NOT NULL,
        cve                     VARCHAR(40) NOT NULL,
        tipo                    VARCHAR(40) NOT NULL,
        resumen                 VARCHAR(90) NOT NULL,
        descripcion             TEXT NOT NULL,
        fecha_reportada         TIMESTAMP WITH TIME ZONE NOT NULL,
        palabras_clave          VARCHAR(90) NOT NULL,
        registrar_fecha         TIMESTAMP WITH TIME ZONE NOT NULL,
        estatus                 VARCHAR(20) NOT NULL,
        notas                   TEXT NULL
);


ALTER TABLE vulnerabilidad
        ADD PRIMARY KEY (id_vulnerabilidad, id_equipo, id_ubicacion,
                id_soporte_tecnico, id_servicio);


ALTER TABLE actualizacion
        ADD FOREIGN KEY (id_servicio, id_equipo, id_ubicacion,
                id_soporte_tecnico)
                        REFERENCES servicio;


ALTER TABLE actualizacion
        ADD FOREIGN KEY (id_equipo, id_ubicacion,
                id_soporte_tecnico)
                        REFERENCES equipo;


ALTER TABLE administrador
        ADD FOREIGN KEY (id_ubicacion)
                        REFERENCES ubicacion;


ALTER TABLE administrador
        ADD FOREIGN KEY (id_ubicacion, id_administrador)
                        REFERENCES administrador;
```

```
ALTER TABLE bitacora
       ADD FOREIGN KEY (id_servicio, id_equipo, id_ubicacion,
              id_soporte_tecnico)
                            REFERENCES servicio;


ALTER TABLE bitacora
       ADD FOREIGN KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico)
                            REFERENCES equipo;


ALTER TABLE correlacion
       ADD FOREIGN KEY (id_vulnerabilidad, id_equipo, id_ubicacion,
              id_soporte_tecnico, id_servicio)
                            REFERENCES vulnerabilidad;


ALTER TABLE correlacion
       ADD FOREIGN KEY (id_inactividad, id_equipo, id_ubicacion,
              id_soporte_tecnico, id_servicio)
                            REFERENCES inactividad;


ALTER TABLE correlacion
       ADD FOREIGN KEY (id_politica)
                            REFERENCES politica;


ALTER TABLE correlacion
       ADD FOREIGN KEY (id_servicio, id_equipo, id_ubicacion,
              id_soporte_tecnico)
                            REFERENCES servicio;


ALTER TABLE correlacion
       ADD FOREIGN KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico)
                            REFERENCES equipo;


ALTER TABLE correo_electronico
       ADD FOREIGN KEY (id_soporte_tecnico)
                            REFERENCES soporte_tecnico;


ALTER TABLE correo_electronico
       ADD FOREIGN KEY (id_ubicacion, id_administrador)
                            REFERENCES administrador;


ALTER TABLE direccion_ip
       ADD FOREIGN KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico)
```

```
                        REFERENCES equipo;


ALTER TABLE equipo
       ADD FOREIGN KEY (id_soporte_tecnico)
                        REFERENCES soporte_tecnico;


ALTER TABLE equipo
       ADD FOREIGN KEY (id_ubicacion)
                        REFERENCES ubicacion;


ALTER TABLE equipo_administrador
       ADD FOREIGN KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico)
                        REFERENCES equipo;


ALTER TABLE equipo_administrador
       ADD FOREIGN KEY (id_ubicacion, id_administrador)
                        REFERENCES administrador;


ALTER TABLE inactividad
       ADD FOREIGN KEY (id_servicio, id_equipo, id_ubicacion,
              id_soporte_tecnico)
                        REFERENCES servicio;


ALTER TABLE inactividad
       ADD FOREIGN KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico)
                        REFERENCES equipo;


ALTER TABLE mensaje
       ADD FOREIGN KEY (id_ubicacion, id_administrador)
                        REFERENCES administrador;


ALTER TABLE politica_equipo
       ADD FOREIGN KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico)
                        REFERENCES equipo;


ALTER TABLE politica_equipo
       ADD FOREIGN KEY (id_politica)
                        REFERENCES politica;


ALTER TABLE respaldo
       ADD FOREIGN KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico)
                        REFERENCES equipo;
```

```
ALTER TABLE servicio
       ADD FOREIGN KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico)
                            REFERENCES equipo;


ALTER TABLE telefono
       ADD FOREIGN KEY (id_soporte_tecnico)
                            REFERENCES soporte_tecnico;


ALTER TABLE telefono
       ADD FOREIGN KEY (id_ubicacion, id_administrador)
                            REFERENCES administrador;


ALTER TABLE telefono
       ADD FOREIGN KEY (id_ubicacion)
                            REFERENCES ubicacion;


ALTER TABLE vulnerabilidad
       ADD FOREIGN KEY (id_servicio, id_equipo, id_ubicacion,
              id_soporte_tecnico)
                            REFERENCES servicio;


ALTER TABLE vulnerabilidad
       ADD FOREIGN KEY (id_equipo, id_ubicacion,
              id_soporte_tecnico)
                            REFERENCES equipo;
```