INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY
CAMPUS ESTADO DE MÉXICO

"CERTIFICACIÓN DEL NIVEL DE SEGURIDAD EN APLICACIONES"

TESIS QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACIÓN
PRESENTA

HUGO CORTÉS MONROY

Asesor: DR. JOSÉ DE JESÚS VÁZQUEZ GÓMEZ

Comité de tesis: DR. ROBERTO GÓMEZ CÁRDENAS

M. C. ALFONSO ESPARZA BETANCOURTH

Jurado:	DR. ROBERTO GÓMEZ CÁRDENAS	Presidente
	DR. JOSÉ DE JESÚS VÁZQUEZ GÓMEZ	Vocal
	M. C. ALFONSO ESPARZA BETANCOURTH	Secretario

Atizapán de Zaragoza, Edo. Méx., Mayo de 2002.

RESUMEN

El presente documento tiene como objetivo establecer una métrica para determinar el nivel de seguridad de una aplicación. Se ilustra la hipótesis mediante un caso de estudio; dicho caso es Jaguar, un servidor de aplicaciones que se desea implantar para realizar las transacciones cotidianas que se realizan en una institución financiera.

Para ello realizaremos un recorrido conceptual por los aspectos más importantes sobre seguridad computacional, las necesidades de seguridad informática, la situación computacional actual en nuestro entorno y el mundo, los organismos de certificación de seguridad informática (corrientes americanas y europeas), intentos de estandarización en lo que al tema de certificación de seguridad refieren, mejores prácticas de desarrollo, nuevas metodologías de implementación de sistemas, dispositivos de ataque y defensa, etc.

Crear una guía para la validación de aplicaciones es tema central durante esta tesis, ya que ésta, nos permitirá atribuirle un nivel de seguridad a la aplicación en turno evaluada. Se realiza un mapeo de dicha guía de verificación al caso de estudio, con la finalidad de aplicar de manera práctica la hipótesis planteada, así como vislumbrar trabajos futuros y propuestas de mejora.

Hay que reconocer que la rápida evolución de los productos y entornos, introducen la posibilidad de que no todas las características de seguridad sean evaluadas en el tiempo que originalmente se estableció para la certificación.

LISTA DE FIGURAS

LISTA DE TABLAS

ABREVIATURAS Y SÍMBOLOS

Abreviatura	Significado
IT	Information Technology
PP	Protection Profile
ST	Security Target
CC	Common Criteria
ITSEC	Information Technology Security Evaluation Criteria
TCSEC	Trusted Computer System Evaluation Criteria
OSSTMM	Open-Source Security Testing Methodology Manual
TOE	Target of Evaluation
TNI	Trusted Network Interpretation
Hw	Hardware
Sw	Software
TCB	Trusted Computer Base
ACL	Access Control List
MAC	Control de acceso mandatorio
CLEF	Comercial Licensed Evaluation Facilities
CB	Certification Body
EAL	Evaluation Assurance Level
IDS	Intrusion Detection System
PBX	Private Branch Exchange
PAC	Certificado de Atributos de Privilegio
CA	Autoridad Certificadora
RACF	Resource Access Control Facility
DCE	Data Communications Equipment
GSS	General Security Services

API	Applications Program Interface
PKI	Public Key Infraestructure
SCOMP	Secure Communications Processor
HTML	Hyper Text Markup Language
SSL	Secure Socket Layer
OLTP	Procesamiento de Transacciones intensivas en línea
CTS	Component Transaction Server
DB	Data Base
EAS	Enterprise Aplication Server
CORBA	Common Object Request Broker Architecture
OMG	Object Management Group
COM	Component Object Model
ORB	Object Request Broker
JCM	Java Connection Management
CPU	Central Processor Unit
IOP	Referencia de Objeto interoperable
IPSec	Seguridad en IP
DES	Data Encryption System
IDEA	International Data Encryption Algorithm
RC4	Rivest Cipher
RSA	Rivest Shamir Adlemn
HTTP	Hyper Text Transfer Protocol
VPN	Virtual Private Network
FIFO	Fist Input First Output
DoS	Denial of Service
IIOP	Internet Inter Orb Protocol

1 INTRODUCCIÓN

Se ha dado en considerar a la computadora como la maravilla del siglo XX. A medida que ha emergido de sus años de infancia, la computadora ha ido asumiendo responsabilidades de trabajo cada vez mayores hasta convertirse, hoy día, en el caballo de batalla de nuestros negocios. Hospitales, empresas, agencias gubernamentales, de transportación, financieras y en general cualquier tipo de organización, usan aplicaciones en computadoras para registrar y mantener el control de sus activos informáticos, como son: nóminas, inventarios y otros valores. No sabemos hasta donde llegará la importancia del uso de la computadora en la sociedad. Lo que sí sabemos es que su función será cada vez más importante, igualmente importante será entonces, protegerlas de fallas, catástrofes, criminales, etc. Sin ellas, empresas y gobiernos podrían quedar totalmente varados. [1]

Se puede decir que durante las dos pasadas décadas, el procesamiento de datos se ha convertido en el centro nervioso de la comunidad financiera. La automatización de actividades cotidianas dentro del mundo financiero ha provocado una dependencia de la computadora, y las instituciones han encontrado en ella una herramienta que simplifica, acelera y da confiabilidad a las operaciones. A medida que esta dependencia se incrementa, también lo hace el riesgo financiero asociado a una pérdida de la capacidad de procesamiento.

Además, un factor más reciente que afecta la seguridad en forma creciente, es la introducción de sistemas distribuidos y el uso de redes e instalaciones de comunicaciones para transmitir datos entre terminales de usuario y computadoras y entre computadoras y computadoras en Internet o localmente usando los protocolos de la red de Internet.

Por lo anterior, no es exagerado decir, que el significado real de ***seguridad*** en las computadoras es la *supervivencia*: por un lado, la supervivencia de una operación vital dentro de una organización, y por otro lado la supervivencia de la empresa misma. Sin embargo, esta fuerte dependencia rara vez es notada por aquellas personas que no tienen contacto directo con el procesamiento de la información a través de las computadoras, o con el desarrollo de las

aplicaciones que llevan al cabo este procesamiento, aún estando dentro de la organización. Las consecuencias de una pérdida de la información del negocio, muchas veces no se consideran sino hasta que ocurre un desastre. La mayoría de la gente no piensa en esta posibilidad. Sin embargo, los desastres ocurren, y por eso las empresas deben estar preparadas para tratar de evitar al máximo su ocurrencia o para minimizar las pérdidas causadas por ellos. La presencia o ausencia de medidas de seguridad de la información es el factor más importante para que una empresa se vea o no impactada por una eventualidad negativa. De manera similar, la presencia o ausencia de un plan de contingencia es el factor más importante para que una empresa sobreviva o desaparezca después de un desastre. Las compañías que desarrollan sus planes de contingencia, que diseñan procedimientos de emergencia y de recuperación, que entrenan a su personal en estos procedimientos, y que los prueban y efectúan simulacros, tienen mayor probabilidad de sobrevivir a los desastres.

Actualmente en México, como en el resto de los países del mundo, el gran incremento en el manejo de la información a través de redes de computadoras, ha implicado la protección de éstos valiosos recursos en su totalidad, cuidando su integridad, disponibilidad y confidencialidad. No debemos olvidar que en un mundo globalizado, como el que se vive, la información es un recurso que tiene un valor incalculable y es de una importancia crítica y vital para el funcionamiento y competencia de toda organización productiva y de servicios. Una de las principales preocupaciones de los administradores de sistemas y de los responsables de la seguridad informática, es que el usuario, en su mayoría, desconoce la inseguridad que priva en las redes cuando realiza el envío y recepción de información, y durante el manejo que de ella hace en su máquina.

Como primera medida, se recomienda el uso de antivirus.

Otro aspecto que resulta relevante para la protección de nuestra información, es que, resulta indispensable estar al día, no sólo en lo relacionado con los avances tecnológicos para tal efecto, sino que se hace necesario conocer, cuáles son los adelantos en nuestro país, en lo relativo a legislación en materia informática y a las sanciones en materia de daños a los sistemas y a la información contenida y transportada por este medio.

Es conveniente hacer notar que con el fin de desincentivar a las personas que pretendieran llevar al cabo conductas que pudieran dañar los recursos mencionados anteriormente, se realizaron reformas a diversas disposiciones en materia penal, las cuales fueron publicadas en el Diario Oficial de la Federación el 17 de mayo de 1999, entre las que destaca lo relacionado a los delitos informáticos. Por ejemplo; "la persona que modifique, destruya, o provoque la pérdida de información se hace acreedora a una sanción penal". Esa sanción puede ir de seis meses a dos años de prisión y no sólo eso, además puede llevar aparejada una sanción que va de cien a trescientos días de salario mínimo como multa.

Por otro lado, la información respaldada por productos o sistemas que utilizan tecnologías de información (IT) es una fuente válida, que permite que las organizaciones logren su misión. Adicionalmente, particulares tienen una sensata expectación sobre si su información personal contenida en productos o sistemas IT permanece segura, sobre si estará disponible cuando se requiera de dicha información y si no va estar sujeta a modificación no autorizada. Los productos o sistemas IT deben realizar sus funciones mientras que se ejercita el control apropiado de la información, para asegurarla se protege contra peligros tales como difusión, alteración, o pérdida

indeseada o injustificable. El término seguridad se utiliza para cubrir la prevención y la mitigación de estos y de peligros similares.

Muchos consumidores de IT carecen del conocimiento o los recursos necesarios para juzgar si su confianza en la seguridad de los productos o sistemas con tecnología en la información, es apropiada, y pueden no desear confiar solamente en las aseveraciones de los desarrolladores. Los consumidores pueden por lo tanto elegir aumentar su confianza en las medidas de seguridad de productos o sistemas, pidiendo un análisis de su seguridad (es decir una evaluación de seguridad).

En la actualidad, existe un potencial de mercado ilimitado para un certificador o proveedor, ya sea de información segura o de productos que brindan características seguras.

La mayoría de organizaciones y compañías, en cualquier ramo de la industria, ahora dependen de la tecnología de la información, para una adecuada gestión de sus negocios.

En Europa, la IT es esencial para los negocios que desean desarrollar y mantener sus ventajas competitivas. Pero también existen desventajas, las cuales aparecen como amenazas sobre la seguridad de la información, que antes no existían. Así, podemos mencionar que la mayoría de los sistemas actuales debe de ser diseñados para operar sobre una red, los programas que se desarrollan "reutilizando" muchas veces códigos que no fueron verificados por la organización. La seguridad de una organización, ya no sólo depende de sus esfuerzos por fortalecer su infraestructura, sino que ahora *depende de qué tan seguros son los sistemas vecinos en su entorno*. Así la mayoría de las veces, la seguridad puede ser un punto difícil de afrontar, pero que tampoco se debe ignorar.

<<La seguridad en la tecnología de la información es la oportunidad de los mercadólogos y certificadores>> [1]

La mayoría de las compañías necesitan conservar la *integridad* (prevención sobre la modificación no autorizada), *confidencialidad* (prevención sobre la divulgación no autorizada) y *disponibilidad* (prevención sobre la retención no autorizada de información o recursos) de su información. La seguridad informática busca cubrir estas necesidades, además de cuidar aspectos como el control de acceso y la autenticación de los usuarios.

El término "seguridad" es muy amplio, e incluye varios aspectos. Si la información no esta disponible en un tiempo razonable para la gente que la necesita, será casi como si la información hubiese sido alterada o robada. La disponibilidad puede ser fácilmente afectada por una inadecuada seguridad, y más sí la mala intención esta involucrada. Cada día, se incrementa la necesidad por la seguridad de la información, al aumentar también, las compañías que manejan sistemas distribuidos. La seguridad de estos vulnerables sistemas, depende cada vez menos de los procedimientos de administración tradicional (es decir, salvaguardas físicos) y más de las técnicas de seguridad informática. Con el avance de la tecnología, la seguridad se ha complicado más. Las compañías son más cautas al elegir su software de seguridad.

La información, de igual manera puede llegar a ser inservible por una pérdida en la integridad de los datos. Obviamente, la integridad se ve amenazada, desde un simple error en la introducción de información en una base de datos vital, que provocaría un grave problema para quienes requieran de esta información; hasta los problemas más complejos y significativos.

Un producto o sistema debe necesariamente contener una tecnología de la información (IT), para que ésta, le provea de *confidencialidad, integridad y disponibilidad* de la información a la empresa poseedora de dicha información. Por tanto, la empresa requiere contar además, con características seguras en diversas áreas como: *control de acceso, auditoria y recuperación de errores.*

Los usuarios de aplicaciones y sistemas de información, necesitan contar con una garantía de la seguridad de la información que los sistemas que utilizan les brindan. Para ello, se requiere establecer una medida de comparación o evaluación de la seguridad de los sistemas de información, que se deseen adquirir o desarrollar. En la vida cotidiana, los usuarios o dueños de empresas que requieren de sistemas seguros para garantizar los aspectos antes mencionados en su información, tal vez confíen en la palabra o buena reputación de los creadores o vendedores de "sistemas seguros", tal vez prefieran su propia evaluación o incluso recurran a un grupo especializado independiente, para recibir una opinión imparcial.

Tal valoración de un sistema o producto requiere de *objetivos y criterios de evaluación de seguridad* bien definidos, así como la existencia de una "Autoridad Certificadora" (reglamentado por normas o estándares aceptados internacionalmente) que pueda avalar que la evaluación realizada a un sistema o producto fue realizada de forma apropiada.[2]

El resto de capítulos de este trabajo contienen lo que a continuación se describe: Los dos siguientes capítulos son la base teórica de la propuesta que se hace en esta tesis; el capítulo 2, menciona los aspectos más relevantes establecidos por los organismos internacionales certificadores de seguridad, así como las diferencias de enfoque principales entre ellos. El capítulo 3, contiene una descripción de las metodologías propuestas por diversas organizaciones en materia de seguridad computacional así como las mejores prácticas de desarrollo, nuevas tendencias de programación y diversos aspectos que permean en ideas claras con rumbo a la consecución de los objetivos propuestos por esta tesis. El capítulo 4, propone una extensa lista de chequeo, la cual establece los aspectos más importantes para determinar si un sistema computacional es considerado seguro o no, en base al cumplimiento de los puntos en dicha lista establecidos. El capítulo 5, muestra el mapeo de la lista de chequeo propuesta en el capitulo anterior, a una aplicación que se ejecuta sobre JAGUAR, un servidor de aplicaciones que una institución financiera requiere auditar para conocer el grado de certidumbre al realizar sus operaciones económicas y de intercambio de datos con otras entidades financieras.

2 CARACTERÍSTICAS DE LOS CRITERIOS DE EVALUACIÓN DE SISTEMAS

Como se mencionó anteriormente, el presente trabajo pretende establecer una serie de criterios para evaluar el nivel de seguridad de una aplicación, tomando como referencia Criterios de Evaluación de Seguridad en Sistemas, propuestos y establecidos por organismos internacionales como el TCSEC, ITSEC, COMMON CRITERIA y OSSTMM, por ello a continuación se presenta una visión general de los criterios de evaluación de los sistemas mencionados.

2.1 TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA (TCSEC)

Propuesto inicialmente en 1983. Su propósito es proveer criterios técnicos de seguridad de Hardware, Firmware y Software, así como metodologías y técnicas de evaluación que den soporte a la política de seguridad de un sistema, su evaluación y su acreditación.

De acuerdo con estos criterios de seguridad en computadoras desarrollado por el Departamento de la Defensa de los Estados Unidos (DoD), se asignan varios niveles de seguridad que determinan qué tan bien está protegido el "hardware", "software" y la información almacenada en ellos. Estos niveles describen diferentes tipos de seguridad física, autenticación del usuario y confiabilidad del software, para el sistema operativo. Estos estándares también imponen límites en los diferentes tipos de componentes que podrán ser conectados en algún sistema. El color de su cubierta le da el nombre de "Orange Book". [3]

Las características principales de este conjunto de criterios son:

- Guía objetiva para la evaluación de sistemas
- Define conceptos de seguridad
- Guía de implementación de conceptos de seguridad de computadoras

- Se enfoca principalmente en sistemas operativos de propósito general (sistemas multiusuarios)
- Sirve como base al TNI (Trusted Network Interpretation) [4]

2.1.2 FORMAS DE EVALUAR EN EL TCSEC

- Evaluación de un *producto* excluyendo el ambiente de aplicación. Donde sólo se pueden hacer suposiciones respecto al ambiente en que operará.
- Evaluación de un *sistema* operativo y su ambiente para determinar si el sistema satisface los requisitos de seguridad de la organización.

Este conjunto de criterios se compone de 4 divisiones (A, B, C y D), donde A es la división en que se cuenta con mayor nivel de seguridad, mientras que la división D corresponde a un sistema con seguridad nula o para el que no se requiere seguridad, que a su vez son integradas por clases de seguridad como se muestra en los párrafos siguientes:

2.1.3 DIVISIONES DE EVALUACIÓN

División D: Protección Mínima de Seguridad

De acuerdo con este organismo, la división D está reservada para sistemas que no tienen que ser evaluados. No existe listado de requerimientos para esta división.

La clase D1 es la forma más elemental de seguridad disponible. Este estándar parte de la base que asegura que todo sistema no es confiable. No hay protección disponible para el Hardware; el sistema operativo se compromete con facilidad, y no hay autenticación con respecto a los usuarios y sus derechos, así como para tener acceso a la información que se encuentra en la computadora. Este nivel de seguridad se refiere por lo general a los sistemas operativos como MS-DOS, MS-Windows, etc. Estos sistemas operativos no distinguen entre usuarios y carecen de un sistema definido para determinar quién trabaja en el teclado. Tampoco tienen control sobre la información que puede introducirse en los discos duros.

División C: Protección Discrecionaria

El nivel C tiene dos subniveles de seguridad: las clases C1 y C2.

Clase C1 Protección de seguridad Discrecionaria

Estos sistemas proporcionan características limitadas de seguridad. El Orange Book describe a los sistemas de nivel C1, como un ambiente en el que los usuarios comparten el procesamiento de datos con usuarios del mismo nivel. A este nivel, las características de seguridad, pretenden que los usuarios no generen errores deshonestos que puedan dañar el sistema (Ej., escribir en memoria del sistema o software crítico) o que pueda interferir con el trabajo de otros usuarios (al

borrar o modificar sus programas o datos). Las características de seguridad de estos sistemas no son suficientes para "arrojar" a un intruso del sistema.

Las dos principales características requeridas en usuarios visibles son:

Contraseñas (o algún otro mecanismo). Estas identifican y autentican a un usuario antes de que se le permita el acceso o uso del sistema.

Protección discrecionaria de archivos u otros objetos. Con sistemas de este tipo, se pueden proteger los archivos, para decidir quien tiene permiso de acceso a ellos. Existen diversos métodos de protección, los cuales incluyen protección de clases, grupos, controles públicos y listas de control de acceso.

Estos sistemas pueden también proporcionar una arquitectura de sistema que sea capaz de proteger al código del sistema, de saboteos por usuarios de programas. El sistema puede ser evaluado para asegurarse que esta trabajando propiamente y que las características de seguridad no pueden ser evitadas de manera obvia. Existen también requerimientos específicos sobre la documentación.

Describe la seguridad disponible en un sistema típico Unix. Existe algún nivel de protección para el Hardware, puesto que no puede comprometerse tan fácil, aunque todavía es posible. Los usuarios deberán identificarse a sí mismos con el sistema por medio de un nombre de registro del usuario y una contraseña. Esta combinación se utiliza para determinar qué derechos de acceso a los programas e información tiene cada usuario.

Estos derechos de acceso son permisos para archivos y directorios. Estos controles de acceso discrecional habilitan al dueño del archivo o directorio, o al administrador del sistema, a evitar que algunas personas tengan acceso a los programas e información de otras personas. Sin embargo, la cuenta de la administración del sistema no está restringida a realizar cualquier actividad. En consecuencia, un administrador de sistema sin escrúpulos puede comprometer con facilidad la seguridad del sistema sin que nadie se entere.

Además varias de las tareas cotidianas de administración del sistema sólo pueden realizarse al registrarse el usuario conocido como ***root***. Con la centralización de los sistemas de computadoras de hoy día, no es raro entrar a una organización y encontrar a dos o tres personas que saben la contraseña ***root***.

Pocos proveedores consideran tener sus sistemas evaluados en un nivel más bajo de seguridad que el C1. El sistema IBM MVS/RACF esta evaluado en C1, pero la última versión de este sistema, calificó en nivel C2. Mucha gente considera que un sistema Unix ordinario (sin características mejoradas) calificaría en nivel C1. (Otros afirman que califica en nivel C2).

Clase C2 Protección de acceso controlado

Los sistemas C2 proporcionan mucha más rigurosa seguridad que los sistemas C1. Además de proporcionar todas las características requeridas por los sistemas C1, los sistemas C2 ofrecen las siguientes características para usuarios visibles:

Responsabilidad para todos los usuarios. Control y auditoria individual a través de contraseñas (manteniendo un registro de todas las acciones que ejecuta un usuario y que están relacionadas con la seguridad), el sistema es capaz de seguir el rastro de quienes y que hacen en el sistema.

Control Discrecionario más detallado. El Orange Book describe los requerimientos de sistemas C2 de la siguiente manera: "Esos controles de acceso, serán capaces de incluir o excluir a la granularidad de un usuario". A través de listas de control de acceso (ACL) o algún otro mecanismo, se puede ser capaz de especificar, por ejemplo, que solo X y Y pueden leer de un archivo y que sólo W puede modificarlo.

Objetos de reúso. Esta característica asegura que ningún dato restante en memoria, disco o cualquier otro dispositivo en el sistema, sea accesible accidentalmente por un usuario.

La arquitectura de los sistemas C2, permite que los recursos del sistema sean protegidos vía características de control de acceso. Los sistemas C2 requieren de evaluación más rigurosa y documentación.

Pocos sistemas han sido calificados como C2 (tal vez porque realmente no representa un alto grado de seguridad) aunque más recientemente sistemas sometidos a evaluación persiguen ser calificados a un nivel más alto. (Algunos de esos sistemas ofrecen la habilidad para adecuar el sistema para proporcionar seguridad C2 o B1). Ejemplos de calificaciones C2, incluyen: Digital Equipment Corporation´s VAX/VMS 4.3, Gould´s UTX/32S, Wang Laboratories´ SVS/OS CAP 1.0 (Protección de Acceso Controlado), Control Data Corporation´s Network Operating System (NOS), and Prime´s Primos revisión 21.0. Auditoria de los usuarios. [4]

La auditoria es una característica fundamental de esta clase y se utiliza para mantener los registros (logs) de todos los eventos relacionados con la seguridad, como aquellas actividades practicadas por el administrador del sistema. La auditoria requiere de autenticación adicional porque, si no, ¿cómo se puede estar seguro de que la persona que ejecuta el comando es realmente quién dice ser?

Con el uso de autorizaciones adicionales, es posible que los usuarios de un sistema C2 tengan la autoridad para realizar tareas de manejo de sistema sin necesidad de una contraseña de root. Esto mejora el rastreo de las tareas relativas a la administración, puesto que cada usuario realiza el trabajo en lugar del administrador del sistema.

Estas son autorizaciones particulares que permiten al usuario ejecutar comandos específicos o tener acceso a las tablas de acceso restringido. Por ejemplo, los usuarios que no tengan la autoridad necesaria para analizar la tabla de procesos, verán sólo los procesos al ejecutar el comando ps.

División B Protección Mandatoria

El nivel B de seguridad tiene tres niveles.

Clase B1 Seguridad Etiquetada

Sistemas B1 y superiores, soportan controles de acceso mandatarios u obligatorios (MAC). En sistemas MAC, todos los archivos (y otros objetos importantes) en el sistema son etiquetados. El sistema utiliza esas etiquetas de sensibilidad y los niveles de seguridad de los usuarios del sistema, para hacer cumplir las políticas de seguridad del mismo. No se puede otorgar a un usuario acceso a alguno de los archivos, a menos que el usuario tenga la clasificación necesaria sobre el archivo. El Orange Book impone bastantes requerimientos de etiquetado específicos en sistemas B1, incluyendo etiquetado de toda la información que es exportada del sistema.

Los sistemas B1 pueden tener una arquitectura del sistema que separe de manera más estricta las partes relacionadas a la seguridad del sistema, de las que no lo son. Los sistemas B1 también requieren estricta evaluación y documentación. En esta clase, la documentación debe incluir un modelo de la política de seguridad que soporta el sistema. Esta política no es matemática, pero debe ser una expresión clara de las reglas implementadas por las características de seguridad del sistema.

En resumen, la Clase B1 establece lo siguiente:

- Puente entre las divisiones C y B
- Todo objeto y sujeto tiene asociada una etiqueta de seguridad
- Todo cambio de etiqueta es controlado de forma estricta
- Modelo informal de la política de seguridad
- Control de acceso obligatorio de sujetos y objetos
- Control de etiquetado de la información exportada

El nivel B1, o protección de seguridad etiquetada, es el primer nivel que soporta clasificaciones de seguridad multinivel, como la secreta y la ultra secreta. Este nivel parte del principio de que un objeto bajo control de acceso obligatorio no puede aceptar cambios en los permisos hechos por el dueño del archivo.

Ejemplos de sistemas ya evaluados en nivel B1: SecureWare´s CMW+, AT&T´s System V/MLS, IBM´s MVS/ESA y UNISYS´ OS 1100. The Open Software Foundation´s OSF/1, ofrece a los sistemas de sus clientes la opción de configurarse para soportar cualquiera de los niveles de seguridad C2 o B1. [4]

Clase B2 Protección Estructurada

Además de las características de la clase B1, la clase B2 posee los siguientes atributos:

- Categorización de la información
- Modelo formal de la política de seguridad
- Controles discrecionario y obligatorio
- Control en canales ocultos
- Separar elementos críticos y no críticos en la TCB (Trusted Computer Base)
- La TCB es respaldada por pruebas más rigurosas
- Los mecanismos de autenticación son fortalecidos

Los sistemas B2 y más altos, no adicionan realmente más características de seguridad a las requeridas por el nivel B1. En lugar de eso, se extienden y mejoran esas características y requieren aseguramiento adicional para realizar apropiadamente su trabajo. El Orange Book dice que los sistemas en nivel B2 son "relativamente resistentes a penetración"

En la clase B2, el etiquetado de características es extendido hasta incluir todos los objetos en el sistema, incluyendo dispositivos. Los sistemas B2 adicionan también la característica de ruta confiable, permitiendo a los usuarios comunicarse con el sistema de manera directa e inequívoca, por ejemplo, al presionar cierta tecla o interactuar con un determinado menú. El sistema debe ofrecer una prueba de que el intruso no puede interferir directamente con el canal mediante un engaño.

Los sistemas B2 soportan la menor cantidad de privilegios, el concepto es, que los usuarios y programas deben poseer el mínimo número de privilegios que ellos necesiten y el menor tiempo necesario, para ejecutar las funciones del sistema. Este concepto es implementado vía una combinación de diseño, implementación y requerimientos de seguridad. Un ejemplo, es el requerimiento que separa al administrador y operador del sistema.

A partir del diseño de un sistema, el sistema B2 requiere sustancialmente más modularidad y uso de características de hardware para aislar funciones relacionadas a la seguridad de las que no lo están. Los sistemas B2 requieren además de una expresión matemática formal de las políticas de seguridad del sistema, así como también de una evaluación más rigurosa y documentación. También se requiere que un administrador de la configuración del sistema, maneje todos los cambios del código y documentación del sistema, y los desarrolladores del sistema conduzcan una investigación de canales ocultos que pudiesen favorecer al filtrado de información del sistema.

Pocos sistemas han sido evaluados exitosamente en nivel B2 y superiores. Honeywell Information System´s Multics system y Trusted Information System´s Trusted XENIX, son dos ejemplos de sistemas calificados en B2. Podríamos decir que en la actualidad, este es el límite superior en cuanto a la seguridad que los sistemas pueden ofrecer [4]

Clase B3 Dominios de Seguridad

No existen requerimientos para nuevas características de usuarios visibles en el nivel B3, pero el diseño del sistema y características de seguridad son sustancialmente más rigurosas. Se requiere Administración confiable de recursos (asignación de un individuo como administrador de la seguridad), así como recuperación de seguridad (procedimientos para asegurarse de que la seguridad no falle, en caso de que el sistema lo haga) y la habilidad para señalar el administrador inmediatamente si el sistema detecta una "violación inminente a la política de seguridad."

El nivel B3, o nivel de dominios de seguridad, refuerza los dominios con la instalación de Hardware. Por ejemplo, el Hardware de administración de memoria se usa para proteger el dominio de seguridad de un acceso no autorizado o la modificación de objetos en diferentes dominios de seguridad. Este nivel requiere que la terminal del usuario se conecte al sistema por medio de una ruta de acceso segura.

Los sistemas B3 no son muy comunes. Es difícil obtener una calificación de nivel B3. El único sistema que actualmente está en nivel de B3 es XTS-2000 por Honeywell Federal Systems. [4]

División A Protección Verificada

Garantía, a través de verificación formal, de que la información sensitiva es protegida.

Clase A1 Protección verificada

Entre las características de esta clase se pueden mencionar:

- Sistema diseñado con la seguridad como elemento prioritario
- Modelo formal de la política de seguridad
- Prueba matemática de que el modelo es consistente con sus axiomas
- Control de distribución de software, desde el punto de manufactura hasta el usuario (evitar virus, caballos de Troya, etc.)
- Técnicas formales para implementar la TCB de forma correcta.

Es hasta el momento el nivel más elevado de seguridad validado por el libro naranja. Incluye un proceso exhaustivo de diseño, control y verificación. Para lograr este nivel de seguridad, todos los componentes de los niveles inferiores deben incluirse; el diseño requiere ser verificado en forma matemática; además, es necesario realizar un análisis de los canales ocultos y de la distribución confiable. *Distribución confiable* significa que el hardware y el software han estado protegidos durante su entrega para evitar violaciones a los sistemas de seguridad.

Los sistemas A1 se asemejan en funcionalidad a los sistemas B3. La única característica proporcionada por los sistemas A1, además de los requerimientos de los sistemas B3 es distribución de seguridad, la cual hace cumplir la seguridad, mientras un sistema seguro es enviado a un cliente. Lo que ofrece un sistema A1, es un análisis formal y pruebas matemáticas que el diseño del sistema hace corresponder con las políticas de seguridad y las especificaciones de diseño.

Los únicos sistemas que ha recibido una calificación de A1 es Honeywell Information System´s Secure Communications Processor (SCOMP) y Boeing Aerospace´s SNS system. [4]

2.1.4 INCONVENIENTES EN EL TCSEC

- Toma aproximadamente uno o dos años el evaluar un sistema operativo
- No es aplicable a microcomputadoras ni a redes locales. No satisface necesidades actuales
- Es unidimensional, pues prioritariamente se preocupa por la confidencialidad, descuidando otros aspectos de seguridad, como la disponibilidad y la integridad.
- Difícil de asimilar, debido al estilo del Inglés militar empleado en su redacción.

En el **Anexo A**, se muestra de manera esquemática, las características seguras de cada uno de los niveles de seguridad, planteados por el TCSEC, desde diversas áreas de los sistemas.

2.2 INFORMATION TECHNOLOGY SECURITY EVALUATION CRITERIA (ITSEC)

El ITSEC o "White Book", es el conjunto de criterios armonizados para la evaluación de la seguridad de sistemas y productos propuestos por algunos países de Europa. El objetivo principal de la evaluación es proveer al usuario final, un cierto grado de confianza (o certeza) de que el sistema que adquirió o que piensa obtener, reúne sus requerimientos de seguridad.

Al igual que el TCSEC, el libro blanco define a la seguridad como un conjunto de tres elementos:

- Confidencialidad
- Integridad
- Disponibilidad

Estos elementos ya fueron introducidos en el capítulo 1 del presente trabajo.

En el ITSEC, todo producto o sistema es referido como: TOE (Target of evaluation). Al igual que el TCSEC, ambos son compuestos por elementos de software o hardware, y la diferencia principal entre ellos radica en su entorno. Para un *producto*, el entorno es algo muy general, de hecho, se hacen muchas suposiciones acerca del mismo. Esta característica, hace que un producto sea susceptible de ser utilizado en una gran variedad de entornos. Para un *sistema*, el entorno es muy específico, real y, en teoría único. Así los sistemas pueden estar constituidos por productos (quizá ya evaluados).

Los componentes de un TOE (producto o sistema a ser evaluado) pueden estar involucrados con la seguridad o no. Al conjunto de componentes del TOE que tienen que ver con la seguridad se les denomina TCB (Trusted Computing Base) En este momento, la pregunta obvia sería: ***¿cómo saber si un componente es relevante para la seguridad?*** Si esto no está muy claro, puede analizarse en la evaluación

En el ITSEC, un proveedor (sponsor), es el que presenta un ST (security target) para ser evaluado. El ST define las funciones de seguridad del TOE, las posibles amenazas esperadas y mecanismos utilizados para definir las funciones de seguridad. Por otro lado, no sólo es necesario que el proveedor del TOE especifique cuales son las funciones de seguridad y los mecanismos que las implementan, también es necesario tener la certeza (Assurance) de que las funciones cumplen con los objetivos de seguridad especificados. La certeza tiene que ver con que las funciones hayan sido correctamente implementadas (Correctness) y que sean efectivas (Effectiveness).

La Fig. 2.2.1 muestra el proceso de evaluación de un TOE según el ITSEC.

De manera imaginaria para describirla, podemos dividir la Fig. 2.2.1 en seis etapas importantes, las cuales participan de manera activa dentro del proceso de evaluación de un producto u objeto de evaluación. En la primera línea ubicamos aquellas características cuyo comportamiento permea de manera directa y que no podemos dejar de lado sobre la evaluación de un sistema o producto, entre las comunes se encuentran: ambiente del mundo real, suposiciones del ambiente y requisitos operacionales; en segundo término o etapa, las amenazas del mundo real, así como las

amenazas supuestas sobre los productos; en tercer lugar, los objetivos de seguridad básicos que siempre deben estar presentes durante cualquier evaluación, además de los límites establecidos por las regulaciones legales, los objetivos de seguridad básicos, siempre contienen las funciones de seguridad, así como los niveles de evaluación solicitados por el cliente o interesado del nivel de seguridad, que obtendrá del producto o sistema a evaluar. Todo lo anterior, se ve reflejado en el resultado de todo el proceso, que es un sistema o producto evaluado, bajo toda la serie de "restricciones" previamente establecidas, originándose de todo esto un documento que concentra de manera escrita la evaluación.

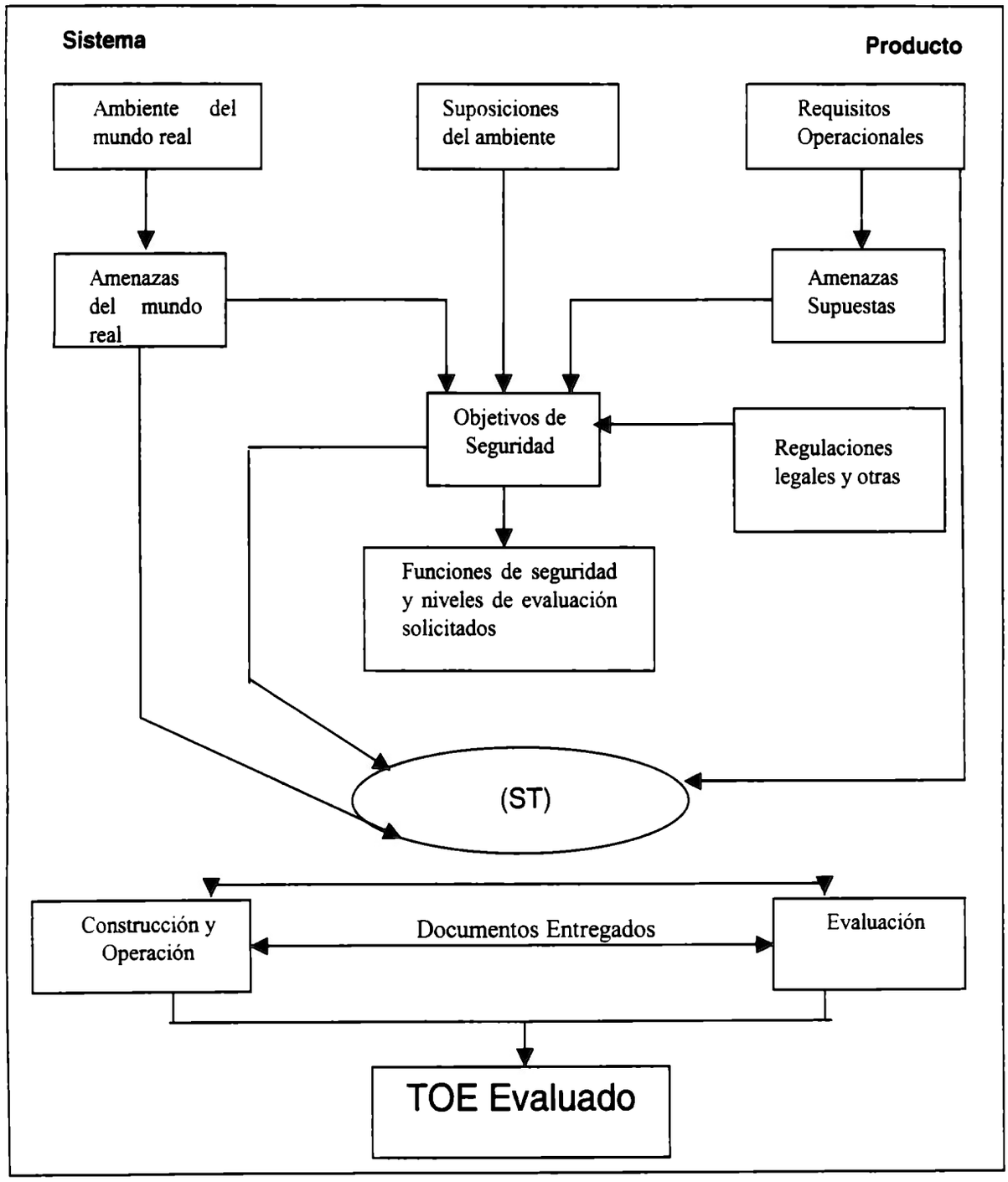


Figura 2.2.1 Proceso de Evaluación del TOE en el ITSEC.

2.2.1 PROCESO DE EVALUACIÓN

Existen usualmente tres participantes en el proceso de evaluación de productos. como se muestra en la figura 2.2.2:

- Sponsor (Proveedor)
- El CLEF (Commercial Licensed Evaluation Facilities)
- Certification Body (El cuerpo de certificación CB)

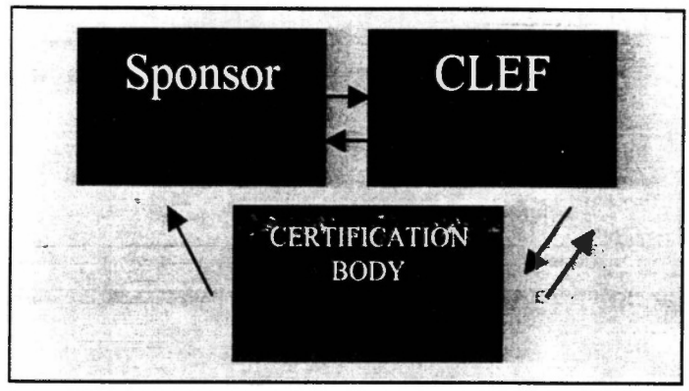


Fig. 2.2.2 Información y flujo de retroalimentación entre las diversas partes del proceso de evaluación [5]

La evaluación es realizada a un nivel convenido de garantía (assurance), que define el grado de confianza requerido por el producto.

Existe varios pasos en el proceso de evaluación:

- Decisión para Evaluar.- ¿es necesaria la evaluación de la seguridad? (responsabilidad del vendedor)
- Preparación de la evaluación.- produce objetivo de seguridad y entregable (vendedor)
- Evaluación.- Medición contra la seguridad objetivo (CLEF)
- Revisión de la certificación y reconocimiento (Cuerpo de la certificación)

El desarrollador genera la decisión de buscar bases de la certificación en un plan de negocio o investigación. Entonces el desarrollador incluye definición del producto a evaluar, especificación de las pretensiones a alcanzar por el producto y nivel de seguridad objetivo. El CLEF y otras consultorías de seguridad especializadas pueden aconsejar y ayudar en el proceso.

Los criterios de evaluación están públicamente disponibles y son utilizados para definir el nivel requerido.

La evaluación formal es programada para un horario aceptable y el trabajo varía de acuerdo al nivel objetivo y a la naturaleza del producto. En el material a examinar, puede quizá incluirse el esquema de documentación, planes de prueba y altos niveles de certificación y códigos fuente. El proceso entero es monitoreado por el Cuerpo de Certificación (CB), quien examina resultados y si es apropiado, emite un certificado.

La certificación aplica sólo a versiones específicas de un producto. Cuando el producto es corregido o actualizado, tal vez sea necesario ampliar la certificación. Este proceso de

evaluación, como se ve en la figura 2.2.3, evalúa productos existentes en el mercado para asignarles un nivel de seguridad, asignando una evaluación final. En resumen, El CLEF mapea sus criterios de seguridad con las características que presenta el producto, en caso de existir diferencias, este equipo elabora un reporte acerca de la problemática, el producto es regresado al proveedor del producto motivo de evaluación para que resuelva esos problemas, cuando ya no existe problemática, notifica al cuerpo de Certificación (CB), el cual revisa el reporte técnico elaborado por el CLEF, y confirma el seguimiento de la evaluación. Finalmente el CB pública el reporte técnico de evaluación al vendedor o proveedor.

Paso 1. La decisión de Evaluar

Para vendedores de productos, la decisión de evaluar es basada en una análisis de costo / beneficio de un negocio. Esto puede ser una importante decisión, en la definición del mercado, tamaño, conocimiento de necesidades, estableciendo un patrón de demanda para productos seguros, o un proyecto ligado principalmente a oportunidades específicas de negocio. En ambos casos se incrementa el potencial de negocio.

Paso 2. Preparación para la Evaluación

El objetivo de seguridad (ST) especifica las características de seguridad del producto y lo relacionado a:

- Los objetivos de seguridad
- Posibles amenazas de seguridad
- Ambientes en los cuales el producto pretende operar

En este estado, es también necesario especificar el nivel de seguridad requerido, medido de E1 a E6, siendo E6 el nivel más alto de seguridad, además de establecer las funciones de seguridad a cumplir por el producto objeto de evaluación. En la tabla T2.1 se establecen las funciones de seguridad del ITSEC, así como su significado de una manera muy general, y estas funciones nos permiten establecer ese nivel de seguridad requerido que se menciona durante éste párrafo.

FUNCIÓN	SIGNIFICADO
Identificación y Autenticación	Las políticas de seguridad de los sistemas, especifican los sujetos y objetos que pueden ser identificados y autenticados. Los criterios identifican tres mecanismos de autenticación: ▪ Autenticación por conocimiento ▪ Autenticación por posesión ▪ Autenticación por características propias
Administración de derechos	Las políticas de seguridad especifican los derechos que poseen cada sujeto y objeto e identifican la relación entre ellos, algunas reglas especiales en el sistema y las reglas para

	garantizar y cambiar esos derechos.
Verificación de derechos	El sistema puede verificar los derechos de un sujeto cuando el sujeto intenta acceder a un objeto
Auditoria	El sistema puede auditar eventos relacionados a la seguridad, registrando información acerca de quiénes y qué hacen en el sistema
Objetos de reúso	Los objetos pueden ser "limpiados" de datos antes de ser utilizados por usuarios, que accidentalmente accedieron a la información, incluso sin tener permitido el acceso
Recuperación de errores	Las políticas de seguridad identifican condiciones de error y como el sistema se recupera de dichos errores.
Continuidad del servicio	El sistema es capaz de continuar prestando servicios clave, para mantener la seguridad del sistema
Seguridad en la comunicación de datos	Debido a las características ordinarias de seguridad en el sistema, éstos permiten adicionar ciertos funciones y mecanismos de seguridad en la comunicación, como: ▪ Autenticación de ambas entidades ▪ Control de acceso ▪ Confidencialidad de datos ▪ Integridad de datos ▪ Autenticación del origen de datos ▪ No repudiación

Tabla T2.1 Funciones de seguridad del ITSEC [6]

Algunos productos pueden ya haber sido evaluados, por ejemplo, de acuerdo a otros criterios como el TCSEC, por tanto, esto puede tomarse en cuenta y reducirse el tiempo y costo de evaluación.

También debe tomarse en cuenta si el producto va a ejecutarse sobre una red. La evaluación sólo aplica a versiones específicas y hardware. En lugares donde los productos (por ejemplo, bases de datos) corren sobre diferentes plataformas de hardware, sólo son evaluados para una plataforma, el costo se reduce para futuras evaluaciones en otros Host.

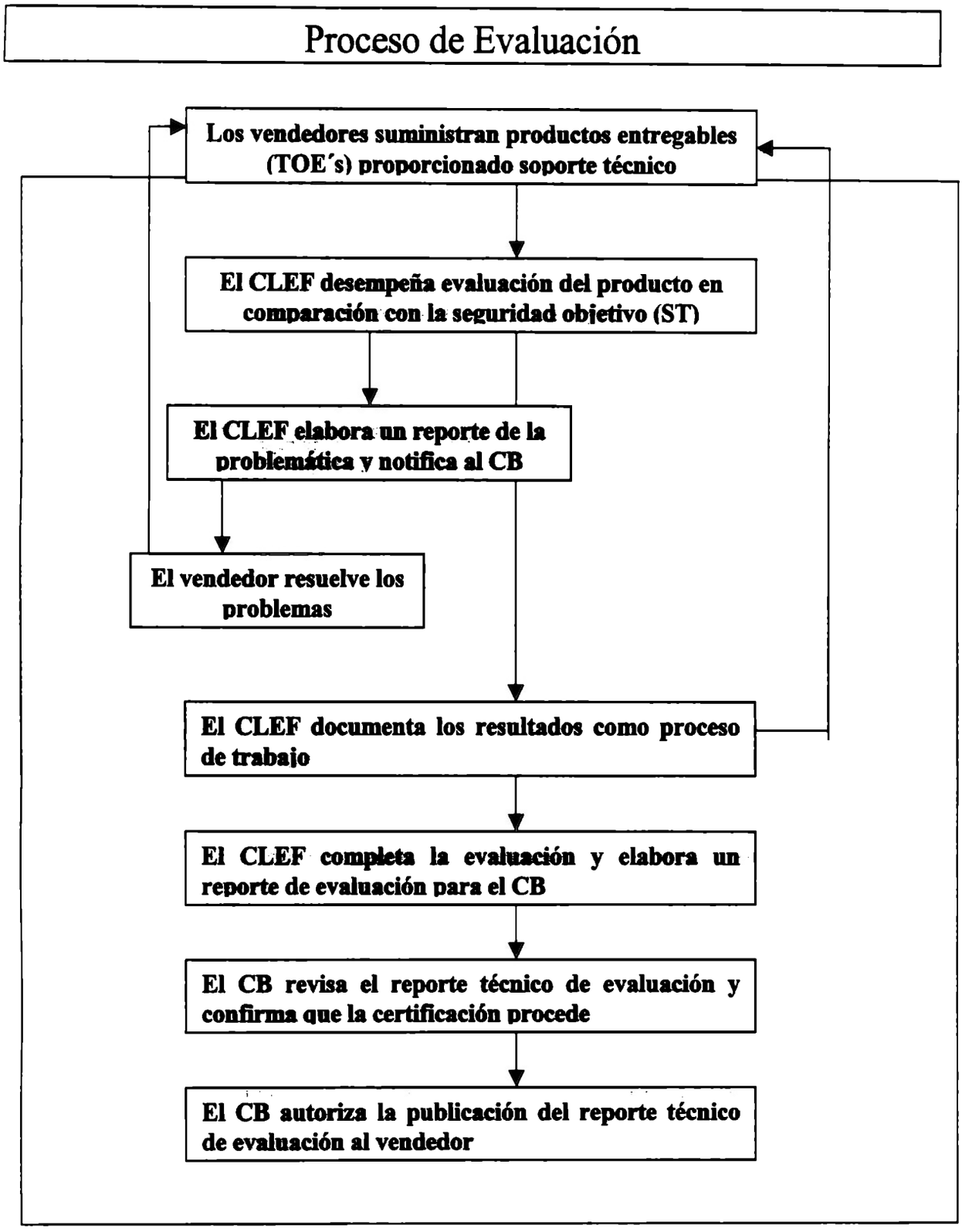


Fig. 2.2.3 Esquema del Proceso de Evaluación bajo ITSEC [5]

Paso 3. Evaluación.

Es un proceso de prueba, en el cual el desarrollo se refleja en sí mismo, verificando que ciertos requerimientos se cumplan.

Para realizar el proceso de la manera más eficaz posible y sin problemas, el CLEF debe trabajar muy de cerca con el proveedor, a través de diversas etapas de la evaluación. Esto se emprende como una fase única, abarcando varias etapas claramente definidas, durante las cuales los evaluadores:

- Evalúan el objetivo de seguridad
- Producen un programa de trabajo de la evaluación
- Convienen una lista de documentos, material y soporte a proporcionar
- Evalúan la exactitud del sistema
- Evalúan el ambiente de operación
- Producen reportes de evaluación comprensivos

Paso 4. Certificación

El cuerpo de Certificación, revisa toda la evidencia de documentación, proporcionada por los evaluadores y determinan que tanto de la seguridad buscada, se ha alcanzado. Si es íntegra, entonces la certificación es otorgada.

Si existe algún defecto explotable en el producto, el proveedor (Sponsor) y el certificador convienen en las modificaciones.

Paso 5. Re-Evaluación

El cuerpo de certificación, aconseja si la re-evaluación es necesaria, si un producto tiene que ser modificado. El trabajo involucrado puede ser mínimo, durante la primera evaluación, para clasificar los componentes del producto de acuerdo a su influencia con las características de seguridad. Por lo tanto, en cualquier momento se realizan cambios en la evaluación del producto, el proveedor, CLEF y certificador pueden utilizar la clasificación para determinar más fácilmente el impacto en la certificación y que acción debe tomarse

La certificación ITSEC de un producto de software significa que los usuarios pueden confiar en el nivel de seguridad asignado a cualquier producto que deseen adquirir.

2.2.2 NIVELES DE SEGURIDAD (CONFIANZA)

Sistemas E0.

Seguridad Inadecuada.

Sistemas E1.

El objetivo de seguridad y diseño de la arquitectura informal deben ser producidos. La documentación del usuario / administrador ofrece una guía para la seguridad en el TOE. La seguridad que hace cumplir las funciones es probada por el evaluador y desarrollador. Son utilizados métodos de distribución seguros.

Es aplicable donde la confianza en la operación correcta es requerida, pero las amenazas existentes son bajas.

Sistemas E2.

Un diseño informal detallado y documentación de la prueba son producidos. La estructura debe mostrar la separación del TOE y otros componentes sobre la seguridad implementada. Existe búsqueda de errores en pruebas de penetración. Se evalúa el control de la configuración y seguridad del desarrollador. Es necesario un seguimiento durante el inicio y operación del producto.

Sistemas E3.

Código fuente o esquema del hardware es producido. Debe haber correspondencia entre el código fuente y el diseño detallado. Los lenguajes implementados deben ser estándares reconocidos. La revisión debe ser después de la corrección de errores

Sistemas E4

Es un modelo formal de seguridad, especificación semi-formal de funciones de seguridad implementadas, deben producirse arquitectura y detalles de diseño. La prueba mostrada debe ser suficiente. El TOE y las herramientas están bajo control de configuración, con los cambios revisados y opciones de compilador documentadas.

Sistemas E5.

El diseño de la arquitectura explica la correlación entre la seguridad que hace que sus componentes la cumplan. Debe producirse información sobre el proceso de integración y tiempo de ejecución. Control de configuración independiente por parte del desarrollador. Identificación de elementos configurados, así como la seguridad que debe hacerse cumplir, con soporte para relaciones viables entre ellos.

Sistemas E6.

Deben producirse descripción formal de la arquitectura y seguridad en funciones implementadas. Se muestra correspondencia a partir de la especificación formal de seguridad en las funciones implementadas a través de pruebas y código fuente. Definición de diferentes configuraciones del TOE, en términos del diseño de la estructura formal. Todas las herramientas con respecto a control de la configuración.

2.3 COMMON CRITERIA

El COMMON CRITERIA (CC) representa el resultado del esfuerzo en el desarrollo de criterios para la evaluación de seguridad de la tecnología de la Información (IT), que son ampliamente usados dentro de la comunidad internacional. Esto es, una alineación y desarrollo de un número de criterios: los Europeos, estadounidenses y canadienses (ITSEC, TCSEC y CTCPEC, respectivamente). Esto es una contribución al desarrollo de un estándar internacional, que busca abrir el camino de reconocimiento mutuo de evaluación de resultados. [7]

El desarrollo de criterios ITSEC en Canadá y países Europeos, han seguido el trabajo original del TCSEC en los EU (Orange Book). El desarrollo de los criterios federales de los EU es un intento por combinar esos otros criterios con el TCSEC, y eventualmente dirigir la asociación de investigación hacia la producción de los Criterios Comunes. (COMMON CRITERIA)

Una gran cualidad en el desarrollo del CC, es la estrecha participación de todas las partes con experiencia en la creación del documento original nacional de Criterios. Los beneficios del CC vienen de su madurez acumulada, y los intentos por una flexibilidad completa para la estandarización de evaluación y funcionalidad segura. El CC ha sido creado suficientemente flexible para permitir la integración con los numerosos sistemas existentes para la evaluación, certificación y acreditación de seguridad de IT.

La estructura del CC también proporciona gran flexibilidad en la especificación de productos seguros. Consumidores y otras partes, pueden especificar la seguridad de la funcionalidad de un producto en términos de perfiles estándar de protección, e independientemente seleccionar el nivel de evaluación de un conjunto definido de siete niveles de seguridad, del EAL1 al EAL7 (niveles de garantía de la evaluación).

La versión 1.0 del CC fue publicada en Enero de 1996, la versión 2.0 apareció a inicios de 1998, y posteriormente la versión 2.1, de la cual a continuación realizaremos una remembranza por sus características más básicas y fundamentales.

El CC presenta requisitos para la seguridad en IT de un producto o sistema dentro de las distintas categorías de las exigencias funcionales y los requerimientos de confianza. Los requisitos funcionales del CC definen el deseo del comportamiento seguro.

Esta versión del CC para la evaluación de la seguridad de la tecnología de la información, es una revisión que se alinea con el estándar internacional ISO/IEC 15408.1999.

Las siete organizaciones gubernamentales (colectivamente llamadas "Organizaciones Patrocinadoras del Proyecto CC") son:

- Canada: Communications Security Establishment
- France: Service Central de la Sécurité des Systèmes d'Information
- Germany: Bundesamt für Sicherheit in der Informationstechnik
- Netherlands: Netherlands National Communications Security Agency
- United Kingdom: Communications-Electronics Security Group
- United States: National Institute of Standards and Technology
- United States: National Security Agency

2.3.1 ORGANIZACIÓN DEL CC

Parte 1, Introducción y modelo general,

Es la introducción al CC. Define conceptos generales y principios sobre evaluación de seguridad en IT y presenta un modelo general de evaluación. También se presenta una construcción para expresar los objetivos de seguridad en IT, para seleccionar y definir requerimientos de seguridad en IT. y para escribir especificaciones de alto nivel para productos y sistemas. Además. la utilidad de cada parte del CC es descrita en términos de cada objetivo.

Alcance

Este estándar multiparte, el COMMON CRITERIA (CC), intenta ser usado como una base para la evaluación de las propiedades de la seguridad en los sistemas y productos IT. Para fundamentar al CC como una base, los resultados de una evaluación de la seguridad de la IT deben ser significativos para todas las partes.

El CC permitirá comparar entre los resultados de evaluaciones de seguridad independientes. También permitirá proveer un conjunto común de requerimientos para las funciones de seguridad de productos y sistemas de IT y para asegurar la aplicación de medidas durante la evaluación de la seguridad. El proceso de evaluación establece un nivel de confidencia para las funciones de seguridad de cada producto y sistema y asegura la aplicación de las medidas de los requerimientos ya conocidos. La evaluación de los resultados puede ayudar a los consumidores a determinar cuándo los productos o sistemas en la tecnología de la información son suficientemente seguros.

La Evaluación tanto en productos como en sistemas es conocida como "Objeto de Evaluación" (TOE) Tales TOE´s incluyen, por ejemplo, sistemas operativos, redes de computadoras, sistemas distribuidos y aplicaciones.

La seguridad del CC está dirigida a la divulgación no autorizada de la información, modificación o pérdida. Las categorías de protección relacionadas con esos tres tipos de fallas de seguridad son comúnmente llamadas: *Confidencialidad, integridad y disponibilidad* respectivamente. El CC puede también ser aplicable en aspectos de seguridad IT fuera de los tres mencionados anteriormente. Se concentra en la amenaza de la información que pudiese originarse en la actividad humana, o maliciosa de alguna manera, pero pude ser aplicable para algunas amenazas no humanas.

El CC es aplicable para medidas de seguridad en IT implementadas en hardware (HW), Firmware (FW) y Software (SW) donde se proponen aspectos particulares de evaluación sólo para aplicar a ciertos métodos de implementación, estos pueden ser indicados dentro de la declaración de criterios relevantes.

Algunos temas, ya sea que involucran técnicas especializadas o que están relacionados con la seguridad de la IT, se encuentran fuera del alcance del CC. Algunos de estos temas son considerados a continuación:

El CC no contiene criterios de seguridad que apliquen a las medidas de seguridad administrativa no relacionadas directamente con las medidas de seguridad de IT. Sin embargo, hay que reconocer que una parte significativa de la seguridad en el TOE puede llevar al cabo frecuentemente, medidas administrativas como son: organizacional, personal, física y control de procedimientos. Las medidas de seguridad administrativa en el ambiente de operación del TOE son tratadas como seguras tradicionalmente, donde éstas tienen un impacto en la habilidad de las medidas seguras en la IT o identificación de amenazas.

La evaluación de aspectos físicos y técnicos de la seguridad en IT, como control de la emanación electromagnética, no es cubierto específicamente, aunque varios de los conceptos relacionados sean aplicables al área. En particular, lo relacionado con algunos aspectos de protección física de los TOE`s.

El CC no relaciona ni a la metodología de la evaluación ni a lo relacionado con las cuestiones legales y administrativas, bajo las cuales el CC puede ser aplicado por evaluación de autoridades. Sin embargo, se espera que el CC sea usado con propósitos de evaluación en el contexto de tales asuntos legales y administrativos y en dicha metodología.

Los procedimientos para la acreditación del uso de los resultados en la evaluación en productos o sistemas se encuentran fuera del alcance del CC. La acreditación de productos o sistemas es un proceso administrativo mediante el cual se concede autoridad para la operación de un producto o sistema IT en un entorno operacional completo. La evaluación se centra en las partes de seguridad de IT de los productos o sistemas y esas partes del entorno operacional que puede directamente afectar el uso seguro de elementos IT. Los resultados del proceso de evaluación son consecuentemente una entrada válida para el proceso de acreditación. Sin embargo, como otras técnicas son más apropiadas para la evaluación de propiedades seguras en productos o sistemas no IT y sus relaciones con las partes seguras de IT.

El tema de los Criterios para la evaluación no incluye las características inherentes a los algoritmos criptográficos. Independientemente se evalúan las propiedades matemáticas de criptografía incluidas en un TOE, la evaluación de sistemas en los que el CC es aplicado, debe tener una estructura para cada evaluación.

Público Objetivo del CC

Existen tres grupos con un interés general en la evaluación de las propiedades de seguridad en productos y sistemas IT: Consumidores TOE, desarrolladores TOE y evaluadores TOE. Los criterios que trata este estándar son estructurados para respaldar las necesidades de los tres grupos. Ellos son los considerados principales usuarios del CC. Los tres grupos pueden beneficiarse de estos criterios

Consumidores

El CC desempeña un papel importante en las técnicas que soportan la selección del consumidor de IT, es decir, los requisitos de seguridad para expresar sus necesidades de organización. El CC describe para asegurarse que la evaluación satisface las necesidades de los consumidores pues éste es el propósito y la justificación fundamental para el proceso de la evaluación.

Los consumidores pueden utilizar los resultados de evaluaciones para ayudarse a decidir si un producto o un sistema evaluado satisfacen sus necesidades de seguridad. Estas necesidades de seguridad se identifican generalmente como resultado del análisis de riesgo y de las políticas de dirección.

Los consumidores pueden también usar la evaluación para comparar diferentes productos o sistemas. La presentación de los requisitos del aseguramiento dentro de una jerarquía, soportan esta necesidad.

Desarrolladores

El CC es pretendido por los desarrolladores para la preparación y asistencia en la evaluación de sus productos o sistemas y en la identificación de los requisitos de seguridad para satisfacer cada uno de sus productos o sistemas. Es también absolutamente posible que resulte una metodología asociada a la evaluación, potencialmente acompañada por un acuerdo de reconocimiento mutuo para la evaluación de resultados, permitiría después que el CC apoyara a cualquier persona, independientemente del desarrollador del TOE, en la preparación y asistencia en la evaluación de un *TOE* desarrollado

Las construcciones del CC se pueden entonces utilizar para argumentar que el TOE se conforma con los requisitos identificados por medio de funciones y de aseguramientos especificados para ser evaluado. Cada uno de los requerimientos de un TOE esta contenido en un Objetivo de seguridad (ST)

El CC describe las funciones de seguridad que un desarrollador puede incluir en el TOE. El CC puede ser usado para determinar las responsabilidades y acciones para soportar las pruebas que son necesarias para apoyar la evaluación del TOE. Además define el contenido y presentación de las pruebas.

Evaluadores

El CC contiene criterios que son utilizados por evaluadores para constituir juicios acerca de los TOE´s en cuanto a sus requerimientos de seguridad. El CC describe el conjunto de acciones comunes para la evaluación, así como las funciones de seguridad sobre las cuales se ejecutan estas acciones. Note que el CC no especifica procedimientos guía para llevar al cabo esas acciones.

Otros

Mientras que el CC esta orientado hacia la especificación y evaluación de las propiedades de la seguridad de los TOE´s, también puede utilizarse como material de referencia para todas áreas que se interesen sobre la seguridad en tecnología de la información (IT) Algunos de los grupos de interés que pueden beneficiarse con el contenido de información del CC son:

- Sistemas guardianes y oficiales de seguridad de sistemas, responsables de determinar y conocer las políticas y requerimientos de seguridad en tecnología de la información en las organizaciones.

- Auditores externos e internos, responsables de evaluar y adecuar seguridad en los sistemas.
- Creadores de seguridad y diseñadores responsables de la especificación del contenido de seguridad en productos y sistemas IT.
- Autoridades evaluadoras responsables de la administración y evaluación de la seguridad en programas.

Contexto de Evaluación

Para lograr una buena comparación entre resultados de evaluación, las evaluaciones se deben ejecutar dentro de los lineamientos de un sistema autorizado que contenga un conjunto estándar de criterios, que realice un monitoreo de la calidad de las evaluaciones y administre las regulaciones con las cuales se facilita la evaluación. (Ver Figura 2.3.1)

El CC no establece requerimientos para los lineamientos regulatorios. Sin embargo, la consistencia de los lineamientos de diferentes autoridades evaluadoras será necesaria para alcanzar las metas de mutuo reconocimiento de los resultados de tal evaluación. En la siguiente figura se muestran los principales elementos que conforman el contexto para la evaluación:

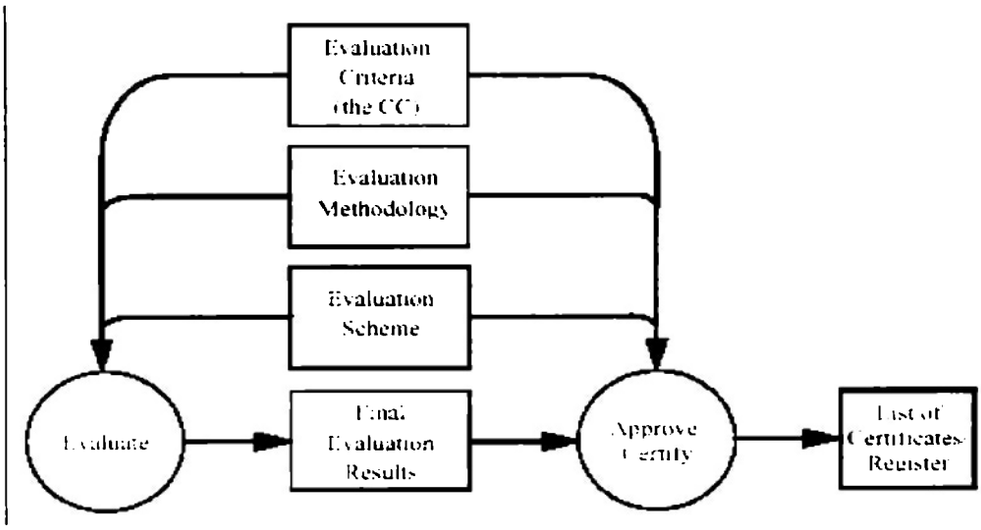


Figura 2.3.1 Proceso de Evaluación bajo CC [7]

El uso de una metodología común de evaluación, contribuye a la repetición y objetividad de los resultados, pero no es por sí mismo suficiente. Algunos de los criterios de evaluación requieren de la aplicación de juicios especializados y conocimiento de respaldo, cuya consistencia es más difícil de alcanzar. De hecho, para resaltar la consistencia de la evaluación final resultante, esta puede ser sometida a un proceso de certificación. El proceso de certificación, es la inspección independiente de los resultados encaminados a la producción de la certificación final o aprobación. El certificado esta normalmente disponible al público. El proceso de certificación es un medio para lograr grandes consistencias en la aplicación de criterios de seguridad en IT.

Los sistemas de evaluación, metodología y procesos de certificación, son la responsabilidad de las autoridades evaluadoras que ejecutan los sistemas de evaluación y están fuera del alcance del CC.

Parte 2. Requisitos funcionales de seguridad

Establece un conjunto de componentes funcionales como un camino estándar para expresar los requerimientos de funcionalidad para los TOE´s. Lista del conjunto de componentes funcionales, familias y clases.

Parte 3, Requisitos de seguridad

Establecer un conjunto de componentes de seguridad como un camino estándar para expresar los requisitos de certidumbre para TOE´s. Lista el conjunto de componentes de seguridad, familias y clases. También define criterios de evaluación para perfiles de protección (PPs) y Seguridad Objetivo (ST) y plantea la evaluación de niveles de seguridad que definan la escala predefinida por el CC para evaluar el aseguramiento para TOE´s, los cuales son conocidos como: **Niveles de evaluación de Seguridad (EALs)**

De lo anterior, se deduce un nuevo cuestionamiento:

2.3.2 ¿QUÉ ES UN PERFIL DE PROTECCIÓN (PP)?

Simplemente es un conjunto de requerimientos diseñados por un conjunto de circunstancias. Esto consiste de:

- lista de amenazas
- lista de requerimientos funcionales
- lista de actividades de seguridad
- justificación sobre la existencia de esas amenazas

Los PP pueden ser diseñados por un grupo de consumidores potenciales quienes tienen similares necesidades de seguridad en IT o por el desarrollador mismo.

Un PP no está relacionado con cualquier producto o sistema dado, más bien, define necesidades de usuario independiente de cualquier producto específico. Contra la certificación, un PP especificará la extensión para la cual sus requerimientos tienen que responsabilizarse. Un PP es particularmente útil en ayudar a formular la especificación de adquisiciones.

Los perfiles de protección ya publicados incluyen:

- Acceso controlado
- Reglas basadas en control de acceso
- Seguridad Etiquetada
- Sistema ORACLE de Administración de Bases de Datos Comerciales
- Sistema ORACLE de Administración de Bases de Datos Gubernamentales
- FIREWALL para el nivel de aplicación
- FIREWALL para filtro de tráfico

2.3.3 ROL DEL CUERPO DE CERTIFICACIÓN CB DURANTE LA EVALUACIÓN

El CB esta activo en todos los estados de la evaluación, aunque el grueso del trabajo es realizado por el CLEF y el desarrollador. El CB confirma la seguridad objetivo ST y los programas de trabajo de la evaluación. Con excepción del nivel EAL1, el certificador cuida el inició de una tarea, de manera conjunta con el CLEF y el desarrollador, en orden, para discutir la evaluación y estar de acuerdo en el programa de actividades. Problemas potenciales pueden ser identificados y ellos determinan las acciones para remediarlos.

Un cuestionamiento más, antes de continuar:

2.3.4 ¿QUÉ ES UN OBJETIVO DE SEGURIDAD O SEGURIDAD OBJETIVO ST?

Es la especificación de las funcionalidades, seguridad y ambiente de trabajo, en el cual deben diseñarse.

Como evaluación de progreso, el Certificador monitorea y examina las actividades emprendidas, generando reportes así como posibles soluciones. Un objetivo clave del CB es, checar que la evaluación sea conducida de acuerdo con la metodología propuesta, presentada en el CC. La evidencia proporcionada debe respaldar las conclusiones de la evaluación y las pruebas apropiadas que justifiquen el nivel de seguridad solicitado.

El certificador puede atender el progreso de una o más evaluaciones, donde la línea de la evaluación es revisada, en evaluaciones complicadas y donde han sido acordados nuevos trabajos. El proceso de evaluación culmina en la preparación por parte del CLEF, del Reporte Técnico de Evaluación (ETR). Este reporte presenta todo lo encontrado por el CLEF y presenta además las evidencias. El ETR es enviado entonces al CB.

2.3.5 CERTIFICACIÓN

El certificador revisa el ETR y realiza comentarios acerca de las áreas en las cuales es necesario agregar explicaciones, pruebas de resultados, o que simplemente no es lo suficientemente claro. Toda la información evidencia proporcionada por los evaluadores es tomada como medida y prueba de los resultados, para comparar con la Seguridad Objetivo, esto para asegurar todos los objetivos a alcanzar. Cuando el certificador esta satisfecho con las evidencias presentadas por el CB, elabora un Reporte de Certificación y garantiza la **Certificación.**

2.3.6 RE-EVALUACIÓN Y CONSERVACIÓN DEL CERTIFICADO

Inevitablemente los productos IT desarrollados, se vuelven sensibles. El CB recomienda si una re-evaluación es necesaria, si un producto ha sido modificado. El trabajo involucrado puede ser mínimo durante la primera evaluación por la clasificación de los componentes del producto de acuerdo a su influencia en las características de seguridad. En todos los cambios hechos al producto evaluado, el desarrollador puede usar la clasificación, para determinar más fácilmente el impacto sobre la certificación e identificar las acciones apropiadas.

Las vulnerabilidades pueden ser descubiertas en productos que ya deben ser evaluados. En esos casos la práctica normal para el desarrollador es publicar una corrección. Cuando un producto se encuentra en el sistema de mantenimiento de certificación, la publicación del parche o corrección, no invalida su certificación. Es consecuencia de un nivel moderado de seguridad, no detectar, ni remover todas las vulnerabilidades. Hay que reconocer que la rápida evolución de los productos y entornos, introducen la posibilidad de que no todas las vulnerabilidades sean revisadas o imaginadas en el tiempo establecido originalmente para la certificación. Los países participantes en el desarrollo del CC están en vías de formalizar un proceso conservación de la certificación, comparable con el Sistema de Conservación del Certificado, ofrecido por el CB, para certificados ITSEC. Este mantenimiento es proyectado, esté bajo el control del desarrollador, directamente o vía el CLEF.

2.3.7 CLASES DE SEGURIDAD Y FUNCIONALIDAD

EL Common Criteria tiene 11 clases de funcionamiento y 10 clases de seguridad

Funcionalidad	***Seguridad***
Auditoria	Evaluación del perfil de protección (PP)
Soporte Criptográfico	Evaluación del objetivo de seguridad (ST)
Comunicaciones	Administración de la configuración
Protección de los datos de usuario	Distribución y Operación
Identificación y autenticación	Documento Guía
Privacidad	Suporte del Ciclo de Vida
Protección de las funciones de seguridad del TOE	Pruebas de conservación de la Seguridad
Utilización de Recursos	Evaluaciones
Administración de la seguridad	Evaluación de la Vulnerabilidad
Correcciones / canales de seguridad	

Tabla T2.2 Clases de Seguridad y Funcionalidad del CC

El CC tiene siete Niveles de Evaluación de la Seguridad (EALs), de EAL1 al máximo nivel EAL7. Estos tienen una correspondencia aproximada a los niveles del ITSEC como se muestra en la tabla T2.3

CC	EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
ITSEC	-	E1	E2	E3	E4	E5	E6

Tabla T2.3 Correspondencia entre los niveles de seguridad del CC e ITSEC

Estos paquetes de seguridad son diseñados para proporcionar un grupo balanceado de elementos de seguridad para uso general. Los niveles representan, niveles ascendentes de confidencia que

pueden ser ubicados en el objetivo de seguridad (TOE). Entre más alto es un nivel de seguridad, más grande es el rigor aplicado al TOE y sus requerimientos de seguridad, por ejemplo, intensificar el análisis y búsqueda de las vulnerabilidades en la seguridad.

En resumen los siete niveles presentan lo siguiente:

Sistemas EAL1. Evaluación de la Funcionalidad.

El análisis es soportado por pruebas independientes de una muestra de las funciones de seguridad, de acuerdo a la comprensión del comportamiento de la seguridad. EAL1 es aplicable a productos, donde la confidencia en correcta operación es requerida, pero la valoración sobre amenazas es baja. Este paquete de seguridad es particularmente adecuado para sistemas que dicen como es factible su funcionamiento sin la asistencia del desarrollador.

Sistemas EAL2. Evaluación Estructural.

Análisis de las actividades de las funciones de seguridad, especificación de interfaces y alto nivel de diseño de los subsistemas del TOE, son algunas de sus características. Existen pruebas independientes de las funciones de seguridad, se requiere evidencia al desarrollador, acerca de las pruebas aplicadas a evidencias obvias y "cajas negras".

Sistemas EAL3. Pruebas y Chequeo Metodológico.

El análisis es soportado por pruebas de "cajas grises", existe confirmación selectiva independiente de los resultados de las pruebas del desarrollador, buscando vulnerabilidades obvias. Es también necesario desarrollar control sobre el entorno y administración de la configuración del TOE. Este nivel es aplicable donde los requerimientos son en un nivel moderado de seguridad independiente, con una investigación minuciosa del TOE y su entorno, sin incurrir en una costosa o sustancial reingeniería.

Sistemas EAL4. Diseño, Evaluación y Revisión Metodológica.

El análisis es soportado por un nivel bajo de diseño de los módulos del TOE y un subconjunto de la implementación. La evaluación es soportada por una investigación independiente sobre vulnerabilidades obvias. Los controles de desarrollo son soportados por un modelo de ciclo de vida, identificación de herramientas y una administración automatizada de la configuración. Este nivel es aplicable donde un grado moderado de alto nivel de seguridad es requerido, aunque se puede incurrir en algunos costos de ingeniería en adiciones específicas de seguridad.

Sistemas EAL5. Diseño y Evaluación Semiformal.

El análisis incluye todo sobre la implementación. El aseguramiento es complementado por un modelo formal, una presentación semiformal de la especificación funcional, diseño de alto nivel y demostración semiformal de correspondencia. La investigación de vulnerabilidades debe incluir resistencia a ataques de penetración con potencial moderado. También es necesario el análisis del

canal secreto y diseño modular. Este nivel es aplicable donde los requerimientos son de un nivel alto de seguridad en un desarrollo planeado, asociado con una rigurosa metodología de desarrollo.

Sistemas EAL6. Evaluación y Verificación del Diseño Semiformal.

El análisis es sustentado por una metodología modular, para diseño y presentación estructurada de la implementación. La investigación independiente de las vulnerabilidades puede asegurar resistencia a ataques de penetración con un alto potencial de ataque. Debe haber una búsqueda sistemática para los canales secretos. Fomentar la consolidación de ambientes de desarrollo y controlar la administración de la configuración. Este nivel es aplicable donde una seguridad especializada del TOE es requerida para situaciones de alto riesgo

Sistemas EAL7. Evaluación y Verificación del Diseño Formal

Aquí, el modelo formal es complementado por una presentación formal de las especificaciones funcionales y alto nivel de diseño, mostrando correspondencia. Se requiere de pruebas sobre las evidencias del desarrollador en "cajas blancas" y confirmación independiente completa de los resultados de las pruebas. Este nivel es apropiado donde un nivel especializado de seguridad del TOE es requerido para situaciones de un extremoso nivel de riesgo.

En apoyo de las tres partes del CC citadas anteriormente, se anticipa, que otros tipos de documentos pudiesen ser publicados, incluyendo material con técnicas fundamentadas y documentos guía.

Finalmente, se muestra de manera tabular (Tabla T2.4, Tabla T2.5 y Tabla T2.6), una comparación o interrelación entre los criterios de los tres más importantes organismos: CC, TCSEC e ITSEC, base de nuestro estudio.

Common Criteria	**US TCSEC** "Orange Book"	**European** **ITSEC**	**Descripción**
EAL0: Seguridad Inadecuada	**D:** Mínima Protección	**E0:** Seguridad Inadecuada	No existe protección contra amenazas
EAL1: Evaluación de la Funcionalidad			Protección contra amenazas de bajo potencial
EAL2: Evaluación de la Estructura	**C1**: Protección de Seguridad Discrecionaria	**E1**	Bajo, para niveles moderados de seguridad. Aseguramiento independiente
EAL3:	**C2:**		Nivel moderado de seguridad. Inspección del análisis realizado por el desarrollador, independientemente de las

Evaluación y Chequeo de la Metodología	Protección de Acceso Controlado	**E2**	pruebas selectivas
EAL4: Diseño, Evaluación y Revisión Metodológica	**B1:** Protección de Seguridad Etiquetada	**E3**	Moderado, para altos niveles de seguridad. Inspección del análisis y bajo nivel de diseño de los módulos. Independiente de la evaluación de vulnerabilidades obvias
EAL5: Diseño y Evaluación Semiformal	**B2:** Protección estructurada	**E4**	Alto nivel de seguridad. Inspección del análisis. Modelo Formal. Especificación funcional y alto nivel de diseño. Evaluación independiente de las vulnerabilidades y canal secreto
EAL6: Diseño y Evaluación Verificado Semiformalmente	**B3:** Dominios de Seguridad	E5	Protección de activos de alto valor. Demostración del diseño modular y por capas e implementación estructurada
EAL7: Diseño y Evaluación Verificado Formalmente	**A1:** Diseño Verificado	**E6**	Riesgo extremadamente alto y/o protección de activos de gran valor, además de las características de niveles previos. Demostración de diseño basado en modelo formal (Ej. matemático) Demostración de mínima complejidad

Tabla T2.4 Interrelación entre ITSEC, TCSEC y CC [6]

Clases	**Propósito**
F1	Derivado de la clase C1 del Orange Book
F2	Derivado de la clase C2 del Orange Book
F3	Derivado de la clase B1 del Orange Book
F4	Derivado de la clase B2 del Orange Book
F5	Derivado de la clase B3/A1 del Orange Book
F6	Clase para sistemas con altos requerimientos de integridad (en contraste a la confidencialidad) para datos y programas. Es particularmente apropiado para bases de datos
F7	Clase para sistemas con altos requerimientos para cualquier sistema completo o una función especial de un sistema. Es apropiado para sistemas de control de proceso.
F8	Clase para sistemas con altos requerimientos de protección de integridad de datos durante las comunicaciones de datos.
F9	Clase para sistemas con alta demanda en la confidencialidad de datos durante comunicaciones de datos. Apropiado en sistemas criptográficos.
F10	Clase para redes de alta demanda en la confidencialidad e integridad de la información ha ser comunicada. Apropiada cuando la información sensitiva necesita ser comunicada sobre redes

	inseguras.

Tabla T2.5 Clases de Funcionalidad del ITSEC en analogía con TCSEC

Nivel de seguridad	Propósito
E1	Evaluación
E2	Control de configuración y distribución controlada: a grandes rasgos equivalente a la clase C2 de seguridad del Orange Book (OB)
E3	Acceso para diseño detallado y código fuente; aproximadamente equivalente a la clase de seguridad B1 del OB.
E4	Análisis riguroso de vulnerabilidad; aproximadamente equivalente a la clase de seguridad B2 del OB.
E5	Demostrar correspondencia entre diseño detallado y código fuente; Aproximadamente equivalente a al clase B3 del OB.
E6	Modelos y descripciones formales, ligados por correspondencias formales; aproximadamente equivalente a la clase de seguridad A1 del OB.

Tabla T2.6 Niveles de Seguridad del ITSEC [6]

2.4 OPEN-SOURCE SECURITY TESTING METHODOLOGY MANUAL

Este nuevo "estándar" pretende serlo para probar la seguridad en Internet.

Es un conjunto de reglas para evaluar una sólida penetración, "*hackeo*" ético y análisis de información segura, incluyendo el uso de herramientas de dominio público, para evaluar la estandarización de la seguridad y la mejora de herramientas automatizadas para la evaluación de vulnerabilidades. Se pretende además proporcionar una metodología estándar, para una minuciosa evaluación de la seguridad en una organización que maneja información a través de Internet. Dentro de esa metodología, se pretende que el estándar sea público, para evaluar la seguridad en Internet, así como utilizarlo como base para todas las metodologías de evaluación de seguridad sobre Internet conocidas y por conocer.

La meta final es establecer un estándar en la metodología de evaluación, el cual cuando sea utilizado, de forma manual o automática, proporcione como resultado los requerimientos de seguridad operacional necesarios para asegurar la presencia en Internet. El resultado indirecto es crear una disciplina que puede actuar como punto central en todas las pruebas de seguridad en Internet sin importar el tamaño de la red, el tipo de sistemas o las aplicaciones de Internet.

En este estándar, el evaluador de la seguridad realiza evaluación sobre seguridad de la información, aislamiento, seguridad de los sistemas, evaluación de políticas, y evaluación de las estrategias de defensa en la mercadotecnia y negocios.

Basado en BS7799 y en su equivalente internacional ISO-17799 sobre evaluación de seguridad en la información

Un inconveniente es que quizá no aplique en todos los países, debido a que en este estándar se considera aislamiento de datos.

Una prueba de la seguridad se realiza con dos tipos de ataque. Un ataque pasivo es a menudo una forma de colección de datos que no influencie ni viole directamente sobre el sistema o la red objetivo, y un ataque intruso sin embargo, viola el sistema o la red objetivo y se puede registrar y utilizar alarma en el sistema en cuestión.[8]

El proceso de una prueba de seguridad se concentra en la evaluación de las áreas siguientes:

Visibilidad

Detecta la presencia en Internet, esto incluye, pero no limita, accesos filtrados, los tipos de sistemas, la configuración, las aplicaciones, direcciones de correo, nombres de empleado, las habilidades del nuevo sistema de administración, la circulación de sus productos de software, los Websites visitados por sus empleados y todo lo que descarga.

Acceso

Visita a Internet. Acceso a una página Web, a un negocio electrónico, a un servidor punto a punto para contestar la correspondencia, a un servidor de DNS, un flujo de video, o cualquier cosa en las cuales un servicio o una aplicación utiliza la definición del casi público, donde una computadora obra recíprocamente con otra computadora dentro de su red. La limitación del acceso significa negar todo, excepto a quien o que expresamente sea parte del negocio.

Seguridad

La confianza es el concepto más importante de la seguridad en Internet. Es una medida de cuánto, la gente puede depender de lo que ofrece el sistema. La confianza depende de la clase y de la cantidad de autenticación, de la no repudiación, del control de acceso, de la responsabilidad, del secreto de los datos, y de la integridad de los datos empleada por el sistema(s). La confianza es a veces la base para un servicio, por ejemplo cuando una computadora se conecta con otra. Algunas asociaciones que proporcionan confianza incluyen VPNs, PKIs, los conectores de HTTPS, de SSH, de B2B, la base de datos a las conexiones del servidor, el E-mail, el Web del empleado que navega, o cualquier comunicación entre dos computadoras que cause interdependencia entre ellas, es decir, servidor a servidor, servidor - cliente, o P2P.

Alarma

La alarma, es la notificación oportuna y apropiada de las actividades que violan o intentan violar visibilidad, el acceso, o la confianza. Esto incluye pero no se limita al análisis de archivos comunes, acceso diario, vigilar el tráfico, los sistemas que detectan intrusión, o "sniffeo". El "alarmar" es a menudo la conexión más débil de medidas de seguridad apropiadas.

La metodología se analiza en parámetros y tareas. Los parámetros son el flujo de la metodología de un punto a otro en Internet. Cada parámetro tiene una entrada y una salida. La entrada es la información utilizada en la ejecución de cada tarea. La salida es el resultado de tareas terminadas. La salida puede o no, ser datos analizados (también conocidos como capacidad) que sirven como entrada para otro parámetro. Puede incluso ser el caso. que la misma salida sirva como entrada para más de un parámetro tal como direcciones IP o nombres del dominio.

Algunas tareas no producen salidas. Los parámetros que no tienen ninguna entrada de información pueden ignorarse durante la evaluación. Los parámetros ignorados no indican necesariamente una evaluación inferior; más bien, podrían indicar seguridad superior.

En los parámetros que no tienen salida, el resultado puede significar cualquiera de las tres cosas siguientes:

- Las tareas no fueron realizadas correctamente.
- Las tareas revelaron seguridad superior.
- Los datos, resultado de la tarea, se han analizado incorrectamente.

Es vital que exista imparcialidad en la ejecución de las tareas de cada parámetro. *"Buscar algo que no se tiene ninguna intención de encontrar, puede conducir a encontrar exactamente lo deseado"*. En esta metodología, cada parámetro comienza exactamente con una entrada y una salida por la razón de mantener baja la predisposición.

El tiempo es relativo. Proyectos más grandes significan más tiempo consumido en cada parámetro y en cada tarea. La cantidad de tiempo permitida antes de regresar datos de salida, depende del evaluador y del alcance de la evaluación. La evaluación apropiada es un equilibrio del tiempo y de la energía, donde tiempo es dinero y la energía es el límite de la potencia del hombre y de la máquina.

La evaluación de la seguridad es un esfuerzo estratégico. Mientras de diferentes maneras y con diferentes herramientas se puedan evaluar muchos de los parámetros, existirán pocas variaciones en el orden en el cual ellos se ejecutan.

Puntos existentes en Internet son en cada uno de los cuales. una organización puede interactuar con Internet. Esos puntos son desarrollados para ofrecerse como parámetros en el flujo de la metodología. Algunos de esos parámetros son:

- Evaluación de la red
- "Escaneo" de puertos
- Sistema de huella digital
- Evaluación de fuga inalámbrica
- Prueba de servicios
- Rastreo de la Web
- Rastreo de correos
- Servicio de nombres
- Documentos visibles
- Antivirus y Troyanos
- "Escaneo" automático de vulnerabilidades redundantes

- Investigación de Exploits
- Manual de Evaluación y verificación de vulnerabilidades
- Aplicación de pruebas
- Firewall y evaluación de la lista de control de acceso (ACL)
- Revisión de políticas de seguridad
- Evaluación del sistema de detección de intrusos
- Evaluación de extensiones telefónicas privadas y envío de voz
- Ingeniería amigable
- Evaluación de los sistemas de seguridad
- "Crackeo" de Password
- Negación de Servicio
- Cookies y análisis de gusanos en la red
- Revisión rutinaria del servidor
- Metodología

El flujo de la metodología parte de la evaluación de la red, hasta el final de reporte. Un ejemplo de este flujo permite una separación entre la colección de datos y la evaluación de los datos colectados. El flujo puede también determinar el punto preciso para saber cuando extraer y cuando insertar datos.

En esta metodología, cada parámetro tiene una relación con el anterior y con el siguiente. La evaluación inicia con una entrada que es normalmente la dirección del sistema a ser evaluado, y finaliza con el inicio de la fase de análisis y el reporte final.

Los parámetros son las variables en la evaluación. Los parámetros requieren una entrada para ejecutar las tareas. Las tareas son las pruebas de seguridad que se ejecutan dependiendo de las entradas. Los resultados de las tareas pueden ser inmediatamente analizados como un resultado procesado. De cualquier manera, son consideradas salidas de los parámetros. Estas salidas son frecuentemente la entrada de los parámetros siguientes, o en ciertos casos pueden ser las entradas de parámetros previos. (Figura 2.4.1)

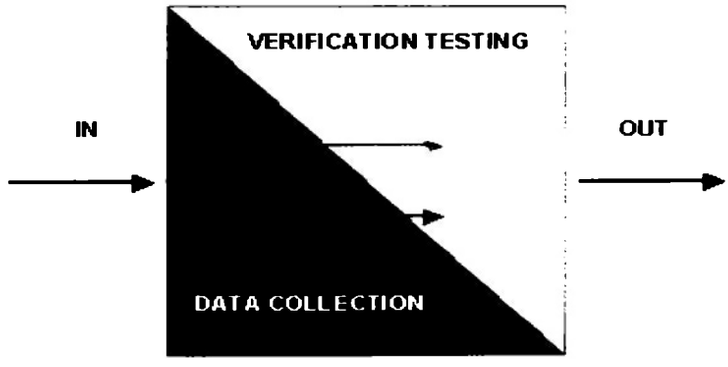


Figura 2.4.1 Flujo de datos en el OSSTMM [8]

2.4.1 INTERDEPENDENCIA DE PARÁMETROS

Existe la posibilidad de ejecutar ciertos parámetros de manera paralela. Por ejemplo, las pruebas de **IDS** no interfieren con el **Wardialing**, además, nunca las pruebas dependen de los resultados

de la otra. Sin embargo, ambos dependen de la revisión que se haga de las políticas de seguridad para definir ciertos parámetros.

Los parámetros son los pasos para una minuciosa y completa evaluación de la seguridad en Internet, en un sistema abierto.

Para llevar al cabo una minuciosa evaluación de la red se debe incluir:

- Respuesta a los servidores de nombres
- Examinar la parte externa de la red
- Examinar las rutas de la organización objetivo
- Fuga de información
- "Escaneo" de Puertos
- Es la prueba de invasión de los puertos del sistema en la capa de transporte en la red.

Algunas de las tareas para desarrollar un minucioso escaneo de puertos son:

- Chequeo de errores
- Enumeración de sistemas
- Enumeración de Puertos
- Encapsulamiento y tuneleo de protocolos
- Protocolos de Ruteo
- Sistema de Huella digital

Es la prueba activa de un sistema, distinguir sistemas únicos de sistemas operativos y niveles de versión

2.4.2 INVESTIGACIÓN DE SERVICIOS

Examen activo de "la escucha" detrás del servicio.

2.4.3 ESCANEO AUTOMATIZADO DE VULNERABILIDADES

La evaluación de las vulnerabilidades utilizando herramientas automatizadas, es una manera eficiente para determinar hoyos existentes y niveles de corrección en un sistema.

2.4.4 INVESTIGACIÓN DE EXPLOITS

Involucra investigación en línea de bases de datos y listas de envío específicos para el sistema a ser examinado.

2.4.5 MANUAL DE EVALUACIÓN Y VERIFICACIÓN DE VULNERABILIDADES

Este parámetro es necesario para eliminar falsos positivos, ampliando el alcance del "hackeo" y descubriendo el flujo de datos en la salida de la red. El manual de prueba está destinado para todo tipo de usuarios de las computadoras: creativos, expertos o ingenuos en la evaluación de la red.

2.4.6 PRUEBAS DE APLICACIÓN

Este parámetro se refiere a la evaluación de aplicaciones (no demonios) accesibles a partir de la red. Esas aplicaciones pueden ser escritas en cualquier lenguaje o script. Generalmente proporcionan un proceso de negociación.

2.4.7 FIREWALL Y PRUEBAS DE LA ACL

Este Parámetro ha sido diseñado para asegurarse de que solo quien tenga acceso permitido a la red puede hacerlo.

2.4.8 EVALUACIÓN DEL SISTEMA DE DETECCIÓN DE INTRUSOS

Prueba enfocada al funcionamiento y sensibilidad de sistema de detección de intrusos

2.4.9 REVISIÓN DE LAS POLÍTICAS DE SEGURIDAD

Son un documento que esquematiza la manera de suavizar los riesgos en una organización, mediante el uso de tipos específicos de tecnología. Esas tareas, requieren que las pruebas de verificación de vulnerabilidades sean completas y que todas las revisiones técnicas sean realizadas.

2.4.10 DOCUMENT GRINDING

Este parámetro es importante en la verificación de gran cantidad de información y pertenece a varios niveles de información considerada segura. Lo que se busca es obtener: un perfil de la organización, perfil de los empleados clave y un perfil de las redes de la organización.

2.4.11 INTELIGENCIA COMPETITIVA

Explorar IC es buscar información de la presencia de Internet que pueda ser analizada como capacidad del negocio. Establece una medida de la política de seguridad en la planeación de una red futura.

2.4.12 CRACKEO DE PASSWORDS

Es el proceso de validación de la fuerza de una contraseña a través del uso de herramientas de recuperación automática de contraseñas, que dan a conocer cualquier debilidad de los algoritmos criptográficos, su implementación incorrecta o contraseñas débiles debidas a factores humanos.

2.4.13 NEGACIÓN DE SERVICIOS

Es una situación, donde cualquier factor, accidental o no, impide el funcionamiento esperado de un sistema. En ocasiones podrá funcionar exactamente como ha sido planeado, pero sin el alcance o aplicando los parámetros impuestos.

2.4.14 REVISIÓN DE POLÍTICAS DE PRIVACIDAD

Este es el punto más importante de las posturas de un cliente en una organización. Esta política puede ser visible. En casos donde la política no existe, es necesario utilizar legislación privada local de la organización objetivo.

La **prueba de PBX** puede ayudar a prevenir el robo de información por fraude.

* Los puntos anteriores describen de manera breve pero concisa, lo más relevante de este nuevo estándar para la evaluación de la seguridad de un sistema, OSSTMM. [8]

De manera resumida podemos decir que cada uno de los organismos internacionales en cuanto a certificación de seguridad computacional, analizados a lo largo de este capítulo, proporcionan una perspectiva y conocimiento más amplio y reforzado en cuanto a verificación o evaluación de la seguridad.

Cada uno de ellos tiene marcadas líneas de acción, pues sus objetivos de seguridad difieren de un organismo a otro, mientras que unos se preocupan por la integridad, otros la dejan de lado por empeñarse en preservar la confidencialidad y/o disponibilidad. Algunos se centran en sistemas sobre ambiente distribuido, otros intentan abarcar redes e Internet, sistemas operativos, etc.

Lo único cierto es que los cinco organismos aquí analizados "cooperaron" para tener ideas básicas fundamentales para la creación de nuestra propuesta de lista de chequeo, cuya lista es uno de los objetivos primordiales de este trabajo, claro además de la puesta en práctica sobre algunas aplicaciones, caso en especifico, aplicaciones ejecutándose sobre Jaguar.

3 "DESARROLLO" DE SISTEMAS SEGUROS

En este capítulo, se pretende mostrar un panorama en cuanto a tipos de ataques y desarrollo de sistemas seguros, así como mejores prácticas de desarrollo, para dar mayor realce a la importancia de la seguridad, tanto física, como lógica de la información. La información que ha continuación se muestra, es extraída principalmente de la red mundial, para percibir los niveles y tipos de ataque existentes hoy día, y en muchas ocasiones inimaginables, incluso los criterios organizacionales que nos están sirviendo de base fundamentada para la generación de nuestros criterios y estudio, recalcan que a los criterios que describen, le tienen asignado un tiempo establecido para realizar un estudio que va desde un análisis de las vulnerabilidades obvias, hasta la resistencia a ataques directos y de alto impacto propios del tipo de aplicación. También se puede enfocar este análisis en una búsqueda sin un plan preestablecido (estándar OSSTMM), porque de esa manera las evaluaciones e investigaciones no son tendenciosas a preferencias y/o experiencias de los evaluadores, provocando con esto, obtener de manera más eficiente el resultado.

Por lo anterior, hemos dividido el presente capítulo en tres temas que consideramos representativos:

- *Mejores prácticas*
- *Sistemas Immunix*
- *Programación Extrema*

3.1 MEJORES PRÁCTICAS

Contar con seguridad en las transacciones es de suma importancia en esta época de rápida expansión comercial, administración de las redes de computadoras y la emergente economía en Internet. Los inherentes cambios en tecnologías de seguridad tienen que ser prioritarios en todas las compañías que utilizan tecnología de la información.

El término *seguridad computacional,* es una generalización para una colección de tecnologías que ejecutan tareas específicas relacionadas a seguridad de datos, información y sistemas. Utilizando esas tecnologías de manera eficiente para asegurar una red corporativa, requiere que estén integradas en un plan completo de seguridad. El proceso de planeación para su correcta implementación involucra:

- Obtener una detallada comprensión de los riesgos potenciales del ambiente (por ejemplo. virus, hackers y desastres naturales).
- Elaborar un análisis proactivo de las consecuencias y contramedidas, para asegurar las fisuras con relación a riesgos.
- Creando una estrategia cuidadosamente planeada para integrar medidas de seguridad de todos los aspectos en redes empresariales, basado en compresión y análisis.

3.1.1 MEJORES PRÁCTICAS PARA SEGURIDAD INFORMÁTICA EMPRESARIAL [9]

Es una colección de documentos ("White Papers"), enfocados en los diferentes aspectos de seguridad en redes empresariales. Los documentos son agrupados dentro de tres categorías generales que reflejan los diferentes niveles de conocimiento necesario para crear e implementar un concepto exitoso de seguridad. La estructura además permite acercarse a la seguridad del sujeto basada en sus áreas individuales de interés. Los niveles y relaciones correspondientes son:

Respaldo de Seguridad.

Compuesta de artículos que proporcionan una comprensión de temas de seguridad y contramedidas generales independientes de la tecnología.

- Amenazas a la seguridad
- Estrategias de Seguridad
- Planeación de la seguridad

Básicas de la Seguridad.

Algunas series que inician con un artículo describiendo un ejemplo de una arquitectura de seguridad y continúa con artículos que tratan cada entidad en la arquitectura de manera separada, discutiendo cómo ellos adecuan la arquitectura de manera general.

- Organizaciones que construyen arquitecturas seguras
- Consideraciones en aseguramiento y sistemas
- Consideraciones de seguridad para la autoridad administrativa

Mejores Prácticas de Seguridad.

Un conjunto final de artículos que discuten algunos escenarios de la vida real e implementan soluciones.

- Autenticación
- Seguridad y disponibilidad de datos en autoridades administrativas
- Administración de resolución de nombres
- Seguridad de IP para sistemas de comunicación local
- Seguridad y disponibilidad de datos para sistemas finales
- Monitoreo y auditoria de sistemas finales

Estos documentos dan a los lectores, una base sólida sobre cómo construir una estrategia de seguridad en su empresa.

Acerca de organizaciones que construyen seguridad

Los artículos sobre las mejores prácticas de seguridad en las empresas están basados en una metodología que considera seguridad de datos en términos de entidades de seguridad de una empresa. Para implementar seguridad efectiva, esta metodología primero considera toda la estructura de red, y de manera separada la estructura dentro de entidades discretas de seguridad, así como determinar la seguridad expuesta e implementar seguridad en cada entidad. Estas entidades forman la base para las organizaciones constructoras de arquitectura de seguridad.

Estas organizaciones dividen la estructura de seguridad de una empresa en las siguientes entidades:

- Sistemas finales (dispositivos de computación con sistema operativo)
- Sistemas de comunicación local (funcionalidad en redes)
- Administración de la Autoridad (Control centralizado de la seguridad)
- Redes privadas (redes compartidas entre compañías)
- El Internet

La figura 3.1.1 muestra la relación entre las anteriores entidades.

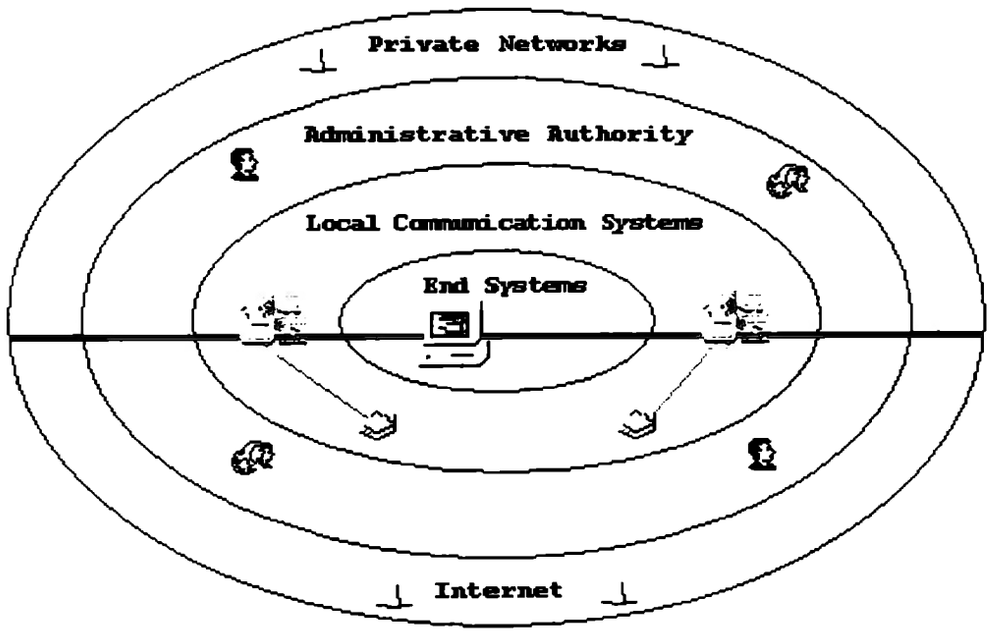


Figura 3.1.1 Estructura de seguridad empresarial [9]

Amenazas a la Seguridad

Las grandes amenazas a los sistemas de computación y su información provienen de humanos, a través de acciones que son maliciosas o ignorantes, o ambas. Cuando la acción es maliciosa, el motivo u objetivo es generalmente el ataque.

Para lograr sus objetivos, los atacantes utilizan técnicas y métodos, cada vez más sofisticados, para explotar vulnerabilidades en las políticas y sistemas de seguridad.

Seguridad computacional

Significa protección a la información y a los sistemas. Es el trato con la prevención y detección de acciones, son autorizadas por usuarios de una computadora. Esta definición además de los aspectos tan conocidos de confidencialidad, privacidad e integridad, implica tener conocimiento de la información y su valor, para tomar medidas de protección. Una clasificación fuerte sobre medidas de protección en seguridad computacional es:

- Prevención. Tomar medias que prevengan que la información sea dañada, alterada o robada. Estas medidas pueden ir desde bloquear el acceso al lugar donde se encuentre el servidor, hasta modificar las políticas de alto nivel de seguridad.
- Detección. Tomar medidas que permitan detectar cuando la información ha sido dañada, alterada o robada, como ha sucedido y quien lo ha provocado. Existen diversas herramientas disponibles para la detección de intrusos, daños o alteraciones y virus.
- Reacción. Tomar medidas que permitan la recuperación de la información, aún si la información fue dañada o perdida.

Estas medidas son buenas, pero si no se conoce como la información puede ser comprometida, no se pueden tomar medidas para protegerla.

Amenazas, Ataques y Vulnerabilidades de la Seguridad

Como se ha venido mencionando, la información es el activo clave de las organizaciones. Las compañías obtienen una ventaja competitiva por conocimiento de cómo utilizar la información. Las amenazas vienen de otros, quienes quieren adquirir la información o limitar las oportunidades de negocios, mediante la interferencia del proceso normal de las empresas.

Los administradores necesitan conocer los diversos aspectos de seguridad, para desarrollar sus medidas y políticas para proteger los activos y limitar las vulnerabilidades.

Objetivo + Método + Vulnerabilidad = Ataque

Fig. 3.1.2 División de las amenazas

Amenazas naturales.

Pocas medidas pueden ser implementadas contra estas amenazas. La mejor metodología contra estos desastres, es establecer planes de recuperación y planes de contingencia.

Amenazas Humanas.

Pueden existir dentro y fuera de la organización. Desde empleados descontentos hasta gente que desea dañar la información de la empresa.

Los más peligrosos son usualmente parte de la organización, ya que ellos conocen varios de los códigos y medidas de seguridad ya establecidas. Las personas de confianza tienen probablemente objetivos específicos y tienen acceso legítimo a los sistemas. Ellos pueden introducir caballos de Troya, virus o *worms*, dado que ellos pueden mirar dentro de los archivos del sistema, pueden afectar todos los componentes. Por un simple vistazo a través del sistema, la información confidencial puede ser revelada. Los caballos de Troya son una amenaza para la integridad y la confidencialidad de la información en el sistema. Los atacantes internos pueden también afectar la disponibilidad por sobrecarga en el procesamiento o capacidad de almacenaje de los sistemas, o incluso por baja total.

Los métodos comunes para obtener acceso a sistemas incluyen, robo de contraseñas, explotación de debilidades conocidas, engaños en la red e ingeniería social.

Usuarios, encargados en la entrada de datos, operadores de sistema y programadores, son los que frecuentemente generan errores no intencionales que contribuyen a los problemas de la seguridad, directa o indirectamente. Algunas veces el error es una amenaza, tal como error en la entrada de

datos o en la programación, lo cual colisiona el sistema. En otros casos, los errores generan vulnerabilidades. Los errores pueden ocurrir en todas las fases del ciclo de vida de un sistema.

Modelos teóricos para la determinación de amenazas

Modelo teórico que puede ser utilizado para determinar las diversas amenazas, objetivos, métodos y vulnerabilidades utilizadas en un ataque.

El **Anexo K,** muestra estos modelos teóricos que pueden ser utilizados para la determinación de amenazas y vulnerabilidades principalmente en un ataque.

Estrategias de Seguridad

Las actividades que a continuación se describen, generalmente requerirán una actualización periódica o revisión apropiada. Esos cambios son necesarios cuando las configuraciones y otras condiciones y circunstancias cambian significativamente, o cuando las políticas o regulaciones de la organización requieren cambios.

Establecer un efectivo conjunto de controles y políticas de seguridad requiere utilizar estrategias para determinar las vulnerabilidades existentes en los sistemas de cómputo. El estado actual de esos elementos puede ser determinado por una revisión de la lista que a continuación se mencionará. La revisión debe prestar atención a las áreas donde se carece de políticas, así como también examinar los documentos ya existentes:

- Políticas de seguridad física en cómputo, tales como controles de acceso
- Políticas de seguridad en redes (por ejemplo, políticas para Internet y correo electrónico)
- Políticas de seguridad de datos (controles de acceso y controles de integridad)
- Planes de contingencia y pruebas de recuperación de desastres
- Conocimiento y conciencia de seguridad en cómputo
- Políticas de administración y coordinación de la seguridad de cómputo

Otros aspectos que en ocasiones son considerados son:

- Contraseñas del BIOS
- Contraseñas para configuración del Router
- Documentos de control de acceso
- Otras contraseñas de administración de dispositivos

Identificación de Activos y vulnerabilidades para amenazas conocidas

Evaluar las necesidades de una organización, también incluye determinar las vulnerabilidades para amenazas conocidas. Esta evaluación conlleva reconocimiento de tipos de activos que una organización posee, los cuales sugieren los tipos de amenazas contra las que deben protegerse.

Identificar probables métodos, herramientas y técnicas de ataque

Listar las amenazas, ayuda a los administradores de seguridad a identificar los diversos métodos, herramientas y técnicas que pueden ser utilizadas en un ataque. Los métodos pueden ir desde virus y worms, hasta "crackeo" de contraseñas y comercio electrónico. Por tanto es importante que los administradores de la seguridad se actualicen en el área, porque nuevos métodos, herramientas y técnicas para evadir las medidas de seguridad son constantemente diseñados.

Establecimiento de Estrategias Proactivas y Reactivas

Para cada método, el plan de seguridad debe de incluir estrategias proactivas así como también estrategias reactivas.

Las estrategias proactivas o preataques, son un conjunto de pasos que ayudan a minimizar las vulnerabilidades existentes en las políticas de seguridad y desarrollo de planes de contingencia. Determinando el daño que un ataque podría causar en un sistema y las debilidades y la explotación de vulnerabilidades durante un ataque (determinar el daño que un ataque puede causar, determinar las vulnerabilidades y debilidades que un ataque explota, minimizar las vulnerabilidades que son determinadas para ser puntos en el sistema, para un tipo de ataque específico).

Las estrategias reactivas o post-ataque ayudan al personal de seguridad a evaluar los daños causados por un ataque, reparando el daño o implementando un plan de contingencia que desarrolle la estrategia reactiva, documento y aprendizaje a partir de la experiencia.

Experimentación (Evaluación)

Es el último elemento de la estrategia de seguridad. Ejecutar simulación de ataques en un laboratorio, permitiría ubicar vulnerabilidades. En ocasiones no existe en todas las organizaciones la posibilidad de hacer pruebas, lo cual disminuye la probabilidad de éxito de las políticas de seguridad. La experimentación es un factor importante que podría incluirse en la evaluación de aplicaciones que propone este trabajo.

Ciertos ataques, tales como desastres naturales, no pueden ser evaluados, pero a través de una simulación, podría evaluarse: que pasaría si todos los servidores son dañados, cual es la capacidad de reacción del personal y administradores de seguridad y averiguar en cuanto tiempo la organización funcionaria de manera normal.

Algunas de las actividades que pueden realizarse, para determinar los posibles daños, como consecuencia de un ataque son:

- Simular un ataque de virus, vía correo electrónico, ver que daño causa y como recuperarse de esa situación.
- Utilizar la ingeniería social para adquirir nombre de usuario y contraseña de un empleado ingenuo y observar que tanto accede.
- Simular un incendio en el "cuarto" donde se encuentra el servidor, medir el tiempo perdido y el tiempo que se tardan en recuperarse.

- Simular un ataque de virus malicioso. Notar el tiempo necesario para recuperar una computadora, y multiplicarlo por el número de computadoras infectadas, para averiguar el tiempo perdido y la baja en productividad.

A continuación mencionaremos una lista de posibles cuestionamientos con respecto a vulnerabilidades. Ésta, representa algunos de los existentes, además incluyen ejemplos en las áreas como: seguridad datos y red:

- ¿Los controles de acceso e integridad, así como los procedimientos de respaldo son los adecuados para limitar los ataques?
- ¿Existen políticas y procedimientos de privacidad, qué los usuarios deben acatar?
- ¿Qué controles de acceso a datos (autorización, autenticación e implementación) existen?
- ¿Qué responsabilidad existe por parte del usuario, sobre administración de datos y aplicaciones?
- ¿Están definidas las técnicas de administración de acceso directo a dispositivos de almacenamiento? ¿Cuál es el impacto en la integridad del archivo del usuario?
- ¿Existen procedimientos para la manipulación de datos sensibles?
- ¿Qué tipos de controles de acceso (Internet, conexiones de red) existen?
- ¿Existen procedimientos de autenticación? ¿qué protocolos de autenticación son utilizados para LAN's, redes de banda ancha y servidores "dialup"? ¿quién es el responsable de la administración de la seguridad?
- ¿Qué tipo de medio de red, por ejemplo, cables, switches y ruteadores son utilizados? ¿Qué tipo de seguridad tienen?
- ¿Existe seguridad implementada en archivos y servidores de impresión?
- ¿La organización utiliza encripción y criptografía para usos sobre Internet, VPN's, sistemas de correo electrónico y accesos remotos?
- ¿Se han establecido las redes conforme a los estándares?

Un detalle que no debe dejarse de lado es, el no ser demasiado riguroso en la implementación de controles, porque la disponibilidad de la información se convertiría en el nuevo problema. Debe haber un balance entre controles de seguridad y acceso a la información. La información debe ser tan libre como sea posible para usuarios autorizados.

El proceso de evaluación y afinación de políticas y controles de seguridad es un proceso iterativo. El equipo encargado de la respuesta a incidentes, debe involucrar esfuerzos proactivos sobre seguridad como:

- Desarrollo de guías sobre el manejo de incidentes
- Identificación de herramientas de software para responder a incidentes
- Investigación y desarrollo de otras herramientas de seguridad computacional
- Dirigir actividades de conocimiento y adiestramiento.
- Desarrollar investigación sobre virus
- Dirigir estudios sobre sistemas de ataque

3.1.2 METODOLOGÍAS PARA LA DEFINICIÓN DE ESTRATEGIAS DE SEGURIDAD

Para mantener la seguridad de los sistemas y aplicaciones, es de suma importancia que la implantación e implementación de estrategias sean con el más alto nivel de diseño, debido que de ello depende "la vida" de todos los activos de la empresa.

El **Anexo I**, se hace una ejemplificación basada en gráficos, que permite visualizar de una manera general, pero de suma ayuda para la protección de la información.

Otros tópicos que consideramos importantes para nuestro estudio, es el saber los principales tipos de políticas de seguridad:

- Políticas de Contraseñas
- Responsabilidades administrativas
- Responsabilidades del usuario
- Políticas de correo electrónico
- Políticas de Internet
- Políticas de respaldo y almacenamiento

En un sistema que tiene implementado un mecanismo de autenticación basado en contraseñas, el que éstas sean vulnerables, comprometen al sistema en cinco aspectos importantes:

- Una contraseña debe ser inicialmente asignada a un usuario al pertenecer al sistema
- La contraseña de un usuario debe cambiar periódicamente
- Los usuarios deben introducir su contraseña, al momento de autenticarse
- Los usuarios no deben revelar sus contraseñas a nadie, incluyendo administradores y ejecutivos.

Estas políticas, normalmente dependen de las necesidades de la organización, tal vez algunas especifiquen, tamaño mínimo, no espacios en blanco, tiempo mínimo y máximo de duración, evitar reúso de contraseñas, si los usuarios generan sus contraseñas con todas estas políticas, se asegura que los usuarios utilizan caracteres específicos en sus contraseñas, lo cual las hace más difícil romperlas.

Por otro lado, las organizaciones también necesitan de políticas para el correo electrónico para establecer una guía general en áreas como:

- Uso del correo electrónico para conducir negocios oficiales
- El uso del correo electrónico para negocios personales
- Control de acceso y protección confidencial de mensajes
- Administración y retención de mensajes por correo electrónico

Algunas formas de prevenir accidentes son:

- Capacitar a los usuarios cuando las cosas funcionan mal y como recuperarse

- Configurar el software de correo de tal manera que el comportamiento por defecto sea él más seguro
- Utilizar software de correo que siga fielmente las convenciones y protocolos de Internet.
- Utilizar algoritmos de encripción, para digitalmente firmar los mensajes, puede prevenir usurpación. Encriptar el contenido de los mensajes o el canal de transmisión puede prevenir escucha.

En cuanto a servidores Web, existen muchas áreas por asegurarse: sistemas operativos, software del servidor de Web, servidor de scripts y otros.

En cuanto a *respaldos*, estos son necesarios, cuando la información almacenada en el sistema es de valor e importancia. Los respaldos son importantes por varias razones:

- Falla en el hardware
- Falla en el Software
- Errores de usuario
- Errores del administrador
- "Hackeo" y vandalismo
- Robo
- Desastres naturales
- Otros desastres

Como pudimos observar a lo largo de este capítulo se han presentado diversidad de conceptos y prácticas necesarias para proporcionar seguridad en nuestro equipo, sistemas y aplicaciones. Todo de manera muy general nos va proporcionando ideas claras acerca de que demos proteger, pero sobre todo contra que debeos protegerlo. Dado que las limitantes en cuanto a conocimiento en estas áreas, es uno de los aspectos que provocan el auge de la seguridad computacional, hoy día, nos ha significado y ha sido de gran valía para nuestra tesis conocer y analizar, toda la gama de información existente y referente al tema que tratamos en cuestión.

3.2 SISTEMAS IMMUNIX

IMMUNIX es una familia de herramientas diseñadas para realzar la integridad de los sistemas mediante el "endurecimiento" de plataformas y componentes del sistema, contra ataques a la seguridad. [10]

El Sistema Operativo Immunix es una plataforma de Linux endurecida con un conjunto de herramientas de Immunix. Además intentan explotar las vulnerabilidades de la seguridad, por ejemplo, el compromiso de detener el proceso en lugar de proporcionar el control al atacante y entonces restaurar. Los componentes de software son efectivamente cubiertos con tecnologías Immunix para "endurecerlos" contra ataques.

Existen diversos productos que Immunix ofrece al mercado y a las diversas aplicaciones y/o giros de las organizaciones, entre muchos de ellos se encuentran:

3.2.1 STACKGUARD

.Es un compilador que emite programas "fuertes" contra ataques que"corrompen la pila". Estos ataques son la forma más común de ataque de penetración. Los programas compilados con StackGuard son "en principio" inmunes a ataques de "corrupción de la pila". Esta protección requiere que el código fuente no sea modificado. Cuando un programa vulnerable es atacado. StackGuard detecta el ataque en progreso, emite una alerta de intrusión e interrumpe el programa víctima. El StackGuard detecta y derrota los ataques antes mencionados, mediante la protección en el regreso a la pila de los datos que han sido alterados.

El Stackguard pone una palabra "marca" junto al remitente, si al regreso de la llamada a una función, esa palabra ha sido alterada, se ha intentado algo contra la pila, se emite una señal de alerta y el sistema es detenido. Para que el ataque fuese efectivo, tendría que engañarse a la palabra "marca", pero contra esto, el StackGuard ofrece varias técnicas para prevenir este engaño ("canary spoofing"). [10]

3.2.2 BASTION SERVER APPLIANCE.

Es un servidor de aplicaciones, el cual está diseñado de tal manera que sea fácil de operar por usuarios no técnicos, y protegido con tecnologías Immunix.

Sabemos que un conjunto de aplicaciones que mejoran la seguridad en un Sitio, por ejemplo, Firewalls, VPN´s, sistemas de detección de intrusión, verificadores de vulnerabilidades de seguridad, autenticación y sistemas de autoridad de certificados, dependen de la integridad de los sistemas que los Host ejecutan, por tanto, la plataforma de "Immunix Server Appliance" emplea una variedad de herramientas para seguridad en Host que lo protegen fuertemente contra las debilidades anteriores y futuras. Immunix también proporciona una amigable interfaz de administración del sistema Web. De esta manera, la plataforma de Software Immunix, puede ser utilizada para transformar fácilmente aplicaciones sensitivas de seguridad que son caras y difíciles de proteger. [10]

Ahora analizaremos una nueva metodología para construir software de manera oportuna, clara, sencilla, pero sobre todo eficiente.

3.3 PROGRAMACIÓN EXTREMA (XP)

3.3.1 ¿QUÉ ES XP?

Es actualmente una bien intencionada y disciplinada metodología para desarrollo de software. Alrededor de cinco años, ha estado proveyendo sistemas seguros a costos aceptables a diversas compañías.

XP es exitoso porque satisface las necesidades del cliente. La metodología es diseñada para entregar el software al cliente cuando lo necesita. XP autoriza a sus desarrolladores para responder confidencialmente a los cambios que requiera el cliente, aún después del ciclo de garantía.

Esta metodología también enfatiza el trabajo en equipo. Administradores, clientes y desarrolladores, son todos parte de un equipo, dedicado a desarrollar software de calidad. XP enriquece un proyecto de software en cuatro formales esenciales:

- Comunicación
- Simplicidad
- Retroalimentación
- Empeño (coraje)

Los programadores XP establecen ***comunicación*** con sus clientes y con sus colegas. Ellos ejecutan un diseño ***simple*** y limpio. Logran ***retroalimentación*** evaluando el software después de haber "iniciado su funcionamiento". Entregan los sistemas a los clientes tan pronto como es posible, incluso ***sugieren*** la implementación de cambios. En esta fundación, los programadores XP son capaces de responder ***valientemente*** a los requerimientos en cuanto a cambios en el sistema y tecnología.

XP es diferente. Es similar a un gran rompecabezas, existen muchas piezas, cada una, de manera individual es insignificante, pero de manera conjunta es realmente valioso.

Esto es un punto de partida que diferencia a XP de los métodos tradicionales de desarrollo de software, es simplemente un cambio en la forma de programar.

¿Cuándo utilizar XP?

XP fue creado en respuesta a los problemas que requieren cambios en alguna área. En ocasiones los clientes pueden no tener una idea clara de cómo necesitan su sistema, pueden tener un sistema cuya funcionalidad se espera cambie cada ciertos meses. En muchos programas con ambientes dinámicos, cambian requerimientos que deben ser solo constantes. Es en ese momento, es precisamente cuando XP es exitoso, mientras que otras metodologías no lo son.

XP fue establecido también, para ubicar los problemas de riesgo en los proyectos. Si los clientes necesitan un nuevo sistema en un período corto de tiempo, el riesgo es alto. En caso de que sea un nuevo cambio para tu grupo de Sw, entonces el riesgo es aún más grande. Si el cambio es para todo un sistema industrial, el riesgo es más aún. Las prácticas XP fueron establecidas para minimizar los riesgos e incrementar la posibilidad de éxito.

XP esta formado por un grupo pequeño de programadores. Entre 2 y 10. Los programadores pueden ser comunes, no es necesario programadores con doctorado, para utilizar XP. Pero no se puede utilizar XP en un proyecto con un staff muy grande. Se ha notado que en proyectos con requerimientos dinámicos o un alto riesgo, es más efectivo un pequeño equipo de programadores XP que un equipo grande de cualesquiera otros programadores.

XP requiere de un amplio equipo de desarrollo. El equipo XP incluye no solo a los desarrolladores sino de igual manera a los clientes y administradores. Dado que los cuestionamientos, negociación del alcance, programación y pruebas de creaciones funcionales, requieren más tamaño s que solo los desarrolladores en la producción del Sw.

Otro requerimiento es la Testabilidad (Comprobabilidad). Se pueden crear unidades automáticas y pruebas de funcionalidad. Mientras muchas áreas pueden estar descalificadas para este requerimiento, sorpresivamente notaremos que para muchas otras no la serán.

En algunas áreas se hace necesario aplicar una pequeña porción de ingenuidad en las pruebas. Es necesario cambiar en ocasiones el diseño del sistema para hacer más fácil la aplicación de las pruebas.

Algo más en XP es, la productividad. Los proyectos XP unánimemente reportan gran productividad en sus programas, cuando son comparados con otros proyectos dentro del mismo ambiente corporativo. Esto último, nunca fue una meta de la metodología XP. El objetivo real fue siempre realizar un Sw preciso, cuando es requerido.

Reglas y prácticas de XP

- Planeación
- Diseño
- Codificación
- Experimentación

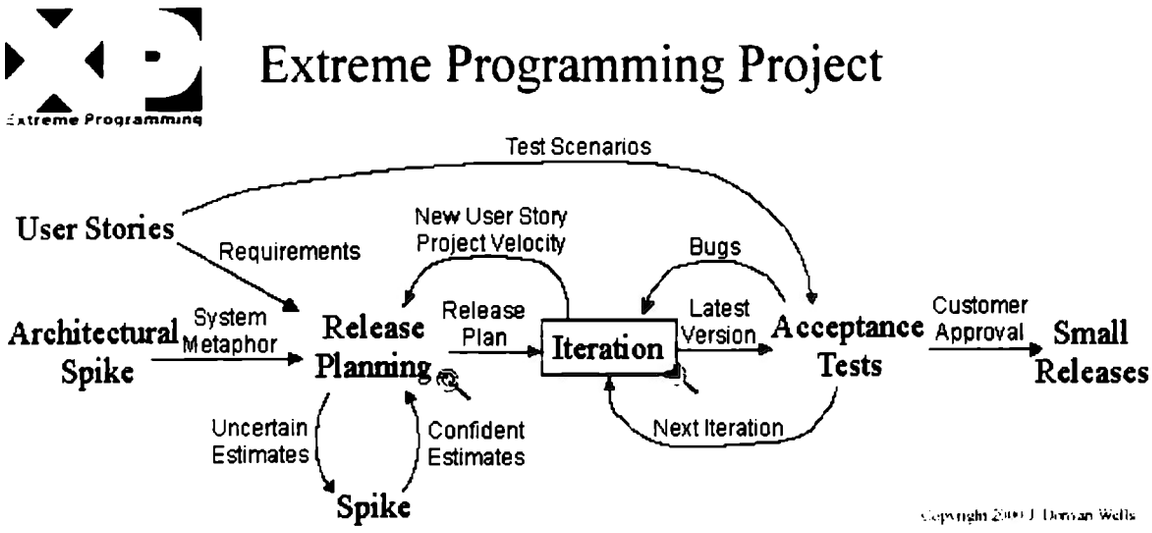


Fig. 3.3.1Esquema XP [12]

3.4 CONCLUSIÓN

Concluyendo los primeros tres capítulos de esta tesis, podemos notar primordialmente los criterios que siguen los organismos internacionales más reconocidos en cuanto a certificación de seguridad en plataformas, redes, bases de datos, incluso en aplicaciones.

Cada uno tiene una tendencia notoria en cuanto alguno de los aspectos a combatir o más bien intentar preservar, para lograr la seguridad: confidencialidad, integridad, autenticidad, etc.

Por tanto cabe hacer mención, que tomaremos diversas ideas de ellos, dado que de manera conjunta, podremos lograr cubrir la mayoría de los aspectos de seguridad. Iniciaremos comentando que el TCSEC nos es de gran utilidad, dada la relevancia que tiene en sus criterios la confidencialidad, poniendo menor énfasis en integridad, disponibilidad y la autenticidad como tal. Pone atención en la protección de accesos no autorizados. Por otro lado no hace ninguna referencia a redes.

Dada la importancia que tienen en nuestra aplicación "EAS JAGUAR" (la cual se explicará más extensamente en el capítulo 5) las comunicaciones o la interrelación con la red mundial, es que varios aspectos de funcionalidad y comunicación han sido tomados con base en el ITSEC y COMMON CRITERIA.

Además estos criterios aportan ideas importantes para nuestro estudio con respecto a control de acceso y autenticación, así como en la parte de integridad de los datos, que sabido es forman parte fundamental en la estratificación de la seguridad de una aplicación en nuestro caso.

Por otro lado, también nos auxiliamos de técnicas nuevas como lo es el Open Source Security Testing Methodology Manual (OSSTMM), el cual propone diversas pruebas de penetración para poner al descubierto vulnerabilidades en los sistemas, los cuales si dejamos de lado, podrían provocar severos daños en la información y en los sistemas mismos, propietarios de esta información y poseedores de los huecos de seguridad reflejados en las pruebas propuestas por este estándar. Además de ser una metodología que pone especial atención en las actividades que tienen relación con la Internet, es decir, el objetivo indirecto de esta metodología, es crear una disciplina que puede actuar como punto central en todas las pruebas de seguridad en Internet sin importar el tamaño de la red, el tipo de sistemas o las aplicaciones de Internet.

Algo que de esta investigación resulta de igual manera interesante es, tomar estrategias para la realización de nuestra "auditoria" a JAGUAR, de una nueva metodología en el diseño de software como lo es Programación Extrema (XP), y no solo como ideas para la certificación del nivel de seguridad de esta aplicación, sino incluso para el desarrollo de software de cualquier tipo, desde sistemas operativos, sistemas distribuidos, de aplicación financiera, etc., dado que los aspectos relevantes de esta metodología se reflejan en cuatro puntos importantes, los cuales de manera superficial parecería que todo mundo los conoce y los comprende, pero no es suficiente, es necesario llevarlos al cabo, es más, generalizando y sin temor a equivocarnos, la gran mayoría de las empresas desde las que manejan información que podría parecer irrelevante para las naciones, hasta las que manejan información financiera nacional como la institución campo de aplicación de nuestro estudio, aún no aplican metodologías para la construcción de software,

objetos, componentes o módulos estructurados que anexan a las aplicaciones comerciales, buscando adecuarlas a las necesidades de su institución, por tal motivo y retomando el aporte de la metodología XP, las cuales se resume en lo siguiente:

- Comunicación
- Simplicidad (Codificación Clara)
- Retroalimentación
- Empeño (coraje)

En conclusión todas y cada una de las metodologías son consideradas de alta importancia en nuestra investigación, dado que de manera conjunta logran darnos un bosquejo general y a su vez particular de cómo lograr obtener, o que debemos verificar para saber que nivel de seguridad tiene u obtiene una aplicación.

4 CARACTERÍSTICAS A EVALUAR EN EL DESARROLLO DE APLICACIONES

Uno de los pasos que consideramos de suma relevancia en el desarrollo del presente trabajo, es la estructuración de una lista de chequeo, la cual nos permita identificar los puntos más importantes a validar, así como "asentar" la existencia o ausencia de características seguras que posee una aplicación, es decir establecer los "puntajes", para cada uno de los rubros considerados en la lista, de tal manera que al final de la revisión se pueda determinar el nivel de seguridad de la aplicación que se evalué.

Es sabido que cualquier aplicación siempre se encuentra relacionada al sistema operativo y/o plataforma sobre la cual funciona, por tanto, nos parece importante no dejar de lado el hecho de valorar dicho sistema para saber hasta que punto también puede ser considerado seguro, ya que si alguien gana acceso al sistema de forma no autorizada, tendrá mayores posibilidades de dañar los componentes de nuestra aplicación.

A continuación mostramos una propuesta de lista de chequeo, sobre la cual nos basaremos para realizar la evaluación de la aplicación, así como una lista para el sistema sobre el cual corre dicha aplicación. Cabe aclarar que no todos los aspectos considerados deben forzosamente formar parte de la evaluación para todos los programas y/o sistemas operativos. Un punto importante a recordar es, que la mayoría de los ataques considerados para elaborar la lista siguiente está basada en el resumen de vulnerabilidades que ilustra la Fig. 4.1.1, tomando como los tres objetivos principales de seguridad al software, hardware y datos:

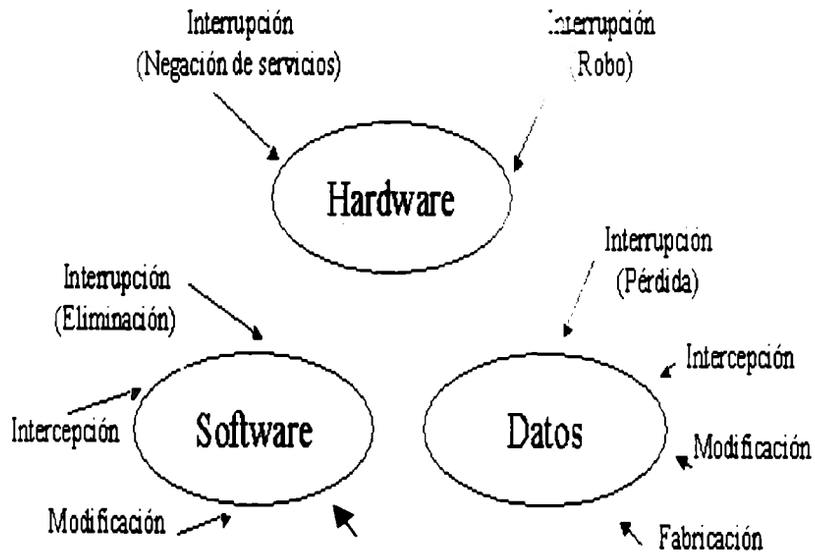


Fig. 4.1.1 Objetivos de Seguridad

Para ejemplificar la figura 4.1.1, podemos decir lo siguiente:

Interrupción

Una parte del sistema resulta destruida o no disponible en un momento dado. Ejemplos de este tipo de ataque pueden ser el daño de una parte del hardware o el corte de una línea de comunicación.

Intercepción

Una entidad no autorizada accede a una parte de la información. La parte no autorizada puede ser una persona, una máquina o un programa. Ejemplos claros de este tipo de ataques son la escucha del canal, ya sea el típico "pinchazo" de la línea telefónica, la intercepción vía radio de comunicaciones móviles o la copia ilícita de archivos o programas transmitidos a través de redes de datos utilizando analizadores de redes.

Modificación

Una entidad no autorizada no sólo accede a una parte de la información, sino que además es capaz de modificar su contenido. Ejemplos de estas modificaciones son la alteración de archivos de datos. alteración de programas y modificación de mensajes mientras son transmitidos por la red.

Fabricación

Una entidad no autorizada envía mensajes haciéndose pasar por un usuario. Cabe mencionar que en el esquema anterior, no es tomada en cuenta la *Usurpación de Personalidad* lo cual es realmente también de gran importancia en cuanto a seguridad se refiere, es más, es uno de los ataques frecuentes.

4.1 LISTA DE CHEQUEO DE DESARROLLOS

Ahora iniciaremos por describir las recomendaciones para validar las características de seguridad, como se había establecido anteriormente la propuesta de lista de chequeo se pretende emplear en aplicaciones.

4.2 APLICACIÓN

En el **Anexo M**, se encuentra nuestra propuesta de chequeo de manera íntegra, recordando que al momento de la elaboración de este documento creemos considerar la mayoría de los aspectos de seguridad existentes o al menos los más conocidos, pero obvio es que pueden agregarse muchos más que surjan recientemente.

4.2.1 AUTENTICACIÓN

La Autenticación es un área en la cual el potencial de la tecnológica de llave pública ha proporcionado grandes beneficios.

Existen varios pasos para proporcionar autenticación en un ambiente distribuido. El primer paso es autenticación o verificación de la identidad del usuario. Existen tres métodos básicos para lograr esto. Primero, una manera de verificar un usuario, es por algo que *el conoce*, tal como un *password*, es el método más popular aunque no necesariamente el más seguro (incluso, hoy en día el uso de contraseñas ya no es considerada una prueba de identidad, dada la facilidad con que puede violarse esta medida de control). El segundo método involucra algo que *el usuario posee* tal como una *llave o una tarjeta inteligente*. El tercer método involucra algo que *el usuario es*, tal como una *huella digital* o patrones de retina, que son los métodos más seguros y caros, aunque los menos estandarizados. [21]

Administración de Passwords

En ocasiones, el establecimiento de políticas de seguridad tanto federales como corporativas, pueden variar dependiendo precisamente de las instituciones en las cuales se aplican, pero lo más común, en cuanto a la administración de passwords, es que exista una administración global y una específica sobre cuentas de usuario.

Algunas características que serían deseables que una aplicación tuviese de manera global son:

- Establecer un tiempo máximo de vida del password para especificar el número de días en que un password es válido antes de que sea necesario cambiarlo.
- Asegurar la unicidad del password, mediante el uso de un historial de passwords. El archivo de historial de passwords en conjunción con el periodo mínimo del password protege contra repetición de password de usuario.
- Establecer un mínimo de longitud para el password.

Por otro lado algunas características aplicables a usuarios en específico son:

- Forzar a los usuarios a cambiar su contraseña para la siguiente vez que interactúen con el servidor (evidentemente antes debió existir un proceso de asignación de cuentas y contraseñas)
- Prevenir a los usuarios de cambiar su password
- Permitir a los usuarios tener password que nunca expiren
- Deshabilitar cuentas de usuario que no son utilizadas

En resumen, las características antes mencionadas, pudiendo haber otras más, son indispensables en un sistema, dado que de ello depende la calificación que dentro del rubro de Autenticación éste pudiese obtener (dentro de la lista de chequeo propuesta para evaluar la seguridad del sistema). Algunos de los rasgos donde se ve involucrada la voluntad del usuario, pueden cubrirse con una exigencia por parte del sistema al modificar, eliminar o crear de determinada manera un password, incluso especificando detalles finos del mismo.

Ahora bien, en algunos sistemas resulta viable un mapeo del password del sistema operativo a la aplicación, lo cual podría beneficiar o fungir como una verificación de que los usuarios que accedieron a la aplicación, hayan de la misma manera tenido acceso al sistema operativo, para de esta manera proteger la aplicación, ya que no necesariamente el administrador de la aplicación es el mismo que el administrador del sistema.

En nuestro estudio, es muy necesario que tanto el sistema operativo como la aplicación posean activas, estas configuraciones y/o atributos, ya que éstos aspectos ayudan de manera significativa al inicio de la interacción plataforma – aplicación.

Validación: desde el momento que una Aplicación requiere autenticación al menos mediante la solicitud de un password, requiere de la administración de ellos, pero además de administrarse, se requiere verificar que se pueden modificar, eliminar, establecer restricciones de su conformación, etc., para considerar la administración de ellos dentro de un nivel aceptable en contribución a la seguridad de la aplicación. Verificando la tabla de password con el administrador del sistema, podríamos determinar la existencia de administración. En cuanto a la autenticación de usuarios, aplicaciones, procesos y cualquier otra entidad de la aplicación o su entorno, el uso de password es el nivel mínimo aceptable de autenticación.

Administración de Usuarios

De manera muy similar a las características de los passwords, las aplicaciones y sistemas operativos deben de tener una administración o control de los usuarios, desde su generación hasta las actividades que ellos tienen permitido realizar, así como la verificación de que efectivamente cumplan con todas y cada unas de las políticas establecidas para el acceso a objetos.

Validación: se debe verificar la existencia de perfiles de usuario, alta y baja de usuarios de manera congruente en la infraestructura de los sistemas de la organización (por ejemplo: si se da de baja a un usuario que renunció, que existan los mecanismos o procedimientos para que esta baja sea efectiva en todos los sistemas de la organización). Debe existir un documento que detalle los perfiles de usuarios del sistema o de la aplicación y la función que tiene asociada cada uno de estos.

Administración de Certificados

Este es un tema bastante relevante en la seguridad de autenticación de un usuario, aplicación o sistema, ante otros usuarios, aplicaciones o sistemas, dado que la protección de una transacción podría depender de la existencia de éstos certificados. Pero al igual que muchas de las características de seguridad, existe la necesidad, de una parte de confianza, dado que se deposita ésta en una entidad o tercera parte que funge como mediador o emisor de certificados, sin embargo, siempre existe la pregunta de ¿quién lo certifica a su vez al tercero, como Entidad Certificadora?.

Uno de los aspectos importantes además de la emisión y distribución, es la vigencia y validez de certificados. Los certificados cuya vigencia ha expirado son registrados en la CRL (lista de revocación de certificados): con la pronta publicación de esta lista se puede evitar que entidades externas establezcan transacciones no válidas mediante certificados caducados.

Por otro lado el nivel de certificación, es decir, el rigor en los procedimientos de certificación y la "reputación intachable" de la autoridad certificadora, también permite ejercer un mayor control sobre las actividades que sobre un sistema o aplicación se pueden realizar, incluso llegar a grados de preocupación que antaño parecerían exagerados; pero que hoy en día son necesarios.

Validación: Se evaluará si se cuenta con procedimientos de administración de los certificados manejados por las aplicaciones, sistemas operativos y lo mismos usuarios.

Es importante mencionar que las aplicaciones que manejen certificados para la autenticación, ya sea de usuarios, procesos u otras aplicaciones, además del esquema tradicional basado en passwords, tendrán un puntaje superior en la evaluación.

Asimismo, el hecho de que exista una CA, inmersa en la aplicación misma o implantada en el entorno que da soporte a ésta, amerita una mayor evaluación que el sólo manejar certificados, ya que implica que la confianza está depositada en la organización misma (en su entorno) y no en autoridades externas.

Administración de Llaves

Otro punto de la Autenticación que podría parecer redundante con los tres puntos mencionados anteriormente, pero cuya especificidad le distingue; por ejemplo, consideremos situaciones como cuando un empleado deja de trabajar en la organización por razones diversas como renuncia, despido, retiro o fallecimiento; ocurre que un sistema falla perdiéndose la información y su respaldo está corrupto; y si sumamos a esto que la información estaba cifrada con la lave privada de algún empleado descontento que ha renunciado, o que el respaldo que no se pudo recuperar contenía las llaves que cifran la información, entonces estamos en verdaderos aprietos. Deben administrarse las llaves para evitar estas situaciones.

Validación: se evaluará si se cuenta con procedimientos de administración de llaves. Debe contarse son un documento que detalle cómo proceder para restituir las llaves de la aplicación o sistema.

Dispositivos Biométricos y Tarjetas Inteligentes

La biometría y las tarjetas inteligentes, son consideradas como un segundo factor o segundo nivel de autenticación. Esto quiere decir, que si sumamos cualquiera de estas dos tecnologías al uso de password, un intruso requeriría además de conocer el password de un usuario, contar con la tarjeta inteligente o alguna medición biométrica que le permitiesen autenticarse.

Existen varios factores para elegir algún dispositivo biométrico en particular: ¿qué aspecto de la constitución humana puede medirse y ser útil para la autenticación a aplicaciones o sistemas? ¿Cómo lograr la identificación con suficiente exactitud, confiabilidad y velocidad?

Los dispositivos biométricos pueden ser económicos (alrededor de 150 dólares) hasta muy costosos (varias decenas de miles de dólares); aunque quizá sean en general, el mecanismo más costoso para implantar la autenticación de una aplicación, es considerado como el único mecanismo en que el usuario que se autentica no puede delegar a otro su identidad, es decir, no es posible que un usuario le preste su "huella digital" o su "retina" a otro, al menos no de una manera sencilla.

Por otra parte, las tarjetas inteligentes llevan un "chip" (circuito integrado) que cuenta con memoria, y capacidad de procesamiento (incluso criptográfico), donde se almacena información del propietario de la tarjeta. Deslizando la tarjeta en una lectora conectada o integrada a una PC, los usuarios consiguen acceso a su información, que pueden ser datos personales, passwords e incluso su propia llave privada, expedida por su autoridad certificadora.

Las lectoras de tarjetas inteligentes tiene un precio menor a los 100 dólares, y muchos sistemas operativos ya las aceptan de forma transparente. Además, los usuarios de tarjetas inteligentes no están limitados a sus propias PC o servidores, sino que pueden obtener acceso fácilmente a cualquier número de máquinas o incluso puntos físicos de entrada. El costo de la tarjeta inteligente está alrededor de los 15 dólares.

En el **Anexo H**, podrá encontrar algunas justificaciones del porqué de estos nuevos controles de acceso

En general, los dispositivos biométricos de identificación de huellas digitales ofrecen seguridad de nivel "uno en 500,000", lo que significa que en 500.000 intentos, logra pasar una persona que no debe. Y al contrario, más o menos una vez por cada 2.000 intentos es rechazado un usuario autorizado. Las compañías pueden colocar el umbral aún más abajo, para evitar más número de lecturas equivocadas, pero con un umbral más estricto el proceso de reconocimiento se vuelve más lento.

Otros dispositivos biométricos ofrecen niveles de seguridad aún más altos. De todos los dispositivos biométricos disponibles, el reconocimiento del iris es considerado más seguro, con una lectura falsa de tan sólo uno en 1.2 millones.

Validación: Se evaluará si la aplicación cuenta con facilidades para integrar el uso de dispositivos biométricos o de tarjeta inteligente en sus procesos de autenticación. El contar con estas facilidades, hace de la aplicación más segura, lo cual se reflejará en la evaluación.

4.2.2 CONTROL DE ACCESO

El control de acceso a las aplicaciones es siempre precedido por una comprobación de la identidad de los usuarios (autenticación). El control de acceso puede tomar dos formas generales:

- Discrecionario, que se basa en el buen criterio de los administradores de las aplicaciones o sistemas operativos. Este control puede tomar la forma de permisos de lectura, escritura, modificación y administración de los recursos. Asimismo las listas de control de acceso pueden ser un ejemplo de este mecanismo.
- Obligatorio (mandatorio), que se basa en niveles, o funciones previamente establecidas en una organización, donde el acceso a los archivos y otros recursos de la aplicación o sistema no dependen del sentido común de los administradores, sino en reglas predefinidas, bien establecidas, la mayoría de las veces, inmersas en las aplicaciones o en el sistema operativo. Este tipo de control es más restrictivo e incluso limita en cierta medida todo "el poder" de lo que puede hacer el administrador. Un sistema que cuenta con este control se considera más seguro que otro que sólo cuente con control discrecionario.

En el **Anexo G**, se encuentra una descripción cronológica del control de acceso.

Este control va muy de la mano con la administración de usuarios y sus perfiles

Validación: Se evaluará si la aplicación cuenta con controles de acceso a los recursos que maneja. El contar con un control mandatario, hace que la aplicación sea más segura, lo cual se reflejará en la evaluación.

4.2.3 INTEGRIDAD

La integridad es mucho más difícil de identificar con exactitud. Integridad significa diferentes cosas en diferentes contextos, algunos significados son los siguientes: [7]

- Precisión

- Exactitud
- No modificado
- Modificado sólo en formas aceptadas
- Modificado sólo por personas autorizadas
- Modificado sólo por procesos autorizados
- Consistente
- Internamente consistente
- Resultados correctos y significativos

Una característica básica en un documento auténtico es su integridad. Por ejemplo: un documento electrónico que por errores de transmisión o fallas en el medio de almacenaje o intencionadamente, se modifica el contenido original del documento entonces el documento pierde su integridad y por tanto su autenticidad. Si un documento es auténtico entonces es integro, pero no viceversa. Para este tipo de fallas, típicamente se requiere de controles como los CRC (Códigos de Redundancia Cíclica) o funciones Hash (también conocidas como compendios de mensajes, funciones de un solo sentido o huellas digitales)

En el Anexo B, puede consultarse información importante y base para una fácil introducción y comprensión del tema de integridad, mediante una reseña conceptual de los algoritmos más comúnmente empleados para asegurar la integridad de la información

Validación: se evaluará si la aplicación cuenta con facilidades para detectar si la información que manipula o produce se conserva íntegra.

4.2.4 DISPONIBILIDAD

La disponibilidad aplica tanto a datos como a servicios (acceso a recursos de cómputo), e incluso personas. Diferentes expectaciones de disponibilidad incluyen:

- Presencia de objetos o servicios de manera utilizable
- Accesibilidad a objetos o servicios por usuarios autorizados
- Capacidad para encontrar servicios necesarios
- Progreso: definir tiempo de espera
- Definir tiempo/oportunidad de servicio

Los objetivos de la disponibilidad son:

- Respuesta oportuna
- Asignación justa
- Tolerancia a fallas
- Utilidad o funcionalidad (puede ser utilizado como se pretende?)
- *Concurrencia controlada:* soporte para acceso simultáneo, administración de Deadlock y acceso exclusivo.

La seguridad colectiva es justamente el inicio del entendimiento de lo que la disponibilidad implica y como asegurarla. Un poco de control de acceso centralizado es fundamental para la

preservación de confidencialidad e integridad, pero no es claro que un punto simple de control de acceso sea capaz de forzar la disponibilidad. Muchos de los progresos en seguridad computacional han sido en las áreas de confidencialidad e integridad; sin embargo, la protección de la disponibilidad no ha sido tan estudiada, de hecho, se puede decir que los problemas y ataques más severos en la actualidad van en contra de la disponibilidad.

Validación: Se evaluará si la aplicación cuenta con facilidades para:

- Proporcionar sus servicios de manera continua
- Mantener accesible la información bajo condiciones extremas de carga
- Cuenta con los enlaces de capacidad suficiente para soportar el servicio que se pretende brindar

4.2.5 CONFIDENCIALIDAD

Criptosistemas

El uso de criptosistemas es hoy día, cada vez más necesario, debido a las características de seguridad que proporcionan. Un criptosistema debe satisfacer los siguientes requisitos para ser utilizado en la práctica:

- Las transformaciones de cifrado y descifrado deben ser computacionalmente eficientes.
- La seguridad del sistema debe depender exclusivamente del secreto de las claves y no del secreto de los algoritmos de cifrado y descifrado. Estos algoritmos pueden conocerse de forma pública y deben ser tales que sin el conocimiento de las claves no pueda descifrarse un mensaje.

Muchos de los sistemas y aplicaciones actuales proporcionan facilidades para cifrar la información que manipulan, tanto para criptosistemas simétricos como asimétricos. La administración de llaves, mencionada en la sección anterior, juega un papel fundamental en la confidencialidad de las aplicaciones.

Validación: se evaluará si la aplicación cuenta con facilidades para cifrar la información en el sistema en que reside. El contar con estas facilidades, hace de la aplicación más segura, lo cual se reflejará en la evaluación.

Canales Seguros

Uno de los aspectos más importantes de la seguridad es este punto que nos ocupa. Entre los aspectos que consideramos con mayor repercusión, dentro de la confidencialidad, como ya hemos mencionado, son: El manejo de criptosistemas simétricos, asimétricos, híbridos, cifrado de archivos y directorios, manejo de módulos VPN, IPSec y transferencia de datos bajo protocolo SSL.

Creemos que si una aplicación cumple con estos aspectos, estaríamos hablando de una aplicación segura en cuanto a confidencialidad.

Cabe mencionar que en el mercado existen un sin número de empresas dedicadas a brindar seguridad en los aspectos anteriormente enlistados, entre algunos se encuentran aquellos dedicados a brindar cifrado seguro en archivos e e-mail, en ocasiones, este cifrado está basado en Hardware.

Entre las tecnologías que permiten contar con canales seguros podemos mencionar:

- Redes virtuales privadas (VPN)
- IPSec
- SSL (Secure Sockets Layer)

Una descripción detallada sobre los anteriores tres puntos, se realiza en el **Anexo C** (Confidencialidad)

Validación: Se evaluará si la aplicación cuenta son facilidades para cifrar la información que transmite o recibe de la red en su entorno. El contar con estas facilidades, hace que la aplicación sea más segura, lo cual se reflejará en la evaluación.

4.2.6 NO REPUDIACIÓN

Existe una diferencia sutil entre el concepto de autenticidad y el de no-repudiación. Un ejemplo claro sería: si presenciamos que alguien escribe un documento; si el documento no es firmado, se podrá comprobar la autenticidad, pues lo vimos escribirlo, pero no será posible probarlo, pues sin la firma es imposible establecer el vínculo entre la voluntad de la persona y el contenido del documento. Si se puede probar a terceros que efectivamente el documento es auténtico, entonces se dice que el documento es no-repudiable. Si el documento es no –repudiable es auténtico, pero no viceversa. La no-repudiación se basa en firmas digitales y en una autoridad certificadora de confianza.

Validación: Se evaluará si la aplicación cuenta con facilidades para permitir firmas la información que manipula. El contar con estas facilidades, hace que la información manejada por la aplicación sea más confiable (si las llaves privadas son generadas por una autoridad reconocida por la aplicación y por las entidades que utilizan sus servicios)

4.2.7 TOLERANCIA A FALLAS

Este, es también un punto importante dado que la continuidad en el servicio repercute en la seguridad de una aplicación. Para lograr esta continuidad, en ocasiones se cuenta con mecanismos como redundancia de los sistemas donde reside la aplicación, el balanceo de cargas entre estos sistemas redundantes, e incluso de un adecuado y permanente suministro de energía eléctrica, lo cual ayuda no sólo a la disponibilidad de la aplicación y de la información que maneja, sino a mantener un gran porcentaje de tolerancia a fallas, porque siempre existirá la infraestructura necesaria, pendiente o emergente en cuanto el sistema falle, y mediante esta técnica se realiza un adecuado manejo de carga.

Validación: se evaluará que el entorno de la aplicación garantice, mediante al menos un método, la continuidad de la operación de ésta.

4.2.8 PARCHES

Las versiones de sistemas y/o aplicaciones siempre llevan consigo una mejora en diversos aspectos de cómputo, entre ellos la eliminación de vulnerabilidades reportadas por los organismos dedicados a esto, y que repercuten directamente en la seguridad de la aplicación y del sistema mismo. Por tanto, contar con las actualizaciones de los sistemas, que incluye la emisión de parches, contribuyen a mantener un mejor nivel de seguridad.

Validación: se evaluará que esté dando soporte, a la aplicación y a los sistemas en su entorno, que garantice su mantenimiento y actualización.

4.2.9 CERTIFICACIÓN

Este aspecto ampliamente mencionado a lo largo de este documento, por obvias razones es importante. El contar con un nivel de certificación por alguno de los organismos más reconocidos (por ejemplo, TCSEC, ITSEC o Common Criteria), siempre será relevante, dada la certeza del nivel certificado. Por tanto en nuestra evaluación es considerado el grado de seguridad en que califica.

Validación: un sistema o aplicación previamente certificado por un organismo reconocido para este fin, será mejor evaluado que uno que sólo promete algunas características funcionales.

4.2.10 RESISTENCIA A ATAQUES

Uno de los aspectos en que más énfasis se hace es la resistencia a ataques, dado lo que significa una falla en estos campos para la estabilidad del sistema, desde un ataque de escucha, intrusión, hasta desbordamiento de pila y buffer. Para este último, dado que es un espacio de memoria, en el cual las posiciones de memoria se encuentran consecutivas, esto permite que al ubicar una de estas posiciones se pueda conocer el resto de la información ahí concentrada mediante una simple indexación.

El problema de los búferes es el desbordamiento, es decir, todo se reduce a una pregunta simple: ¿qué pasaría si metiésemos más datos al buffer de los que puede almacenar? En un esquema tradicional de programación, sobrescribiríamos la dirección de retorno, haciéndola apuntar hacia otra parte de la memoria. Si realmente se ha tratado de una casualidad, lo más probable es que es que la nueva dirección a la que apuntamos esté fuera de nuestro espacio reservado de memoria, produciendo un error de segmento, con lo cual se puede decir que se ha producido un desbordamiento de buffer (buffer overflow).

Ahora que si el objetivo es romper la seguridad, nos interesaría que el programa de buffer ejecutará un código determinado. Obviamente lo que queremos ejecutar no se encuentra dentro de las instrucciones del programa, por tanto, se desbordará el buffer y se provocará que la dirección de retorno apunte hacia el inicio de ese programa que deseamos se ejecute.

Entre los ataques posibles podemos mencionar:

- Desbordamiento de memoria
- Desbordamiento de snack

- IP Spoofing, o falsificación de direcciones de origen o destino
- Escaneo de puertos
- Saturación de conexiones
- Virus

Es necesario mencionar que para resistir a estos ataques debemos contar con elementos monitores que los detecten, y si es posible los alimenten. Es este aspecto juegan un papel muy importante las herramientas para detección de intrusión, los antivirus y los mismos Firewalls.

En el **Anexo D**, se encuentra una descripción más detallada de los aspectos que permiten verificar el nivel de resistencia a ataques presentada por un sistema.

IDS.

Un sistema de detección de intrusión (IDS) inspecciona toda la actividad de entrada y salida de una red e identifica patrones sospechosos que puedan indicar un ataque a la red o sistema que intente romperlo o comprometerlo.

Existen varias formas de categorizar a los IDS:

Detección de mal uso vs. Detección de anomalías:

En detección de mal uso, el IDS analiza la información, recopilando y comparando de grandes bases de datos o firmas de ataques.

Sistemas basados en red vs. Sistemas basados en Host

En un sistema basado en red, o NIDS, el flujo individual de paquete a través de una red es analizado. El NIDS puede detectar paquetes maliciosos que están destinados a ser suprimidos por las reglas de filtrado de un Firewall. En un sistema basado en Host, el IDS examina la actividad en la computadora o Host individual.

Sistemas pasivos vs. Sistemas reactivos

En un sistema pasivo, el IDS detecta un hueco de seguridad potencial, registra la información y emite señales de alerta. En un sistema reactivo, el IDS responde a las actividades sospechosas para registrar salida de un usuario o para reprogramar el Firewall para bloquear el tráfico de la red a partir de un origen sospechosamente malicioso.

Validación: se evaluará que la aplicación cuente en su entorno con herramientas de detección de intrusión para garantizar la reacción adecuada en respuesta a un ataque.

4.2.11 REUSO DE COMPONENTES

Muchas veces los Hackers se valen de elementos que para un usuario "inocente" podrían parecer no problemáticos, es decir, de los cuales no se podría comprometer información alguna, simplemente porque no se tiene conocimiento de que son de uso delicado o lo que pareciera

extremoso, ni siquiera saben de su existencia. Estos elementos comunes son: la información almacenada en el disco duro, la que en determinado momento se encuentra en la memoria caché, en los registros o incluso en la sección de temporales. Se puede reducir el número de elementos problemáticos de una aplicación si se construyen componentes adecuados y evaluados, que vayan mejorándose conforme pasen las actualizaciones y que sirvan para desarrollar otras aplicaciones, con la confianza de que los componentes han sido evaluados.

Validación: Se evaluará si la aplicación ha sido desarrollada o configurada mediante componentes.

4.2.12 AUDITORIA

Un aspecto que también posee relevancia dentro de la seguridad en sistemas es la auditoria. Para ello hemos considerado varios aspectos entre los que se encuentran: el monitoreo a procesos en ejecución, memoria, caché, procesador, servicios en ejecución, gráficas o diagramas, alarmas, reportes, acceso a archivos y subdirectorios, acceso remoto, así como: generación, auditoria y administración de logs. En resumen, el realizar actividades como las anteriores, dentro de una aplicación, permitiría cooperar en la localización y detección de intrusión y la actualización y mejora de la aplicación.

Validación: Se evaluará que exista un plan de auditoria de la aplicación

4.2.13 RUTAS SEGURAS (O DE CONFIANZA)

Las rutas seguras, se refieren a las conexiones entre el usuario y las aplicaciones, hasta las interacciones entre aplicaciones. Estas relaciones evidentemente son de importancia e influyen sustancialmente en la posible calificación del nivel de seguridad de una aplicación de forma general, dado que de estas conexiones podría incluso influir el éxito o fracaso del resto de características seguras que pudiese posee una aplicación, es decir, si la conexión entre un sistema operativo y la aplicación no son seguras, los aspectos considerados específicamente en un sistema podrían estar de más, dado que tendríamos intrusos a nivel operativo, que podrían dañar fácilmente nuestro sistema.

Una ruta segura proporciona una inequívoca manera, mediante la cual, un usuario puede comunicarse directamente con la TCB (Trusted Computer Base) sin tener que interactuar con el sistema a través de aplicaciones o capas no seguras del sistema operativo. Una ruta segura en primera instancia es un requerimiento para sistemas catalogados en nivel C2 o superiores.

Validación: Se evaluará que existan rutas de confianza entre aplicaciones y el sistema operativo que la soporta.

4.2.14 ARRANQUE SEGURO

Otro aspecto que consideramos en esta evaluación, y que al igual que la seguridad entre sistema operativo y aplicaciones y entre aplicación y aplicación, impacta de manera importante en la seguridad de un sistema, es lo que llamamos el Arranque seguro, y es precisamente en este aspecto, donde continuamente se han encontrado huecos de seguridad en los sistemas.

Validación: Se evaluará que la aplicación al ser ejecutada esté realmente en comunicación directa con un sistema operativo y no con un programa intermedio que pudiese incluso estar interceptando toda comunicación entre ellos.

4.2.15 RESPALDO

El respaldo que de la información realice una aplicación, es lo que más importa a las empresas, dado que es el activo que mueve a la gran mayoría de ellas. Como lo establecimos al inició de este capítulo bajo un esquema (Fig. 4.1.1) la información es uno de tres aspectos donde impacta la ausencia de seguridad. Hemos considerado, obviamente, dependido del tipo de sistema y su hardware, verificar, si el respaldo se realiza en cintas, CD, en la misma red u algún otro recurso de almacenamiento, para inspeccionar si se lleva al cabo de manera segura.

Validación: Se evaluará que la aplicación cuente con procedimientos de respaldo bien definidos en su política de seguridad.

4.2.16 COMPARTICIÓN DE RECURSOS

Generalmente este aspecto va muy de la mano con los privilegios que posee un usuario o grupo de usuarios, aún cuando podría presentarse de manera separada. Los recursos computacionales y de información de un equipo podrían permitir dependiendo de la asignación que de ellos se haga a los usuarios y/o grupos de usuarios generar enormes huecos de seguridad al compartir discrecionalmente sus recursos, lo cual hará más probable la intrusión hacia la información. Es aquí donde se maneja comúnmente la asignación de privilegios, tal vez de acuerdo a estrato dentro de la empresa, o sobre todo a necesidad que cada usuario tenga de la información, es decir: "know to need"

Validación: Se evaluará que la aplicación cuente con un documento que justifique cómo debe compartir los recursos que maneja, o que cuente con linimientos que muestren que no se está compartiendo recursos de forma comprometedora.

4.2.17 CLASIFICACIÓN DE LA INFORMACIÓN

Una buena clasificación de la información, significa que podría existir una distribución de acceso por aplicaciones, por usuarios que manejan la información o las aplicaciones mismas, y se podría llevar un mejor control sobre los usuarios permitiéndoles hacer únicamente lo que su perfil requiera.

Validación: Se evaluará que la aplicación separa y da tratamiento adecuadamente a la información y recursos que maneja de acuerdo a la clasificación que le fue asignada a éstos.

4.2.18 ADMINISTRADOR DEDICADO

El hecho que un sistema tenga para su ejecución un administrador dedicado, implica que en ningún momento, y por ninguna razón el servidor deje de desatender las peticiones y/o actividades que de una determinada aplicación emanen, con esto, permitiendo jamás una

intrusión en los cortes para asignación de tiempo de procesamiento a otros procesos. Por tanto, hemos considerado pertinente, tomar nota sobre la existencia de un administrador dedicado o no, para cualquier sistema, y en nuestro caso particular, para la aplicación objeto de nuestro estudio.

Validación: Se evaluará si la aplicación cuenta con un administrador especializado, lo cual aumenta mucho la confianza en dicha aplicación.

4.2.19 EVALUACIÓN PREVIA DEL PRODUCTO POR PARTE DE LA ORGANIZACIÓN

Si las empresas que adquieren productos y sistemas computacionales, realizaran un estudio previo de las ventajas y desventajas que obtendrán con la adquisición de dicho producto o sistemas, así como poner especial atención en el momento de tratar la parte de seguridad con los proveedores de dichos equipos o sistemas, tal vez estaríamos hablando que sólo seríamos vulnerables ante posibles innovaciones a las formas de ataque, y quizá estaríamos preparados para ello, pero no seríamos débiles ante ataques bien conocidos y que siguen teniendo mella en los sistemas que no han sido actualizados, la mayoría de las veces por descuido, desinterés y sobre todo ignorancia.

Validación: Se evaluará que la empresa cuenta con un documento que avale que antes de adquirir el sistema motivo de evaluación, su personal encargado del área realizó un estudio de todas las características relacionadas al producto (nuestro objetivo es analizar la parte de seguridad), y a que nivel fue este estudio previo a la decisión de adquisición y compra del mismo.

4.2.20 POLÍTICA DE RECUPERACIÓN DE DESASTRES Y PLAN DE CONTINGENCIA DEL NEGOCIO

Ante un desastre inminente o ya perpetrado, de la índole que sea o por cualquier causa, las políticas de recuperación de desastres pueden sacar a flote el funcionamiento normal de una empresa, o por otro lado, sumergirla en la más "profunda de las desgracias", dado que esas políticas comprenden o tienen establecidas, las acciones a seguir ante la mayoría de las situaciones que pudiesen presentarse. Estas medidas, nos dicen qué hacer con nuestro equipo (Hw), que acciones tomar con las bases de datos, servidores, datos almacenados, procesos en ejecución, etc.

Dado que difícilmente alguien llámese empresa, sistema o información está preparado para sufrir jamás un ataque que evidencie el nivel de seguridad de cualquiera de ellos.

Validación: Que una aplicación posea una correcta política de recuperación de desastres, y lo que sería mejor, políticas de mantenimiento físico y lógico, predictivos antes de los correctivos.

4.3 ENTORNO DE DESARROLLO

Ya hablamos en los párrafos anteriores acerca del entorno de la aplicación, refiriéndose a toda aquella infraestructura informática y computacional que le permite realizar sus servicios de

manera adecuada. Para nuestro estudio en especial y los casos presentados en el siguiente capítulo, los sistemas sobre los cuales se encuentra instalado Jaguar, para las pruebas de funcionamiento son:

- Windows NT
- Unix

Obvio es que dependiendo de la plataforma, seguramente las condiciones óptimas de funcionamiento pueden variar, dado que cada uno de ellos presenta cierta disponibilidad así como "resistencia" a los sistemas de aplicación.

5 CASO DE ESTUDIO

5.1 INFRAESTRUCTURA DE DESARROLLO DE UNA INSTITUCIÓN FINANCIERA: "JAGUAR"

Iniciemos esta sección comentando que como política en algunas instituciones financieras, se ha buscado que el desempeño y la escalabilidad de los sistemas sea un requisito previo a todo desarrollo. La arquitectura de sistemas distribuidos en n-capas, permite estas ventajas.

Las nuevas tendencias de programación e incluso de desarrollo proponen una arquitectura en la cual, el acceso a los datos y la logística de cualquier negocio resida en un servidor, evitando así que resida en cada uno de los clientes, esto muchas veces o en su totalidad de ocasiones provoca que las aplicaciones en el cliente sean cada vez menos densa, además de que estas acciones ayudan o colaboran en el reúso de código. Además esto permite también crear aplicaciones de forma eficaz y eficiente.

En cuanto a desarrollo de aplicaciones n-capas (Fig. 5.4.1), los servidores de aplicaciones han cobrado relevante importancia, por esto y muchas cosas mas, la organización bajo estudio decidió adquirir el servidor de aplicaciones JAGUAR CTS, cuya arquitectura n-capas se muestra a continuación, como preámbulo a lo que es Jaguar.

Modelo n-tier:

Es el modelo más adecuado para construir aplicaciones distribuidas. Las características más relevantes de dicho modelo son las siguientes:

- La capa intermedia interactúa con el servidor de base de datos por medio de protocolos estándar de comunicación.
- La capa intermedia contiene la mayoría de la lógica del negocio, permitiendo la reutilización de código y el mantenimiento centralizado.

- Es relativamente fácil ampliar el número de capas de acuerdo a las condiciones del sistema.

- Se enfoca principalmente en soportar el desarrollo de aplicaciones Orientadas a Objetos.

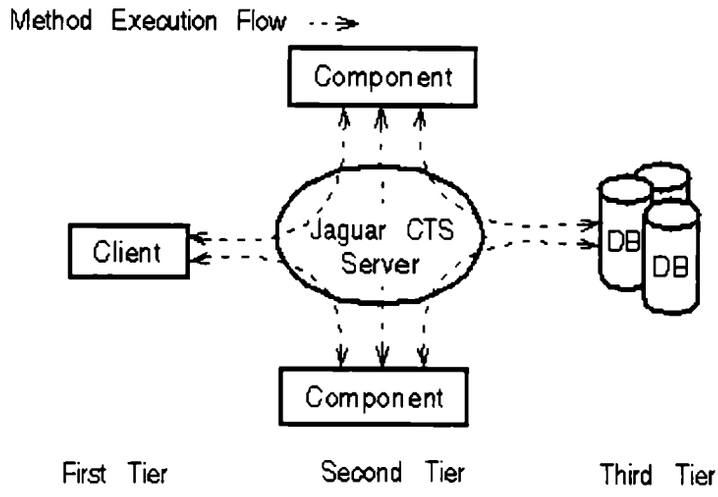


Fig. 5.1.1 Arquitectura Multicapa de Jaguar [23]

En la institución financiera bajo estudio se adquirió un EAServer (Enterprise Application Server), el cual es una integración de un conjunto de aplicaciones que se utilizan para destacar las aplicaciones Web que soportan altos volúmenes de tráfico, contenido dinámico y procesamiento de transacciones intensivas en línea (OLTP). EAServer consiste de un servidor de componentes de transacciones Jaguar (Jaguar CTS), PowerJ, Power Dynamo, Servidor Adaptable e Integrador de Aplicaciones.

Antes de realizar una visión general sobre Jaguar, revisaremos los aspectos más relevantes del Protocolo SSL, dado que es sobre el cual funciona dicho Servidor.

En e **Anexo E**, se concentra información al respecto del protocolo SSL (Secure Socket Layer) de gran importancia en las transacciones seguras ejecutadas por las aplicaciones que funcionan sobre Jaguar.

5.2 ENTERPRISE APPLICATION SERVER JAGUAR

Jaguar CTS simplifica la creación y administración de aplicaciones sobre Internet que dan servicio a miles de clientes simultáneamente. Los componentes Jaguar se ejecutan en la capa intermedia entre aplicaciones cliente del usuario final y bases de datos remotas. Jaguar proporciona control eficiente de sesiones cliente, seguridad, bases de datos de conexión triple y flujo de transacciones sin requerir de conocimiento especializado en la parte de desarrollo de componentes.

La escalabilidad de Jaguar e independencia de plataforma permite desarrollar aplicaciones en máquinas de procesadores simples y económicos y entonces desplegar la aplicación en un servidor multiprocesador de nivel empresarial.

Jaguar CTS proporciona las siguientes características:

- Una ingeniería de ejecución de plataforma independiente, multihilos y escalable
- Transferencia y soporte stub/proxy para la mayoría de modelos de componente, incluyendo JavaBeans, PowerBuilder, Java, Actives y C/C++.
- Soporte dinámico HTML utilizando Java Servlet, Páginas Java Server y sitios Web PowerDynamo.
- Soporte de plataforma Java Edición Empresarial (J2EE)
- Administración gráfica con Sybase Central, incluyendo interfaz de componentes, seguridad basada en reglas, password, características requeridas SSL de sesión, administración de certificados de servidor y usuario, soporte de módulos IDL, monitoreo de transacciones OTS y monitoreo en tiempo de ejecución.
- Integración estrecha con el PowerBuilder y ambientes de desarrollo PowerJ
- Sesiones de cliente transparentes y administración del ciclo de vida de componentes.
- Conexión oculta que permite el reúso de conexiones remotas a bases de datos
- Servidor de nombres de estándar industrial para resolver componentes utilizando nombres lógicos más que direcciones de servidor.
- Administración de transacciones para simplificar el diseño e implementación de unas transacciones de aplicación.
- Características de transacciones de hilo seguro para simplificar el uso de datos y recursos compartidos
- Soporte de conjunto de resultados para permitir recuperación eficiente de datos tabulares en aplicaciones cliente
- Declarativas, seguridad basada en roles para restringir conexiones de cliente y los componentes que pueden ser invocados por una sesión de cliente específica.
- Seguridad basada en identidad para restringir llamadas ínter componentes
- Soporte para mensajes asíncronos

Jaguar es un servidor de aplicación, que sirve para dar de alta componentes lógicos de un negocio y HTML dinámicos generados por Servlets y páginas JavaServer (JSPs). Jaguar soporta "pequeños" clientes de Web, utilizando formularios de HTML o Java Applets, y procesadores independientes.

Los clientes pueden ser implementados utilizando PowerBuilder, Java (CORBA o Enterprise JavaBeans), Actives, C++, o incluso interfaces MASP (Methods as Stores Procedures).

Los componentes son módulos reusables de código que combinan tareas relacionadas (métodos) dentro de una interfase bien definida. Los componentes de Jaguar son instalados en un servidor Jaguar y contiene los métodos que ejecutan los negocios lógicos y acceso a datos.

El administrador instala los componentes de código ejecutable en el servidor Jaguar. Los componentes pueden ser distribuidos a través de una red (incluyendo el Internet o una Intranet) o diferentes servidores. Los componentes instalados pueden ser utilizados por cualquier número de aplicaciones independientes.

EAServer contiene características que permiten mejorar los Web Sites y el Procesamiento de Transacciones en línea (OLTP), de manera rápida y fácil:

ESTÁNDARES ABIERTOS

EAServer esta comprometido con estándares abiertos, al soportar más bases de datos, tanto para plataformas Unix como para NT, además:

- La amplia variedad de clientes, incluyendo navegadores de solo HTML (ultradelgados), applets de Java (delgados), EJB y programas cliente de PowerBuilder y CORBA.
- La amplia variedad de componentes y herramientas desarrolladas, incluyendo PowerBuilder, PowerJ, Visual C++ y Visual Basic.
- La mayoría de interfaces para servidores Web, tales como: ISAPI, NSAPI (Netscape Server API) y CGI.

ALTO RENDIMIENTO

El servidor multihilos proporciona un número de funciones sofisticadas que sincronizan las capacidades de bases de datos relacionales, páginas dinámicas secretas y programación script, permitiendo el EAServer soportar la mayoría de la demanda de aplicaciones Web, en el mundo.

SOPORTE PARA APLICACIONES WEB

Una aplicación Web, es una unidad de desarrollo, para contenido relacionado a Web, páginas JavaServer (JSPs) y Java Servlets. Una aplicación Web contiene archivos estáticos, implementación de clases JSP y servlets y un descriptor que describe cuales de los archivos, servlets, y JSPs deben ser configurados en el servidor anfitrión.

ESCALABILIDAD AVANZADA

Un robusto multiprocesador en ejecución, ejecuta componentes y procesos de transacción, a alta velocidad, mientras se generan los usos más eficientes de recursos, CPU, memoria y redes de los sistemas disponibles.

ADMINISTRACIÓN DE TRANSACCIONES FLEXIBLES

Un administrador de transacciones, oculta virtualmente toda complejidad del control y coordinación de aplicaciones desarrolladas implícitamente con las transacciones.

Lo anterior representa las principales funciones que componen JAGUAR, de manera muy general, ahora se hará un recuento de las características que hacen de Jaguar, un servidor de aplicaciones que presenta ciertas ventajas sobre el resto de servidores de su tipo.

Jaguar es una plataforma para transacciones multinivel, en aplicaciones intensivas de negocio en el Internet. Estas aplicaciones se mueven más allá de actualizaciones o de la colección de datos dinámicas unidireccionales a las actualizaciones de dos vías en tiempo real de la información

crítica del negocio. También se pueden migrar aplicaciones tradicionales de transacciones cliente / servidor a aplicaciones multicapa de Jaguar.

Jaguar proporciona una estructura para desarrollar la lógica multicapas de aplicaciones basadas en componentes distribuidos. Jaguar es el único servidor de transacción de componentes, basado en estándares abiertos. Combina las características de un monitor de procesamiento de transacciones y un manejador de peticiones (ORB), para proporcionar un paquete de fácil uso y que permite desarrollar rápidamente desarrollo de aplicaciones para transacciones. Con Jaguar, los desarrolladores pueden enfocarse en la solución de problemas de negocios, en lugar de programar infraestructura de aplicaciones.

El "corazón" de Jaguar es un servidor de transacciones de alto rendimiento que proporcionan un control eficiente de las sesiones de clientes, seguridad, hilos, conexiones a bases de datos y flujo de transacciones.

La escalabilidad de Jaguar y la independencia de plataforma, permite desarrollar la aplicación en una máquina de un procesador barato, además de desarrollarlo en un servidor de multiprocesadores a un nivel empresarial.

La lógica por parte del cliente en aplicaciones empresariales debe ser tan pequeña y eficiente como el ancho de banda de la red permita. Para lograr esta meta, las aplicaciones son divididas en tres partes: lógica de presentación, lógica del negocio y lógica de la base de datos. La base de datos reside en la capa de abajo del sistema empresarial, para mantener y asegurar los activos de información de la organización. La lógica del negocio, reside en la capa de en medio o servidor de transacción de componentes. La lógica de presentación esta en la computadora de escritorio de los usuarios, en el nivel más alto o dinámicamente se encuentra en su computadora.

El servidor de transacciones, es el responsable de ejecutar y asegurar la inmensa mayoría de la lógica de negocio de la corporación. Esto genera un componente crítico en la arquitectura de las redes centralizadas. El navegador Web conecta a Jaguar o aun servidor Web, vía http para bajar una página HTML, la cual contiene un applet de Java que ejecuta que ejecuta funciones de presentación. El applet se comunica con Jaguar, llamando a componentes middletier, que ejecutan lógica de negocio. Los componentes middletier pueden utilizar Librerías Cliente de Sybase, ODBC (Open Data Base Connectivity), o JDBC (Java Data Base Connectivity) para comunicarse con un tercer hilo DBMS (Data Base Management System). El DBMS almacena, procesa y protege los datos corporativos. Jaguar controla un grupo de conexiones para el respaldo de la base de datos y coordinación del procesamiento de transacciones a esos servidores.

Los componentes son objetos que se encuentran en un servidor (por ejemplo, Servidor Jaguar) y pueden ser utilizados por diferentes programas, independientemente de los lenguajes de programación. Un cliente ejecuta los métodos en un componente. En lugar de crear un programa masivo, se genera un cliente que contiene el GUI, (Graphical User Interface) código validado y varios componentes individuales que contienen la funcionalidad (lógica de negocio) del programa.

Debido a la separación de la funcionalidad de la GUI, se puede fácilmente actualizar y cambiar la funcionalidad de un programa sin tener que cambiar la GUI. Además, múltiples clientes (incluyendo Dynamo) pueden utilizarse al mismo tiempo.

Con Jaguar se puede:

- Implementar Java / Enterprise JavaBeans, Java Servlets componentes nativos de PowerBuilder, componentes ActiveX y componentes C / C++ para el servidor de transacciones de la middletier.
- Administrar el servidor de transacciones para utilizarse Sybase Central (con el Jaguar Manager plugin), el cual soporta un navegador de interfaces para componentes, declaración de reglas basadas en seguridad y monitoreo en tiempo de ejecución.
- Generar Clusters de servidores Jaguar, para lo cual uno pueda ser capaz de balancear cargas de peticiones de clientes, generar componentes altamente disponibles, sincronización de componentes, paquetes y otras configuraciones de servidor, a través de cluster de servidores y automáticamente fallan sobre los componentes en un servidor caído a componentes en otro servidor. Permite configurar medidas y políticas para controlar dinámicamente el balanceo de cargas de componentes ejecutados sobre un servidor en un cluster de Jaguar.
- Crear y establecer aplicaciones Web que contengan archivos estáticos, Servlets, JSPs y especificar las propiedades necesarias para que puedan trabajar de manera conjunta.

Jaguar proporciona además los siguientes servicios:

- Sesión transparente del cliente y administración de componentes del ciclo de vida.
- Conexiones secretas, que permiten el rehúso de conexiones con bases de datos.
- Administración de transacciones para simplificar el uso de datos y recursos compartidos. Un modelo de transacción distribuida de Jaguar es el CORBA OTS (Object Transaction Service) y XA (arquitectura Xopen). Jaguar incrusta un coordinador de transacciones, para administrar las transacciones OTS / XA.
- Soporte en conjuntos resultado, permitiendo la recuperación eficiente de datos tabulares en aplicaciones cliente.
- Soporte para componentes EJB (Enterprise JavaBean) desarrollados de acuerdo con la versión 1.1 de la especificación EJB.
- Hospeda a Web Sites Dynamo a fin de acceder esos sitios desde un navegador.
- Declaración de reglas básicas de seguridad para restringir conexiones cliente y los componentes que puedan estar involucrados por una sesión específica de un cliente.
- Identidades para mapear nombres de identidades lógicas hacia nombre de usuario, contraseña y características requeridas por una sesión SSL. Las identidades nombre son configuradas en modo de ejecución para componentes y métodos de componentes.
- El nombramiento de servicios permite asociar un nombre lógico con un objeto tal como un componente. Permite a los clientes, fácil ubicación de un componente en cualquier parte de una red y ejecutar sus componentes.
- SSL soporta clientes Java, ActiveX y C++. Además, se puede instalar un administrador Jaguar, independientemente del administrador de Seguridad de un servidor Jaguar, el cual permite el manejo de certificados en máquinas donde los clientes utilizando SSL se conecta.
- Capacidad para agregar certificados digitales IDs, para reglas de Jaguar, a fin de que el acceso a componentes sea autorizado basado en certificados cliente bajo SSL.
- El cliente y servidor apoyan la seguridad en PKI. Pueden utilizarse certificados de llave pública para autenticar servidores Jaguar y conexiones cliente.

- Una modalidad en la característica de protección, permite establecer requerimientos mínimos de seguridad en paquetes, componentes y nivel del método.
- Los servidores Jaguar, pueden utilizar una maquina virtual de Java, con excepción del valor por defecto.
- Instalación de un servidor Jaguar como un servicio NT, a fin que el servidor Jaguar inicie automáticamente cuantas veces Windows NT inicie.
- Un servidor JagRepair, para reparar los errores de configuración que previenen desde el inicio de un servidor Jaguar.
- Implantación plena de componentes Java desde el Power IDE
- Implantación y sincronización de archivos de aplicación (tales como applets Java y archivos HTML) asociados con un componente, servlet o paquete.
- Renovar componentes Java e implementación de archivos servlets sin cerrar o restaurar el servidor Jaguar.
- Un servicio de mensaje, que puede utilizarse para la notificación asíncrona del acontecimiento
- JavaMail para enviar correos desde un componente Java, JSP o servlet.

DYNAMO

Dynamo es perfecto para implementar aplicaciones Web para clientes delgados (o ligeros), ya sea en Internet o Intranet, puede soportar millones de "golpes" por día. Dynamo proporciona las herramientas necesarias para construir y controlar un cliente delgado en aplicación Web, conteniendo ambos, estáticos y dinámicos HTML.

El servidor de aplicación Dynamo, actúa como un intermediario entre el servidor Web y el sistema manejador de bases de datos DBMS. El servidor de aplicación procesa templates, las cuales son páginas HTML con incrustaciones de SQL o enunciado COMPONENT, scripts DynaScripts.

Con Dynamo se puede:

- Construir templates, enunciados SQL y scripts para páginas Web mediante el uso de una colección de poderosos asistentes personalizados.
- Modificar el código fuente de las páginas Web mediante el uso de un editor que realza la sintaxis.
- Escribir scripts en Dynascript, el cual es diseñado específicamente para escribir en el servidor con el servidor de aplicación Dynamo. DynaScript es totalmente compatible con ECMAScript. ECMAScript es el estándar de JavaScript y lenguajes Jscript.
- Almacenar y ejecutar la aplicación Web del thin-client, en una base de datos y controlarlo utilizando un Sybase Central, una herramienta gráfica de administración del servidor.
- Utilizar tecnologías de replicación de bases de datos existentes, para distribuir soluciones Web (incluyendo aplicación y datos) donde la aplicación sea necesaria. Soluciones Web pueden ser distribuidas por múltiples servidores para balanceo de cargas, varios grupos de trabajo fuera de línea o incluso para laptops de usuarios móviles.
- Utilizar el servidor PowerDynamo

Dentro de la documentación que contiene todo el sistema Jaguar, se puede dividir en 4 áreas diferentes:

- Jaguar CTS Getting Started. Describe como iniciar la preconfiguración del servidor Jaguar.
- Jaguar CTS System Administration Guide. Proporciona información detallada en cuanto a configuración del servidor Jaguar utilizando el plugin del administrador de Jaguar para Sybase Central. También describe la configuración HTTP, administración de conexión caché, seguridad, servicios de nombres y administración de archivo de aplicación.
- Jaguar CTS Programmer`s Guide. Introduce la arquitectura de Jaguar y contiene instrucciones paso a paso para construir componentes Jaguar y aplicaciones cliente que ejecutan componentes Jaguar.
- Jaguar CTS AP Reference. Contiene páginas de referencia para Clases de Java, objetos Actives y rutinas de C para Jaguar.

En el **Anexo I**, se encuentra una descripción de los capítulos que comprenden cada uno de estos documentos, y que conocerlo permite tener una idea más clara del lugar a recurrir en caso de una complicación en la configuración de un servidor Jaguar, y principalmente la parte segura..

5.3 AMBIENTE DE DESARROLLO

La plataforma sobre la que se encuentra instalado Jaguar en la institución bajo estudio, es una red Novell y Windows NT, con Fast-Ethernet a 100MB, donde la infraestructura de desarrollo es Jaguar sobre Windows NT (Clientes) y clientes en Windows 98 y Windows 2000 Pro., así como el manejo de programación en lenguajes C, C++, Java, Delphi y PowerBuilder

5.3.1 CARACTERÍSTICAS DE SEGURIDAD

Configuración de Seguridad

Para proteger los datos en Internet, los Servidores Jaguar proporcionan las facilidades de SSL, para autenticar clientes y servidores y encriptar datos.

Establecer seguridad en el sitio del cliente, mediante el uso del módulo de PKCS #11, proporcionado tanto por Netscape como por Sybase. Jaguar proporciona muestra de certificados y muestra de applet de seguridad que permite evaluar las conexiones.

Se puede también establecer una administración de password y generar reglas que permitan limitar el acceso a paquetes y sus componentes.

Las aplicaciones multicapa distribuidas están sujetas a una variedad de riesgos como:

- Saboteo de datos.- alguien deliberadamente modifica la información que viaja entre un cliente y un Host.

- "Escucha" .- una tercera parte escucha en una sesión cliente-host
- Enmascaramiento.- un sujeto suplanta a otro y accede información que no le corresponde

5.3.2 CRIPTOGRAFÍA DE LLAVE PÚBLICA

Para mantener comunicaciones seguras entre un cliente y un Host, las técnicas de criptografía de llave pública son utilizadas para:

- Autenticación. Verificando la identidad de ambos, cliente y servidor: las técnicas de criptografía de llave pública usan certificados de firma digital que identifican entidades de red.
- Encripción. Modificación de datos que sólo deben ser leídos por quien se pretende.

En el **Anexo F**, se pueden encontrar algunos aspectos específicos sobre criptografía en Jaguar.

5.3.3 SSL, HTTPS E IIPOS

SSL proporciona seguridad en conexiones de red. Especialmente, SSL utiliza encripción de llave pública para proporcionar:

- *Autenticación* de cliente y servidor utilizando certificados
- *Encripción*, la cual previene que terceras partes conozcan los datos transmitidos.
- *Verificación de integridad*, la cual detecta si los datos transmitidos son alterados

Paquetes de otros protocolos pueden ser encapsulados dentro de paquetes SSL. Una conexión en la cual el protocolo de aplicación es incrustado dentro de SSL, es una conexión con túnel SSL.

Ambos IIOP y HTTP pueden ser tuneleados dentro de SSL, con lo cual esos protocolos toman ventaja de las características de seguridad de SSL. Por ejemplo, conexiones HTTP encapsulan paquetes HTTP dentro de paquetes SSL. Un Web Browser crea una conexión segura http cada que se cargue una página desde un URL que inicie con "HTTPS".

5.3.4 ADMINISTRACIÓN DE SEGURIDAD DE JAGUAR

La configuración de Jaguar para aceptar conexiones de cliente sobre protocolos seguros IIPOS y HTTPS utilizan:

- Administración de la seguridad mediante control de llaves y certificados de Jaguar
- El administrador Jaguar define perfiles de seguridad que establecen varios niveles de seguridad en Jaguar y los asignan a un "receptor". Los perfiles permiten determinar:
  - Requerimientos de autenticación de cliente y servidor
  - Algoritmos de encripción y decripción
- Jaguar utiliza certificados y "receptor" para autenticar clientes, si es necesario encriptar y decriptar datos.

- Integración Segura: Jaguar integra una infraestructura segura de llave pública (PKI) que habilita servidores y clientes Jaguar para utilizar identificadores seguros de cliente / servidor.

Existen tres escenarios involucrados en ID's confiables y certificados no seguros:

- Cliente y servidor Jaguar utilizan certificados no confiables
- Cliente confiable y servidor Jaguar no confiable(y viceversa)
- Cliente y servidor Jaguar utilizan certificados confiables

Cliente y servidor utilizan certificados no confiables

En este escenario, el administrador de seguridad de Jaguar, utiliza un token de PKCS #11 de Sybase, para controlar las llaves y certificados de Jaguar. En el cliente, se utiliza un mecanismo Browser para administrar llaves y certificados para applets de Java o administrador de seguridad independiente para acceder el token de PKCS #11 de Sybase para administrar llaves y certificados para C++ y aplicaciones Java.

Cliente Seguro y servidor Jaguar no confiable (y viceversa)

En un ambiente mixto de ID's seguros y certificados no confiables, cada sitio (cliente y servidor) debe importar el certificado de otra CA, y debe reconocerla como si fuese de una CA confiable.

Cliente y servidor Jaguar utilizan certificados seguros

Cuando ambos utilizan ID's seguros, utilizan Confianza para manipular los ID's y utilizar el administrador Jaguar para establecer un perfil de seguridad que utiliza esos ID's.

Uso de Netscape para administrar certificados en el cliente

PKCS #11 es un estándar RSA que especifica un API (Programa de Aplicación de Interfaz) llamado Criptoki (crypto-key, abreviatura de cryptographic token interface) que ejecuta funciones criptográficas, tales como administración de par de llaves y certificados.

Netscape 4.0x suministra un módulo de PKCS #11 que permite administrar los certificados en el sitio del cliente. Sybase también proporciona un módulo de PKCS #11 que permite administrar sus certificados. Sybase recomienda que se instalen los módulos de PKCS #11 dentro de Netscape, el cual proporciona acceso inmediato a los certificados muestra del servidor Jaguar.

Tareas del Administrador de Seguridad

El administrador de seguridad permite controlar llaves y certificados utilizados por Jaguar:

- Control de administrador de seguridad
- Control de evaluación de la CA
- Administración de llaves

- Administración de certificados
- Se puede instalar y usar el administrador de seguridad en una máquina independiente o en una máquina cliente manejar llaves de cliente, certificados e información segura en una base de datos local.
  El administrador de seguridad en el cliente, permite clientes C++, CORBA y aplicaciones Java, para acceder a servidores Jaguar utilizando características SSL sobre conexiones IIOP.
- La evaluación de CA, es una autoridad reconocida que firma peticiones de usuarios certificados. Esos certificados pueden ser utilizados por clientes y Jaguar para probar las características de seguridad de sus aplicaciones. Los certificados firmados por la CA no son propuestos para aplicaciones comerciales. Si ya se cuenta con una certificación no es necesario este paso.
  - Creación de una CA
  - Generación de un Certificado firmado de usuario
  - Procesamiento de una petición de certificado
  - Exportación de un certificado por parte de la CA

El administrador de seguridad despliega cualquier llave privada que no tenga un certificado asociado con ella, incluyendo llaves privadas que tienen una petición de certificado expirado.

Sybase recomienda que se borre cualquier llave privada que no tenga un certificado de petición asociado con ella.

- El administrador de seguridad viene con varios certificados preinstalados. Jaguar acepta certificados de cliente solo si ellos han sido firmados por una CA. El usuario puede modificar los atributos de seguridad para cualquiera de esos certificados.

Cuando se instala o se exporta un certificado, el administrador de seguridad determina el tipo de certificado basado en la extensión.

PKCS #12 es un estándar de RSA que especifica una sintaxis de transferencia para identidad personal de la información. Jaguar soporta del estándar PKCS #12, lo cual permite mover los certificados de usuario y llaves secretas entre sistemas y programas que soportan el estándar mencionado, tal como Netscape Communicator y Microsoft Internet Explorer.

La transferencia de certificados de usuario y llaves secretas, permite utilizar certificado y llave privada en el ambiente de seguridad del objetivo. Exportando, instalando y etiquetando un certificado de una CA el ambiente de seguridad del objetivo, simplemente permite aceptar certificados que han sido firmados por la CA.

La implementación de PKCS #12 de Sybase, permite transferir certificados y llaves secretas ya sea en formato tradicional (encripción de 128 bits) o formato internacional (encripción de 40 bits)

El administrador de seguridad verifica la firma, fecha de expedición y validez del certificado. Si el certificado es parte de un canal de certificados, verifica cada uno de ellos.

Un canal involucra más que un certificado. Cada certificado en el canal es firmado por el certificado antecedente.

El administrador de seguridad permite borrar los certificados propios, llaves privadas asociadas, unidad certificadora y certificados obtenidos de otros.

5.3.5 PERFILES DE SEGURIDAD

Definen las características de seguridad de una sesión cliente-Jaguar. Se asigna un perfil de seguridad a un "listener" es cual es un puerto que acepta peticiones de conexión de cliente de varios protocolos. Un servidor Jaguar soporta múltiples "listeners". Los clientes que soportan las mismas características pueden comunicarse son Jaguar vía el puerto definido en el "listener".

Cada *perfil de seguridad* tiene asociado una característica de seguridad, la cual es un nombre que tiene un conjunto de CipherSuites asociados con él. Una característica de seguridad junto con el CipherSuites, define esas características de una conexión cliente/servidor:

- *Protocolo.* Todos los perfiles utilizan SSL versión 3 como protocolo base. El tráfico IIOPS y HTTPS es tuneleado a través de SSL.
- *Autenticación.* Sea o no utilizada la autenticación, los perfiles pueden soportar:
  - *No autenticación.* Ningún cliente o servidor necesita proporcionar certificado para autenticación.
  - *Autenticación de servidor.* Solo el servidor necesita proporcionar un certificado, el cual será aceptado o rechazado por el cliente.
  - *Autenticación de cliente y servidor.* Ambos proporcionan certificados, los cuales serán aceptados o rechazados por el otro.
  - *Resistencia y método de encripción.* Si son o no los datos encriptados, y si la llave y método son fuertes.
  - *Uso internacional.* Todos los CipherSuites están disponibles localmente, pero no todos son adecuados para exportar fuera de USA y Canadá.
  - *Método de Hashing.* El método utilizado para crear el "*message digest*".

Ejemplo: el CipherSuite SSL_RSA_UIT_NULL_MD5 puede ser interpretado como:

- **SSL** el protocolo utilizado. Todos los perfiles son SSL
- **RSA** el algoritmo utilizado para intercambio de llaves
- **NULL** no encripción
- **MD5** el método Hash utilizado para computar el message digest.

Existen cuatro categorías de características de seguridad:

- *Simple*
- *Fuerte*
- *Nacional*
- *Internacional*

Un "listener" es un puerto Jaguar que comunica a los clientes utilizando varios protocolos. Para protocolos que utilizan características de seguridad de SSL (HTTPS y IIOPS), se les puede asignar un perfil de seguridad para el "listener". El perfil define características de seguridad del

"listener". Para protocolos que no utilizan SSL (HTTP, IIOP y TDS), el perfil de no seguridad es requerido.

Calidad de protección Jaguar

El administrador de seguridad permite agrupar la calidad de protección (QOP) para paquetes, componentes y métodos de Jaguar. QOP establece un mínimo nivel de encripción y autenticación que un cliente debe conocer antes de acceder a un negocio lógico

Ahora bien, después de esta descripción a grandes rasgos de las capacidades de Jaguar, así como de algunas metodologías de configuración, verificación y aseguramientos de las principales características de seguridad: confidencialidad, disponibilidad e integridad, haremos una validación de código, acentuando las características seguras, que proporciona Jaguar, con la implementación de componentes en Java principalmente, así como algunas diferencias o dificultades para compaginar seguridad de aplicación y de sistema operativo.

De manera muy especial trataremos los principales rasgos de seguridad proporcionados por J2EE y CORBA, quienes inicialmente fueron los principales motivos para la adquisición de este producto (JAGUAR), es decir, la compatibilidad o interoperabilidad de componentes en Java. CORBA, ActiveX y C++ fue uno de los principales motivos por los cuales se opto por adquirir Jaguar.

En primera instancia, realizaremos una descripción de cada uno de los puntos considerados en nuestra *lista de chequeo* para concluir esta sección determinando la calificación resultado de la evaluación a las aplicaciones que se ejecutan sobre JAGUAR.

5.4 EVALUACIÓN

Lista de Chequeo				
	Aplicación	**Si**	**No**	**No aplica**
Secc. I	**Autenticación**			
I.1	*Administración de Password*			
a	**Usuarios**			
1	Generación			
2	Modificación			
3	Eliminación			

4	Características de Password			
4.1	***longitud***			
4.2	***formación***			
4.3	***tiempo de vida***			
b	**Mapeo a Password de sistema**			

Cómo se puede observar hemos determinado que en cuanto a la autenticación, JAGUAR y a la vez sus aplicaciones siguen estrictas "normas" para cumplir válidamente con este aspecto.

En primer lugar sólo existe acceso no limitado para el administrador de Jaguar, y para adicionar seguridad, es posible establecer administración de Passwords y autenticación del sistema operativo, esto se logra desde al administrador de Jaguar, en las opciones de *File-Propiedades de Servidor-Seguridad-Administración de Password*, es precisamente en esta ubicación donde se puede establecer una administración de Password para los usuarios del servidor, pero sólo el administrador de Jaguar (*jagadmin*) puede: acceder al Administrador de Jaguar, asignar o quitar un Password, habilitar o deshabilitar una autenticación de usuario.

Además Jaguar permite autenticación del Sistema Operativo: es decir, selecciona esta opción, que se encuentra en la misma ubicación de la anteriormente señalada, Jaguar *mapea* los usuarios cliente a los nombres de usuario y Password del sistema operativo, y estas características protegen también a las aplicaciones que se ejecutan sobre Jaguar.

En resumen el soporte de autenticación que ofrece Jaguar sus aplicaciones son:

- *Autenticación del sistema operativo*: los nombres de usuario para una conexión de Jaguar, son mapeados directamente a su nombre de login en el sistema operativo. Se puede habilitar la autenticación nativa con el administrador de Jaguar utilizando la hoja de propiedades de servidor.
- *No autenticación*: ni nombres de usuario ni password son requeridos por la configuración default
- *Autenticación basada en certificados SSL*: se puede configurar un puerto seguro IIOP que requiere autenticación mutua (cliente y servidor). Los clientes pueden tener un certificado válido SSL, a fin de conectarse a un puerto, y el certificado debe ser publicado por una autoridad certificadora, que es reconocida por el servidor de Jaguar.

Cuando los clientes se conectan con un certificado SSL, el cliente también proporciona un nombre de usuario y password para la conexión, además del certificado. Jaguar ejecuta un chequeo de autorización basada en nombre de usuario de Jaguar. El nombre de usuario e información de certificado SSL, están disponibles para los componentes a través de la incorporación de componentes CtsSecurity/SessionInfo.

El servidor de Jaguar proporciona soporte nativo SSL sin el uso de Proxies. En el sitio del cliente, el Jaguar Java ORB soporta SSL, cuando se ejecuta Netscape 4.0, Java applets y aplicaciones Java, C++, PowerBuilder y componentes ActiveX, pueden utilizar nativamente el SSL.

Clientes C++ y PowerBuilder requieren que un sistema de infraestructura de llave pública (PKI) este disponible en el cliente para manejar certificados digitales. Se puede utilizar el administrador de seguridad, el cual administra certificados de bases de datos de Jaguar.

- *Calidad de Protección*: el administrador de Jaguar, permite configurar la calidad de protección de los paquetes, componentes y métodos. QOP establece un mínimo nivel de encriptación y autenticación que un cliente debe tener antes de que pueda acceder a un negocio lógico.

	Aplicación	**Si**	**No**	**No aplica**
Secc. XVII	**Compartición de recursos**			
XVII.1	*Grupos*			
XVII.2	*Usuarios*			
XVII.3	*Con privilegios*			
XVII.4	*Sin privilegios*			
XVII.5	*Públicos*			
Secc. XVIII	**Clasificación de la información**			
XVIII.1	*Por aplicación*			
XVIII.2	*Por los usuarios que la manejan*			
XVIII.3	*Por su sensibilidad*			

Jaguar soporta un tipo de autenticación, en la cual el servidor solo intenta autenticar un cliente, cuando éste, desea acceder recursos restringidos; mientras que el cliente solo acceda recursos que no requieren autorización, el servidor no intentará autenticarlo. Cuando un servidor autentica un cliente, el cliente es autenticado para todas las aplicaciones y referencias sobre el servidor. Se puede implementar autenticación de un cliente para todo un servidor mediante el uso de *cookies* o reescribibles de la URL. Una referencia a las credenciales de seguridad de los clientes es guardada en un cookie o codificada en el URL.

	Aplicación	Si	No	No aplica
I.2	*Administración de usuarios*			
a	**Lista**			
b	**Agregar**			
c	**Funciones de búsqueda**			
d	**Cambio de características**			
e	**Eliminar**			

Se puede suministrar un nombre de usuario y Password que sea válido para la máquina donde el servidor Jaguar se esta ejecutando.

Otra protección interesante es la que se proporciona a las aplicaciones que "*corren*" sobre Jaguar la opción de: "*Habilitación de usuarios y validación de grupos*", lo cual actúa de la siguiente manera: si se habilita esta opción, los nombre de usuario y grupo antes de ser adicionados a cualquier de las siguientes carpetas: usuarios autorizados, grupos autorizados, usuarios excluidos, grupos excluidos, son validados.

Jaguar, permite la administración de usuarios, genera lista de ellos, permite agregar usuarios, eliminar, cambio de características o atributos, pero no maneja funciones de búsqueda.

Ahora bien, Jaguar permite el manejo de perfiles o identidades, las cuales a su vez permiten definir nombres de usuario y Passwords utilizados por los intercomponentes llamados. Si un componente ejecuta las propiedades de identidad, entonces los intercomponentes llamados, publicados por el componente utilizan el nombre de usuario y Passwords definidos en una identidad del administrador de Jaguar.

Para crear una identidad:

- resaltar la opción de Identidades en el Administrador de Jaguar
- elegir Archivo y nueva identidad
- introducir un nombre para la nueva identidad y pulsa clic en *crear nueva identidad*, entonces introduce las propiedades de identidad descritas en *propiedades de identidad.*

Dada la justificación anterior, Jaguar como servidor de aplicaciones, y por ende las aplicaciones, cumplen la administración de usuarios, en todos los puntos que hemos propuesto para este aspecto.

	Aplicación	Si	No	No aplica
I.3	*Administración de Certificados*			
a	**Autoridad Certificadora Externa**			
b	**Certificación Interna**			
1	Generación			
2	Renovación			
3	Eliminación			
4	Distribución			
5	CRL (Lista de certificados revocados)			
5.1	***Generación***			
5.2	***Distribución***			
5.3	***Mantenimiento***			
c	**Certificados entre atributos de usuario (Extensiones)**			

Jaguar permite el manejo de certificados con entidades externas, así como generar sus propios certificados internos, para utilizarlos entre aplicaciones. Jaguar utiliza la PKCS # 11 que es un estándar de RSA que especifica una API (Programa de Interfaz de Aplicación) llamado *Cryptoki.* esto ejecuta funciones criptográficas tales como una par de llaves y administración de certificados.

Para mayor referencia del proceso de administración de certificados visite el **Anexo F**

Dada toda la información anterior, hemos convenido que en cuanto a Administración de Certificados se encuentra calificada en alto nivel de seguridad cualquier aplicación que se ejecute sobre Jaguar, dado que las características de este las cubre.

I.4	*Administración de llaves*			
a	**Generación**			
b	**Distribución**			
c	**Repositorio**			

Para ver la instalación de las llaves privadas en el módulo de seguridad, seleccione la carpeta de *llaves privadas*. Las llaves privadas se desplegarán del lado derecho de la ventana.

El administrador de seguridad despliega cualquier llave privada que no tenga un certificado asociado con ella, incluyendo llaves privadas que tienen una petición de certificado expirada. Por ejemplo, se puede generar un par de llaves y petición de un certificado desde una CA al mismo tiempo. Puede tomar varios días recibir su certificado. Mientras tanto, la llave privada se despliega cuando se resalta la carpeta de Llaves privadas.

Sybase recomienda borrar cualquier llave privada que no tenga una petición de certificado vigente asociado a ella.

Para ver información acerca de Llave privada:

- Seleccionar la carpeta de Llaves privadas
- Resaltar la llave cuya información se desea ver.
- Seleccionar *Archivo / información de llave*. La caja de diálogo de la información de Llave, despliega la longitud de la llave.

Para eliminar una llave privada:

- Seleccionar la carpeta *Llaves privadas*. Las llaves privadas se despliegan del lado derecho de la ventana
- Seleccionar la llave que se desea eliminar
- Seleccionar *Archivo / Borrar llave*

En conclusión, las aplicaciones sobre Jaguar, permiten la generación y administración de llaves, por tanto en cuanto a este aspecto, cumple con el grado de aseguramiento requerido para ser considerado seguro.

	Aplicación	**Si**	**No**	**No aplica**
I.5	*Dispositivos Biométricos*			
a	**Local**			
b	**Remoto**			
I.6	*Tarjetas Inteligentes*			

El servidor de aplicaciones Jaguar, no se encuentra aún funcionando al exterior dado que se encuentra en éste proceso de evaluación funcional y de seguridad, por tal motivo no se han hecho pruebas con dispositivos biométricos, ni tarjetas inteligentes, pero en este caso, a pesar de ser un aspecto de autenticación y control de acceso, hoy día en vigor y con gran auge, podría dejarse este trabajo al sistema operativo o plataforma sobre la cual Jaguar se encuentre funcionando, para

determinar dicho acceso al sistema utilizando ambos factores: dispositivos biométricos y tarjetas inteligentes. Cabe mencionar que para ambos sistemas donde se desea implantar Jaguar, NT y Unix, estos dos factores están ampliamente desarrollados. Por tanto, a pesar de que directamente las aplicaciones de Jaguar y el mismo, no permiten el manejo de ellos, consideramos no relevante la calificación obtenida.

	Aplicación	**Si**	**No**	**No aplica**
I.7	*Criptosistemas*			
a	**Local**			
b	**Red**			

En cuanto a estos puntos, se ha hablado dentro de este capítulo ampliamente acerca de las potencialidades de Jaguar, al respecto, por tanto se ha considerado pertinente únicamente mencionar que es, también robusto en esta característica de seguridad.

	Aplicación	**Si**	**No**	**No aplica**
Secc. II	**Control de Acceso**			
II.1	*Listas de control de acceso*			
II.2	*Listas de Capacidad*			
II.3	*Discrecionario*			
II.4	*Mandatario*			
II.5	*Basado en mínimos privilegios*			
II.6	*Confirmación periódica de los derechos de acceso*			
II.7	*Remoto*			

Jaguar en una sección llamada *Declarativas de Seguridad*, y con lo cual corroboraremos que también en este aspecto ha acreditado la evaluación, menciona lo siguiente:

Jaguar CTS proporciona soporte para integrar autenticación y autorización de usuarios. Ellos son autenticados cuando una aplicación cliente genera un proxy u objeto stub (una conexión es realizada cuando la aplicación genera el primer proxy o stub; otros proxies o stubs pueden utilizar las mismas conexiones o asignar espacio como sea necesario). Cada componente tiene una LISTA DE CONTROL DE ACCESO que determina que usuarios tienen permitido invocar los

componentes, si un usuario no tiene autorizado el uso de un componente, intentará crear stubs o proxies con nombre de usuario falso.

	Aplicación	Si	No	No aplica
Secc. III	**Integridad**			
	¿Utiliza método para revisión de integridad?			
	** En caso de responder NO. pase a la Secc. IV*			
III.1	*CheckSum*			
III.2	*MD2*			
III.3	*MD4*			
III.4	*MD5*			
III.5	*SHA-1*			
III.6	*Otro*			

La integridad de la información que entre las aplicaciones es intercambiada, es considera de óptimo nivel, debido al manejo de MD5, dicho algoritmo también fue ampliamente explicado en capítulos anteriores, mencionando las fortalezas que éste presenta a la integridad de los mensajes, mediante un compendio de 56 bits, del mismo. Por lo que no nos queda más que dictaminar que en esta evaluación, todas las aplicaciones que se ejecuten sobre Jaguar, están protegidas en cuanto a integridad.

	Aplicación	Si	No	No aplica
Secc. IV	**Disponibilidad**			
IV.1	*Información disponible cuando es requerida*			
IV.2	*Cluster*			
IV.3	*Manejo de carga*			
Secc. VII	**Tolerancia a fallas**			
VII.1	*Cluster*			
VII.2	*Manejo de carga*			

Partiendo del hecho que un *cluster* es un grupo de servidores que comparten información duplicada en un repositorio o almacén. La sincronización permite conexiones al servidor Jaguar primario en un cluster y distribuir información en un repositorio para "sincronizar" una o más información de otro servidor(es) en el cluster. Se puede también sincronizar servidores Jaguar no "clusterizados". La sincronización proporciona una rápida y fácil forma de distribuir paquetes, servlets y alguna otra configuración de información entre servidores.

Cada cluster incluye un servidor primario, un grupo de servidores participantes y un conjunto de servidores de nombres:

- *El servidor primario* contiene la copia maestra de la configuración confidencial para todos los servidores en el cluster. Éste servidor distribuye (sincroniza) la configuración hacia otros servidores en el cluster.

- *Servidores participantes* o servidores no primarios, comparten un "nombre lógico de servidor", el cual corresponde a un servidor definido por Jaguar en el servidor repositorio primario. Varios servidores físicos en un cluster, comparten un nombre lógico de servidor; cada nombre de servidor comparte componentes y servlets, y utiliza la misma conexión caché y otra configuración de información. Cuando se configura un cluster, se pueden utilizar, múltiples nombres lógicos de servidor para partes de componentes. Para asegurar alta disponibilidad, cada nombre lógico de servidor, puede ser compartido por los últimos dos servidores físicos en el cluster.

Todos los servidores dentro de un cluster pueden compartir el mismo nombre siempre y cuando no este particionado.

Un cluster consiste de los últimos dos servidores de nombres. Cada servidor de Jaguar en un cluster conoce todos los servidores de nombre del cluster. Es obligatorio que todos los componentes de los servidores agrupados de Jaguar, multipliquen los servidores de nombre para proporcionar alta disponibilidad de los componentes del negocio, así como redundancia, si el servidor Jaguar es dado de baja, o incluso si es un servidor de nombres.

Típicamente cada servidor en un cluster, se ejecuta en diferente Host, de manera que cada servidor tiene su propia copia del repositorio entero y de todos los archivos requeridos para la ejecución de componentes. Sybase recomienda que se ejecuten los miembros del cluster desde su propio directorio de instalación.

La figura 5.4.1 muestra un cluster de Jaguar que utiliza al Host A como el servidor primario, para sincronizar a los servidores participantes, incluyendo el servidor de nombres. Cada servidor en el cluster es llamado Jaguar.

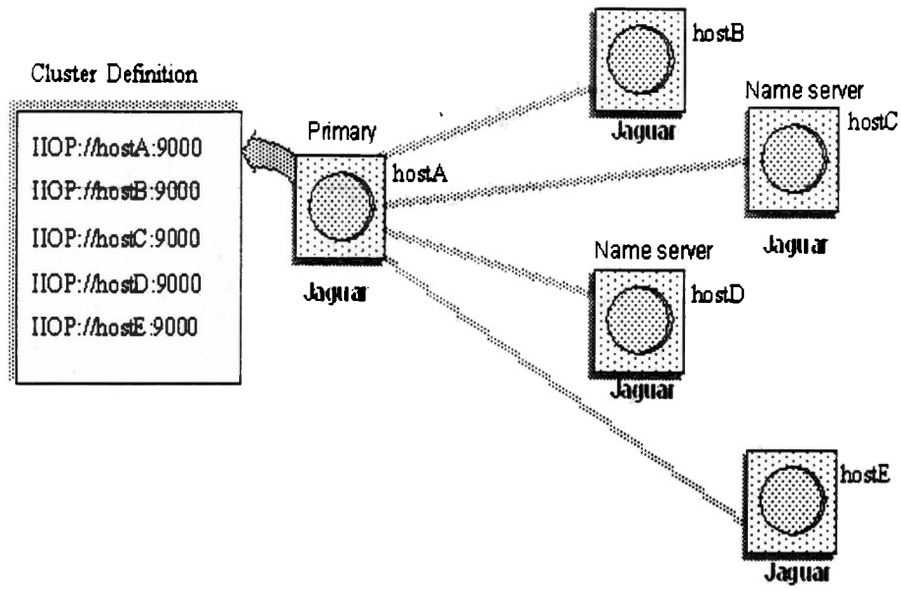


Fig. 5.4.1 Sincronización de Servidores [23]

Todos los Hosts de un cluster deben ser de plataforma del mismo tipo; es decir, no se pueden tener máquinas NT y Unix en el mismo cluster.

Es importante resaltar y repetir que, un servidor puede ser miembro de sólo un cluster. Para proporcionar alta disponibilidad, deben ser definidos al menos dos servidores de nombres por cluster.

Los servidores son definidos por URL, más que por el nombre de servidor de Jaguar. Cada servidor en un cluster puede ser llamado Jaguar. Si no se utilizan particiones, es fácil adicionar máquinas a un cluster, simplemente cambiando "*localhost*" en cada receptor para el nombre de Host Internet o para la dirección IP.

Existe un aspecto importante que debe considerarse en la creación y configuración de un cluster; después de adicionar un servidor no primario a un cluster, el administrador de Jaguar genera una alerta al conectarse directamente al servidor. El usuario puede actualizar directamente la configuración del servidor y sobre escribir cuando el cluster sea sincronizado, siempre y cuando, el nuevo servidor haya sido objetivo de al menos una sincronización antes de haber sido adicionado como miembro del cluster.

Otro aspecto que incluso repercute en el control de acceso es que el servidor de nombres en un cluster, utiliza detección, para periódicamente verificar qué servidores miembro son aceptados en conexiones cliente o han tenido falla. Si un servidor no acepta conexiones, el servidor de nombres no regresa información del perfil (Host y puerto) al cliente, y rutea la petición a otros servidores del cluster. El servidor de nombres también detecta cuando un servidor que falló está listo para aceptar conexiones nuevamente e inicia ruteando peticiones de cliente hacia ese servidor.

Si un servidor de nombres que utiliza almacenamiento temporal falla, el cluster automáticamente "sujeta" al reiniciar el servidor que ha fallado. De otra manera, el cluster proporciona acceso a componentes gracias a que el servidor de nombres permanece en el cluster.

Si un servidor de nombres utiliza almacenamiento permanente y soporte a fallas LDAP, el cluster no necesita "sujetar", pues LDAP puede dejar atrás los perfiles averiados resultado de un cliente innecesario de prueba y error.

En resumen se pueden definir cinco importantes aspectos que Jaguar ofrece en cuanto a disponibilidad preponderantemente:

- Balanceo de cargas: optimiza el funcionamiento del cluster de Jaguar mediante el ajuste de cargas a través de los servidores.
- Realce de componentes: se puede restringir el acceso a componentes mediante un subconjunto de servidores dentro del cluster, o generando disponibilidad desde todos los servidores.
- Alta disponibilidad: un cluster de Jaguar proporciona redundancia (alta disponibilidad) de componentes de negocio y servicios de Jaguar en caso de que un servidor dentro de un cluster falle.
- Componente automático sobre fallas: permite que la referencia del objeto de un cliente sea utilizable a través de los servidores, cuando un servidor dentro de un cluster falla.
- Sistemas de alta disponibilidad sobre fallas en Sybase: se pueden implementar las características sobre fallas en ASE 12.0 con conectividad de bases de datos del servidor de Jaguar, utilizando JCM (Java Connection Management)

Por otro lado, el balanceo de cargas en un cluster de Jaguar es determinado por tres factores, cada uno de los cuales trata, en resumen, de lo siguiente:

- Medida carga: si se selecciona una política de carga dinámica, la medida de carga determina, la carga en los servidores y abastece a cada servidor de una carga numérica, la cual es utilizada para distribuir peticiones entrantes de clientes y permite el funcionamiento óptimo del cluster. La medida de carga, es una colección de sistemas estáticos que definen una carga en un servidor Jaguar. A cada servidor en un cluster, le es asignado un valor, o carga normal, basada en una medida de carga. Varios factores afectan el funcionamiento y rendimiento de un servidor Jaguar en el sistema. Las medidas de carga que Jaguar utiliza para determinar la carga normal son:

    - Utilización del CPU
    - Tiempo de respuesta del método
    - Conexiones IIOP

Cuando la carga en el cluster es ligera, las peticiones entrantes son eventualmente distribuidas para todos los servidores miembros, esto es, todos los servidores miembro tienen la misma carga. Pero cuando al cluster le llega más carga, ésta, es distribuida de acuerdo con la carga actual de los servidores.

Políticas de distribución de carga: cuando se configura el balanceo de cargas, se selecciona la política de distribución que mejor compagine con el entorno y situación
Referencia de objeto interoperable (IOR): contiene un perfil que el cliente utiliza para bloquear un componente. El perfil contiene el servidor y número de puerto que el cliente utiliza para

acceder el componente. La política de distribución determina el orden en el cual los perfiles son distribuidos hacia los clientes.

	Aplicación	**Si**	**No**	**No aplica**
Secc. V	**Confidencialidad**			
V.1	*Módulo de VPN*			
V.2	*El sistema maneja IPSec*			
V.3	*La transferencia de datos se realiza bajo SSL*			
V.4	*Criptosistemas Simétricos*			
	** En caso de responder NO. pase a la Secc. V.5*			
a	**DES**			
b	**3DES**			
c	**IDEA**			
d	**RC2**			
e	**RC4**			
f	**Otro**			
V.5	*Criptosistemas Asimétricos*			
	** en caso de responder NO. pase a la Secc. VI*			
a	**RSA**			
b	**Otro**			
V.6	*Criptosistemas Híbridos*			
V.7	*Cifrado de Archivos*			
V.8	*Cifrado de directorios*			

Con respecto al manejo de VPN e IPSec, no se pudo investigar al respecto, incluso personal que labora en el área de sistemas en la institución bajo estudio, no pudo ofrecerme información concreta, aunque dada la naturaleza del sistema, es de esperarse que VPN sea también una característica de él; pero en cuanto a la transferencia de datos, si se realiza bajo SSL, como se

especificó en la sección de Autenticación. Además el manejo de criptosistemas también se da, mezclando la velocidad y la eficiencia de uno y otro, dando lugar así a los criptosistemas híbridos. Cabe mencionar de manera certera que el tuneleo de la información se lleva al cabo mediante protocolos como el IIOP y HTTP, ambos seguros, es decir, IIOPS y HTTPS, los cuales permiten generar un túnel, bajo el cual fluye la información encriptada, con SSL. En la Fig. 5.4.2 se intenta ejemplificar estas palabras.

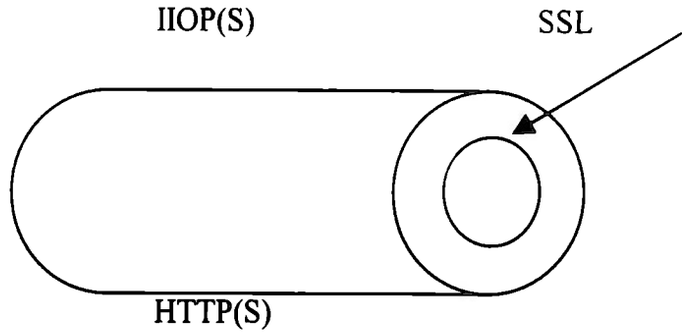


Fig. 5.4.2 Transmisión Segura de Información

Para los dos últimos aspectos que son, el cifrado de directorios, archivos y disco duro, no pudimos comprobarlo, pero dada la naturaleza del servidor de aplicaciones, es de esperarse que permita hacerlo, ya que hoy en día, incluso Antivirus comunes permiten realizar un cifrado de directorios, incluso de un disco duro completo, como lo hace Panda Antivirus.

	Aplicación	**Si**	**No**	**No aplica**
Secc. VI	**No repudiación**			
VI.1	*Autorización del servicio*			
VI.2	*Envío de servicio proporcionado*			
VI.3	*Origen del servicio*			
VI.4	*Recepción del servicio*			
VI.5	*Conocimiento del servicio o contenido del mensaje*			
VI.6	*Firmas Digitales*			

La no repudiación es un factor considerado en las firmas digitales que pueden establecerse entre las aplicaciones que se ejecutan sobre Jaguar, incluso mediante la asignación de atributos a usuarios, campo de acción de los mismos, de los objetos, de componentes, paquetes, etc., permiten que la no repudiación se lleve al cabo. Cabe señalar que el hecho de tener conectividad con componentes CORBA, estos les proveen a las aplicaciones un grado más alto de no repudiación.

Los servicios de no repudiación generados por CORBA trabajan para generar evidencias de acciones de sujetos. Cuando una disputa aparece, la evidencia de no repudiación puede ser utilizada para ayudar a resolver el asunto.

La protección de no repudiación será efectiva si las transacciones son diseñadas para asegurar lo siguiente:

- Cada parte tiene evidencia para defenderse en toda acusación falsa que quizá se haga en su contra.
- Cada parte tiene evidencia para argumentar todas las acusaciones verdaderas que quizá quiera generar.
- Ninguna parte tiene evidencia para argumentar cualquier acusación falsa que tal vez desee realizar.

Cuando surge un desacuerdo y existen evidencia disponibles, la evidencia será evaluada por quienquiera que sea responsable de la decisión. Normalmente será una imparcial tercera parte, un mediador o juez.

La tercer parte de confianza, quien evalúa las evidencias de no repudiación es llamado *arbitro*. Los árbitros tienen reglas para evaluar evidencias. Usando las reglas de evidencias como guía, el árbitro requiere que las partes en desacuerdo le envíen varios tipos de evidencias. Después de enviar las evidencias, el árbitro decide si es confiable o no.

Cantidad de cosas pueden ser enviadas a un árbitro como evidencia. Pero no todo lo enviado es útil. Algunas cosas quizá, incluso, sean rechazadas porque las reglas de evidencia no permitan que sean utilizadas. Cuando un árbitro necesita establecer un desacuerdo acerca de transacciones comerciales, pocos tipos de evidencias son especialmente útiles, como por ejemplo:

- Evidencias de origen: establece que un sujeto originó el texto de un particular mensaje (por ejemplo, un contrato, una autorización de pago, un pagaré, etc.)
- Evidencias de envío: establece que n sujeto propuso un mensaje y o envío
- Evidencias de recepción: establece que un sujeto recibió un mensaje en particular.

Cada tipo de estas evidencias necesitan identificar que sujeto o sujetos estuvieron involucrados en la acción.

La figura 5.4.3 muestra como la no repudiación no puede proporcionarse como un servicio ORB transparente; la seguridad de CORBA proporciona un servicio de aplicación. Las aplicaciones tienen que llamar a un servicio de no repudiación cuantas veces deseen generar evidencias.

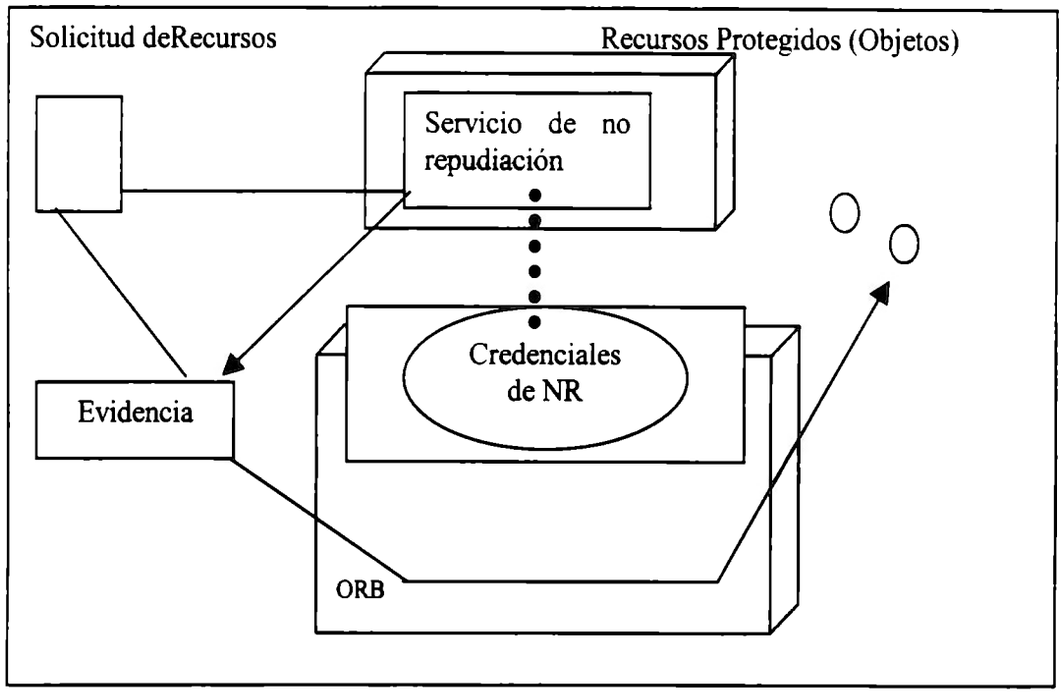


Fig. 5.4.3 Servicio de No Repudiación

Después de la información anterior podemos decir que la seguridad en cuanto a no repudiación por parte de las aplicaciones de Jaguar, es basta.

	Aplicación	**Si**	**No**	**No aplica**
Secc. VIII	**Parches**			
VIII.1	*Último parche*			
VIII.2	*Última versión*			

En este sentido, la organización bajo estudio ha mantenido actualizado el Jaguar CTS, desde la versión 3.0 que permitía interactuar con Java 1.8, hasta la versión actual 3.6.1 que interactúa con J2EE, al igual que la versión 3.6, por tanto también en este punto aprueba la verificación de seguridad.

	Aplicación	**Si**	**No**	**No aplica**
Secc. X	**Certificación**			
	¿La aplicación esta certificada?			
	*En caso de responder "NO", pase a la Secc. XI			

X.1	*TCSEC*		
X.2	*ITSEC*		
X.3	*ICSA*		
X.4	*Otro*		

Éste, fue otro de los factores que intentamos saber enviando un correo electrónico a Sybase, valiéndonos de las personas que laboran para la institución y que tenían el contacto con dicha empresa, dado que dentro de sus actividades esta el realizar pruebas de funcionamiento de Jaguar, pero también es preciso mencionar que después de alrededor de 6 meses, no obtuvimos una respuesta favorable, tal vez justificando un poco, la política de Sybase para responder a preguntas de sus clientes, es FIFO, es decir, los primeros cuestionamientos en llegar son los primeros en recibir respuesta, tal vez nuestra petición aún se encuentra en cola de espera, lo cual suena un poco exagerado. Evidente es, que no tenemos una base para marcar algunas de las opciones que hemos establecido para este aspecto, en nuestra lista de chequeo. Siguiendo un esquema realista, en que no existe ninguna evidencia, debe considerarse que Jaguar no ha seguido ningún tipo de certificación.

	Aplicación	**Si**	**No**	**No aplica**
Secc. XI	**Resistencia a ataques**			
XI.1	*Escucha*			
XI.2	*Negación de Servicio (DoS)*			
XI.3	*Software de IDS*			
a	**Firewalls**			
b	**Otros filtros de direcciones IP**			
c	**Bloqueo Automatizado**			
d	**Estándar RFC2267 (limitadores de flujo de datos)**			
e	**Uso de iTrace**			
XI.4	*Activos*			
XI.5	*Stack Overflow*			
XI.6	*Buffer Overflow*			

Al respecto, Jaguar no maneja estas características de seguridad como tal, es decir, de manera independiente, sólo que el resto de aspectos seguros que ha mostrado, no permiten, que ataques de este estilo ocurran, incluso la parte de Firewalls, funcionan o están implementados de manera independiente a la aplicación, sobre la plataforma, lo que de alguna manera asegura que ataques de "escucha" o de intrusión no alcancen sus objetivos. Con respeto a negación de servicios, en la sección que cubre balanceo de cargas, disponibilidad, etc., se comentaba acerca del manejo de cluster, lo cual permitía que siempre la información estuviese disponible en el momento que es requerida, por tal motivo, de este detalle nos hemos validado para certificar que en cuanto a negación de servicios, es ampliamente robusto, dado que prácticamente esto no ocurre, dada la distribución que de peticiones de clientes se hace, entre dichos clústeres.

Evidentemente por ser una institución financiera, la resistencia a ataques físicos es robusta, los equipos clave que manejan información confidencial, que soportan las conexiones externas a la organización, todo esta perfectamente bajo resguardo, incluso sabemos que desde el ingreso a las instalaciones comunes o generales, el acceso es restringido, habiendo siempre un responsable por cada persona ajena al personal que labora para esta institución.

Por tanto, además de no tener conocimiento real a tal nivel de detalle de seguridad en cuanto a resistencia a ataques, hemos determinado asigna una calificación alta, debido a la protección que al respecto, el entorno le provee.

	Aplicación	**Si**	**No**	**No aplica**
Secc. XII	**Reúso de Componentes**			
XII.1	*Protección de la información almacenada*			
XII.2	*Caché*			
XII.3	*Registros*			
XII.4	/temp			

En este sentido, también Jaguar es considerado robusto, dado que presenta mecanismos que procuran la seguridad de los datos, tanto de los que están almacenados, como de los que viajan a través de algún medio. Además para proteger los datos de Internet, Jaguar proporciona características de SSL para autenticar clientes y servidores y encriptar datos.

Establece seguridad en el sitio del cliente mediante el uso del módulo PKCS #11 proporcionado ya sea por Sybase o por Netscape. Jaguar proporciona pruebas o muestras de certificados y pruebas de applet de seguridad que permiten evaluar las conexiones. Es posible establecer y administrar passwords y generar reglas que permitan limitar el acceso a paquetes y componentes.

Por lo anterior, además de la gran capacidad que tiene para manejo de información en la caché, hemos determinado asignar buen puntaje a este rubro, no sin antes mencionar algunos huecos de seguridad que existen cuando se habla de archivos HTML, por tanto, las aplicaciones pueden

tener un "hoyo de seguridad" potencial, si las clases implementadas en los componentes de Java son desarrollados bajo el directorio de HTML que tiene Jaguar. ya que un usuario no autorizado puede implantar un programa que se conecte a Jaguar por el puerto http, bajando los componentes implementados y las clases. El usuario puede decompilar las clases y ganar acceso a información sensitiva como passwords de una base de datos. obviamente Sybase a detectado tal desventaja y ha propuesto lo siguiente:

- Destacar la implementación de componentes Java bajo el subdirectorio /java/classes de Jaguar.
- Codificar componentes que recuperen conexiones caches para utilizar el API getCacheByName, en lugar del API que requiere password de base de datos.
- Implementar los componentes de Java para recuperar potencial y sensitiva información de propiedades de archivo que no es localizado bajo el subdirectorio HTML de Jaguar.

Hablando un poco más sobre las secciones de clasificación de la información y seguridad de la misma, y que incluso repercuten en esta sección; Jaguar proporciona interfaces de compartición de datos en cada componente modelo. Esas interfaces permiten que los componentes almacenar referencias para compartir datos y bloquear o desbloquear una pieza específica de datos; por ejemplo, si todas las instancias de un componente escriben en el mismo archivo, se puede almacenar un puntero de archivo como un objeto compartido, entonces bloquea el archivo antes de escribir y lo desbloquea cuando ha terminado.

La interfaz de datos compartidos tiene las siguientes restricciones:

- El almacenamiento de datos no es constante: cualquier dato almacenado como un objeto compartido de Jaguar es perdido cuando el servidor es restaurado
- Componentes C y ActiveX pueden solo compartir datos con otros componentes C y ActiveX, respectivamente.
- Componentes Java pueden solo compartir datos entre instancias del mismo componente.

Ahora bien, para vencer esas limitaciones, Sybase recomienda implementar un componente compartido que sea llamado por otros componentes que requieren compartir datos. Cada componente puede ser implementado para almacenar datos compartidos en instancias variables o en una base de datos remota. Para asegurarse que sólo una instancia o componente ha sido generado, configurar la opción de *compartir* en la etiqueta *instancias* de la ventana de propiedades de componente.

Múltiples instancias de un componente pueden leer y actualizar el mismo dato, esto es, un dato compartido. Sólo instancias del mismo componente pueden compartir datos. Datos compartidos entre componentes son contenidos en un objeto compartido. Cada pieza de datos es llamada una propiedad y es identificada por un número de índice. Un objeto compartido puede contener cualquier número de propiedades compartidas. Un número de índice es un entero arbitrario que se le asigna a la propiedad. Cada componente puede tener cualquier número de propiedades. Un valor de propiedad no perdura después de que el servidor fue dado de baja.

Es importante mantener la integridad de las propiedades de los datos compartidos, una simple lectura u operación de actualización en una propiedad es *atómica*. Atómico significa que una

operación sobre datos será completada antes de que cualquier otra operación pueda acceder los datos. Múltiples lecturas y actualizaciones sobre propiedades simples pueden ser sincronizadas mediante un bloqueo de propiedades.

	Aplicación	**Si**	**No**	**No aplica**
Secc. XIII	**Auditoria**			
XIII.1	*Monitoreo*			
a	**Procesador**			
b	**Memoria**			
c	**Caché**			
d	**Procesos**			
e	**Servicios en ejecución**			
f	**Gráficas o diagramas**			
g	**Alarmas**			
h	**Reportes**			
i	**Acceso a archivos y subdirectorios**			
j	**Acceso remoto**			
XIII.2	*Generación de Logs*			
XIII.3	*Auditoria de Logs*			
XIII.4	*Administración de Logs (reconstrucción)*			

En la parte de auditoria que es el tema de esta sección, lo siguiente es una muestra de las capacidades de Jaguar al respecto:

El visualizador y el monitor en tiempo de ejecución, los cuales permiten seguir la pista del funcionamiento y estadísticas del servidor Jaguar.

El Visualizador permite que el monitor:

- REQUESTLOG.- Jaguar mantiene dos archivos que permiten monitorear eventos http.

Información acerca de las peticiones de acceso es registrada en *httprequest.log*. las estadísticas http son registradas en el archivo *httpstat.dat*. Ambos archivos están ubicados por *default* en el subdirectorio *bin* de Jaguar.

Las peticiones de acceso registran información acerca de cada petición http. Si se definen servidores adicionales, el nombre del archivo *httprequest.log* es anexado al nombre del servidor.

Las estadísticas de acceso registran el número total de hits en el servidor y el número total de hits por página.

- SRVLOG. El archivo *srv.log* sigue la pista de evento sen el servidor y cualquier evento de configuración.

Proporciona información acerca de actividades que llevan hacia fuera de la aplicación. Seguir la huella de la salida es enviar al archivo de accesos de Jaguar. La tabla siguiente describe los registros y propiedades de las pistas.

Property	**Description**
Log File Name	The name of the Jaguar log file. This file defaults to *srv.log* in the Jaguar *bin* subdirectory. The *srv.log* logs a wide range of information and is helpful in isolating problems. You can create the log file in an alternate directory by prefixing a full path to the file name you enter. If you do not enter a full path, the file is created in the Jaguar *bin* subdirectory. You cannot use environment variables when specifying a full path.
Log File Size (Bytes)	The size, in bytes, to which the log file grows before it is truncated.
Truncate Log on Startup	When this flag is set, the log truncates every time the server is restarted. Keep in mind that if the server crashes and this flag is set, you will lose the log file and the information it contains.
Trace Attentions	If set, traces attentions received or acknowledged by Jaguar.
Trace Network Driver APIs	If set, traces Net-Lib driver requests.
Trace Network Driver Requests	If set, traces network layer protocol requests.
Trace Protocol Data	If set, traces TDS packet content (the actual TDS traffic between a client and Jaguar) in hexadecimal and ASCII format.
Protocol Headers	If set, traces TDS protocol packet header information, such as packet type and length.

Trace Servlets	If set, traces the execution of Jaguar's servlet execution engine.

Tabla T5.4.1 Auditoria de Logs [23]

- ERRORLOG: el archivo *httperror.log* sigue la pista de los errores en HTTP, tal como una petición para un archivo HTML que no existe. Si se definen servidores adicionales, el nombre del archivo de registro es anexado al nombre del servidor.

El monitor en tiempo de ejecución permite monitorear eventos y estadísticas del servidor, los cuales pueden ayudar a anticipar y prevenir problemas en el servidor.

Para iniciar el monitor:

- Doble clic en el icono de *servidores*
- Doble clic en el servidor que se desee monitorear

El monitor puede conectar a otros servidores Jaguar vía receptor IIOP. La configuración del servidor identifica el Host y el número de puerto, al cual el monitor intenta conectarse.

- Clic en el icono de monitor en tiempo de ejecución.

El monitor despliega las carpetas listadas abajo. Para cada uno de los grupos de alto nivel, existen subgrupos de carpetas. Clic en la carpeta del subgrupo cuya estadística se quiere ver.

- Paquetes: monitoreo de eventos y estadísticas para un paquete específico o para todos los paquetes en el servidor
- Conexiones Caché: monitorea una conexión caché específica o estadísticas para todas las caché.
- Red: monitoreo de protocolos específicos de sesiones de información

La información anterior nos ha demostrado que en cuanto auditoria, las aplicaciones sobre Jaguar que cumplen con la mayoría de los aspectos importantes de un monitoreo de procesos, memoria, registros, etc.

Secc. XIV	**Rutas seguras**			
XIV.1	*Conexión directa entre S.O y aplicación*			
XIV.2	*Negación de Servicio (DoS)*			

- No hubo evidencia de existiese conexión directa entre aplicación y aplicación, aún cuando la documentación de Jaguar insinua que asi sucede.

Secc. XV	**Arranque Seguro**			
Secc. XIX	**Administrador Dedicado**			

Cuando Jaguar se ejecuta no existe ningún proceso en ejecución además de él, lo cual de alguna manera nos permite realizar una conexión directa entre sistema y aplicación, y a su vez entre Jaguar y las aplicaciones que "corren" sobre él. Aunque de manera certera no tengo fundamentos para determina el grado de seguridad de las rutas entre sistema operativo y aplicación, y entre aplicaciones, además el proveedor no dio más información al respecto.

En cuanto a servidor dedicado, si podemos decir que el servidor de aplicaciones Jaguar si corre sobre un servidor dedicado, logrando con esto una asignación total del tiempo de procesamiento.

Secc. XX	**Evaluación del producto por parte de la organización**			

Sí se realizó una evaluación previa sobre las características de Jaguar, en comparación con otros servidores de aplicaciones existentes en el mercado. Aunque a juicio personal considero que fue un tanto escueta y tendenciosa, dado que algunos factores no económicos por los que se tomó la decisión de adquirir el producto, hoy en día ya no son tomados en cuenta en la implementación de aplicaciones y componentes que se ejecutan sobre Jaguar. Aún así cumplieron con este punto.

Secc. XXI	**Política de recuperación de desastres**			
Secc. XXII	**Plan de contingencia del negocio**			

No tuvimos acceso a las políticas de recuperación de desastres ni de los planes de contingencia, pero sabemos que existen, por tanto, y dada la naturaleza del negocio, creemos que en este aspecto también acredita Jaguar y la institución financiera objeto de nuestro estudio..

5.5 EVALUACIÓN DE CÓDIGO DESARROLLADO PREVIAMENTE

```
/*

  vacation-Logon.java


  NOTE: This file is a generated file.
```

```
    Do not modify it by hand!
*/

package vacation;

// custom imports for Logon
// add your custom import statements here
```

Interfaz pública de Servelet. Define métodos que todos los servlets pueden implementar. Un servlet es un pequeño programa de java que corre dentro de un servidor Web. Los servlets reciben y responden a peticiones de clientes Web, usualmente a través de HTTP, el protocolo de trasferencia de Hipertexto. Para implementar esta interfaz, se pude escribir un servlet genérico que extiende javax.servlet.GenericServlet o un servlet http que extiende a javax.servlet.http.HttpServlet, implementando otras clases.

```
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class Logon extends javax.servlet.http.HttpServlet
{
  protected boolean create() throws java.lang.Exception
  {
    return true;
  }
  public Logon()
  {  // Constructor.
```

```
}

private void unhandledEvent( String listenerName, String methodName, java.lang.Object event )

{

}

/**

 * destroy Method

 */

public void destroy()

{

   super.destroy();

   // TODO: implement

}

/**

 * doGet Method

 */

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException

{

   response.setContentType("text/html");

            PrintWriter out = response.getWriter();

            out.println("<HTML>");

            out.println("<HEAD>");

            out.println("<TITLE>Logon Servlet</TITLE>");

            out.println("</HEAD>");

            out.println("<BODY>");

            String user = request.getParameter("user");
```

```
                String password = request.getParameter("password");
                HttpSession session = request.getSession(true);
                if (user.equals(password)) {            //Logon successful
                        session.setAttribute("user", user);
                        response.sendRedirect("/Vacation/ListDest.jsp");
                }else{          //Logon failed
                        session.removeAttribute("user");
                        response.sendRedirect("LogonError.htm");
                }
                out.println("</BODY>");
                out.println("</HTML>");
                out.close();
    }
    /**
     * doPost Method
     */
protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException
{
doGet( request, response );
// TODO: implement
}
/**
     * init Method
     */
    public void init(ServletConfig config) throws ServletException
    {
```

```
        super.init(config);
        // TODO: implement
    }
    /**************************************
     * data members
     **************************************/
    // add your data members here
}
```

Aquí mostramos un pequeño código desarrollado en Java, pero cabe mencionar que en estos dos ejemplos que presentaremos, y en cualquier otro, sería difícil visualizar la utilización de componentes seguros, dado que cualquier código o componente, con el simple hecho de correr sobre Jaguar, le son heredadas todas las características de conectividad, de transacción y de seguridad del mismo Jaguar.

5.6 EVALUACIÓN DE CÓDIGO UTILIZANDO COMPONENTES EVALUADOS

```
/*
   Sie_EJB-Dircuasie_EJBBean.java
   NOTE: This file is a generated file.
       Do not modify it by hand!
*/
package Sie_EJB;
// custom imports for Dircuasie_EJBBean
// add your custom import statements here
import java.io.*;
import java.util.Date;
```

```
public class Dircuasie_EJBBean extends java.lang.Object implements
javax.ejb.SessionBean
{
   protected boolean create() throws java.lang.Exception
   {
      // Connection source: Jaguar cache
      transaction_sie.setTraceToLog( false );
      transaction_sie.setRegisterName( true );
      transaction_sie.setName( "Dircuasie_EJBBean.transaction_sie" );
      transaction_sie.setConnectionSource( new powersoft.powerj.db.JaguarConnectionSource(
"Sie_jdbc", com.sybase.jaguar.jcm.JCMCache.JCM_FORCE, false ) );
      transaction_sie.setUseInitialSettings( true );
      transaction_sie.setLoginTimeout( 0 );
      transaction_sie.setOwner( this );
      transaction_sie.setRestoreInitialSettings( false );
      query_sie.setTraceToLog( false );
      query_sie.setName( "Dircuasie_EJBBean.query_sie" );
      query_sie.setTransactionObject( transaction_sie );
      query_sie.setOwner( this );
      query_todos.setTraceToLog( false );
      query_todos.setName( "Dircuasie_EJBBean.query_todos" );
      query_todos.setTransactionObject( transaction_sie );
      query_todos.setOwner( this );
.
.
.
```

```
public Dircuasie_EJBBean()
{  // EJB constructors don't have a server context.
}
private void unhandledEvent( String listenerName, String methodName, java.lang.Object event )
{
}
// method for interface javax.ejb.SessionBean
public void ejbActivate() throws javax.ejb.EJBException, java.rmi.RemoteException
{
  // To Do
}
// method for interface javax.ejb.SessionBean
public void ejbPassivate() throws javax.ejb.EJBException, java.rmi.RemoteException
{
  // To Do
}
// method for interface javax.ejb.SessionBean
public void ejbRemove() throws javax.ejb.EJBException, java.rmi.RemoteException
{
  // To Do
}
// method for interface javax.ejb.SessionBean
public void setSessionContext( javax.ejb.SessionContext parm0 ) throws javax.ejb.EJBException, java.rmi.RemoteException
{
  this._sessionContext = parm0;          // generated helper code
```

```
    // To Do
}
private int CreaDirectorio()
{
    try{
                fileOut = new BufferedWriter(new FileWriter(nombreArchivo));
        }
        catch(IOException e){
                System.out.println("Error al crear archivo de salida: " + nombreArchivo);
        }
    return 0;
}
private int doCompute(int entrada)
{
        query_sie.setSQL(select);
        if(!query_sie.open())
        {
                System.err.println("Error en el query");
                return -1;
        }
        ex_fetch_data(entrada);
        return 0;
}
private void ejecuta_todos()
{
    String sql;
```

```
            //sql = "select numsec, sector from RegeneraSector";
            sql = "SELECT idxsec, sectorvis, numsec, sector, regenerar FROM
RegeneraSector WHERE numcuadros<>0 order by idxsec";
            query_todos.setSQL(sql);
            if(!query_todos.open())
            {
                System.err.println("Error en el query: " + sql);
                return;
            }
            try
            {
                while (query_todos.next())
                {
                    setTitsec(query_todos.getStringValue(4));
                    numsec = Integer.parseInt(query_todos.getStringValue(3));
                    setNumsec(numsec);
                    /* Se da nombre al Archvo de Salida */
                    setNombreArchivo("");
                    /* Se crea el archivo de salida*/
                    CreaDirectorio();
                    /* Se escribe el encabezado del archivo      */
                    encabezado();
                    encabezado2();
                    try{
                        //System.out.println("Se cierra archivo");
/**
 * Regenera
```

```
 */
private int regenera;
public int getRegenera() {
   return regenera;
}
public void setRegenera(int newValue) {
   regenera = newValue;
}
/**
 * Select
 */
private String select = "";
public String getSelect() {
   return select;
}
public void setSelect(String newValue) {
   select = newValue;
}
/**
 * Titsec
 */
private String titsec = "";
public String getTitsec() {
   return titsec;
}
```

```
public void setTitsec(String newValue) {
    titsec = newValue;
}
/**
 * Tsec
 */
private String tsec = "Externo";
public String getTsec() {
    return tsec;
}
public void setTsec(String newValue) {
    tsec = newValue;
}
/*****************************************
 * data members
 *****************************************/
protected powersoft.powerj.db.java_sql.Transaction transaction_sie = new powersoft.powerj.db.java_sql.Transaction();
protected powersoft.powerj.db.java_sql.Query query_sie = new powersoft.powerj.db.java_sql.Query();
protected powersoft.powerj.db.java_sql.Query query_todos = new powersoft.powerj.db.java_sql.Query();
// add your data members here
```

Los fragmentos de código anteriores con dificultad nos permiten ver la incursión de las propiedades seguras de Jaguar, dado que estos fueron generados de manera automática por el mismo servidor de aplicaciones, aún así se logra percibir algunas instrucciones de autenticación y de certificación en la parte final, así como de manejo de la caché.

5.7 REPORTE DE RESULTADOS Y PROPUESTAS DE MEJORA

Lista de Chequeo				
	Aplicación	**Si**	**No**	**No aplica**
Secc. I	**Autenticación**			
I.1	*Administración de Password*	√		
a	**Usuarios**	√		
1	Generación	√		
2	Modificación	√		
3	Eliminación	√		
4	Características de Password	√		
4.1	***longitud***	√		
4.2	***formación***	√		
4.3	***tiempo de vida***	√		
b	**Mapeo a Password de sistema**	√		
I.2	*Administración de usuarios*	√		
a	**Lista**	√		
b	**Agregar**	√		
c	**Funciones de búsqueda**		X	
d	**Cambio de características**	√		
e	**Eliminar**	√		
I.3	*Administración de Certificados*	√		
a	**Autoridad Certificadora Externa**	√		

	Aplicación	Si	No	No aplica
b	**Certificación Interna**	√		
1	Generación	√		
2	Renovación	√		
3	Eliminación	√		
4	Distribución	√		
5	CRL (Lista de certificados revocados)	√		
5.1	***Generación***	√		
5.2	***Distribución***	√		
5.3	***Mantenimiento***	√		
c	**Certificados entre atributos de usuario (Extensiones)**	√		
I.4	*Administración de llaves*	√		
a	**Generación**	√		
b	**Distribución**	√		
c	**Repositorio**	√		
I.5	*Dispositivos Biométricos*		X	
a	**Local**		X	
b	**Remoto**		X	
I.6	*Tarjetas Inteligentes*		X	
I.7	*Criptosistemas*	√		
a	**Local**	√		
b	**Red**	√		
Secc. II	**Control de Acceso**	√		
II.1	*Listas de control de acceso*	√		

II.2	*Listas de Capacidad*	√		
II.3	*Discrecionario*	√		
II.4	*Mandatario*	√		
II.5	*Basado en mínimos privilegios*		x	
II.6	*Confirmación periódica de los derechos de acceso*	√		
II.7	*Remoto*	√		
Secc. III	**Integridad**			
	¿Utiliza método para revisión de integridad?	√		
	** En caso de responder NO. pase a la Secc. IV*			
III.1	*CheckSum*	x		
III.2	*MD2*	x		
III.3	*MD4*	x		
III.4	*MD5*		√	
III.5	*SHA-1*			
III.6	*Otro*			
Secc. IV	**Disponibilidad**			
IV.1	*Información disponible cuando es requerida*	√		
IV.2	*Cluster*	√		
IV.3	*Manejo de carga*	√		
Secc. V	**Confidencialidad**			
V.1	*Módulo de VPN*	x		
V.2	*El sistema maneja IPSec*	x		

	Aplicación	Si	No	No aplica
V.3	*La transferencia de datos se realiza bajo SSL*		√	
V.4	*Criptosistemas Simétricos*		√	
	* *En caso de responder NO, pase a la Secc. V.5*			
a	**DES**	√		
b	**3DES**	√		
c	**IDEA**		X	
d	**RC2**		X	
e	**RC4**		X	
f	**Otro**		X	
V.5	*Criptosistemas Asimétricos*	√		
	* *en caso de responder NO, pase a la Secc. VI*			
a	**RSA**	√		
b	**Otro**		x	
V.6	*Criptosistemas Híbridos*	√		
V.7	*Cifrado de Archivos*	√		
V.8	*Cifrado de directorios*	√		
Secc. VI	**No repudiación**	√		
VI.1	*Autorización del servicio*	√		
VI.2	*Envío de servicio proporcionado*	√		
VI.3	*Origen del servicio*	√		
VI.4	*Recepción del servicio*	√		

VI.5	*Conocimiento del servicio o contenido del mensaje*	√		
VI.6	*Firmas Digitales*	√		
Secc. VII	**Tolerancia a fallas**	√		
VII.1	*Cluster*	√		
VII.2	*Manejo de carga*	√		
Secc. VIII	**Parches**	√		
VIII.1	*Último parche*	√		
VIII.2	*Última versión*	√		
Secc. IX	**Análisis de Contenido**		**X**	
Secc. X	**Certificación**			
	¿La aplicación esta certificada?			
	*En caso de responder "NO", pase a la Secc. XI			
X.1	*TCSEC*			
X.2	*ITSEC*			
X.3	*ICSA*			
X.4	*Otro*			
Secc. XI	**Resistencia a ataques**	√		
XI.1	*Escucha*	√		
XI.2	*Negación de Servicio (DoS)*	√		
XI.3	*Software de IDS*	√		

	Aplicación	Si	No	No aplica
a	**Firewalls**			
b	**Otros filtros de direcciones IP**			
c	**Bloqueo Automatizado**			
d	**Estándar RFC2267 (limitadores de flujo de datos)**			
e	**Uso de iTrace**			
XI.4	*Activos*	√		
XI.5	*Stack Overflow*	√		
XI.6	*Buffer Overflow*	√		
Secc. XII	**Reúso de Componentes**			
XII.1	*Protección de la información almacenada*	√		
XII.2	*Caché*	√		
XII.3	*Registros*	√		
XII.4	/temp	√		
Secc. XIII	**Auditoria**			
XIII.1	*Monitoreo*			
a	**Procesador**			
b	**Memoria**	√		
c	**Caché**	√		
d	**Procesos**	√		
e	**Servicios en ejecución**	√		
f	**Gráficas o diagramas**			

g	**Alarmas**			
h	**Reportes**	√		
i	**Acceso a archivos y subdirectorios**	√		
j	**Acceso remoto**	√		
XIII.2	*Generación de Logs*	√		
XIII.3	*Auditoria de Logs*	√		
XIII.4	*Administración de Logs (reconstrucción)*	√		
Secc. XIV	**Rutas seguras**			
XIV.1	*Conexión directa entre S.O y aplicación*			
XIV.2	*Negación de Servicio (DoS)*			
Secc. XV	**Arranque Seguro**	√		
Secc. XVI	**Respaldo**			
XVI.1	*Cintas*			
XVI.2	*Red*	√		
XVI.3	*CD's*	√		
XVI.4	Otra			
Secc. XVII	**Compartición de recursos**	√		
XVII.1	*Grupos*	√		
XVII.2	*Usuarios*	√		

XVII.3	*Con privilegios*	√		
XVII.4	*Sin privilegios*	√		
XVII.5	*Públicos*	√		
Secc. XVIII	**Clasificación de la información**	√		
XVIII.1	*Por aplicación*	√		
XVIII.2	*Por los usuarios que la manejan*	√		
XVIII.3	*Por su sensibilidad*	√		
Secc. XIX	**Administrador Dedicado**	√		
Secc. XX	**Evaluación del producto por parte de la organización**	√		
Secc. XXI	**Política de recuperación de desastres**	√		
Secc. XXII	**Plan de contingencia del negocio**	√		
	Total a obtener: 102 Obtenidos: 84			
	Calificación obtenida: 84%			

En toda organización del estilo o giro que esta sea, siempre existen cosas a mejorar o detalles a superar. En la institución motivo de nuestro estudio, fueron evidentes durante este período de análisis, diversas limitantes tanto en software como en logística, es decir, Jaguar por si solo, tiene deficiencias, así como las aplicaciones que son implementadas.

Podríamos redactar todo un tratado acerca de cada una de las características a mejorar, y evidentemente sería de gran utilidad y objetividad, pero lo que realmente resulta importante, es darnos cuenta que en todas las organizaciones, hace falta cultura al respecto. Muchas de las personas que laboramos para una institución donde se maneja equipo de cómputo no tenemos cultura de seguridad informática, pero no solo eso, sino que desconocemos la mayoría de los aspectos que debemos proteger, y debido a esto, como podemos participar en proteger los activos físicos y lógicos si ni siquiera sabemos de ello.

En resumen, la más clara e importante propuesta de mejora es el conocimiento del personal que labora en la institución en conceptos informáticos, y de esto dependerá la mejora palpable en cuestión de seguridad informática, porque repito, hasta entonces, podremos opinar, construir, diseñar componentes, códigos y/o aplicaciones con grandes niveles de seguridad, derivadas de ese conocimiento al respecto

Hablando de la calificación obtenida para Jaguar CTS, esta resulta un tanto subjetiva, dadas las limitantes del análisis, porque en ocasiones faltaba recurso informático para alcanzar certeza en las pruebas, y en otras conocimiento al respecto, con lo cual podemos obtener un resultado que podría tener una desviación estándar fácilmente de 15 puntos porcentuales. Pero independientemente de esto, la calificación obtenida por Jaguar puede considerarse alta.

El presente trabajo podría tener realmente utilidad, así como permitirnos ver los trabajos futuros y propuestas de mejora, en la medida que la propuesta de lista de chequeo se aplica a varias aplicaciones, pues en esa medida podremos verificar de manera real la verdadera ayuda que puede proporcionarnos, y cuyo aspecto es precisamente el objetivo de éste trabajo.

Redundando, debemos conocer qué proteger, para saber de qué lo protegemos. Debemos codificar de manera muy diferente a la tradicional, debemos pensar de manera segura, esto implica que todo es vulnerable a menos que se demuestre lo contrario, con un nuevo modelo de programación segura, podríamos estar abatiendo uno de los principales centros de generación de vulnerabilidades. Para no solo es la gente que codifica, sino la parte de diseño, pues en las metodologías de diseño de estrategias de defensa ante los ataques habidos y por haber, podremos amainar también en buen grado de vulnerabilidades y posibilidades al exterior para causar daño en nuestros sistemas y equipos de cómputo. Generalizando, esto es cultura de seguridad computacional, la cual considero en esta institución es por arriba del promedio, pero no suficiente para poder considerar a la institución y a todos los sistemas seguros.

Por último, en la medida que se vaya aplicando ésta técnica de verificación de la seguridad en aplicaciones, podrían empezarse a definir niveles mínimos de seguridad, éstos, nos permitirían saber, que al alcanzarlos o rebasar los requerimientos establecidos por estos niveles mínimos, una aplicación podría ser liberada o considerada segura. Siguiendo los pasos de los criterios de certificación analizados en el capítulo 2 de este trabajo, podríamos incluso designar una escala de cumplimiento, basada principalmente en confidencialidad, integridad y autenticidad, pero dependiendo de la aplicación y de la institución, podría ser que las características seguras de su información fuesen sensibles en diferentes áreas, por tanto, en la medida que se apliquen, esos niveles de seguridad podrían establecerse de manera estándar para poder liberar a la mayoría de aplicaciones.

Finalmente, nos gustaría mencionar que evaluar la seguridad de aplicaciones y su entorno resulta una tarea extenuante y aún cuando quisimos intencionalmente realzar los chequeos de forma

superficial, esta labor puede tomar de semanas a meses. Aún así, las instituciones financieras en nuestro país requieren pruebas como la presentada en este capítulo, y sólo concretarse a probar que el programa o aplicación "funcionen".

6 RESULTADOS Y CONCLUSIONES

La evaluación de un sistema sin importar la magnitud y tipo del mismo, siempre involucra además de conocimiento y tecnología, una gran dosis de tiempo y dedicación, por tanto es que consideramos que el presente estudio puede ser de gran interés para diversos sectores donde la seguridad informática en el desarrollo o implantación de aplicaciones tenga injerencia. A diferencia de la evaluación de sistemas operativos, las aplicaciones, versiones de las aplicaciones y actualizaciones de las mismas, surgen aún durante el ciclo de vida de estas aplicaciones; es ésta la principal razón que nos llevó a pensar que, aunque es posible, no sería conveniente desde el punto de vista de costo/beneficio el evaluar las aplicaciones de uso exclusivo de una organización, con base en estándares tan rigurosos como el ITSEC, TCSEC o CC. En lugar de esto último, se prefirió elaborar una lista de chequeo que permitirá conocer al desarrollador el estado de seguridad actual que guardan las aplicaciones, permitiéndole identificar dónde mejorar para obtener una mejor puntuación en su software y, sobre todo, generando paulatinamente una conciencia de seguridad en los futuros desarrollos de la empresa. Al observar la lista generada, podría resultar demasiado exhaustiva para muchos de los casos; sin embargo, ésta puede ser adaptada de acuerdo a las necesidades de seguridad de la organización. Evidentemente, este trabajo de investigación se basa en los criterios de evaluación de productos y sistemas y no pretende sustituirlos.

6.1 APLICABILIDAD

La aplicabilidad de la evaluación de una aplicación para medir el grado de seguridad y certificar el nivel al que se encuentre, es de gran interés dada la escasez de instituciones y/o organismos dedicados a realizar este tipo de trabajos. Existe certificación del nivel de seguridad de sistemas operativos primordialmente, incluso bases de datos de gran proliferación en el mundo de la informática, pero difícilmente con respecto a aplicaciones "desarrolladas en casa". Cabe mencionar que los organismos encargados de determinar mediante criterios el lugar donde se puede ubicar a un sistema dependiendo el grado de seguridad que presente, han establecido criterios y normativas para clasificarlo, pero haciendo énfasis en que sus trabajos se enfocan a las

plataformas y redes de computadoras primordialmente. Por tanto consideramos de gran aplicación el presente trabajo, por todo lo anterior y dado el constante avance la tecnología informática y computacional, lo cual provoca el desarrollo de ataques computacionales más efectivos y llenos de infraestructura, esto como consecuencia genera que las empresas cada vez más estén preocupadas y empeñadas en mantener la seguridad de sus activos, dado que la información que viaja hoy en día a través de las redes públicas y privadas es crecientemente sensible para todas y cada una de esas empresas.

6.2 TIEMPO DE EVALUACIÓN

El tiempo de evaluación podría depender de la aplicación a evaluar y sobre todo de la sensibilidad de su información, es decir, de la repercusión que una pérdida de ésta, ocasione en la empresa, pues en esa medida, los criterios y/o aspectos considerados en la evaluación, podrían perder vigencia o interés, o por otro lado cobrar mayor fuerza en cuanto a importancia en su cumplimiento.

EL tiempo de evaluación podría oscilar de 6 meses a 1 año dependiendo, ya decíamos, de la sensibilidad de la información. Sin embargo, si el chequeo se va sistematizando y realizando periódicamente, por ejemplo, sobre nuevas versiones de aplicaciones previamente evaluadas, pueden mejorarse los tiempos de chequeo.

Cabe mencionar que evaluar una aplicación de esta naturaleza es un trabajo nada sencillo, dado que implica además de tiempo, disponibilidad de equipo, como en logística, conocimiento preciso de las actividades a realizar, ya que cada uno de los criterios propuestos para desarrollar una certificación de seguridad de cualquier sistema, requiere de recursos, lógicos y físicos. Estos aspectos son de suma importancia para poder evaluar de manera objetiva una aplicación, pues se requiere de gran capacidad de cómputo, recursos de software y hardware que permitan verificar de manera verídica cada uno de los aspectos propuestos en la lista de chequeos

6.3 TRABAJOS FUTUROS

Los trabajos futuros que pueden derivarse del actual, también tienen una amplia y favorable perspectiva, porqué ¿a qué desarrollador, pero sobre todo a qué empresa que requiere de seguridad computacional no le agradaría tener el más alto nivel de seguridad en sus sistemas y aplicaciones? El tener un nivel aceptable le proporciona un porcentaje de seguridad de sus activos (de manera momentánea al menos) más alto que el resto de las organizaciones que no lo posean. El constante avance tecnológico en el área como se mencionó anteriormente, provoca que temas de este estilo no pierdan vigencia, por el contrario, día a día el interés crece, porque lo que hoy está bien protegido mañana difícilmente lo estará. Cada día es más difícil mantener la integridad, disponibilidad y confidencialidad de un sistema de computo, por lo que reincidimos en marcar la importancia de esta tesis, así como la continuidad que a ella pueda darse, dado que la certidumbre de las herramientas que se emplean para audita determinado aspecto en ocasiones está supedita o más bien depende del grado de complejidad de la misma herramienta, lo cual también por obvias razones repercute en el costo de ésta, lo cual en ocasiones no pudiese ser alcanzable para los

recursos de una determinada empresa, dejándola de esta manera atenida al reporte de herramientas de evaluación más comerciales, accesibles, pero que no reportan "huecos" de seguridad de gran nivel, exponiendo de esta manera ante gente especializada y con recursos los activos del sistema, es decir, la información sensible de ésta. Dentro de los trabajos futuros podemos prever que será necesario realizar un sinnúmero de evaluaciones para poder definir una escala que indique el grado de confianza y seguridad que podemos depositar en las aplicaciones evaluadas; buscando acercarnos a los esquemas propuestos en criterios tradicionales y más orientados a sistemas operativos, como son los criterios descritos en el capítulo 2.

Realmente hay mucho por hacer, dadas las expectativas que con respecto a seguridad computacional se han generado, pero particularmente, si la verificación a un sistema o producto se realiza de manera correcta, (de manera verídica y auxiliados por recursos; tanto intelectuales, de hardware y software) la calificación obtenida al nivel de seguridad, será lo más apegado a la realidad. Además, en la medida que se incrementen los puntos de la lista de chequeo, ya sea porque algunos hayan sido omitidos en esta primera versión de lista, o por el surgimiento de nuevos aspectos a evaluar, y éstos sean aplicados dependiendo del producto o sistema, existirá siempre un trabajo futuro, el cual permitirá, vislumbrar de una manera más exacta el nivel de seguridad de dicho sistema o producto.

BIBLIOGRAFÍA

[1] RODRÍGUEZ LUIS ÁNGEL
"Seguridad de la Información en Sistemas de Cómputo"
Ediciones Ventura
1995

[2] KARANJIT SIYAN, PH. D.
"Internet y Seguridad en Redes"
Prentice Hall Hispanoamericana.
2000 p. 56-58, 133-135

[3] DEPARTMENT OF DEFENSE TRUSTED COMPUTER SYSTEM EVALUATION CRITERIA (TCSEC)
"Criterios de Evaluación de Sistemas de Cómputo Seguros"
Diciembre de 1985
http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html

[4] VÁZQUEZ GÓMEZ JOSÉ DE JESÚS, DR.,
"Seguridad Computacional", Manual de Curso
Instituto Tecnológico y de Estudios Superiores de Monterrey
1997

[5] ITSEC "White Book" (Information Technology Security Evaluation Criteria)
"Criterios de Evaluación de la seguridad en tecnologías de la información"
Octubre de 2000
http://itsec.gov.uk/

[6] RUSELL DEBORAH &. GANGEMI G.T SR
"Computer Security Basics"
O'REILLY
Julio de 1992

[7] COMMON CRITERIA
"Criterios de Evaluación de Seguridad"
30 de Abril, 2001
http://csrc.nist.gov/cc//

[8] HERZOG PETE
OSSTMM (Open Source Security Testing Methodology Manual)
Mayo 5, 2001
http://www.ideahamster.org/osstmm.htm & http://uk.osstmm.org/osstmm.pdf

[9] Best Practices for Enterprise Security
"Mejores Practicas de Seguridad Computacional en Empresas: Metodologías"
www.microsoft.com/technet/security/bestIprac

[10] ADAPTIVE SYSTEM SURVIVABILITY
"The Immunix research project has ended"
<http://www.cse.ogi.edu/DISC/projects/immunix/>

[11] EXTREME PROGRAMMING PROJECT
"Programación Extrema"
Enero 5, 2002
http://www.extremeprogramming.org/

[12] GRISSONNANCHE, ANDRÉ
"Security and Protection in Information Systems"
North Holland

[13] PFLEEGER CHARLES P.
"Security in Computing"
Prentice May

[14] BIRD, TINA
"Redes Privadas Virtuales"
Agosto de 2000
http://kubarb.phsx.ukans.edu/~tbird/vpn/FAQ.html

[15] ÁLVAREZ MARAÑÓN, GONZALO
"Autorización y Autenticación"
2000
< http://www.iec.csic.es/criptonomicon/autenticacion/control.html>

[16] RODRÍGUEZ BARBOSA, LUIS
2000, "Control de Acceso"
<http://www.iec.csic.es/criptonomicon/articulos/expertos69.html>

[17] COMPUTER SECURITY RESOURCES CENTER
"Security Testing"
http://csrc.nist.gov

[18] SYMANTEC RESEARCH
"Denial of Services Attack (DoS)"
23 de febrero de 2000
< http://www.sarc.com/avcenter/venc/data/dos.attack.html>

[19] CERT ADVISORY
"Reportes de Ataque"
<http://www.kb.cert.org/vuls/id/38950> (2001 - 2002)

[20] SEARCH SECURITY
"Hoax"
2000 - 2002
<http://searchsecurity.techtarget.com/>

[21] RFC FIPS 190
"Guía para el uso de tecnología avanzada de autenticación"
Septiembre 28, 1994
http://www.parallaxresearch.com/dataclips/pub/government/nist_fips/fip190.htm

[22] KURT DILLARD
"What is a Bastion Host?"
Abril 1998

[23] "Documentacion de Jaguar"
2000
http://www.sybase.com/products/applicationservers/easerver/techsupport

[24] "Comparativa de Antivirus"
HispaSec
2001
http://www.hispasec.com/comparativa.asp

ANEXO A TABLA RESUMEN DEL TCSEC

La siguiente tabla muestra los requerimientos de seguridad del TCSEC. Dichos requerimientos están "acomodados" en forma tabular, de tal manera que más claramente se observen los cambios de una clase a otra.

Note lo siguiente:

- "No requerimientos" es una columna que significa que el Orange Book, no define requerimientos para esa característica en esa clase.
- "No requerimientos adicionales" significa que los requerimientos para esta clase son los mismos, que para la clase previa.
- Cada columna muestra sólo nuevos requerimientos para esa clase.[6]

C1	C2	B1	B2	B3	A1
Control de Acceso Discrecionario					
La TCB define y controla el acceso entre nombres de usuarios y nombres de objetos. Los mecanismos de ejecución, permiten a usuarios especificar y controlar los objetos compartidos, por nombres individuales y definir grupos de ambos.	**Requerimientos adicionales**: definir grupos específicos e individuales. Mecanismos de ejecución que proporcionen más control y que limiten la propagación de accesos. Protección a objetos por acceso no autorizado o que haya expirado	Sin requerimientos adicionales	Sin requerimientos adicionales	**Requerimientos adicionales**: Los mecanismos de ejecución son una lista de control de accesos, estos controles son específicos para cada objeto, lista de nombres de objetos y modos de acceso	Sin requerimientos adicionales
Reutilización de Objetos					
No requerimientos	Todas las autorizaciones están contenidas en un objeto compartido, éstas serán canceladas por las asignaciones iniciales, ubicadas o reubicadas a sujetos	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales
Etiquetas					

No requerimientos	Sin requerimientos	Etiquetas de sensibilidad asociadas con cada sujeto y objeto almacenados bajo control. Estas etiquetas son utilizadas como la base de las decisiones del control de acceso mandatorio	Requerimientos adicionales: Etiquetas de sensibilidad asociadas con cada recurso del sistema ADP. Esto es directa o indirectamente accesible por sujetos externos al TCB	Sin requerimientos adicionales	Sin requerimientos adicionales
Integridad de etiquetas					
Sin requerimientos	Sin requerimientos	Las etiquetas de manera precisa representan los niveles de seguridad de un sujeto u objeto con el cual ellos se relacionan	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales
Exportación de información etiquetada					
Sin requerimientos	Sin requerimientos	El TCB designa cada canal de comunicación y dispositivos de I/O como un nivel o multinivel. Cualquier cambio sobre esa designación es realizado manualmente. La TCB se mantiene y es capaz de auditar cualquier cambio en el nivel de seguridad o niveles asociados con un canal de comunicación o dispositivo de I/O	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales
Exportación a Dispositivos Multinivel					
Sin requerimientos	Sin requerimientos	Cuando el TCB exporta un objeto a un dispositivo de I/O multinivel, la etiqueta de sensibilidad	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales

		asociada al objeto, también es exportada y reside en el mismo medio físico, como y de la misma forma que la información...			
Exportación a Dispositivos de un solo nivel					
Sin requerimientos	Sin requerimientos	Dispositivos de I/O y canales de comunicación de un nivel, no son requeridos para mantener las etiquetas de sensibilidad de la información de su proceso. Sin embargo, el TCB incluye un mecanismo mediante el cual, él y un usuario autorizado designan el nivel de seguridad de la información importada o exportada vía canales simples de comunicación o dispositivos de I/O.	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales
Salida Etiquetada Humanamente Legible					
Sin requerimientos	Sin requerimientos	El sistema administrador ADP, debe ser capaz de publicar los nombres de etiqueta asociados con etiquetas de sensibilidad exportadas. EL TCB marca el inició y el fin de las etiquetas de sensibilidad que propiamente representan la sensibilidad de la salida.	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales

Etiquetas sujetas a sensibilidad					
Sin requerimientos	Sin requerimientos	La TCB ejecuta una política de control de acceso mandatorio sobre todos los sujetos y objetos almacenados bajo su control. A esos sujetos y objetos les son asignadas etiquetas de sensibilidad que son una combinación de niveles de clasificación jerárquicos y categorías no jerárquicas. Y las etiquetas son utilizadas como la base de las decisiones en el control mandatorio. Los siguientes requerimientos controlan el acceso entre sujetos y objetos: un sujeto puede leer un objeto siempre y cuando el nivel jerárquico del sujeto sea mayor o igual al del objeto, así como las categorías no jerárquicas del sujeto incluyan todas las no jerárquicas del objeto.	Requerimientos adicionales: La TCB ejecuta una política de control de acceso mandatorio sobre todos los recursos, que son accesibles de manera externa por sujetos de manera directa o indirecta.	Sin requerimientos adicionales	Sin requerimientos adicionales
Identificación y Autenticación					
La TCB requiere que los usuarios se autentiquen antes de iniciar cualquier acción. La TCB protege la autenticidad de los	**Requerimientos adicionales:** La TCB asigna responsabilidad de manera individual	**Requerimientos adicionales:** La TCB mantiene autenticación de datos que incluyen información para verificar la	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales

datos y no pueden accesarlos usuarios no autorizados.		identidad de los usuarios			
Camino Seguro					
Sin requerimientos	Sin requerimientos	Sin requerimientos	La TCB soportará un camino seguro entre ella y los usuarios, para iniciar "logeo" y autenticación. Las comunicaciones por esta ruta son iniciadas exclusivamente por un usuario	**Nuevos requerimientos para B3:** La TCB soporta una ruta segura de comunicación entre ella y usuarios, cuando una conexión positiva entre TCB-usuario es requerida	Sin requerimientos adicionales
Seguridad Operacional					
Arquitectura del Sistema					
La TCB mantendrá un dominio sobre la ejecución que protege de interferencias externas y saboteos. Recursos controlados por la TCB pueden ser definidos subconjunto de los sujetos y objetos en el sistema ADP	**Requerimientos adicionales**: la TCB aisla los recursos que se protegerán de tal manera que estén sujetos a control de acceso y auditoria de requerimientos	**Requerimientos adicionales:** La TCB mantendrá procesos aislados a través de la provisión de distintas direcciones de espacio bajo su control	**Nuevos requerimientos para B2**: La TCB mantendrá en dominio su propia ejecución que lo protege de interferencias externas y saboteo. Generará uso efectivo del Hw disponible, para separar los elementos que son considerados de protección crítica, de los que no lo son.	**Requerimientos adicionales:** La TCB será diseñada y estructurada para usar mecanismos de protección simple, completos y conceptuales. Este mecanismo jugará un papel central en la ejecución de estructuras internas del sistema y la TCB.	Sin requerimientos adicionales
Integridad del Sistema					
Características de Hw y/o Sw serán proporcionadas tanto como puedan ser usadas y validadas periódicamente en operación correcta de los elementos del Hw y Firmware del TCB.	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales	Sin requerimientos adicionales

Análisis del Canal Secreto					
Sin requerimientos	Sin requerimientos	Sin requerimientos	El desarrollador del sistema conducirá la búsqueda a través de canales de almacenaje seguros y generará una determinación del máximo ancho de banda de cada canal identificado	**Requerimientos adicionales**: Investigar para todo canal secreto (almacenaje y tiempo)	**Requerimientos adicionales**: Métodos formales serán utilizados en el análisis.
Facilidad en la Administración de la Seguridad					
Sin requerimientos	Sin requerimientos	Sin requerimientos	La TCB soportará operación separada y administración de funciones	**Requerimientos adicionales**: Las funciones ejecutadas en el rol del administrador de seguridad serán identificadas	Sin requerimientos adicionales
Recuperación de la Seguridad					
Sin requerimientos	Sin requerimientos	Sin requerimientos	Sin requerimientos	Mecanismos y/o procedimientos serán proporcionados para asegurar que después de que un sistema ADP falla, se recuperará.	Sin requerimientos adicionales
Seguridad del Ciclo de Vida					
Evaluación de la Seguridad					
Los mecanismos de seguridad del sistema ADP serán evaluados y localizados para trabajar como petición en la documentación del sistema	**Requerimientos adicionales**: La evaluación incluirá una investigación de fallas obvias que pudiesen permitir la violación de recursos aislados o que permitieran acceso no autorizado a la auditoria o datos autenticados.	**Nuevos requerimientos para B1** un equipo de individuos quien a conciencia conocerá la implementación específica del TCB, sujeta a diseño documental, código fuente y código objeto a través del análisis y evaluación.	**Requerimientos adicionales:** La TCB será investigada en cuanto a resistencia a penetración, para demostrar que la TCB es consistente con lo descrito en la especificación del nivel superior	**Requerimientos adicionales:** La TCB será investigada en cuanto a resistencia a penetración. Pocas correcciones a fallas deberán existir durante la evaluación.	**Requerimientos adicionales:** La evaluación demostrará que la TCB es consistente con la especificación formal del nivel superior. Mapeo de FTLS a el código fuente, puede formar parte de la base para la evaluación de la

					penetración
Especificación del diseño y verificación					
Sin requerimientos	Sin requerimientos	Un modelo formal e informal de las políticas de seguridad será mantenido por el TCB después del ciclo de vida del sistema ADP y demostrará que es consistente con los axiomas.	Un modelo formal o informal de políticas de seguridad soportadas por el TCB serán mantenidas después del ciclo de vida del sistema ADP que es probado consistente con sus axiomas.	**Nuevos requerimientos para B2**: Un convincente argumento será proporcionado acerca de que la DTLS es consistente con el modelo	**Requerimientos adicionales**: una especificación formal del nivel superior de (FTLS) de la TCB será sostener que exactamente describe el TCB en términos de excepciones, mensajes de error y efectos.
Administración de la Configuración					
Sin requerimientos	Sin requerimientos	Sin requerimientos	Durante el desarrollo y mantenimiento del TCB un sistema de administración de la configuración mantendrá el control de los cambios para describir la especificación del nivel superior, otros diseños de datos, implementación de la documentación, código fuente, la ejecución de la versión del código objeto y pruebas fijas.	Sin requerimientos adicionales	**Nuevos requerimientos para A1**: durante todo el ciclo de vida, por ejemplo, durante el diseño., desarrollo y mantenimiento del TCB, un sistema administrador dela configuración ubicará lo relevante de la seguridad del Hw, Fw y Sw que mantiene el control de los cambios para el modelo formal, la descripción formal del nivel superior. Una combinación de defensas físicas, técnicas y procedurales serán utilizadas para proteger de modificación no autorizada o destrucción de la copia maestra o copias de todo material utilizado para generar el

					TCB.
Distribución de la Seguridad					
Sin requerimientos	Sin requerimientos	Sin requerimientos	Sin requerimientos	Sin requerimientos	Un sistema ADP de control y distribución seguro facilita el mantenimiento de la integridad del mapeo entre los datos maestros descritos en la versión actual y la copia maestra de del código para la versión actual.

Tabla A1Características de seguridad en el TCSEC

ANEXO B ALGORITMOS QUE PRESERVAN LA INTEGRIDAD

Entre los algoritmos más comunes para preservación de integridad en los últimos años, se encuentran:

MD5

Un algoritmo creado en 1991 por el profesor Ronald Rivest que es utilizado para crear firmas digitales. Es pretendido para máquinas de 32 bits y es seguro a diferencia del algoritmo de MD4, el cual ha sido roto. MD5 es una forma de función HASH, lo que significa que toma un mensaje y lo convierte en una cadena fija de dígitos, también llamada "message digest" (mensaje digital).

Cuando se emplea una forma de función HASH, uno puede comparar a un mensaje digital calculado con el mensaje digital que es decriptado con una llave pública para verificar que el mensaje no ha sido alterado. Esta comparación es también llamada "*hashcheck*".

Los "Hashes" y mensajes digitales son la parte más fuerte para comprobar la integridad de la información que viaja al través de una red y medios de comunicación, por tanto, hemos considerado diversos niveles de compendio de mensajes, para verificar el nivel se confianza en este aspecto, claro esta que esto tiene la relevancia de acuerdo a la importancia que para la empresa dueña de la información tenga la integridad de ésta.

CHECKSUM

Es un resumen del número de bits en una unidad de transmisión, incluyendo la misma información, lo cual el receptor puede checar para ver si el mismo número de bits que se envia es el que se recibe. Si coinciden las cantidades, se asume que una transmisión completa se ha realizado. Ambas capas TCP y UDP proporcionan un resumen y verificación de CHECKSUM como uno de sus servicios.

Es pocas palabras, es un sistema simple de detección de error, en el cual los mensajes transmitidos son acompañados por un valor numérico basado en el número de un conjunto de bits en el mensaje. La estación receptora entonces aplica la misma fórmula para el mensaje y verifica que se ha generado de manera segura el mismo valor. Si no es así, el receptor puede asumir que el mensaje ha sido alterado.

MD2

Es una versión anterior de 8 bits del MD5, un algoritmo utilizado para verificar la integridad de datos a través de la creación de un compendio de mensaje de 128 bits de una entrada de datos (el cual puede ser un mensaje de cualquier magnitud) que se pretende sea única. Este fue desarrollado con la intención de ser utilizado con aplicaciones de firma digital, las cuales requieren que grandes archivos puedan ser comprimidos por un método seguro antes de iniciar la encripción con una llave secreta, bajo un criptosistema de llave pública. De acuerdo con los documentos RFC, es computacionalmente imposible que cualesquiera dos mensajes que son pasados por un algoritmo MD5 puedan tener el mismo compendió de mensaje; o que un mensaje falso pueda ser generado a través de una comprensión del compendio de mensaje. MD2, MD4 y MD5 tienen una estructura similar, pero MD2 fue optimizado o diseñado para máquinas de 8 bits, a diferencia de las dos siguientes, las cuales fueron diseñados para maquinas de 32 bits. El algoritmo MD5 es una extensión del MD4, el cual tuvo una severa revisión pero es más rápido, aunque posiblemente no absolutamente seguro. A diferencia del MD5 que no es tan rápido como el MD4. pero ofrece mucha más confianza en cuanto a seguridad de datos.

Ahora bien, para no describir de manera explícita el MD4, lo hacemos, describiendo de manera rápida algunas diferencias sustanciales entre MD4 y MD5 a continuación:

ALGUNAS DE LAS DIFERENCIAS ENTRE EL MD4 Y EL MD5

- Una cuarta ronda ha sido adicionada.
- Cada paso ahora tiene una única constante aditiva.
- La función *g* en la ronda 2 es cambiada de (XY v XZ v YZ) a (XZ v Y not(Z)), para generar una *g* menos simétrica.
- A cada paso, ahora se adiciona en el resultado del paso previo. Esto promueve un rápido "efecto avalancha".
- El orden en el cual las palabras de entrada son accedidas en la ronda 2 y 3 es cambiado, para generar menos patrones semejantes al otro.
- El cambiar cantidades en cada ronda es para optimizar, y producir un rápido "efecto avalancha". Los cambios en diferentes rondas son distintos.

SHS

El *Secure Hash Standard* fue propuesto por NIST como una función para compendio de mensajes. SHS toma un mensaje de una longitud hasta de 264 bits y produce una salida de 160 bits. Es similar a la función MD5, pero es ligeramente más lento de ejecución, pero presumiblemente más seguro. MD4 hacia tres pasos sobre cada bloque de datos; MD5 hace cuatro, SHS realiza 5. Produce un mensaje de 160 bits a diferencia de los MD´s que generan 128 bits.

RFC 1321

El algoritmo *message digest* MD5, es simple para implementar y proporciona una huella digital o mensaje digital de un mensaje de longitud arbitraria. Se especula que la dificultad de que dos mensajes tengan el mismo mensaje digital es en el orden de 2^{64} operaciones, y que la dificultad de originarse con cualquier mensaje teniendo un mensaje digital dado es del orden de 2^{128} operaciones. El algoritmo MD5 ha sido cuidadosamente examinado en cuanto a sus debilidades. Es sin embargo, un algoritmo relativamente nuevo y fomentador del análisis de la seguridad.

ANEXO C CANALES SEGUROS

Normalmente un criptosistema se ve atacado de tres maneras diferentes:

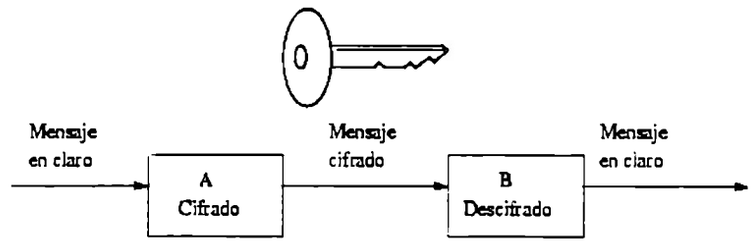
1. Ataque a partir sólo del texto cifrado
   - se intenta conseguir la clave
   - el criptosistema debe resistir este tipo de ataque aunque se consiga la función de cifrado y el lenguaje o contexto del mensaje
2. Ataque a partir de un mensaje conocido
   - el criptoanalista ha interceptado un mensaje cifrado y conoce con certeza la posición de determinadas palabras en claro en el mismo
   - el criptosistema debe resistir el ataque aunque se conozcan muchas parejas mensaje-cifrado.
3. Ataque por elección de mensaje
   - el criptoanalista puede introducir mensajes en el criptosistema y ver el resultado cifrado.

Los tipos de criptosistemas son principalmente dos:

1. *Criptosistemas Simétricos*. Llamados de clave única o clave privada.
   - La fortaleza del cifrado reside en el secreto de la clave *K*.
   - Las entidades que se comunican comparten la clave *K*
   - Debe ser computacionalmente imposible determinar, mediante pruebas sistemáticas, la clave, auque se conozcan las funciones de cifrado y descifrado y se tengan muchas parejas de mensaje-cifrado.
   - El secreto y la autenticidad se obtienen al mismo tiempo.

2. *Criptosistemas Asimétricos*. Llamados de dos claves o de clave pública.
   - Cada usuario U, dispone de dos claves ***ub*** (pública) y ***uv*** (privada)
   - Cualquier usuario puede enviar mensajes a *U* utilizando *Eub*
   - El usuario U no sabe quien le envía el mensaje porque *ub* es pública
   - La fortaleza del sistema reside en la imposibilidad de determinar uv a partir de *ub*.
   - Para asegurar la autenticidad hay que aplicar la transformación *Duv* conocida únicamente por propietario de la clave *uv*
3. La fortaleza pasa por el secreto de las claves, pero aparece un nuevo problema: la generación, administración y distribución de las claves (protocolos).

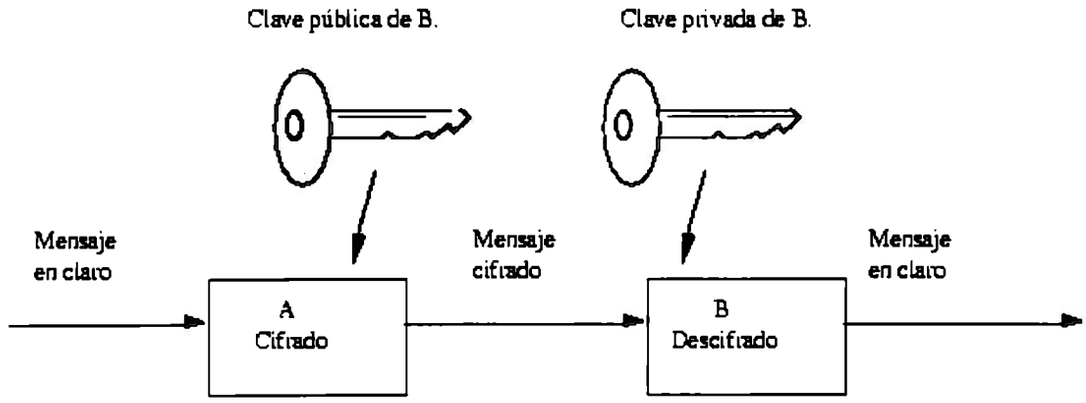
Dicho de otra manera, los criptosistemas simétricos se caracterizan por el hecho que se emplea la misma clave en las transformaciones de cifrado y descifrado. Para proporcionar confidencialidad, un criptosistema simétrico actúa de la siguiente forma: dos sistemas A y B desean comunicarse de forma segura, y mediante un proceso de distribución de claves, ambos comparten un conjunto de bits que será usado como clave. Esta clave será secreta para cualquier otro individuo, entidad, etc., distinto de A y B. Así pues, cualquier mensaje intercambiado entre A y B irá cifrado usando dicha clave.



C1 Criptosistemas Simétricos [12]

Los criptosistemas de clave pública, a diferencia de los simétricos, utilizan pares de claves complementarias para separar los procesos de cifrado y descifrado. Una clave, la privada, se mantiene secreta, mientras que la clave pública puede ser conocida. El sistema tiene la propiedad que a partir del conocimiento de la clave pública no es posible determinar la clave privada. Este enfoque con dos claves permite simplificar la gestión de claves, minimizando el número de claves que deben ser gestionadas y permitiendo su distribución a través de sistemas no protegidos. En una red con *n* usuarios, si se usa cifrado de clave simétrica se precisan *n(n-1)/2* claves, mientras que si se emplea cifrado de clave pública bastan *2n* claves.

Potencialmente hay dos modos de uso de los criptosistemas de clave pública, dependiendo del uso que se haga de la clave privada (cifrado o descifrado). Por una parte cualquier usuario puede enviar un mensaje de forma confidencial a un receptor (p.e. B) cifrando su contenido con la clave pública del receptor, que será el único capaz de descifrarlo por ser el único conocedor de la clave privada (en caso contrario la gestión de claves estaría mal hecha. Por otro lado cualquier usuario (p.e. A) puede autenticar el origen y contenido de un mensaje cifrándolo con su clave secreta, ya que prueba su identidad como único poseedor de esta clave. Cualquier receptor puede verificar la autenticidad del mensaje descifrándolo con la clave pública del emisor. La siguiente figura muestra el proceso.



C2 Criptosistemas Asimétricos (Llave Pública) [12]

De manera muy resumida ese es el funcionamiento de los criptosistemas empleados en la actualidad para proporcionar control de acceso, autenticación, pero sobre todo confidencialidad de la información.

Ahora bien, en cuanto a tecnologías que permiten contar con canales seguros, podemos mencionar:

- Redes Virtuale s Privadas Virtuales (VPN)
- IPSec
- SSL (Secure Socket Layer)

VPN

Aunque algunos proveedores y prestadores de servicios quizá estén en desacuerdo, comúnmente una red privada virtual (VPN) es un grupo de dos o más sistemas de cómputo, típicamente conectados a una red privada (una red construida y mantenida por una sola organización, para uso exclusivo) con limitado acceso a red pública, esto permite comunicar "de manera segura" sobre una red pública.

Las VPN´s pueden existir entre una máquina individual y una red privada (cliente-servidor) o una LAN remota y una red privada (servidor a servidor). Las características de seguridad difieren de un producto a otro, pero la mayoría de los expertos de seguridad están de acuerdo en que las VPN´s incluyen encripción, autenticación robusta de usuarios remotos o hosts y mecanismos para ocultamiento o enmascaramiento de información acerca de la topología de la red privada a atacantes potenciales en la red pública.

Independientemente del gran número de productos VPN´s, todo recae en tres amplias categorías: sistemas basados en HW, VPN´s basados en Firewall y paquetes de aplicación VPN independientes.

La mayoría de sistemas VPN basados en HW son ruteadores de encripción. Son seguros y fáciles de utilizar, dado que proporcionan una cercanía al equipo de encripción "plug and play" disponible. Proporcionan el más alto rendimiento de red de todos los sistemas VPN, puesto que no consumen recursos fijos de procesador en ejecución de sistema operativo u otras aplicaciones. Sin embargo, pueden no ser flexibles como los sistemas basados en SW. El mejor conjunto de VPN basado en HW ofrece a SW solo a clientes para instalación remota e incorpora algunas características de control de acceso más tradicionalmente manejadas por Firewalls u otros dispositivos periféricos de seguridad.

Los VPN´s basados en Firewall toman ventaja de los mecanismos de seguridad de Firewalls, incluyendo restricciones de acceso a la red interna. Ellos también desempeñan interpretación de direcciones, satisfacen requerimientos para autenticación robusta y levantan el servicio de alarma en tiempo real y registro extenso. La mayoría de Firewalls comerciales, adicionalmente fortalecen el núcleo del host donde se encuentra el sistema operativo para poner la vista en peligros o servicios innecesarios, proporcionando seguridad adicional para el servido VPN. La protección del S.O. es un valor agregado, dado que muy pocos proveedores de aplicaciones VPN, proporcionan una guía de seguridad sobre S.O. el funcionamiento puede ser una inquietud, especialmente si el Firewall esta ya cargado, sin embargo, los proveedores de Firewalls ofrecen procesadores de encripción basados en HW para minimizar el impacto del control de VPN en el sistema. [14]

Los VPN´s basados en Sw, son ideales en situaciones donde ambos puntos finales del VPN no son controlados por alguna organización (típicamente para requerimientos de ayuda al cliente o sociedades comerciales), o cuando diferentes Firewalls y ruteadores son implementados dentro de la misma organización. Hasta el momento, los VPN´s independientes ofrecen la mayor

flexibilidad en cuanto a administración de tráfico en la red. Muchos productos basados en Sw permiten tráfico basado en direcciones o protocolo, a diferencia de los productos basados en Hw, los cuales generalmente "tunelean" todo tráfico que ellos manejan, independientemente del protocolo. El "tuneleo" de tipos específico de tráfico, es una ventaja en situaciones donde sitios remotos pueden distinguir una mezcla o tráfico, algunos que necesitan transportarse sobre un VPN (así como acceso a bases de datos o centro de operaciones) y algunos que no lo necesitan (así como navegar en la Web). En situaciones donde los requerimientos de funcionamiento son bajos (tal como usuarios conectados sobre "dial up links"), los VPN´s basados en Sw, pueden ser la mejor opción.

Los sistemas basados en Sw, son generalmente robustos para controlar los ruteadores de encripción. Requieren compatibilidad con el sistema operativo del Host, la aplicación por sí misma debe contar con sus mecanismos de seguridad. Algunos paquetes de VPN basado en Sw requieren cambios en tablas de ruteo y sistemas de direccionamiento de la red. [10]

IPSEC

Es un evolucionado estándar para comunicaciones privadas seguras sobre internet. Los paquetes normales IPv4 consisten de encabezados y carga útil, ambos contienen información de valor para un atacante. El encabezado contiene dirección IP, origen y destino, los cuales son requeridos para rutear, pero pueden ser engañados o alterados en lo que es conocido como ataques "*man in the middle*"; la carga útil, consiste de información, la cual puede ser confidencial para una organización en particular. El IPSec proporciona mecanismos para proteger tanto los datos del encabezado como de la carga útil. El IPSec autentifica el encabezado digitalmente (AH), firmando el paquete de salida, tanto los datos del encabezado como de la carga útil, con un valor Hash anexado al paquete, verifican la identidad de las máquinas origen y destino, así como la integridad de la carga útil. El ESP (Encapsulating Security Payload) de IPSec garantiza que la integridad y confidencialidad de los datos en el mensaje original en combinación con un Hash seguro y encripción de ambos, permiten que el mensaje original se mantenga.

NAT (netwok address traslation) es incompatible con la autenticación de protocolo, si es utilizado en trasportación o modo "tuneleado". Un IPSec VPN utiliza protocolos (AH) firmando digitalmente los paquetes de salida, tanto de datos de carga útil como de encabezado, con un valor Hash anexado al paquete. Cuando se utiliza el protocolo AH, el paquete que contiene la carga útil no es encriptado.

DES

Este algoritmo emplea una clave de 56 bits y opera con bloques de datos de 64 bits. El proceso de cifrado ejecuta una permutación inicial al texto en claro, y aplica 16 veces una función que depende de la clave. Una de las controversias generadas con el DES es precisamente si la longitud de la clave es suficientemente grande o no. El algoritmo se basa en permutaciones, sustituciones y sumas módulo 2. Las permutaciones son de tres tipos:

- Directas: reordenamiento de bits
- Expandidas: algunos bits se duplican y el conjunto se reordena
- Selecciones permutadas: algunos bits se deprecian y el resto se reordena.

Las sustituciones en el DES son conocidas como cajas S y están especificadas en ocho tablas diferentes. El algoritmo es el mismo para cifrar que para descifrar.

IDEA (INTERNATIONAL DATA ENCRYPTION ALGORITHM)

Fue originalmente llamado IPES (Improved Proposed Encrytion Standard), fue diseñado para ser eficiente computando en Software. Encripta bloques de 64 bits de texto plano dentro de un bloque de 64 bits de texto cifrado utilizando una llave de 128 bits es relativamente nuevo, pero suficiente publicado, que los criptoanalistas han tenido suficiente tiempo para detectar vulnerabilidades. Hasta el momento ninguna ha sido encontrada.

IDEA es similar a DES en algunas formas. Ambos operan a base de rondas, tienen una complicada función "destrozadora". La diferencia radica en la expansión de la llave. Con DES, las mismas llaves son utilizadas en orden inverso; con IDEA, las llaves de encripción y decripción son relacionadas de una manera compleja.

ANEXO D RESISTENCIA A ATAQUES

Entre los ataques posibles podemos mencionar algunos de ellos:

- Desbordamiento de memoria
- Desbordamiento de stack
- IP Spoofing, o falsificación de direcciones de origen o destino
- Escaneo de puertos
- Saturación de conexiones
- Virus

DESBORDAMIENTO DE MEMORIA

La Mayoría de antivirus hoy día ofrecen protección en cuanto a desbordamiento de memoria se refiere. permitiendo de esta manera reducir los riesgos en cuanto a inseguridad computacional se refiere. provocada por desbordamientos de este estilo.

DESBORDAMIENTO DE STACK

Este es también uno de los aspectos que se ha comentado en el párrafo anterior son controlados desde un sistema o antivirus externo al tipo de aplicación, por eso es que este aspecto no es del todo preocupante, dados los avances que la respecto existen, y los cuales no permiten que gente externa a la empresa o institución dueña del sistema objetivo de protección, pueda valerse de la información almacenada en determinado momento en una pila o que ha sido arrojado al provocarse el desbordamiento de una pila. Sin embargo, no debemos olvidar que son estos ataques los que tradicionalmente le permiten al atacante obtener privilegios de root.

IP SPOOFING

Una técnica utilizada para ganar acceso a computadoras, por lo cual el intruso envía mensajes a una computadora con una dirección IP, indicando que el mensaje viene de un puerto seguro. Para involucrar una IP errónea, un Hacker puede primero utilizar una variedad de técnicas para encontrar una dirección IP de un puerto seguro y entonces modificar los encabezados de paquete, a fin de que parezca que los paquetes proceden de ese puerto.

Utilizando el IPSpoofing un servidor o host no conectado a esa red se puede conectar a ella por medio del Internet cambiando su IP, por una que pertenezca a la red, por lo que aparentemente cambia su IP suplantando la dirección IP de la red local, de esta manera engaña a los servidores de esa red para que no le soliciten autenticación

ESCANEO DE PUERTOS

Es el hecho de sistemáticamente escanear los puertos de una computadora. Dado que un puerto es un lugar por donde la información sale de una computadora, escanear puertos identifica las

puertas abiertas a una computadora. Escaneo de puertos es legítimamente usado en la administración de redes, pero escanear los puertos también puede ser malo, si alguien observa una debilidad en un punto de acceso para irrumpir dentro de tu computadora.

Tipos de escaneo de puertos:

- *Vainilla*: el escáner a un intento de conexión a todos los puertos 65535
- *Strobe*: un enfoque más del escaneo, es para conocer servicios a explotar
- *Fragmentación de paquetes*: el escáner envía fragmentos de paquetes que se obtienen a través de filtrados simples de paquetes en un Firewall.
- *UDP*: este escaneo busca puertos UDP abiertos
- *Sweep*: el escaneo es de conexiones al mismo puerto en más de una máquina
- *Salto FTP*: el escaneo viene a través de un servidor FTP a fin de disfrazar la procedencia del escaneo.

VIRUS

Evidente es que hoy en día, los virus representan una amenaza palpabale y cotidiana a todos los sistemas y equipos computacionales, dadas la "mejoras" que día con día los creadores de ellos han realizado, siempre con la finalidad de causar el daño para el cual fueron creados, por eso es que resulta interesante, saber y comparar de entre los antivirus, cuál seria una de las mejores opciones para adquirir, instalar y al mismo tiempo saber que en la medida de lo posible, se esta seguro contra los virus mas conocidos. Por ello es que se investigó y en el **Anexo L** se plasma una comparativa realizada por HISPASEC la cual a juicio de los conocedores, es considerada como una de las comparaciones más exhaustivas, serias y honestas, realizadas por cualquier tipo de organización o publicación en español.

ANEXO E SECURE SOCKET LAYER (SSL)

El protocolo **SSL** es un sistema diseñado y propuesto por *Netscape Communications Corporation*, con la idea de proporcionar seguridad y privacidad sobre Internet. Se encuentra en la pila OSI entre los niveles de **TCP/IP** y de los protocolos HTTP, FTP, SMTP, etc. Proporciona sus servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico, típicamente el RC4, y cifrando la clave de sesión de RC4 mediante un algoritmo de cifrado de clave pública, comúnmente el RSA. La clave de sesión es la que se utiliza para cifrar los datos que vienen de, y hacia el servidor seguro. Se genera una clave de sesión distinta para cada transacción, lo cual permite que aunque sea violada por un atacante en una transacción dada, no sirva para descifrar futuras transacciones. **MD5** se usa como algoritmo de Hash.

Proporciona cifrado de datos, autenticación de servidores, integridad de mensajes y, opcionalmente, autenticación de cliente para conexiones TCP/IP.

Cuando el cliente pide al servidor seguro una comunicación segura, el servidor abre un puerto cifrado, gestionado por un software llamado Protocolo SSL Record, situado encima de TCP. Será el software de alto nivel, Protocolo SSL Handshake, quien utilice el Protocolo SSL Record y el puerto abierto para comunicarse de forma segura con el cliente.

SSL se divide en dos capas, cada una utiliza servicios proporcionados por una capa más baja y proporcionando funcionalidad a las de más arriba. El *SSL Record* proporciona confidencialidad, autenticidad y protección repetida sobre una conexión orientada confiable, protocolo de transporte similar a TCP. El anterior sistema de capas es el protocolo *SSL Handshake*, un protocolo secreto de intercambio, con el cual se inicializa y sincroniza un estado criptográfico entre dos puntos. Después que el protocolo de intercambio secreto se completa, los datos sensibles pueden ser enviados, por medio de la capa SSL Record.

LA CAPA DE REGISTRO (THE RECORD LAYER)

La capa de SSL Record, direcciona claramente los problemas estándar, que deben recibir bastante atención en criptografía.

CONFIDENCIALIDAD: *"ESCUCHA"*

El SSL Record encripta todos los datos de la capa de aplicación, con un cifrado y pequeña sesión de negociación de llaves mediante el protocolo *Handshake*. Una amplia variedad de algoritmos robustos, están disponibles para ser utilizados de manera estándar; las aplicaciones deben ser capaces de encontrar un algoritmo de encripción que cumpla los requerimientos de seguridad y las leyes de exportación de los USA. La administración de llaves es realizada de la siguiente manera: pequeñas sesiones de llaves son generadas por un "Hashing" aleatorio y de manera compartida altamente secreta. Independientemente las llaves son usadas para cada sentido de una conexión, así como para diferentes instancias de una conexión.

CONFIDENCIALIDAD: "*ANÁLISIS DE TRÁFICO*"

Este es otro ataque que es digno de considerar. Este ataque ambiciona recuperar información confidencial acerca de sesiones de protección examinando campos de paquetes no encriptados y atributos de paquete no protegidos. Por ejemplo, para examinar direcciones IP origen y destino no encriptadas (incluso puertos TCP), o para examinar el volumen del flujo de datos en la red, un analista de tráfico puede determinar que partes interactúan, que servicios utilizan e incluso, algunas veces recuperar información acerca de sus relaciones de negocio o personales.

Esta vulnerabilidad es presentada porque la longitud del texto cifrado revela la longitud del texto plano. SSL incluye soporte para relleno aleatorio de bloques en modo cifrado, pero no para flujos en modo cifrado. Por tanto es recomendable utilizar lo menos posible SSL para relleno aleatorio en modo cifrado.

CONFIDENCIALIDAD: "*ATAQUES ACTIVOS*"

Un ataque activo importante es el perpetrado sobre IPSec. Ya que este ataque conocido como "cortar y pegar", explota los puntos finales de las aplicaciones intentando de entrada, encriptar los datos de manera diferente y dependiendo del contexto. Este tipo de ataque también toma ventaja de las propiedades básicas del cifrado de bloques. Por ejemplo, si este ataque se realiza sobre la capa de registro del SSL, es factible, que pueda comprometerse la seguridad del sitio. La versión SSL 3.0 detiene este tipo de ataque. El ataque del "pequeño bloque", es otro ataque activo sobre IPSec, aunque sobre SSL, no se han encontrado evidencias de este. En resumen no se conoce un ataque activo a la protección de confidencialidad de la capa de registro de SSL 3.0.

AUTENTICACIÓN DE MENSAJES

Además de proteger la confidencialidad de los datos, SSL, criptográficamente autentica las comunicaciones sensitivas

ATAQUES DE REPETICIÓN

El uso ingenuo de una MAC, no necesariamente detiene a un adversario de replicar un paquete dañado. SSL protege contra ataques de replicación, mediante la inclusión de una secuencia de números en los datos MAC. Este mecanismo también permite protegerse contra retardo, reordenamiento o borrado de datos.

EL PROTOCOLO SSL HANDSHAKE

Durante el protocolo SSL Handshake, el cliente y el servidor intercambian una serie de mensajes para negociar las mejoras de seguridad. Este protocolo sigue las siguientes seis fases:

La fase Hola, usada para ponerse de acuerdo sobre el conjunto de algoritmos para *mantener la intimidad y para la autenticación.*

La fase de intercambio de claves, en la que intercambia información sobre las claves, de *modo que al final ambas partes comparten una clave maestra.*

La fase de producción de clave de sesión, que será la usada para cifrar los datos *intercambiados.*

*La fase de **verificación del servidor**, presente sólo cuando se usa RSA como algoritmo de intercambio de claves, y sirve para que el cliente autentique al servidor.*

*La fase de **autenticación del cliente**, en la que el servidor solicita al cliente un certificado X.509 (si es necesaria la autenticación de cliente).*

*Por último, la fase de **fin**, que indica que ya se puede comenzar la sesión segura.*

Dicho de otra manera, se puede dividir el proceso de recepción de mensajes del servidor en tres partes:

En la primera, después de que el cliente envía la lista de algoritmos y la petición, espera del servidor: el identificador de conexión, el certificado y la nueva lista. Entonces en esta parte se probará que eventualmente recibe respuesta del servidor.

Después el cliente debe recibir del servidor, la petición que hizo pero de forma encriptada para asegurarse que fue la petición original y que la recibió el servidor que supone el cliente. Entonces de la misma forma que en la parte anterior, ahora también se validará que eventualmente el cliente recibe esta respuesta del servidor.

Por último el cliente necesita una nueva llave de sesión para terminar el Handshake, e iniciar el proceso de envío de datos, el recibir esta llave es de vital importancia por tanto debemos validar que esta llave sea recibida.

Por otro lado, el servidor debe poder atender las peticiones de los clientes, por su parte el cliente debe verificar que eventualmente haga los tres envíos que espera cada cliente.

En ambos casos, tanto en el cliente como en el servidor, hay secciones en las que no deseamos que el proceso se cicle, como por ejemplo: que el proceso se encuentre iterando seguidamente por la parte de time-out o recepción errónea.

En conclusión, SSL 3.0 proporciona una seguridad excelente contra "escucha" y otros ataques pasivos. Aunque los modos de exportación débiles, ofrecen solo mínima protección en cuanto a confidencialidad. [23]

Existen diversas formas en las cuales las robustez del protocolo SSL puede ser mejorado.

ANEXO F CRIPTOGRAFÍA EN JAGUAR

Los mensajes no encriptados son conocidos como *texto plano*. La codificación del contenido de un mensaje es llamado encripción. El mensaje encriptado es el *texto cifrado*. Una llave es usualmente requerida para ejecutar la encripción y decripción. Un *CipherSuite* define los parámetros y métodos soportados por ambos, cliente y servidor que ejecutan la encripción y decripción.

La encriptación de llave pública utiliza un par de llaves para encriptar y decriptar. Una llave es secreta (la llave privada) y la otra es distribuida (llave pública). Se envía la llave pública firmada digitalmente (certificado) a alguien con quien se desee comunicar utilizando datos codificados.

Los mensajes que son enviados, son encriptados con la llave pública distribuida y decriptados con la llave privada, mientras que los mensajes que son encriptados con la llave privada, deben ser decriptados con la llave pública. La encripción con RSA es un sistema de encripción de llave pública; ampliamente usado.

CERTIFICADOS DE LLAVE PÚBLICA

Los certificados de llave pública proporcionan un método para identificar y autenticar clientes y servidores en Internet. Los certificados de llave pública son administrados y emitidos por una tercera parte conocida como *Autoridad Certificadora* (CA). Un sujeto (sistema individual u otra entidad en la red) utiliza un programa para generar un par de llaves y enviar la llave pública a la CA, junto con información de identificación (como nombre, organización, dirección de correo electrónico, etc.). Esto es conocido como *petición de certificación*. La CA emite un certificado con firma digital. Una firma digital es un bloque de datos que es creado utilizando una llave pública.

La CA "ata" el certificado del propietario a la llave pública dentro del certificado. El sujeto entonces utiliza el certificado, junto con su llave pública para establecer su identidad. Una vez realizado esto, cualquier sujeto se puede comunicar sabiendo que una tercera parte puede garantizar su identidad.

La figura F.1 ilustra un escenario en el cual un servidor requiere autenticación al cliente:

Un cliente emite una petición y recibe un certificado de la CA.

Un administrador instala el certificado de la CA en el servidor e indica confianza. Cualquier cliente certificado por la CA será conocido y aceptado por el servidor.

El cliente proporciona su certificado y negocia una conexión segura con el servidor.

Después de la autenticación, una conexión segura es establecida. El cliente y servidor pueden utilizar los certificados de cliente para encriptar y decriptar datos para prevenir saboteos.

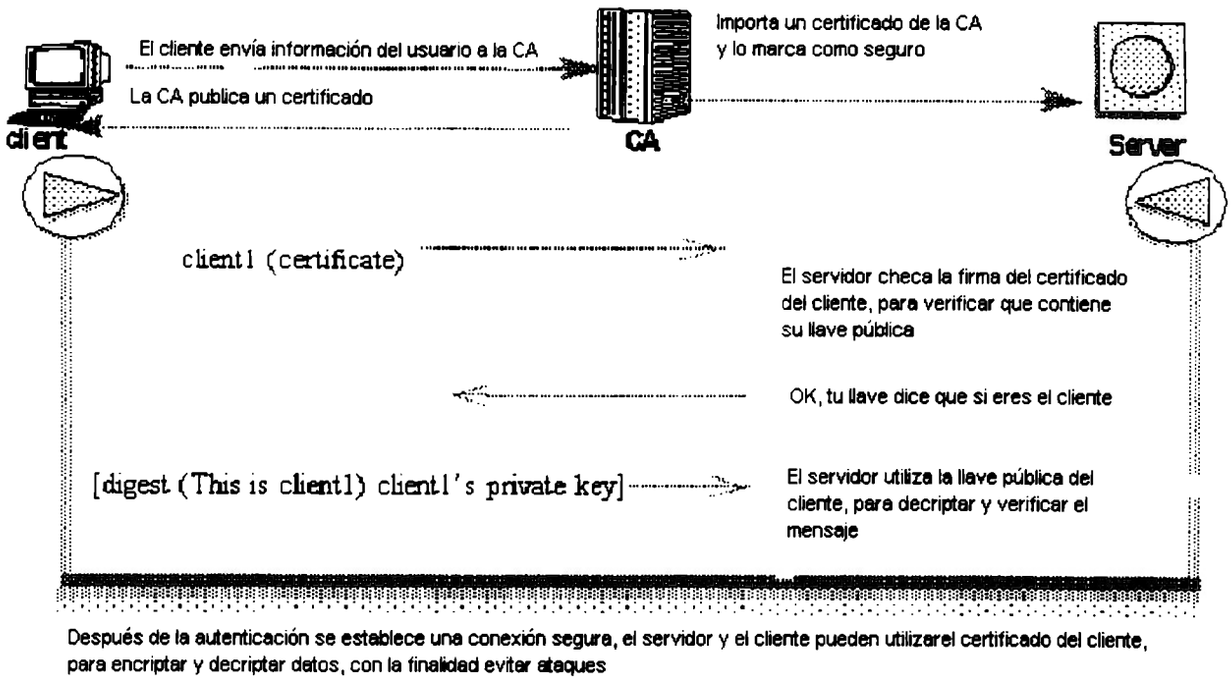


Fig.F1 Escenario de Autenticación del cliente ante el servidor [23]

ADMINISTRACIÓN DE CERTIFICADOS

Netscape 4.0x abastece un módulo de PKCS #11 que permite administrar los certificados del cliente. Sybase también proporciona un módulo de PKCS #11 que permite administrar sus certificados. Sybase recomienda que se instale el módulo de PKCS #11 de Sybase dentro de Netscape, el cual permite el acceso inmediato al ejemplo de los certificados del servidor de Jaguar.

La instalación del módulo de PKCS #11 de Sybase dentro de Netscape 4.0x

Inicie Netscape y:

- seleccione *Comunicación – Información* de Seguridad o, se puede dar un clic en el ícono de seguridad en la barra de herramientas
- clic en los módulos de criptografía
- Click en Add. Ahí se verá un nuevo diálogo, crear un nuevo módulo de seguridad.
- Para el nombre del módulo de seguridad, introduce "Sybase PKCS"
- Para el archivo de módulo de seguridad, introduce la ruta completa del archivo libjsybcki y pulsa O.k. Para el caso de NT, introduce:
  /work/JagPKS/lib/libjsybcki_r.so

Después se verá un prompt preguntando por un Password o PIN de Sybase. Introduzca "sybase". Si no se ve ese prompt, verificar la ruta hacia el objeto compartido DLL.

Después de introducir el Password, se visualizará un listado de Sybase PKCS, como un módulo de seguridad Pulse sobre el módulo de Sybase PKCS, entonces seleccione View/Edit. Una nueva ventana, la del módulo de edición de seguridad, se desplegará. Esta ventana contiene controles del módulo de PKCS de Sybase.

Clic en Más información de la nueva ventana y verifique que el estado es "Ready" en el token/slot de la ventana de información. Pulse O.K. para cerrar la ventana antes mencionada.

Se puede cambiar el Password del módulo de PKCS de Sybase pulsando sobre Cambio de Password en la ventana del módulo de edición de seguridad.

Cuando ambos Jaguar y Netscape, se ejecutan en la misma máquina, comparten archivos de bases de datos de PKCS #11 de Sybase. Si se cambia el PIN, se puede utilizar el nuevo PIN cuando se identifique en cualquiera, Jaguar o Netscape. Sybase sugiere cambiar el PIN a través del administrador de Jaguar, el cual automáticamente propaga los cambios de PIN hacia los perfiles de seguridad. Si se cambia el PIN a través de Netscape, se puede también cambiar el PIN en todos los perfiles de seguridad; de otro modo los receptores de seguridad de Jaguar utilizan esos perfiles de seguridad y no poder iniciar la siguiente vez que se restaure el servidor.

Ahora bien, para cambiar el PIN de usuario se los pasos siguientes deben tomarse en cuenta:

- Seleccionar la carpeta de *Llaves Privadas*
- Seleccionar *Archivo / cambio de PIN*
- Introducir y verificar el nuevo PIN

Restaurar Netscape para que el nuevo PNI se propague hacia las muestras de PKCS #11.

Desplegar el módulo de información de PKCS #11

- Seleccionar la carpeta de llaves privadas
- Para ver la información acerca del módulo de PKCS #11 de Sybase, incluyendo la versión de librería y la versión de *Cryptoki*, seleccionar *File / Módulo de Información.*

Para ver información acerca del token PKCS #11 de Sybase, que administra su llave y la información de certificado, incluyendo nivel y versión de información, seleccionar *File / Información del Token.*

Para salir del módulo de PKCS # 11:

- Seleccionar la carpeta de llaves privadas
- Seleccionar *Archivo / Salir*

Administración de la Autoridad Certificadora (CA)

La CA es un ente con autoridad que firma las peticiones de certificación de usuario. Esos certificados pueden ser utilizados por clientes, y Jaguar evalúa las características de seguridad de sus aplicaciones. Los certificados firmados por la CA no son interpretados por aplicaciones comerciales. Si ya se tiene una CA u otra firma autorizada, podría no ser necesario valerse de esta CA ofrecida por Jaguar.

Las tareas involucradas en la administración de una CA, incluye:

- Creación de la CA

- Generación de un certificado de usuario, firmado por la CA
- Procesamiento de una petición de certificado
- Exportar un certificado desde la CA

Creación de CA

Para verificar que la CA esta disponible, resaltar la carpeta de *Certificados CA.* Se visualizará *usuario CA de Sybase Jaguar* del lado derecho de la ventana. Si no sucede eso, se debe general la CA.

- Seleccionar la carpeta de *Certificados CA*
- Seleccionar *archivo / generar CA*

Generación de un certificado de usuario, firmado por la CA

- Seleccionar la carpeta de *Certificados CA*
- Seleccionar *Archivo / Generación de certificados de usuario.* Se desplegará el asistente par generación de certificados de usuario.
- Proporcionar la información requerida, descrita en la tabla F2. Pulse retroceso y siguiente para revisar y/o modificar información.
- Se puede utilizar cualquiera de los siguientes caracteres para la etiqueta:
  - Letras de la A-Z y a-z
  - Valores numéricos 0-9
  - (espacio) ' ( ) + , - . / : = ?
- Pulsa Finalizar para salir del asistente y generar el certificado
- Clic O.K en la caja de diálogo *Info.* El certificado se desplegará cuando se resalte la carpeta de *Certificados de usuario.*

Table 5-1: User test certificate information

Property	Description	Comments/example
Key Strength	Select the authentication key strength. The greater the number, the stronger the encryption. Your options are: 512 bits 768 bits 1024 bits	For international users, key strength is 512.
Key Label	The name that identifies the certificate.	Required field. The label must be unique among all labels used for all certificates.

Validity Period	From the drop-down list, select the length of time that the certificate is valid.	When a client (or Jaguar) presents a certificate for authentication, Jaguar (or the browser) checks to see if the certificate has expired.
Cert Usage	Click the check box for either or both: **SSL Client** **SSL Server**	The same certificate can be used by a client and/or a Jaguar Server.
Common Name	Your first and last name.	Required field.
User ID	Any ID that would further identify you.	
Organization	The name of your company, university, or other organization.	Required field.
Organization Unit	The name of a department within your organization.	
Locality	The location of your organization.	You must supply at least one of: **Locality** **State/Province** **Country**
State/Province	State or province name.	
Country	Your two-digit country code; for example, "US."	
Requester Name	The person requesting the certificate.	
Server Admin	The name, if any, of the server administrator.	
E-Mail	Your e-mail address.	
Mark Private Key Exportable	Checked by default, this property allows you to export this certificate along with its private key.	See "Installing and exporting certificates" for more information. If checked, you can later uncheck this property. Once unchecked, you cannot change this property. If unchecked, you

		cannot export this certificate and private key.

Tabla. F2 Procesamiento de una petición de certificado [23]

El administrador de seguridad puede procesar una petición de certificado generado desde cualquier sitio. La CA firma la petición y genera el certificado.

- Seleccionar la carpeta de certificados CA
- Seleccionar *Archivo / proceso de petición de certificado*
- Pegar la petición de certificado dentro de la ventana como se indica, a continuación se muestra un ejemplo de una petición de certificado base64. Se puede incluir el contenido completo, así como las líneas de inicio y fin.

-----BEGIN NEW CERTIFICATE REQUEST-----

MIH4MIGjAgEAMD4xCjAIBgNVBAMTAWExCjAIBgNVBAoTAWExCjAIBgNVBAcTAW
Ex
CzAJBgNVBAgTAmNhMQswCQYDVQQGEwJ1czBcMA0GCSqGSIb3DQEBAQUAA0sAM
EgC
QQC9Yn9AOzflqIarPCC7eRdr3C0wrIG+3B2T+pEs9sdgEjnc/bw1GfxcZKYamWXg
G1KQycFqkdrFNP79fgRCOd3xAgMBAAGgADANBgkqhkiG9w0BAQQFAANBAIEljmCB
HbFdNj0MtFDa002f/Trl6FtGCh7Gs23pZlWIUzDlGFowiuJY6iMDzd/1bJz5yYB+
IvlM9Ath/zTF2eY=

-----END NEW CERTIFICATE REQUEST-----

- Configurar las propiedades de certificado siguientes:
  - Tipo de formato.- identificar el tipo de formato de la petición, ya sea "Binario" o "Base64"
  - Uso.- dependiendo de cómo se utilice el certificado, seleccionar cliente SSL, Servidor SSL o ambos.
  - Período de validación.- seleccionar la longitud del tiempo que el certificado tendrá validez.
- Pulsar *Next.* El certificado es generado y desplegado en la caja de diálogo. A continuación se muestra un certificado firmado en Base64:

----BEGIN CERTIFICATE-----

MIICYTCCAcqgAwIBAgIBBzANBgkqhkiG9w0BAQQFADCBgjEzMDEGA1UEAxMqU3li
YXNlIEphZ3VhciBVc2VyIFRlc3QgQ0EgKFRFU1QgVVNFIE9OTFkpMSAwHgYDVQQK
ExdTeWJhc2UgSmFndWFyIFVzZXIgVGVzdDEpMCcGA1UEBxMgU3liYXNlIEphZ3Vh
ciBVc2VyIFRlc3QgTG9jYWxpdHkwHhcNOTgwNzAyMDIzOTEzWhcNOTgwOTAyMDIz
OTEzWjBHMQ0wCwYDVQQDEwR0ZXN0MQ0wCwYDVQQKEwR0ZXN0MQ0wCwYDV
QQHEwR0
ZXN0MQswCQYDVQQIEwJjYTELMAkGA1UEBhMCdXMwXDANBgkqhkiG9w0BAQEFA
ANL
ADBIAkEAvzvqs9yjW/PDCt/Rotp9x9PHrULLeGOLlVSubo9poY1f5OYwsrjfaOtT

bkhWDrakuwJJk8smDNSAl93tdP9r8wIDAQABo2UwYzAMBgNVHRMEBTADAQEAMB0G
A1UdDgQWBBTAT0n9qsvdfqc9NzGPA5oLKsMzJjAhBgNVHSMEGjAYoBYEFGLT8qZb
3LtGjw84nxna9YBHb7q6MBEGCWCGSAGG+EIBAQQEAwIAwDANBgkqhkiG9w0BAQQF
AAOBgQB3OStVqhoWT66yXNsrznCg9t8yNClobnKGOJTqt+VbhV7BUgBH+fVSjf7v
xJyV4twwlBvU08PsKYQGj4sJ1Ao3lsOXWrr6YZIHZZ6p9P8JXjY016Vg9g5SDmEV
jgGbwy6ZOZYx27npp4X31WXY27KDZrV/FrwvF6/Pv6mZY7ijUw==

-----END CERTIFICATE-----

- seleccionar *guardar archivo* e introducir el nombre de la ruta completa, para guardar el certificado generado, como un archivo. También se puede hacer esto, seleccionando *Examinar* para especifica la ubicación del archivo.

Si se desea utilizar un certificado para Autenticación, se debe instalar el certificado en la misma máquina donde la petición de certificado fue generada, dado que es ahí dónde la llave privada esta almacenada.

Los certificados firmados por la CA solo son pretendidos para evaluación. En una situación real, la CA debe verificar la información de usuario para establecer identidad.

Exportar un certificado desde la CA

Se pueden exportar certificados, incluyendo el certificado de la CA. Exportar el certificado de la CA, permite cargarlo dentro del browser de Netscape y los marca como seguro.

- Seleccionar la carpeta de *certificados CA*
- Resaltar usuarios de CA de Jaguar Sybase
- Seleccionar *Archivo / exportación de certificados*
- Desde el asistente de exportación de certificados, seleccionar el tipo de formato para el certificado exportado. Para la CA, seleccionar *Binary Encode X509 Certificate*. Pulsar *Next*.
- Seleccionar Guardar archivo e introducir el nombre de la ruta completa para el archivo que contiene la CA.

No se debe adicionar ninguna extensión al final del nombre de archivo. La extensión *A.crt* es automáticamente adicionada al certificado exportado por el administrador de seguridad. Netscape 4.0 reconoce esta extensión como un certificado X.509 y por consecuencia los manipula.

- Pulsar *Finish* para exportar el certificado desde el archivo especificado

ANEXO G CONTROL DE ACCESO

Desde los orígenes del tiempo (informático) se ha tratado de controlar el acceso a los recursos de información. La tecnología utilizada para este control de accesos ha evolucionado con los propios sistemas de información protegidos.

La autenticación pretende establecer quién eres. La autorización (o control de accesos) establece qué puedes hacer con el sistema. Ambos conceptos parecen ir ligados de forma indisoluble, pero no siempre. La posesión de la llave de mi coche le parece autoridad suficiente para dejarse conducir, incluso por un individuo cuyo pasaporte poco tiene que ver con el mío. Estos dos conceptos, que a menudo se mezclan de manera difusa, son completamente independientes. Sin embargo, especialmente cuando se accede a un recurso de información vía red, sin protección física, existen tres actividades que irán siempre ligadas: La autenticación (quién soy), la autorización (qué puedo hacer) y el registro de auditoria (qué he hecho).

En un mundo cada vez más dependiente de las redes, es vital no sólo proteger el acceso a los recursos por los 'hackers, sino también evitar una manipulación accidental con fatales consecuencias.

Existen tradicionalmente dos tipos básicos de controles de acceso con filosofías diametralmente opuestas:

En el modelo de control de acceso discrecional (DAC), un usuario bien identificado (típicamente, el creador o propietario del recurso) decide cómo protegerlo estableciendo cómo compartirlo, mediante controles de acceso impuestos por el sistema. Este es el modelo habitual en buena parte de los sistemas operativos más habituales. Lo esencial es que el propietario del recurso puede cederlo a un tercero.

En sus inicios estos sistemas eran excesivamente simples, al permitir un conjunto limitado de operaciones posibles sobre un recurso (rwx por propietario, grupo o resto de usuarios, como en Unix), si bien pronto se añadieron las famosas listas de control de accesos (ACLs), listas de usuarios y grupos con sus permisos específicos. Las ACLs permiten un nivel de granularidad que, en ocasiones, es inconveniente, por cuanto complica la administración de la seguridad.

En el modelo de control de acceso mandatorio (MAC), es el sistema quién protege los recursos. Todo recurso del sistema, y todo principal (usuario o entidad del sistema que represente a un usuario) tiene una etiqueta de seguridad. Esta etiqueta de seguridad sigue el modelo de clasificación de la información militar, en donde la confidencialidad de la información es lo más relevante, formando lo que se conoce como política de seguridad multinivel. Una etiqueta de seguridad se compone de una clasificación o nivel de seguridad (número en un rango, o un conjunto de clasificaciones discretas, desde DESCLASIFICADO hasta ALTO SECRETO) y una o más categorías o compartimentos de seguridad (CONTABILIDAD, VENTAS, etc.). En este tipo de sistemas, todas las decisiones de seguridad las impone el sistema, comparando las etiquetas del usuario frente al recurso accedido, siguiendo un modelo matemático (Bell-LaPadula, 1973). Los criterios de seguridad TCSEC correspondientes al nivel de seguridad B1 o superior incluyen este modelo.

El modelo DAC se ha venido usando profusamente en sistemas operativos de propósito general con clasificación de seguridad TCSEC nivel C1 o superior, y en virtualmente todos los sistemas de bases de datos, aplicaciones y sistemas de comunicaciones de propósito comercial.

El modelo MAC no ha salido habitualmente del entorno militar, donde la clasificación de la información es lo más relevante.

Los modelos DAC y MAC son inadecuados para cubrir las necesidades de la mayor parte de las organizaciones. El modelo DAC es demasiado débil para controlar el acceso a los recursos de información de forma efectiva, en tanto que el MAC es demasiado rígido. Desde los 80 se ha propuesto el modelo de control de accesos basado en roles (RBAC), como intento de unificar los modelos clásicos DAC y MAC, consiguiendo un sistema donde el sistema impone el control de accesos, pero sin las restricciones rígidas impuestas por las etiquetas de seguridad.

Básicamente, un rol establece un nivel de indirección entre los usuarios y los derechos de acceso, a través de un par de relaciones: asignación de roles a usuarios, y asignación de permisos y privilegios a roles. Las políticas de control de accesos basado en roles regulan el acceso de los usuarios a la información en términos de sus actividades y funciones de trabajo, representándose así de forma natural la estructura de las organizaciones.

Uno de los problemas más apremiantes en la gestión de grandes sistemas de información heterogéneos es la complejidad de la administración de seguridad. La aproximación RBAC, intuitivamente, modela de forma natural la estructura de autorización en las organizaciones del mundo real, facilitando las tareas administrativas al separar la asignación de individuos a funciones o perfiles de trabajo, y la definición de políticas de acceso (definición de roles en términos de lo que pueden hacer en el sistema). Permiten asimismo la construcción jerárquica de estas políticas de acceso, por herencia o especialización. Así, la política de control de accesos para un supervisor de planta puede ser una especialización de la del operador de planta. Por ello, la tecnología RBAC tiene el potencial de reducir la complejidad y el coste de la administración de seguridad en estos entornos heterogéneos.

Además, dada la alta integración entre los roles y las responsabilidades de los usuarios, pueden seguirse los principios del ***mínimo privilegio*** y de la separación de responsabilidades. Estos principios son vitales para alcanzar el objetivo de integridad, al requerir que a un usuario no se le otorguen mayores privilegios que los necesarios para efectuar su trabajo, y que para completar una transacción de cierta seguridad (por ejemplo, la autorización de un pago) se requiera la culminación de una cadena de transacciones simples por más de un usuario.

El modelo RBAC es hoy día ubicuo: desde sistemas de base de datos relacionales, pasando por sistemas operativos de red, firewalls, productos de seguridad Mainframe y entornos abiertos, sistemas de Sign-On único y de seguridad web...

La industria ha venido usando otros modelos de control de accesos, como complemento de estos tres básicos, como el modelo de capacidades, en donde parte de las decisiones de autorización se toman a partir de "capacidades", "derechos efectivos" o atributos de privilegio, contenidos en las credenciales que un usuario adquiere durante la autenticación. DCE, por ejemplo, incorpora este modelo en su servicio de seguridad. A menudo estos atributos de autorización se integran en un objeto conocido como "Certificado de Atributos de Privilegio" (PAC, por sus siglas en inglés). Un PAC es como una acreditación de visitante: Se obtiene tras prueba de identidad, está emitida

por una autoridad en la que se confía, no identifica al individuo pero sí lo categoriza (Ej. "Visitante"), es de duración limitada, y no encierra en sí mismo información de privilegios o permisos (qué puertas puedo franquear, por ejemplo).

Escenificación Escrita de la Historia del Control de Acceso

La "Edad de Piedra": Sistemas centralizados

Érase una vez una época en la que todo era mucho más simple. El Mainframe era la pieza angular de los sistemas de información (algo que, en muchos casos, sigue teniendo vigencia). Ya en 1976 IBM lanza RACF (Resource Access Control Facility), sistema de seguridad que, pese a su espartana simplicidad inicial, evolucionó, revolucionando la concepción sobre los sistemas de control de accesos. Casi al tiempo, surgieron otros productos de seguridad análogos, como CA-ACF2, TopSecret, y un largo etcétera.

Unix se encontraba en sus balbucientes inicios. Su seguridad se basaba en el modelo DAC simple, donde en esencia los recursos podían leerse, escribirse y ejecutarse, y donde se concedían permisos para el propietario, para el grupo y para el resto de usuarios del sistema.

La "Edad de Bronce" Sistemas distribuidos

Y es cuando hace su aparición la informática distribuida. Se empieza a hablar de cliente/servidor. Se introduce el concepto de Middleware transaccional. Las bases de datos relacionales implementan controles de acceso a datos basados en perfiles de usuario o roles, con sistemas bastante sofisticados. Aparece DCE como plataforma de informática distribuida, donde el servicio de seguridad se basa en una autenticación distribuida (Kerberos) y la emisión de certificados de atributos de privilegio (tickets que encierran los atributos de autorización de un principal DCE).

DCE, pese a su elegante diseño en cuanto a seguridad, no cobra la difusión que merece (debido a su fama de complejo). En informática distribuida, estamos en el punto de no retorno: Hay ya demasiados sistemas de seguridad heterogéneos como para pensar en integrarlos en una única infraestructura de seguridad, sea DCE o no.

Aparecen en la industria soluciones que prometen, además de facilidades como Sign-On único o administración centralizada de la seguridad, un primer control de accesos a las aplicaciones corporativas. Estos sistemas, que se interponen entre los usuarios y los sistemas de seguridad de los diferentes componentes de una aplicación distribuida, aportan más a la facilidad de administración de seguridad y la conveniencia de los usuarios (lo cual no es poco) que a la seguridad en sí misma. Aparecen perfiles específicos del estándar GSS-API, como, respuesta europea a DCE, que a menudo conforman la infraestructura en la que se apoyan este tipo de productos.

La gestión distribuida de permisos cobra relevancia. Los datos de control de accesos (permisos, ACLs) se almacenan localmente, pero deben ser gestionados de forma centralizada. Aparecen una nueva clase de herramientas, directamente encaminadas no ya a proporcionar seguridad, sino a facilitar la administración de los múltiples sistemas de seguridad existentes.

Y la actual "Edad de Hierro": Las PKI

Entonces surge la Internet, las arquitecturas de objetos y componentes distribuidas (**CORBA, EJB, DCOM**). El lenguaje Java cobra un éxito sin precedentes por su promesa de lenguaje independiente de plataforma. Todas estas plataformas ofrecen sistemas de seguridad que recogen la experiencia previa. Así, el servicio de seguridad CORBA se basa en un paradigma que se parece sospechosamente al de DCE. El modelo de seguridad Java, en cuanto a autorización, se basa en un modelo de dominios protegidos, donde los permisos de asignan en el sistema local para un dominio que agrupa las clases Java cargadas desde unos determinados sitios y firmados por unos determinados individuos.

Pero lo más visible y novedoso en el presente quizá sea el fenómeno PKI. Ya no es posible controlar un grupo más o menos numeroso de empleados: en los negocios electrónicos se requiere controlar la seguridad en los accesos de usuarios bajo los que no se tiene control administrativo, como proveedores, clientes, consumidores, etc. La PKI es, sobre todo, infraestructura, que las aplicaciones tanto orientadas a Internet como orientadas al "mundo interior" (bases de datos, sistemas operativos) están tan sólo comenzando a aprovechar.

Bien, ya tenemos una PKI, surge una interrogante, ¿Qué aporta de cara al control de accesos? La tecnología de clave pública ofrece grandes ventajas en cuanto a autenticación fuerte de usuarios (su combinación con dispositivos como las tarjetas inteligentes, de hecho, constituye el estado del arte en cuanto a autenticación de usuarios). Protocolos como SSL o IPSEC se apoyan en la PKI para ofrecer servicios añadidos de confidencialidad e integridad en las comunicaciones. La cuestión es: ¿tienen las PKI algo que ofrecer en cuanto al control de accesos?

Los certificados de atributos

Los certificados de atributos no son sino la respuesta de clave pública a los "certificados de atributos de privilegio" usados en el pasado. Los certificados de clave pública X.509 proporcionan evidencia de la identidad de una persona (aunque incluso esto puede ponerse en cuestión). Pero en entornos de comercio electrónico, se precisa más información que la mera identidad, en especial cuando las partes involucradas en una transacción no han tenido contacto previo. En este caso, la información sobre los atributos de privilegio de una persona (por ejemplo, su capacidad de firmar un contrato, o su límite de crédito) es mucho más relevante que su mera identidad.

X.509 v3 introdujo el útil concepto de extensión. Lo más lógico pareció en ese momento, añadir al certificado de identidad extensiones que recogieran estos atributos. Pero es como mezclar agua y aceite. Los atributos de privilegio cambian mucho más a menudo que la identidad de los individuos, lo cual obliga a revocar el certificado antiguo y emitir uno nuevo. Y la validación del certificado, que ya era tema delicado por las revocaciones debidas, se convierte en esta situación en tema trascendental (evidentes cuando se otorga un certificado al director de compras, y a continuación se le despide). Las extensiones dedicadas al control de accesos son propietarias, y dificultan la interoperabilidad. Y, por si fuera poco, tenemos un problema de jurisdicción: el certificado de identidad podrá ser emitido por una autoridad de certificación que dé fe de la identidad del usuario (como Verisign); pero, ¿puede la Verisign erigirse como depositaria de la concesión de privilegios de utilización de los recursos de información de mi empresa X?

La respuesta a estas dificultades es inmediata: ¿Por qué no dividir el certificado X.509 en dos, uno para la información de identidad, certificado de identidad, y el otro para la información ligada con el control de accesos, certificado de atributos? Esto simplifica el proceso de emisión y,

si los certificados de atributos se emiten con una duración limitada, eliminar en ciertas circunstancias el problema de revocación. Si los certificados de atributos duran un día, o una hora, podría no ser necesaria su revocación: simplemente expiran.

¿Cuáles pueden ser los atributos encerrados en un certificado de esta clase? Roles, grupos, identidades de acceso y/o auditoria (para aplicaciones de sign-on único), restricciones. Por ejemplo, un atributo puede expresar el límite de crédito concedido a un determinado suscriptor a un servicio de tienda virtual. Las posibilidades son ilimitadas. Podemos contemplar así los certificados de atributos como un "manojo de llaves" que se añaden a un certificado de identidad ("pasaporte digital") para abrir determinadas puertas.

Ejemplos de aplicaciones de esta tecnología son:

Servicios de suscripción ("pago por evento"). Los usuarios se registrarían de forma gratuita, pero sólo obtendrían un certificado de atributos tras el pago de una cuota de suscripción. La duración del certificado correspondería a la de disfrute del servicio subscrito.

Control de accesos basado en roles a servicios en red cuya autenticación se realiza usando una PKI (Web, FTP, SMTP, etc.). El protocolo SSL autentica al cliente; un certificado de atributos permite al servicio determinar qué puede hacer el usuario dentro del mismo.

Control de accesos a redes privadas virtuales (VPN). Piénsese en los usuarios "itinerantes" de una gran organización. En vez de mantener controles de acceso replicados en cada puerta de acceso VPN, mediante un servicio centralizado se concede al usuario un certificado de atributos que le permita el acceso a través del puerto de acceso.

Sign-on único basado en autenticación fuerte mediante tarjeta inteligente. Las credenciales necesarias para el acceso a aplicaciones (usuario / password) pueden ir cifradas en el certificado de atributos concedido al usuario tras el proceso de autenticación basado en la posesión de la clave privada en tarjeta inteligente.

Certificación cruzada y traducción de políticas

Otro medio de controlar el acceso a servicios de alto nivel mediante una PKI lo tenemos en el concepto de "políticas de utilización" y "traducción de políticas". A menudo la autoridad de certificación emisora sí puede hacer establecer condiciones sobre la utilización de las claves y el certificado emitido, a través de las extensiones de política de uso. El banco X puede emitir un certificado para un cliente, y establecer a nivel de certificado de identidad el tipo de servicio al que el cliente tiene derecho ("tarjeta oro" y "superlibretón", por ejemplo). Esto no es tan flexible como los certificados de atributos, pero puede ser perfectamente aprovechable bajo ciertas condiciones.

Imaginemos que el banco X firma un acuerdo de colaboración (o se fusiona, según cierta moda imperante) con el banco Y, que tiene su propio servicio de banca virtual y emite certificados para dos tipos de servicios, denominados "tarjeta platino' y "superlibretónplus". Ambos bancos emiten certificados cruzados que, algún día, las aplicaciones de banca virtual de cada entidad aprovecharán para otorgar su confianza a los poseedores de certificados emitidos por el otro banco. El problema es: ¿Equivale el usuario de una "tarjeta oro" del banco X a un usuario de la "tarjeta platino" del banco Y?

Este tipo de relaciones entre políticas de uso de certificados pueden hacerse explícitas en los certificados cruzados mediante las extensiones estándar de traducción de políticas (policy mapping). Así, nuestros bancos podrán equipar las respectivas tarjetas oro y platino, y no hacer lo propio con las 'superlibretón' y 'superlibretónplus'. ¿Aprovecharán algún día las aplicaciones esta información?

Autorización en autoridades de validación

Sin validación (comprobación de que un certificado está firmado por una cadena de autoridades entre las que se haya una en la que se confía, que el certificado esté dentro de su período de validez, y que no haya sido revocado), el receptor de un certificado digital en una transacción de comercio electrónico sólo podrá confiar parcialmente en la misma. En el mundo de las tarjetas de crédito, cuando uno se haya enfrente del terminal punto de venta, tiene lugar una validación en línea de la tarjeta, en la que la disponibilidad es esencial (la emisión de la tarjeta, por contra, es una función fuera de línea: poco importa que la tarjeta me llegue un día antes o un día después). Lo mismo ocurre con los certificados digitales: su emisión suele ser un proceso offline, mientras que su validación es un proceso online.

Para entornos de producción, o para aplicaciones de alta sensibilidad (no ya el pago del alquiler de una película en el videoclub de la esquina), una PKI sin validación es fundamentalmente incompleta e insegura. Para proporcionar esta función surgen las Autoridades de Validación (CA), cuya competencia es la validación de los certificados, no su emisión; y añadir a la validación "intrínseca" (firma correcta por una cadena de autoridades en la que se confía, validez temporal, no revocación/suspensión) una validación extrínseca: A partir de información de política de accesos se establece si un certificado es válido si, además de ser válido intrínsecamente, verifica además las condiciones de la política de accesos impuestas por la organización. Y si el certificado no es "válido", la aplicación se niega a ofrecer el servicio solicitado.

El imponer un punto de control de accesos en el proceso de validación tiene sustanciales ventajas: La validación es un proceso independiente de la emisión del certificado, de forma que los parámetros para la autorización pueden modificarse posteriormente, tras la emisión del certificado: Cambios en la política de accesos no requieren la revocación del certificado.

En Conclusión, lejos quedan ya los tiempos en los que todo se reducía a asegurar el Mainframe corporativo. La evolución de los sistemas de control de accesos ha ido pareja con la de los sistemas de información. Vivimos en un mundo, dirigido a los sistemas ya no distribuidos, y se hacen necesarios los modelos de control de accesos para estos nuevos sistemas. La tecnología de clave pública, como viene siendo habitual, tiene mucho que decir al respecto. [12]

El control de acceso constituye una poderosa herramienta para proteger la entrada a un Web completa o sólo a ciertos directorios concretos e incluso a archivos o programas individuales. Este control consta generalmente de dos pasos:

En primer lugar, la autenticación, que identifica al usuario o a la máquina que trata de acceder a los recursos, protegidos o no.

En segundo lugar, procede la cesión de derechos, es decir, la autorización, que dota al usuario de privilegios para poder efectuar ciertas operaciones con los datos protegidos, tales como leerlos,

modificarlos, crearlos, etc. Por defecto, todas las páginas y servicios de un servidor web se pueden acceder anónimamente, es decir, sin necesidad de identificarse ante el servidor y sin ningún tipo de restricción. [11].

ANEXO H ¿INTELIGENCIA O BIOMETRÍA?

La ventaja de las tarjetas inteligentes es que almacenan las claves privadas de los usuarios para el uso de sistemas de encriptación de clave pública o PKI. Estos sistemas utilizan cifrado asimétrico, además de certificados digitales para lograr transacciones seguras en Internet. Por este medio, la clave pública queda a disposición de todas las personas que intervienen en transacciones de comercio electrónico, como bancos o comerciantes a través de Internet, pero la clave privada sólo la sabe el usuario y la utiliza para descifrar un documento.

Los certificados digitales son documentos electrónicos, emitidos por una tercera parte de confianza, que identifican al propietario de una clave privada.

Desde luego, las tarjetas inteligentes tienen sus desventajas: pueden perderse o ser robadas o que el usuario olvide el NIP. Por esto, algunas compañías están recurriendo a **dispositivos biométricos** como medida de seguridad más automática. Para autenticar y dar acceso a datos seguros, los dispositivos biométricos leen rasgos únicos de la anatomía de una persona (por lo general una huella digital, pero a veces el iris del ojo).

El software de identificación uno-a-uno es más veloz y se utiliza principalmente en oficinas donde una persona utiliza la misma PC todo el día, mientras que el sistema "de uno a muchos" conviene en oficinas donde varias personas pueden tener acceso al mismo sistema.

La queja que expresan las compañías que han comenzado a utilizar sistemas de identificación de huellas digitales es que los usuarios se resisten a la tecnología, debido a que les preocupa ser excesivamente supervisados o porque eso de las huellas digitales les resulta incómodo.

ANEXO I ESTRATEGIAS DE SEGURIDAD

METODOLOGÍAS PARA LA DEFINICIÓN DE ESTRATEGIAS DE SEGURIDAD

Estrategias de Seguridad

Una metodología para definir políticas y controles de seguridad

..Predicción de ataques / Valoración de riesgos

------ Para cada tipo de amenaza (e.g. ataque malicioso)

------- Para cada tipo o método de ataque (e.g. virus)

---------**Estrategia Proactiva**

---------- Predicción de daños

---------- Determinación de Vulnerabilidades

---------- Minimizar Vulnerabilidades

Implementar plan y desarrollar políticas y controles de seguridad

---------- Generar planes de contingencia

----------**Estrategia Reactiva**

---------- Valoración de daños

---------- Determinar causas del daño

---------- Reparación de daños

Implementar plan y desarrollar políticas y controles de seguridad

---------- Documentos

---------- Implementar planes de contingencia

..Evaluación de Resultados / Realización de Simulación

..Evaluación de las políticas de seguridad

..Ajuste de acuerdo a la política

Fig. I1 Metodología para la definición de estrategias de seguridad [9]

Se presentan ahora cuatro ejemplos de posibles ataques y cuál sería una metodología de estrategia a seguir para clasificarlo y tratarlo, o más bien prevenirlo.

Ejemplo 1: *Ataque No Malicioso*

Juan Pérez desea respaldar en el servidor, la información de 6.4 GB que tiene en su disco duro, el cual tiene además un espacio libre de 6.5 GB, al querer hacerlo, surge un mensaje de advertencia, acerca de la no capacidad del servidor para realizar dicha operación. Abajo se muestra la metodología que debería seguirse, antes de realizar el respaldo.

Estrategias de Seguridad

Una metodología para definir políticas y controles de seguridad

..Predicción de ataque

Negación de servicios cuando el usuario abuse de los recursos

------ Para cada tipo de amenaza

Empleado no malicioso: Juan Pérez

------- Para cada tipo o método de ataque

Juan no tiene motivo para desestabilizar los servicios, por lo que no utiliza método alguno

---------**Estrategia Proactiva**

---------- Predicción de daños

Sino existe espacio en el disco duro, posiblemente cause baja en la productividad

---------- Determinación de Vulnerabilidades

No espacio en disco y falta de entrenamiento al personal

---------- Minimizar Vulnerabilidades

Implementar espacio en disco y generar conciencia de la necesidad de conocimiento en seguridad

---------- Planes de contingencia

Posiblemente tener un servidor en "standby"

----------**Estrategia Reactiva**

---------- Valoración de daños

Baja en la producción

---------- Determinar causas del daño

Falla en el servidor, cuando el usuario quiso copiar todo el disco dentro de una carpeta

---------- Reparación de daños

Borrar la información dentro de la carpeta del usuario

---------- Documentos

---------- Planes de contingencia

Implementar un servidor "standby" mientras que el servidor de producción esta el linea

..Evaluación de Resultados: baja en el rendimiento

..Evaluación de las políticas de seguridad: revisión de las políticas de asignación de recursos, así como de las políticas de capacitación al personal

..Ajuste de acuerdo a la política

Fig. I2 Ataque No Malicioso [9]

Ejemplo 2: *Amenaza Maliciosa* (atacante externo)

Juan Pérez como Hobbie introduce virus al sistema, dicho virus desestabiliza los sistemas de correo electrónico. En este caso la metodología a seguir sería:

Estrategias de Seguridad

Una metodología para definir políticas y controles de seguridad

..Predicción de ataque

Si el servicio de e-mail es negado cuando se utiliza para e-commerce y otras funciones financieras

------ Para cada tipo de amenaza

Atacante malicioso: Juan Pérez

------- Para cada tipo o método de ataque

Virus por e-mail

---------**Estrategia Proactiva**

---------- Predicción de daños

Si se detiene el servicio de e-mail, se detiene la producción

---------- Determinación de Vulnerabilidades

No existe escaneo de virus o no existe antivirus en la base de datos para realizar dicho escaneo

---------- Minimizar Vulnerabilidades

Implementar escaneo de virus o actualizar el antivirus contacto con el fabricante

---------- Planes de contingencia

Posiblemente tener un servidor en "standby"

----------**Estrategia Reactiva**

---------- Valoración de daños

Baja en la producción

---------- Determinar causas del daño

Colisión en el servidor de correo, debido a un virus

---------- Reparación de daños

Implementar escaneo de virus, actualización del antivirus e informar al personal

---------- Documentos

---------- Planes de contingencia

Implementar un servidor "standby" mientras que el servidor de producción esta el línea

..Evaluación de Rsultados: baja en el rendimiento

..Evaluación de las políticas de seguridad: revisión de las políticas de escaneo y detección de virus

..Ajuste de acuerdo a la política

Fig. I3 Ataque Externo [9]

Ejemplo 3: *Amenaza Maliciosa* (atacante interno)

Juan Camaney es un empleado de una empresa elaboradora de naves espaciales, y la competencia le ofrece una gran suma, por robar el último diseño. Como él no tiene los derechos de acceso necesarios, se hace pasar por administrador y pide vía telefónica a un usuario con los derechos suficientes su nombre de usuario y contraseña.

Estrategias de Seguridad

Una metodología para definir políticas y controles de seguridad

..Predicción de ataque

Mediante el uso de ingeniería social, la información puede ser robada

------ Para cada tipo de amenaza

Atacante: empleado malicioso Juan Camaney

------- Para cada tipo o método de ataque

Ingeniería Social

---------**Estrategia Proactiva**

---------- Predicción de daños

Si la información es robada, existe disminución en las ganancias

---------- Determinación de Vulnerabilidades

No conciencia de seguridad en los empleados

---------- Minimizar Vulnerabilidades

Implementar capacitación para la generación de conciencia de seguridad en los empleados

---------- Planes de contingencia

---------**Estrategia Reactiva**

---------- Valoración de daños

Baja en las ganancias e información sensible

---------- Determinar causas del daño

Dar a conocer nombres de usuario y contraseñas de los empleados

---------- Reparación de daños

Implementar capacitación en cuanto a seguridad, así como conocer a las autoridades relevantes de la institución

---------- Documentos

---------- Planes de contingencia

Implementar un servidor "standby" mientras que el servidor de producción esta el línea

..Evaluación de Resultados: Información sensible puede ser robada debido a la pobre conciencia de seguridad, provocando además baja en los ingresos

..Evaluación de las políticas de seguridad: revisión de las políticas de capacitación en seguridad computacional

..Ajuste de acuerdo a la política

Fig. I4 Ataque Interno [9]

Ejemplo 4: *Amenaza No maliciosa* (desastre natural)

La compañía XYZ no tiene sistema de protección y detección de incendios, en el área donde se encuentra el servidor. Alguna ocasión el encargado del área, olvida apagar los "aparatos" del aire acondicionado, los cuales se sobrecalientas originando un incendio que acaba hasta con el último detalle.

Estrategias de Seguridad

Una metodología para definir políticas y controles de seguridad

..Predicción de ataque

Fuego

------Para cada tipo de amenaza

Desastre natural: fuego

--------Para cada tipo o método de ataque

Sin método

--------- **Estrategia Proactiva**

----------Predicción de daños

EL fuego puede causar pérdida de la información, hardware y en la producción.

----------Determinación de Vulnerabilidades

No existencia de sistemas de detección y protección de incendios o mal funcionamiento de ellos

----------Minimizar Vulnerabilidades

Implementar políticas para la detección y prevención de incendios, así como mantenimiento regular de los mismos

----------Planes de contingencia

Generar respaldos seguros sobre funcionamiento y almacenarlos fuera del sitio y de ser posible tener hardware de repuesto

--------- Estrategia Reactiva

----------Valoración de daños

Baja en la información, hardware y productividad

----------Determinar causas del daño

Fuego, causado por bloqueo en el aire acondicionado

----------Reparación de daños

Reincorporación de los servidores y restauración de los respaldos más recientes

----------Documentos

----------Planes de contingencia

...Evaluación de Resultados: Los incendios pueden también tener efectos catastróficos en los sistemas de cómputo

..Evaluación de las políticas de seguridad: revisión o implantación de políticas de detección de incendios

..Ajuste de acuerdo a la política

Fig. 15 Desastre Natural [9]

Planeación de la seguridad

Visión general. La parte más importante del desarrollo es la planeación. La planeación de la seguridad involucra desarrollar políticas de seguridad e implementación de controles para prevenir que los riesgos en las computadoras se hagan realidad.

El bosquejo que a continuación se presenta es una simple guía. Cada organización es diferente y necesitará planear y crear políticas basadas en metas y necesidades individuales.

Las herramientas y tecnología son enfocadas sobre características. Este énfasis permite que los funcionarios y administradores de la seguridad elijan que herramientas o técnicas con las que mejor satisfagan las necesidades de seguridad de la organización.

Valoración básica de los riesgos. La valoración de los riesgos es una parte muy importante en la planeación de la seguridad en las computadoras. La valoración del riesgo proporciona el inició de la implementación de la planeación de seguridad para proteger los activos contra diversos ataques. Existen tres preguntas básicas que debemos respondernos para mejorar la seguridad de un sistema:

- ¿Qué activos dentro de la organización requieren de protección?
- ¿Cuáles son los riesgos para cada unos de esos activos?
- ¿Qué tanto tiempo, esfuerzo y dinero estamos dispuestos a gastar en ampliar y obtener una nueva y adecuada protección contra esas amenazas?

No se pueden proteger activos, sino se conoce contra que protegerlos.

Las computadoras necesitan protección contra riesgos, pero ¿qué tipo de riesgos?. En términos simples, un riesgo es realizado cuando una amenaza toma ventaja de una vulnerabilidad para dañar nuestro sistema. Después de conocer los riesgos, se pueden crear políticas y planes para reducir esos riesgos.

Existen muchas formas de identificar el riesgo de los activos. Una forma de obtener los riesgos, es del personal que labora dentro de la organización, tal vez tendríamos una gran lista de activos y riesgos para ellos, incluso ideas geniales. Esto además incrementaría la conciencia de seguridad dentro de la organización.

Los riesgos, como se menciono anteriormente, pueden venir de tres maneras: riesgos por desastres naturales, riesgos intencionales y riesgos no intencionales. La figura 3.3.12 los muestra.

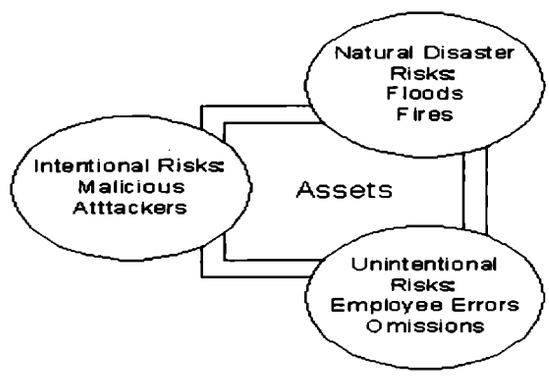


Fig. I6 Tipos de Riesgos [14]

El proceso de valoración de los riesgos en el flujo de estrategias de seguridad, puede ser dividido en los siguientes pasos:

- Identificar los activos a proteger y su valor
- Identificar los riesgos de cada activo
- Determinar la categoría de la causa del riego (riesgo por desastre natural, intencional o no intencional)
- Identificar los métodos, herramientas o técnicas que la amenaza utiliza.

Después de evaluar los riesgos, el siguiente paso es la planeación proactiva., lo cual implica desarrollo de políticas y controles de seguridad, así como implementación de herramientas y técnicas que ayuden a la seguridad. El plan proactivo es desarrollado para proteger activos, prevenir ataques y errores de empleados. El plan reactivo es un plan de contingencia para cuando, el plan proactivo ha fallado.

Estrategias de Seguridad

-------Para cada tipo de amenaza (e.g. ataque malicioso)

--------Para cada tipo o método de ataque (e.g. virus)

--------- **Planeación Proactiva**

----------Desarrollo de políticas y controles de seguridad

----------Implementación de herramientas y técnicas para ayudar a la seguridad

Acceso seguro, datos seguros, código seguro

Tecnologías para asegurar la conectividad en redes

Herramientas para detección

----------Tecnologías para "levantar" los sistemas en el momento de la falla.

--------- **Estrategia Reactiva**

----------Implementar planes de contingencia

.. Evaluación de Resultados / Realización de Simulación

.. Evaluación de las políticas de seguridad

.. Ajuste de acuerdo a la política

Fig. I7 Planeación de la Seguridad [9]

Es muy importante realizar una buena valoración de los activos, así como sus correspondientes políticas y controles de seguridad. La Fig. 3.3.14 muestra la relación entre una buena valoración de activos y unos buenos controles y políticas de seguridad.

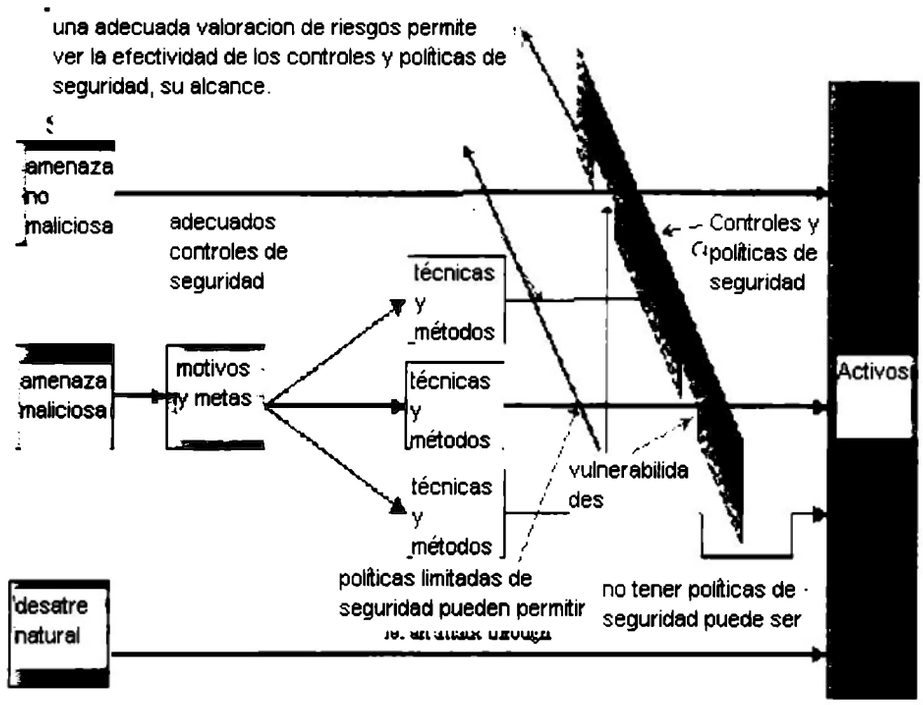


Fig. 18 Relación entre valoración de activos y controles y políticas de seguridad [9]

ANEXO J DOCUMENTOS DE JAGUAR

JAGUAR CTS GETTING STARTED

Terminología y conceptos

Una aplicación Jaguar consiste de uno más paquetes y aplicaciones clientes o applet. Los paquetes están formados de componentes y los componentes a su vez están compuestos de uno o más métodos.

- Un Host Jaguar, puede controlar y ejecutar componentes tales como objetos programables Actives, JavaBeans o componentes CORBA. En el ambiente de Jaguar un componente es simplemente un objeto de aplicación que consiste de uno o más métodos. Los componentes Jaguar, típicamente ejecutan asuntos lógicos, acceden a recursos de datos y regresan resultados al cliente. Los clientes (applets) crean una instancia de un componente y ejecutan métodos asociados con esos componentes. Los componentes se ejecutan solo dentro de un servidor Jaguar.
- Un paquete es una colección de componentes que trabajan de manera conjunta para proporcionar un servicio o algunos aspectos de los asuntos lógicos de la aplicación. Un paquete define un "límite de confianza" dentro del cual, los componentes pueden fácilmente comunicarse. Cada paquete actúa como una unidad de distribución, agrupando recursos de aplicación para facilitar el despliegue y control.

Jaguar soporta los siguientes tipos de componentes:

- ActiveX
- CORBA
- C++
- Java

- Un stub es una clase de Java o C++ generado por el administrador de Jaguar y actúa como un objeto proxy para un componente de Jaguar. Un stub es compilado y ligado con sus applets de Java o aplicaciones cliente. Un stub se comunica con Jaguar para "instanciar" e invocar un método o un componente en la capa intermedia. Los Stubs generan un componente de Jaguar de manera remota que aparece de manera local ante el cliente.
- Un skeleton actúa como la interfaz entre el entorno de ejecución de Jaguar y el código del usuario que implementa el método. Los skeletons son compilados y ligados con cada uno de los componentes y en tiempo de ejecución habilitan a Jaguar para que localice o invoque un método apropiado.
- Jaguar transparentemente mantiene una sesión entre una aplicación cliente y el servidor de Jaguar. Diferente a un escenario típico http, donde una nueva conexión es generada para cada petición y respuesta, sesiones que permite un browser para mantener una conexión con el servidor a través de un ciclo múltiple de petición-respuesta.

Es posible desarrollar y distribuir una aplicación Jaguar a través de la red.

Jaguar implementa una arquitectura de cómputo de capa triple o multicapa. En este modelo tres distintos elementos trabajan conjuntamente para proporcionar a usuarios acceso a datos:

- Applet o aplicación en el sitio del Cliente
- Componente de capa intermedia
- Base de datos de respaldo

Los applets de Java son descargados por los clientes, los cuales "instancian" componentes sobre el servidor. Las aplicaciones cliente son instaladas en las máquinas cliente, de las cuales ellos también instancian componentes en el servidor.

Un applet o aplicación controla la presentación e interacción con el usuario final. Los componentes de la capa intermedia, los cuales ejecutan Jaguar, manipulan la mayoría del procesamiento de aplicación.

Finalmente, las bases de datos almacenan, administran y procesan los datos.

Si el cliente es un applet, los usuarios encuentran un inicio de aplicaciones a partir de páginas tradicionales HTML. En lugar de simplemente cargar una página estática, Jaguar descarga un applet ejecutable para browser de manera individual. Si el cliente es una aplicación ya instalada, el usuario inicia la aplicación a partir de su máquina. Los clientes se comunican directamente con un componente de aplicación, ejecutándose en la capa intermedia. El servidor de componentes accede datos de una o más bases de datos, aplicando negocios lógicos y regresando resultados al applet del cliente para su presentación.

Cuando un objeto proxy es generado en un applet cliente, instancia su componente correspondiente, registrado en el servidor de Jaguar. En el sitio del servidor un componente es instanciado en respuesta a una petición de un objeto proxy ejecutado en el ambiente del cliente. Un método en un componente es ejecutado cuando es invocado por un objeto proxy en el applet del cliente.

Existen tres pasos básicos involucrados en la creación y despliegue de una aplicación Jaguar que emplea applet Java como un cliente.

Para generar y desplegar una aplicación Jaguar:

- definir paquetes, componentes y métodos el administrador de Jaguar es una interfaz GUI de Jaguar que permite definir de manera fácil los paquetes. Componentes y métodos que los clientes de Jaguar utilizan para ejecutar una aplicación. El administrador de Jaguar genera:
    - Archivos *Stub* en el cliente. Los stubs contienen interfaz de información utilizada por el cliente para invocar métodos de componentes de Jaguar.
    - Archivos *Skeleton* en el Servidor. Los Skeletons proporcionan la interfaz de información de cada método de un componente.
- Una vez que se han generado los stubs y los skeletons, se escriben las clases de Java que una vez ligadas con los archivos stub, forman la base de los applets descargables.

- Desarrollar los componentes en el servidor que se conectan con los skeletons para formar negocios lógicos de tu servlet. Jaguar soporta un gran número de las herramientas de ambientes de desarrollo integrados (IDE) disponibles hoy día.
- Desplegando la aplicación. Se pueden registrar componentes en cualquier servidor Jaguar. Debido a que Jaguar CTS es también un servidor Web, se puede escribir una página HTML para tú applet e instalarlo sobre Jaguar.

Ambiente de ejecución de Jaguar

Una aplicación típica de Jaguar tiene un applet o página HTML asociada con él. Una vez que se ha construido y establecido una aplicación, se ejecuta de la forma siguiente:

- Jaguar recibe una petición HTTP y descarga la página HTML o applet de la petición. Incluidos con el applet están los Java Stubs, los cuales a través de un proxy, instancian componentes e invocan métodos en esos componentes.
- El cliente establece una sesión con Jaguar. La sesión es diferente a una conexión http, pues permite al cliente y Jaguar mantener una conexión durante la transacción.
- El cliente genera una instancia del componente a través del proxy en el cliente. El proxy se utiliza dependiendo del tipo de componente a ser instanciado. Jaguar valida comparando con la lista de acceso a componentes. Si el usuario es validado, el despachador checa la localidad y estado del componente y genera la instancia.
- El cliente invoca el negocio lógico del componente mediante la ejecución de sus métodos.
- Los componentes pueden interactuar con bases de datos remotas
- Jaguar regresa los resultados desde la base de datos al cliente.
- El cliente indica que ha sido completada la operación. Jaguar destruye la instancia del componente o regresa una señal para futuros instancias del cliente. El cliente se desconecta de Jaguar.

En cuanto a este aspecto, Jaguar, mediante lo que llama: "usuario de jagadmin" tiene un acceso ilimitado al administrador de Jaguar. Para adicionar seguridad, se puede establecer mediante una administración de password para un "user jagadmin" y habilitar autenticación del sistema operativo. Para acceder a esas propiedades:

- desde el administrador de Jaguar, doble clic en el servidor para poder configurarlo.
- seleccionar File y propiedades de servidor
- seleccionar la etiqueta seguridad

En la administración de password, se puede establecer un password de administración para el "jagadmin user" en cada servidor. Solo el jagadmin puede:

- acceder al administración de Jaguar
- poner o quitar el password de jagadmin
- habilitar o deshabilitar la autenticación de usuario

Para fijar el password de administración:

- Seleccionar el conjunto de password jagadmin
- En el diálogo de administrador de password, introducir dos veces el password y pulsar OK. Las convenciones y restricciones del password de administración, son las mismas que para los password de usuario del sistema.

Habilitación de la autenticación del SO: si se selecciona esta opción, Jaguar "mapea" los nombres y password de usuario del sistema operativo al sistema. También se puede suministrar un nombre de usuario y password que es válido para la máquina donde el servidor Jaguar esta corriendo. Por ejemplo, para Unís, se puede utilizar password del servicio de información de la red (NIS) y para NT, se puede utilizar el password de dominio de NT. Los usuarios de NT pueden proporcionar un nombre de dominio como parte de su nombre de usuario; por ejemplo, *domain_name\username*

Para habilitar autenticación en NT:

- Desde NT, inicie el administración de usuario
- seleccionar políticas / user rights
- Pulse en Show Advanced User Rights
- desplazarse sobre la lista de la derecha y seleccionar "actuar como parte del sistema operativo"
- utiliza el botón "Add", para adicionar usuarios y grupos
- Salir y regresar al sistema NT, para habilitar autenticación
- desde Jaguar, seleccionar la opción de "habilitar autenticación de OS", dentro de las opciones de la etiqueta: propiedades de seguridad.

Para habilitar la autenticación OS en Unix, seleccionar la opción de "Habilitar autenticación OS", en la etiqueta Seguridad.

El password para la cuenta de Jagadmin, es siempre definido en el administrador de Jaguar. Aun cuando el Jagadmin es definido como un nombre de usuario en OS y la autenticación del OS sea habilitada, el password definido en el Administrador de Jaguar requiere de "logearse" como Jagadmin.

Habilitación de Usuario y validación de grupos. Si se permite, los nombres de usuario y grupo, son validados contra los nombre de usuario y grupo del sistema operativo antes de adicionarlo a cualquiera de los siguientes fólder:

- Usuario autorizado
- Grupo autorizado
- Usuario excluido
- Grupo excluido

Para habilitar la validación de usuario o grupo, seleccionar la opción de "validación de usuario y grupo" en la etiqueta Seguridad.

JAGUAR CTS SYSTEM ADMINISTRATION GUIDE

Esta sección esta compuesta de los siguientes capítulos:

Capítulo 1: "Configuración de Jaguar" contiene información acerca de la configuración de Jaguar, incluyendo:

- Creación de servidores Jaguar
- Configuración de propiedades de servidor
- Configuración de propiedades http
- Configuración de conexiones caché
- Configuración de servicios de mensaje

Capítulo 2: "Servidor de Nombres Jaguar" contiene información acerca del uso de servicio de nombres de Jaguar para localizar objetos, tales como: paquetes, componentes y servidores en cualquier parte de la red.

Capítulo 3: "Cluster y sincronización de Jaguar" contiene información acerca de la creación de cluster de servidores de Jaguar, los cuales proporcionan alta disponibilidad para los servicios y componentes de Jaguar, así como la sincronización de repositorios desde un servidor primario entre un cluster a otro cluster de servidores, los cuales ejecutan todo lo que se actualiza dentro de los cluster.

Capítulo 4: "Balanceo de Cargas, Sobre fallas y disponibilidad de componentes" contiene información acerca de cómo balancear cargas a en un cluster de servidores de Jaguar y como configurar e implementar sobre fallas de componentes.

Capítulo 5: "Configuración de Seguridad" contiene información acerca de la configuración delas características de SSL para Jaguar, incluyendo:

- Generación de Certificados cliente / servidor
- Asignación de certificados para perfiles de seguridad
- Asignación de perfiles de seguridad para *listeners*

- También incluye información acerca de:
- Cambio de la administración de Password
- Uso de la autenticación del Sistema Operativo
- Asignación de reglas a paquetes y componentes

JAGUAR CTS PROGRAMMER'S GUIDE

Presenta una visión general de las características de Jaguar, diseño de conceptos y la aplicación del desarrollo de procesos, todo esto dividido en los siguientes capítulos:

Capítulo 1: "Introducción al Jaguar CTS" contiene una visión general de la información acerca de la construcción de aplicaciones de Jaguar, clientes y componentes.

Capítulo 2: "Creación de Aplicaciones basadas en componentes" proporciona un nivel de conocimiento del desarrollo típico de un proceso

Capítulo 3: "Asimilación de transacciones y ciclos de vida de componentes" describe como Jaguar manipula transacciones multicomponente y ciclos de vida multicomponente.

Ahora bien, para información sobre desarrollo de componentes con Jaguar, los siguientes capítulos son los más adecuados:

Capitulo 4: "Administración de Aplicaciones y Paquetes con el Administrador de Jaguar" describe como crear aplicaciones y paquetes con el administrador de Jaguar. Esos elementos son requeridos para el desarrollo de componentes para un servidor de Jaguar

Capítulo 5: "Definición de Componentes" describe como definir paquetes y componentes en el administrador de Jaguar y como configurar las propiedades de los componetes

Capítulo 6: "definición de Interfaces de Componentes" describe como crear, ver y editar interfaces de componentes en el administrador de Jaguar.

Para información sobre desarrollo de componentes con Enterprise Java Beans (EJB), los siguientes capítulos:

Capítulo 7:"Visión General de Enterprise JavaBeans" introducción del modelo de componentes EJB

Capítulo 8: "Creación de Componentes Enterprise JavaBean" describe como crear componentes EJB

Capítulo 9: "Creación de Clientes Enterprise JavaBean" describe como implementar un cliente que utiliza interfaces de clientes EJB para llamar métodos de componentes de Jaguar.

Capítulo 10: "Interoperabilidad de Jaguar EJB" describe como llamar componentes que no son EJB desde clientes o componentes EJB, y como llamar componentes EJB desde cliente no EJB.

Capítulo 11: "Creación de aplicaciones cliente" describe como crear y destacar aplicaciones cliente EJB

Para información o desarrollo de componentes utilizando modelos Java / CORBA, los siguientes capítulos:

Capítulo 12: "Creación de Componentes CORBA/Java" contiene información acerca de la construcción de componentes CORBA-Java

Capítulo 13: "Creación de clientes Java, compatibles con CORBA" describe como implementar un cliente que utiliza objetos ORB (Object Request Broker) para llamar a métodos de componentes Jaguar.

Capítulo 14: "Uso de SSL y Conexiones Proxy en Clientes Java" describe como crear conexiones seguras desde una aplicación Cliente en Java

Para información sobre desarrollo de componentes utilizando modelos CORBA/C++, los siguientes capítulos:

Capítulo 15: "Visión general de CORBA C++" describe el soporte de C++ en Jaguar, y explica como Jaguar mapea los tipos de datos de CORBA IDL a los tipos de datos de C++.

Capítulo 16: "Creación de componentes CORBA C++" contiene información acerca de la construcción de componentes C++

Capítulo 17: "Creación de Clientes CORBA C++" describe como desarrollar clientes C++ con conexión a Jaguar

Para información sobre desarrollo de componentes Actives, los siguientes capítulos:

Capítulo 18: "Visión general sobre ActiveX" describe el soporte para Actives de Jaguar, incluyendo como Jaguar mapea los tipos de datos de CORBA IDL a los tipos de datos de Actives

Capítulo19: "Creación de componentes ActivceX" contiene información acerca de la construcción de componentes Actives

Capítulo 20: "Creación de clientes ActiveX" describe como desarrollar clientes que se conectan a Jaguar utilizando el servidor Proxy para clientes de Jaguar

Capítulo 21: "Usando SSL en clientes ActiveX" describe como crear conexiones seguras desde una aplicación Cliente de ActiveX

Para información sobre desarrollo, configuración y ejecución de aplicaciones Web, servlets y servidor de páginas Java, ver los siguientes capítulos:

Capítulo 22: "Creación de aplicaciones Web" describe como definir y configurar aplicaciones Web

Capítulo 23: "Creación de Java Servlets" describe como crear y ejecutar servlets Java en Jaguar

Capítulo 24: "Páginas JavaServer" describe como crear y ejecutar Páginas JavaServer en Jaguar

Capítulo 25: "Configuración de seguridad en aplicaciones Web" describe el modelo Web de seguridad en Jaguar, y como implementar sus propias políticas de seguridad.

Capítulo 26: "Uso de conexiones HTTP y HTTPS en Java" describe como crear cleitnes Java para conectar a Jaguar con cualquier otro servidor de Web utilizando protocolos HTTP y HTTPS

Para información o características avanzadas sobre componentes, los siguientes capítulos:

Capítulo 27: "Envío de conjunto de resultados" describe como enviar conjuntos de resultados desde un método codificado en C, C++ o Java

Capítulo 28: "Uso del Administrador de Conexiones" describe como acceder a conexiones caché desde un método codificado en C, C++ o Java

Capítulo 29: "Administración de la condición de Componentes constantes" describe como crear componentes CORBA o EJB que almacenen información de estado en una base de datos remota

Capítulo 30: "Creación de JavaMail" describe como utilizar el API de JavaMail para acceder a un servidor de mail desde componentes o servlets de Java

Capítulo 31: "Uso de servicio de mensajes" describe como utilizar el servicio de mensajería asíncrona de Jaguar, para implementar eventos o mensajes que manejan aplicaciones lógicas en clientes y componentes.

Capítulo 32: "Uso de administración de Hilos" describe como crear hilos para ejecutar procesamiento asincrónico en componentes Jaguar

Capítulo 33: "Creación de componentes de Servicio" describe como generar componentes que ejecuten servicios de Jaguar

Capítulo 34: "Creación y Uso de Pseudo componentes de Jaguar" describe soporte para pseudo componentes C++ y Java en Jaguar.

JAGUAR CTS API REFERENCE

Esta formado por cuatro capítulos que contiene:

Capítulo 1: "Clases e Interfaces de Java" documentos sobre clases e interfaces de Java en Jaguar. Esta información es necesaria para implementar componentes Java o Clientes Java.

Capítulo 2: "Interfaz de Referencia ActiveX C++ " documentos sobre interfaces ActiveX C++ en Jaguar. Esta información es necesaria para implementar componentes ActiveX utilizando C++.

Capítulo 3: "Interfaz de Referencia Actives IDispatch" documentos sobre interfaces de automatización en ActiveX de Jaguar. Esta información es necesaria para implementar componentes ActiveX utilizando IDes que utilizan automatización Actives tal como PowerBuilder.

Capítulo 4: "Interfaces Cliente ActiveX" documentos sobre las interfaces de clientes ActiveX que utilizan procesos resultado del conjunto retornado por una invocación a un método de Jaguar.

Capítulo 5: "Referencia sobre Rutinas en C" documentos sobre rutinas de librerías en C. para Jaguar. Esta información es necesaria para implementar componentes C. [23]

ANEXO K MODELOS PARA LA DETERMINACIÓN DE AMENAZAS

MODELOS TEÓRICOS PARA LA DETERMINACIÓN DE AMENAZAS

Modelo teórico que puede ser utilizado para determinar las diversas amenazas, objetivos, métodos y vulnerabilidades utilizadas en un ataque.

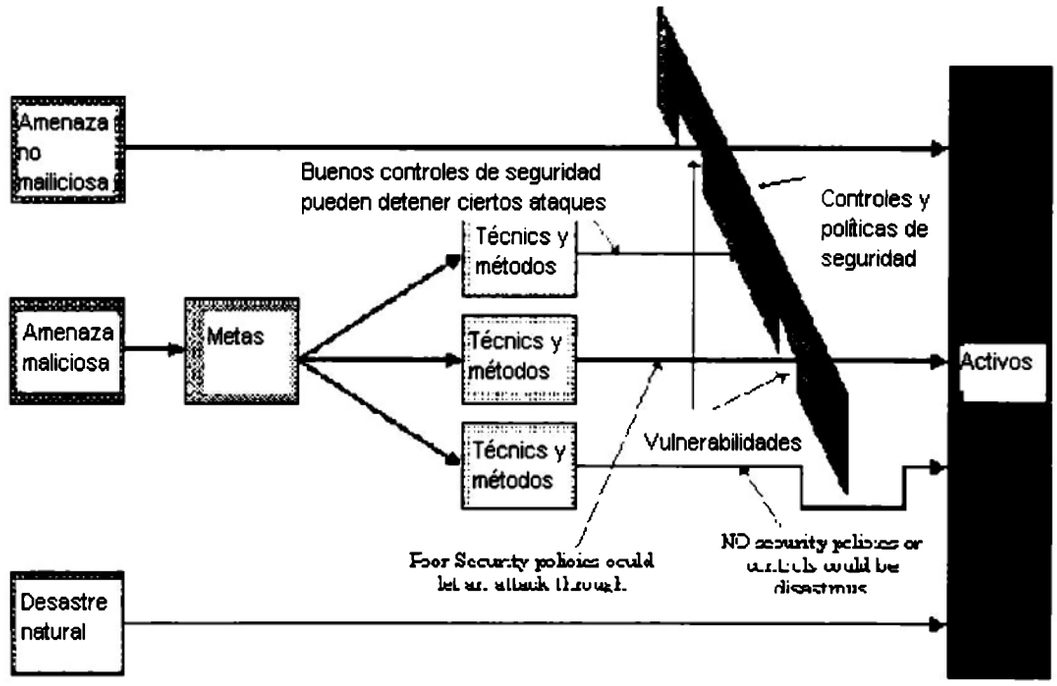


Fig. K1 Aspectos empleados en un Ataque [9]

Algunos de los aspectos mencionados arriba podemos resumirlos en la tabla TK1

Amenazas	Motivos / objetivos	Métodos	Políticas de Seguridad
Desastres No naturales	Negación de servicios	Ingeniería social	Vulnerabilidades
Empleados	Robo	Virus, caballos de Troya, worms	Activos
Atacantes	Alteración	Inf. Duplicada	Información y datos
Ignorantes	Eliminación	Inf. Modificada	Productividad
No-empleados	Generación de trampas	IP falsa	Hardware
Atacantes externos		Bombardeo de correos	Personal
Desastres Naturales		Herramientas de "Hackeo"	
Inundaciones			

Terremotos Huracanes Disturbios y Guerras		Contraseñas "Crakeadas"	

Tabla TK1 Aspectos empleados en un Ataque

Los empleados ignorantes usualmente no tienen motivos u objetivos para causar daño. El daño es accidental. De otro modo, los atacantes maliciosos pueden engañar a los empleados ignorantes, mediante la ingeniería social para obtener entrada al sistema. Los atacantes pueden "enmascararse" como un administrador y solicitar contraseñas y nombres de usuario.

Motivos, objetivos y metas de atacantes maliciosos

Existe un fuerte traslape entre la seguridad física y la integridad y privacidad de datos. Ciertamente, el objetivo de un atacante no es la destrucción física de los sistemas, pero sí la penetración, eliminación o copia de la información sensible.

Algunos de los métodos que los atacantes utilizan son:

Eliminación y alteración de la información. Estos atacantes maliciosos parte de la empresa, normalmente lo hacen para cobrar venganza contra la organización, tal vez porque se encuentran descontentos por algo o contra alguien. Los atacantes externos, quizá lo hacen para probar que pueden ingresar al sistema de una determinada organización o simplemente por diversión.

Robo y fraude de la información. Sistemas de cómputo son explotados de numerosas maneras, ya sea por métodos de fraude automatizados tradicionales o por métodos nuevos. Los sistemas financieros no son los únicos sujetos a fraude. Otros objetivos son, los sistemas de control de acceso a recursos, sistemas concurrentes o sistemas de telefonía de larga distancia.

Desestabilización de las operaciones comerciales normales. Aquí, el atacante ya tiene definido su objetivo, y puede lograrlo, mediante diferentes métodos de ataque de negación de servicio.

Métodos, herramientas y técnicas de ataque

Ataques = motivo + **método** + **vulnerabilidad**

El método es la fórmula de explotar las vulnerabilidades en una organización, como se muestra en la figura anterior.

Vulnerabilidades en la Seguridad

Son puntos débiles o huecos en la seguridad, que son explotados por los atacantes. Algunos de esos puntos débiles son:

- Contraseñas

- Diseño del protocolo. Los protocolos de comunicación son algunas veces puntos débiles. Los atacantes se valen de eso, para ganar información y eventualmente lograr acceso a los sistemas. Algunos conocidos son:

    - *TCP/IP.* La pila de este protocolo tiene algunas debilidades como:
        - *Engaño en las direcciones IP*
        - *Ataques en la petición* de conexión TCP.
    - *Protocolo de TELNET.* Este puede ser utilizado para administrar sistemas ejecutándose desde Microsoft Windows 2000 y Unix. Cuando se utiliza el cliente de TELNET para conectarse a un sistema Unix y viceversa, los nombres de usuario y contraseñas, son transmitidos en texto claro.
    - *Protocolo de Transferencia de Archivos (FTP).* Si el servicio de FTP se ejecuta y los usuarios necesitan enviar información desde una ubicación segura, entonces, el nombre de usuario y contraseña, al igual que en TELNET son enviados en texto claro.
      Además algunos comandos revelan información del usuario, esto no es raro encontrarlo, dada la interoperabilidad entre los productos de Microsoft y versiones de Unix. Estos comandos que revelan información del usuario representan cierta amenaza. Algunos de esos comandos son: *Finger y Rexec.*

        - *Modo de Transferencia Asíncrona (ATM)* y *FRAME RELAY.* Acceso directo al cableado de red y conexiones subterráneas. Existe una vulnerabilidad en la arquitectura de administración remota para dispositivos conectados a redes ATM, los cuales permiten la configuración de información en accesos no permitidos.
        - *Administración de Dispositivos.* Switches y ruteadores son fácilmente manejados por una interfaz de HTTP o a través de comandos en línea. Combinado además con una debilidad en contraseñas, permite a cualquiera tomar el control de un dispositivo.
        - *Módems.* Un módem no autorizado es un serio problema para la seguridad. Son utilizados para conectarse a Internet, para conectarse a la oficina desde la casa, pero un módem es un medio de bordear un FIREWALL que protege una red de intrusos externos. Un hacker utilizando una herramienta "war dialer", para identificar el número telefónico del módem y una herramienta de "crackeo" de contraseñas, para romper una contraseña débil, puede ganar acceso a un sistema. Ahora, debido a la naturaleza de una red, el hacker puede frecuentemente conectarse a cualquier computadora de la red.

Las siguientes figuras explican y ejemplifican las diversas vulnerabilidades existentes y la pérdida en activos involucrados.

Ejemplo 1: *Amenaza no maliciosa (empleados ignorantes)*

Francisco introduce algunos juegos a su equipo, a partir de una unidad de disco, dichos juegos contienen algunos virus y troyanos. Después de un corto tiempo notan comportamientos extraños y discontinuidad en el servicio, incluso algunos daños en los datos.

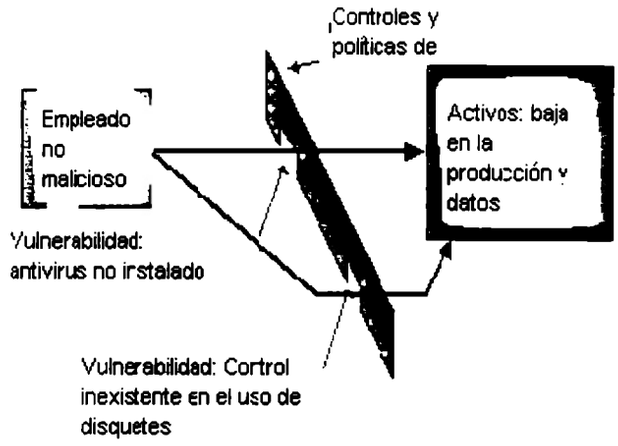


Fig. K2 Amenaza no maliciosa [9]

Ejemplo 2: *Amenaza maliciosa (empleados maliciosos)*

Una empleada conocida como Sally, no fue considerada para una promoción en su trabajo. Ella piensa que pone el extra en su trabajo, además de brindar horas extraordinarias en él. Por tanto, cree que no fue considerada por ser joven, lo cual piensa es demasiado injusto. Se ha resignado, pero a decidido vengarse contra la compañía, deteniendo las peticiones de servicio, al servidor Web de la compañía. Sally utilizará una herramienta de ataque de negación de servicios llamada Trin00.

La mayor parte de los negocios de la compañía son conducidos vía comercio electrónico y los clientes se quejan que no pueden conectarse al servidor Web. El siguiente diagrama bosqueja las diversas herramientas y vulnerabilidades que Sally utilizó para lograr su objetivo.

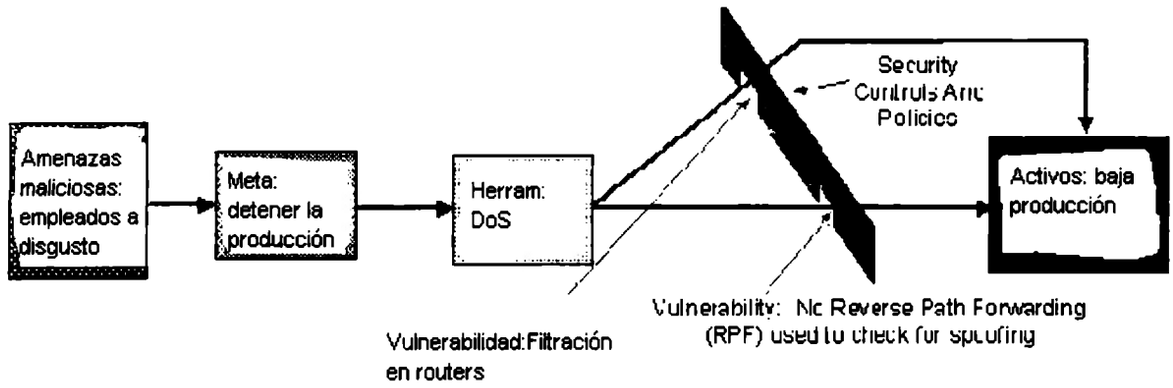


Fig. K3 Amenaza Maliciosa [9]

Ejemplo 3: *Desastres naturales*

Una organización tiene varios módems y ruteadores instalados en una ISDN (Red Digital de Servicios Integrados), pero no cuentan con protección a la corriente eléctrica. Durante una tormenta eléctrica, algunos relámpagos dañan el teléfono y las líneas ISDN. Todos los módems y ruteadores en las ISDN son destrozados, tomando con ellos un par de tarjetas madre. La siguiente figura muestra la vulnerabilidad y la pérdida de activos.

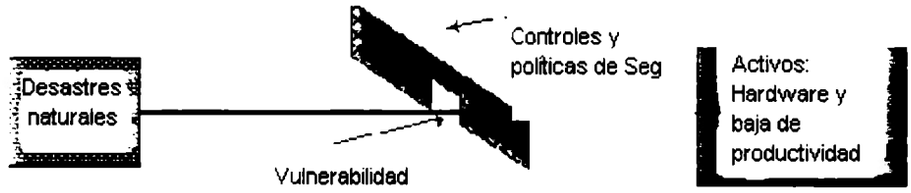


Fig. K4 Desastres Naturales [9]

En conclusión los hackers pueden utilizar diversos métodos, herramientas y técnicas para explotar las vulnerabilidades en políticas de seguridad y controles, para lograr un objetivo. Ataques no maliciosos ocurren debido a políticas y controles de seguridad muy limitados, que permiten que vulnerabilidades y errores tomen lugar. Desastres naturales pueden ocurrir en cualquier momento, de tal manera, las organizaciones deben tomar medidas para intentar prevenir el daño que esos elementos puedan causar.

ANEXO L COMPARATIVA DE ANTIVIRUS

Las comparativas como todas, generan descontentos, y este campo no fue la excepción, generando los propios descontentos entre ciertos fabricantes de antivirus, pero en realidad no se hizo más que mostrar con pruebas contundentes, los resultados de un riguroso análisis.
Anualmente se generan pruebas sumamente extensas y detalladas, que no dejan lugar a dudas o confusiones. Además se han corrigido otras para afinar aún más sus resultados finales. Como en otras oportunidades, se creó un virus totalmente desconocido, que fue enviado en forma anónima a más de 30 fabricantes de antivirus, como si de un usuario común se tratara. Se evaluó la respuesta de estas empresas, la velocidad para crear un antídoto, y el servicio prestado al cliente. Algo que jamás había sido tenido en cuenta por evaluaciones de otros expertos hasta ese momento.

Test ITW (In The Wild). Con esta prueba salió a la luz la capacidad de los 30 productos evaluados para detectar virus que aparecen en la lista de igual nombre mensualmente, donde se reflejan los virus en actividad en el mundo entero, punto fundamental como argumentábamos más arriba.
Test MACRO, donde se evaluaron virus de macros para Word, Excel, Access, PowerPoint, AmiPro, WordPro, Lotus 1-2-3 y CorelDraw.
Test TROYA. Se seleccionaron y probaron las respuestas a 150 troyanos y backdoors, desde los más conocidos como BackOrifice, NetBus, SubSeven, hasta otros de más reciente creación.
Test BIN-BOOT, es el que incluye desde los ya poco comunes virus de booteo, hasta los más sofisticados creados para Win32, pasando por los clásicos virus que afectan ejecutables bajo MS-DOS o Win16, en un total de más de 14.000.
Test Archivos de Internet. Involucra sin dudas a la generación más reciente de virus, tanto los que infectan código HTML, como a los applets de Java, controles ActiveX (OCX), gusanos de IRC, de e-mail y de diversos scripts (VBS, etc.)

También se tuvo en cuenta la capacidad de detectar virus en formatos comprimidos (y sus encadenados, o sea comprimidos dentro de otros comprimidos, etc.). Un talón de Aquiles que solo dos antivirus lograron superar casi en un 100%, menos en algunos formatos poco conocidos (AIN, ARC, HA, PAK, ZOO) que no fueron soportados por ninguno de los productos evaluados.

Antivirus	Sistemas Operativos probados	Puntaje
Antivirus Toolkit Pro (AVP)	DOS/95/98/NT	14
Panda Antivirus (Platinum)	3.x 95/98/2000 NT	12
McAfee (VirusScan)	DOS/95/98/2000/NT	11
Norton Antivirus	DOS 3.x 95/98 2000/NT	10

InoculateIT	3.x/95/98/2000/NT	9
Command Software Antivirus	DOS/95/98/NT	7
Norman Antivirus	DOS/95/98/NT	4
PC-Cillin	95/98/NT	3
F-Secure	95/98	3
Sophos Anti-Virus	DOS/3.x/95/98/NT	3
InoculateIT Personal Edition	95/98	2

Tabla L1. Comparativa de virus [24]

Consideraciones finales

Es importante resaltar que algunos productos no aparecen en la lista, simplemente porque no fueron evaluados por los expertos, o porque su uso está restringido a un mayor conocimiento del sistema operativo por parte del usuario. Un ejemplo de esto es la versión para DOS del F-Prot (producto gratuito si es para uso personal), que sigue siendo una excelente opción como segundo antivirus para revisar nuestra máquina, incluso desde un disquete. Y finalmente, es evidente que lo más importante en este tema es mantener un antivirus actualizado. Pero insistimos en que igual de recomendable es no confiar en un solo antivirus. Sin embargo, jamás mantenga dos antivirus monitoreando. Las consecuencias en estos casos serían imprevisibles para su sistema operativo. Lo ideal sería un antivirus para monitorear, y otro a mano para revisar archivos y carpetas antes de ejecutar algún software por primera vez.

ANEXO M PROPUESTA DE LISTA DE CHEQUEO

Lista de Chequeo				
	Aplicación	**Si**	**No**	**No aplica**
Secc. I	**Autenticación**			
I.1	*Administración de Password*			
a	**Usuarios**			
1	Generación			
2	Modificación			
3	Eliminación			
4	Características de Password			
4.1	***longitud***			
4.2	***formación***			
4.3	***tiempo de vida***			
b	**Mapeo a Password de sistema**			
I.2	*Administración de usuarios*			
a	**Lista**			
b	**Agregar**			
c	**Funciones de búsqueda**			
d	**Cambio de características**			
e	**Eliminar**			

	Aplicación	Si	No	No aplica
I.3	*Administración de Certificados*			
a	**Autoridad Certificadora Externa**			
b	**Certificación Interna**			
1	Generación			
2	Renovación			
3	Eliminación			
4	Distribución			
5	CRL (Lista de certificados revocados)			
5.1	***Generación***			
5.2	***Distribución***			
5.3	***Mantenimiento***			
c	**Certificados entre atributos de usuario (Extensiones)**			
I.4	*Administración de llaves*			
a	**Generación**			
b	**Distribución**			
c	**Repositorio**			
I.5	*Dispositivos Biométricos*			
a	**Local**			
b	**Remoto**			
I.6	*Tarjetas Inteligentes*			
I.7	*Criptosistemas*			
a	**Local**			
b	**Red**			

	Aplicación	Si	No	No aplica
Secc. II	**Control de Acceso**			
II.1	*Listas de control de acceso*			
II.2	*Listas de Capacidad*			
II.3	*Discrecionario*			
II.4	*Mandatario*			
II.5	*Basado en mínimos privilegios*			
II.6	*Confirmación periódica de los derechos de acceso*			
II.7	*Remoto*			
Secc. III	**Integridad**			
	¿Utiliza método para revisión de integridad?			
	** En caso de responder NO, pase a la Secc. IV*			
III.1	*CheckSum*			
III.2	*MD2*			
III.3	*MD4*			
III.4	*MD5*			
III.5	*SHA-1*			
III.6	*Otro*			
Secc. IV	**Disponibilidad**			
IV.1	*Información disponible cuando es requerida*			
IV.2	*Cluster*			
IV.3	*Manejo de carga*			
Secc. V	**Confidencialidad**			
V.1	*Módulo de VPN*			
V.2	*El sistema maneja IPSec*			

	Aplicación	Si	No	No aplica
V.3	*La transferencia de datos se realiza bajo SSL*			
V.4	*Criptosistemas Simétricos*			
	* *En caso de responder NO, pase a la Secc. V.5*			
a	**DES**			
b	**3DES**			
c	**IDEA**			
d	**RC2**			
e	**RC4**			
f	**Otro**			
V.5	*Criptosistemas Asimétricos*			
	* *en caso de responder NO, pase a la Secc. VI*			
a	**RSA**			
b	**Otro**			
V.6	*Criptosistemas Híbridos*			
V.7	*Cifrado de Archivos*			
V.8	*Cifrado de directorios*			
Secc. VI	**No repudiación**			
VI.1	*Autorización del servicio*			
VI.2	*Envío de servicio proporcionado*			
VI.3	*Origen del servicio*			
VI.4	*Recepción del servicio*			
VI.5	*Conocimiento del servicio o contenido del mensaje*			
VI.6	*Firmas Digitales*			

	Aplicación	Si	No	No aplica
Secc. VII	**Tolerancia a fallas**			
VII.1	*Cluster*			
VII.2	*Manejo de carga*			
Secc. VIII	**Parches**			
VIII.1	*Último parche*			
VIII.2	*Última versión*			
Secc. IX	**Análisis de Contenido**			
Secc. X	**Certificación**			
	¿La aplicación esta certificada?			
	*En caso de responder "NO", pase a la Secc. XI			
X.1	*TCSEC*			
X.2	*ITSEC*			
X.3	*ICSA*			
X.4	*Otro*			
Secc. XI	**Resistencia a ataques**			
XI.1	*Escucha*			
XI.2	*Negación de Servicio (DoS)*			
XI.3	*Software de IDS*			
a	**Firewalls**			
b	**Otros filtros de direcciones IP**			
c	**Bloqueo Automatizado**			
d	**Estándar RFC2267 (limitadores de flujo de datos)**			
e	**Uso de iTrace**			

	Aplicación	Si	No	No aplica
XI.4	*Activos*			
XI.5	*Stack Overflow*			
XI.6	*Buffer Overflow*			
Secc. XII	**Reúso de Componentes**			
XII.1	*Protección de la información almacenada*			
XII.2	*Caché*			
XII.3	*Registros*			
XII.4	/temp			
Secc. XIII	**Auditoria**			
XIII.1	*Monitoreo*			
a	**Procesador**			
b	**Memoria**			
c	**Caché**			
d	**Procesos**			
e	**Servicios en ejecución**			
f	**Gráficas o diagramas**			
g	**Alarmas**			
h	**Reportes**			
i	**Acceso a archivos y subdirectorios**			
j	**Acceso remoto**			
XIII.2	*Generación de Logs*			
XIII.3	*Auditoria de Logs*			
XIII.4	*Administración de Logs (reconstrucción)*			

	Aplicación	Si	No	No aplica
Secc. XIV	**Rutas seguras**			
XIV.1	*Conexión directa entre S.O y aplicación*			
XIV.2	*Conexión directa entre aplicación y aplicación*			
Secc. XV	**Arranque Seguro**			
Secc. XVI	**Respaldo**			
XVI.1	*Cintas*			
XVI.2	*Red*			
XVI.3	*CD´s*			
XVI.4	Otra			
Secc. XVII	**Compartición de recursos**			
XVII.1	*Grupos*			
XVII.2	*Usuarios*			
XVII.3	*Con privilegios*			
XVII.4	*Sin privilegios*			
XVII.5	*Públicos*			
Secc. XVIII	**Clasificación de la información**			
XVIII.1	*Por aplicación*			
XVIII.2	*Por los usuarios que la manejan*			
XVIII.3	*Por su sensibilidad*			
Secc. XIX	**Administrador Dedicado**			

	Aplicación	Si	No	No aplica
Secc. XX	**Evaluación del producto por parte de la organización**			
Secc. XXI	**Política de recuperación de desastres**			
Secc. XXII	**Plan de contingencia del negocio**			