Universidad Latinoamericana de Ciencia y Tecnología
ULACIT

"Construyendo aplicaciones Web seguras"

Presentado por: Ing. Robert Valerio Ávila

Profesor: Lic. Guillermo Oviedo

San José, Costa Rica
Diciembre, 2005

Índice de contenido

Resumen ejecutivo

La seguridad de las aplicaciones no es algo que se añade simplemente al final del proceso de desarrollo o que se olvide en las fases de diseño. La misma debe ser inherente a la funcionalidad de la aplicación y debería incorporarse desde las primeras etapas del proyecto, por ejemplo: en la etapa de diseño debe definirse cuáles son los requerimientos de seguridad a ser implementados y el impacto que tendrá en la infraestructura, especialmente cuando existe una fuerte presión por parte del usuario por construir y liberar el software de una manera ágil, flexible y sobre todo económica.

No hay aplicaciones seguras en una infraestructura insegura y viceversa, no se trata de desarrollar un software insuperable o bien de una infraestructura con niveles de seguridad perfectos, sino de minimizar el riesgo, desarrollar software lo suficientemente seguro y confiable para soportar ataques en el ambiente en el cual está operando.

El Windows Server System Reference Architecture (WSSRA) surge como respuesta a las necesidades de implementación de patrones y mejores prácticas de la industria que buscan las compañías. Fue creado con el aporte de los líderes de la industria en sus diferentes áreas: redes y comunicaciones, proveedores de hardware, almacenamiento y servicios de consultaría. Dichos patrones ayudan a reducir la complejidad de la planeación e implementación de soluciones basadas en tecnologías Microsoft y con ello promueven la estandarización.

Es evidente que todas las empresas son diferentes, por lo que no se puede generalizar una metodología para implementar los lineamientos en la construcción del software, sin embargo, el contar con una metodología adecuada, definitivamente facilitará la disminución del riesgo en la creación de las aplicaciones. Una metodología fácil de implementar es la aplicación de una lista de verificación (“Checklist”), la misma se aplicará en las diferentes etapas del proceso de desarrollo y consiste de una serie de evaluaciones que se activan en las diferentes etapas del ciclo de vida de la aplicación y no interfiere con el proceso de desarrollo.

Abstract

The security of the applications is something that you can't simply add at the end of the process development or something to be forgotten in the design phases.

In the same way, it is something that must be inherent to the functionality of the application and should be incorporated from the first stages of the project. For example, in the design stage it must be defined which are the security requirements to be implemented and the impact that it will have in the infrastructure, especially when the user exerts a strong pressure to build and release the software in an agile, flexible and mainly economic way.

There are no safe applications in an unsecure infrastructure and vice versa. Moreover, it does not have to do with the idea of developing an unbeatable software or build a infrastructure with perfect levels of security. The idea is to diminish the risk, to develop a secure enough and reliable software to support attacks in the place in which it is operating.

The Windows Server System Reference Architecture (WSSRA) arises like an answer to the implementation of patterns required and better practices in the industry that companies are looking for. It was created with the help of industry-leading partners in the areas of networking and communications, server hardware, storage, and consultant services. These patterns help to reduce the complexity of the planning and implementation of solutions based on Microsoft technologies and promotes standardization.

It's evident that all the companies are different. That is the reason why a methodology cannot be generalized to implement the standards for the software construction. Nevertheless, counting with a suitable methodology will definitively make things easier in the diminution of the risk when creating applications. An easy methodology to implement is the application of a verification list ("Checklist"). The same one will be applied in the different stages of the development process and it consists of a series of evaluations that are activated in the different stages of the cycle of life of the application, and these evaluations do not interfere with the development process.

Palabras claves

- Seguridad Web
- Infraestructura
- Arquitectura
- Lineamientos
- Seguridad en Asp.Net

Introducción

En el inicio, las soluciones de software se construyeron utilizando un modelo centralizado, en el cual se asumía que todo el procesamiento se realizaba consumiendo recursos locales. Este esquema de trabajo originó múltiples aplicaciones que si bien resolvieron problemas de negocios específicos, no eran fáciles de integrar con otras similares, sobre todo al no contar con mecanismos de fácil de utilización para exponer su funcionalidad.

La evolución tecnológica condujo hacia los modelos: cliente - servidor y de múltiples capas, cuyos procesamientos se distribuían entre diversos servidores y donde cada uno cumple funciones específicas, por ejemplo el servidor Web o de Base de Datos.

El modelo de múltiples capas se orienta hacia la reutilización, de tal forma que permitió mejoras en la productividad. Sin embargo, ambos modelos estuvieron sumamente ligados a tecnologías específicas, por ejemplo: una aplicación basada en “Microsoft COM +” no puede utilizar una construida en el lenguaje Java y viceversa, esta problemática dificultó la reutilización a nivel corporativo, dado que los componentes pertenecían a dominios tecnológicos[1] diferentes.

Debido a lo anterior, en la industria se lanzó una iniciativa de software con el fin de buscar tecnologías que permitieran la reutilización de servicios en forma independiente y la comunicación con diferentes tecnologías, la cual dio origen al Web Service, concepto que permite crear componentes y por ende exponer servicios utilizando estándares aceptados en la industria.

Por lo expuesto anteriormente, las compañías se encuentran bajo una presión debido al ofrecimiento de servicios por los diferentes canales (Internet, intranet y telefonía celular entre otros), de una forma segura y controlada. Al existir esta interoperabilidad no hay límites para que las compañías creen alianzas, se comuniquen entre sí o entre

[1] Ordenamiento lógico de una tecnología específica.

sus diferentes sistemas internos, en la operación de las organizaciones dichos servicios se tornan cada vez más críticos, esto como consecuencia de una estrategia en la cual se desea posicionar productos o servicios lo más rápido posible, compitiendo por salir de primero en un nicho de mercado específico.

Conforme crece ésta necesidad de negocio, los requerimientos para el área de tecnología aumentan, por ende muchas de las soluciones de misión crítica se construyen pensando en responder a una necesidad del usuario de una manera ágil, flexible y sobre todo económica. Sin embargo, el objetivo de la parte comercial y de negocio en las empresas pretende aprovechar esos cortos lapsos u oportunidades en el mercado, por lo que esperan que el desarrollo del software se construya en un tiempo corto, sin importar la tecnología que se utilice, - independientemente del impacto que la solución de software tendrá en la infraestructura de la empresa - , la arquitectura existente, los servidores en que se instalará la solución o bien el monitoreo de la misma.

Tal y como se mencionó anteriormente, el área de negocio espera que se construya las aplicaciones rápidamente y por consecuencia el tiempo destinado para el desarrollo de las soluciones de software es limitado en el cronograma, considerando que se debe cumplir con una fecha de lanzamiento asignada por la administración. Evidentemente, el área administrativa o de negocio se preocupa por posicionar un producto o servicio en el mercado lo antes posible y el área tecnológica por responder oportunamente a las necesidades del negocio sin descuidar la infraestructura existente, pero ¿Qué hay acerca del desarrollo del software? En esta urgencia por entregar una solución quién se pregunta: ¿Escriben los desarrolladores código seguro? o bien ¿Existe una infraestructura segura? ¿Cuál metodología es fácil de implementar?

Por consiguiente, lo que se pretende es que tanto la infraestructura como el código estén diseñados para minimizar el riesgo del ataque de un “Hacker”, en este sentido el objetivo principal del área tecnológica es construir un código robusto, segmentar correctamente las aplicaciones y minimizar las vulnerabilidades que permitan la

intrusión de personas no deseadas, especialmente cuando las aplicaciones estén conectadas en una Intranet corporativa o peor aún, cuando las mismas se exponen por Internet. Definitivamente, existe un riesgo muy alto de que la aplicación sea atacada por vulnerabilidades en su entorno o por deficiencias en el código.

Tomando como base lo anterior, cuando una aplicación es comprometida se produce una gran variedad de consecuencias para el negocio, desde salir de producción (su operación normal), perder la confianza del cliente, dañar la imagen de la empresa, hasta pérdidas de dinero por concepto de un fraude.

Por lo tanto, se creará un lineamiento en el que se recopile una serie de controles y recomendaciones, con base en las mejores prácticas del mercado, destinados al desarrollo e implementación de un producto de software Microsoft basado en código seguro y bajo una infraestructura de alta disponibilidad, escalable, robusta y sobre todo segura.

Infraestructura

Los sitios Web de las grandes empresas constituyen un ejemplo de cambio dinámico, normalmente comienzan siendo pequeños y crecen exponencialmente con la demanda, se expanden tanto en la cantidad de usuarios como en servicios que brindan a los clientes.

Dichos sitios de mayor éxito canalizan este crecimiento aumentando el número de servidores en su infraestructura, pero en la mayoría de las compañías el hardware y software provienen de diferentes proveedores, de las áreas de redes, almacenamiento, seguridad y servidores. Esta situación provoca que se presenten problemas de integración si no se cuenta con estándares bien definidos para la integración.

El crecimiento no estructurado ha conducido a la proliferación de diferentes soluciones con diversos grados de integración arquitectónica, lo cual no es fácil de mantener y soportar.

Para que la empresa responda con la agilidad requerida a las necesidades cambiantes de los clientes y con la capacidad de lidiar con los cambios tecnológicos, es necesario controlar el crecimiento no estructurado y la proliferación de piezas cuya integración no siempre resulta transparente. Por lo que es necesario definir un modelo de infraestructura estandarizado que dicte las bases sobre las cuales se implementarán las soluciones dentro de la organización.

Por consecuencia, se recomienda el Windows Server System Reference Architecture (WSSRA), el cual fue diseñado para proveer un centro de arquitectura corporativa, surge como respuesta a las necesidades de implementación de patrones y mejores prácticas de la industria que buscan las compañías. Fue creado con el aporte de los líderes de la industria en sus diferentes áreas: redes y comunicaciones, proveedores de hardware, almacenamiento y servicios de consultoria, dentro de los que se destacan: Cisco System Inc, Dell Computer Corporation, EMC Corporation, Emulex, Fujitsu Siemens Computers, Hewlett Packard, Nortel Networks y Unisys Corporation entre otras.

El WSSRA provee la guía necesaria para:

- Reduce la complejidad de la planeación e implementación de soluciones basadas en tecnologías Microsoft.

- Ayuda a asegurar la estandarización entre productos.

- Proporciona una referencia con autoridad para integrar los productos Microsoft con los de sus socios de negocios, lo cual facilita el despliegue en una plataforma optimizada.

El crecimiento se apoya en una base arquitectónica sólida que contemple una alta disponibilidad, una infraestructura segura y una eficiente administración, dicha base debe satisfacer los siguientes objetivos esenciales de diseño.

Objetivos esenciales

Los objetivos esenciales de la arquitectura son:

Escalabilidad: Todos sus componentes deben ser escalables para permitir un crecimiento continuo que satisfaga la demanda del cliente y los requisitos del negocio.

Disponibilidad: Los componentes deben incluir redundancia o especialización funcional para controlar los errores.

Seguridad: La arquitectura debe ofrecer un modelo de seguridad completo que proteja los datos y la infraestructura frente a ataques e intentos de robo.

Fácil administración: La facilidad de configuración, el monitoreo de las soluciones y la detección de errores son vitales para alcanzar los objetivos de disponibilidad, escalabilidad y seguridad, los cuales deben ser capaces de adaptarse al crecimiento del entorno.

De las cuatro grandes áreas anteriores, se profundizará en el tema de seguridad, desde el punto de vista de que las aplicaciones utilicen mecanismos de defensa adecuados y tengan un diseño con capacidad de ampliación, de elevado rendimiento y seguro.

Seguridad

Al enfocarse en el tema de la seguridad, las empresas deben disponer de una estrategia de administración del riesgo que incluya una protección adecuada de la privacidad e integridad de la información, lo cual resulta esencial para el éxito de un sitio empresarial.

Entre los datos confidenciales que deben protegerse, pueden citarse las identificaciones utilizadas para la autenticación, por ejemplo: números de tarjeta de crédito, saldos de cuentas o detalles de otras transacciones financieras; razón por la cual es necesario proteger esa información que viaja por la red. En este sentido, existen dos características principales para que los datos no sean revelados o modificados por terceras personas:

Privacidad. El objetivo es garantizar que los datos permanezcan privados y confidenciales, fuera del alcance de terceras personas, por lo que un quebrantamiento de los mismos, eventualmente, provocaría serios daños a los clientes o usuarios internos y definitivamente un deterioro en la imagen de la compañía.

Integridad. La finalidad de un canal de comunicación seguro es garantizar que los datos estén protegidos contra modificaciones accidentales o deliberadas durante la transmisión de los mismos.

La clave para implementar mecanismos correctos de seguridad consiste en utilizar una estrategia de defensa en profundidad, la cual define múltiples niveles sin suponer que ninguna área por sí sola proteja por completo la infraestructura.

Para implementar esta estrategia de defensa en profundidad, la arquitectura se divide en redes físicas o segmentos de red independientes, de manera tal que se divida el sistema en diferentes capas y por ende exista una exposición parcial en niveles clasificados por la criticidad de los datos.

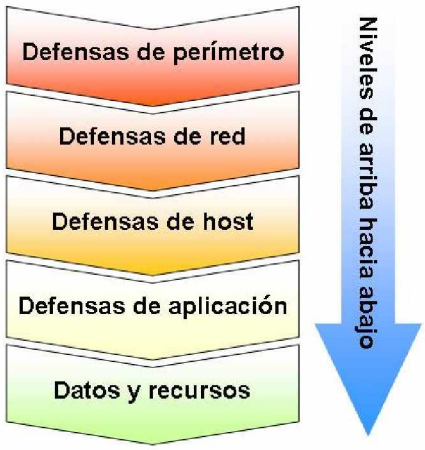

Figura 1. Estrategia de defensa en profundidad

En efecto, en la *Figura 1* la estrategia de defensa en profundidad contribuye a garantizar la seguridad, ya que proporciona diferentes niveles que protegen los recursos en el caso de que se produzca un error en cualquier nivel individual o capa (o en algunos casos, en varios niveles).

Defensas perimetrales

Existen diversos dispositivos seguros que protegen cada punto de acceso a la red. Mediante el uso de la estrategia de defensa en profundidad, se evalúa cada dispositivo, se deciden los tipos de tráfico permitidos y se desarrolla un modelo seguro para bloquear el resto del tráfico. Los puntos de acceso principales al sistema están en los proveedores de servicios Internet (ISP[2]), por lo que hay directivas que sólo permitan el tráfico en los puertos 80 y 443, el resto del tráfico se bloquea por completo.

Por ejemplo si algún usuario intenta tener un acceso no autorizado desde Internet o iniciar un ataque de denegación de servicio (DoS[3]) sólo podrá probar en dichos puertos, los dispositivos detectarán el intento y alertarán al administrador.

Un modelo de seguridad adecuado es fundamental en las redes de las empresas que exponen servicios con intermediación financiera como transferencias de dinero o pagos con tarjetas de crédito, ya que la red de perímetro está expuesta a cualquier usuario de Internet, en este sentido, existe una alta probabilidad que una empresa de esta naturaleza se convierta en blanco de ataques.

Defensas de red

Detrás de los dispositivos perimetrales la siguiente línea de defensa es la propia red, por lo que es necesario evaluar cada segmento con la finalidad de crear una Lista de Control de Acceso conocida por sus siglas en inglés como ACL[4], en la cual se establezcan los permisos que limitan el tipo de tráfico que es absolutamente necesario

[2] ISP, Internet Service Provider – Proveedor de Servicios Internet.
[3] DoS, Deny Of Service – Denegación De Servicio.
[4] ACL, Access Control List – Lista de Control de Acceso.

en la red, el resto del tráfico se bloquea mediante un dispositivo o servidor de seguridad o simplemente se deshabilita.

Defensas de aplicaciones

La protección de las aplicaciones es una parte fundamental en cualquier modelo de seguridad y en este sentido el sistema operativo proporciona el primer nivel de protección. Es responsabilidad del analista o el programador incorporar seguridad adicional en las aplicaciones, minimizando de esta manera las vulnerabilidades. Es evidente que cada aplicación requiere su propio modelo de seguridad, por lo que es indispensable seguir los estándares y lineamientos de desarrollo durante el ciclo de construcción.

Datos y recursos

Se debe evaluar la conexión de todas las aplicaciones y restringir los puertos de cada servidor para eliminar los que no son necesarios, además es imprescindible la activación de bitácoras o pistas de auditoria. Así mismo, se puede complementar con mecanismos disparadores conocidos en inglés como Triggers[5], los cuales funcionarán de forma preventiva y enviarán una alerta en caso de un eventual ataque. Estas recomendaciones deben realizarse con especial cuidado, ya que el objetivo es contar con un mecanismo de alertas que no degrade el rendimiento del sistema.

[5] Trigger – Disparador, se ejecuta ante la activación de un evento previamente programado.

Arquitectura

El diseño de aplicaciones no es una tarea sencilla, para ello es necesario tomar un gran número de decisiones a nivel de arquitectura, diseño e implementación, con el fin de garantizar robustez, escalabilidad, disponibilidad, continuidad del negocio y seguridad en la infraestructura.

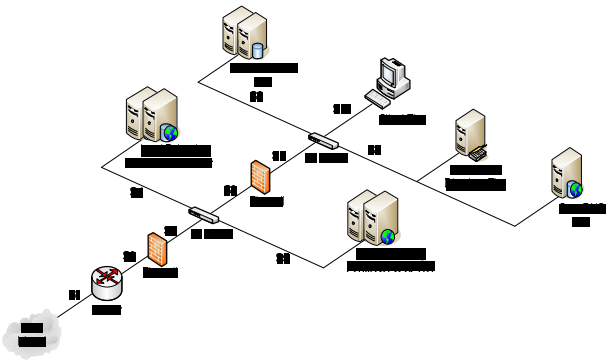

Figura 2. Diseño de las aplicaciones

Estas decisiones tendrán un impacto en las "capacidades" de la aplicación, así como en el diseño e implementación de la infraestructura de destino.

En la actualidad la arquitectura de las aplicaciones está conformada por capas, zonas y segmentos, tal y como se puede observar en la figura 2. Existen varios elementos claves de la arquitectura como lo son los clientes de Internet, los dispositivos situados en la red de perímetro, la llamada zona desmilitarizada (DMZ[6]) y el servidor de seguridad interno entre los principales elementos.

Clientes de Internet

Los clientes de Internet son los usuarios finales, dispositivos inteligentes o aplicaciones a los que se puede tener acceso a través de Internet. Normalmente, los usuarios finales serán personas sentadas ante un equipo con acceso a información a través de Internet mediante sus exploradores locales, pero también puede tratarse de dispositivos inteligentes, como PDA (Personal Digital Assistants) asistentes digitales personales, teléfonos con capacidad de transmisión de datos (tecnología GSM) o teléfonos inteligentes.

[6] DMZ, , Demilitarized Zone – Zona Desmilitarizada.

Red de perímetro

Separa la red externa o red de Internet pública de la red interna de una organización. Entre los dispositivos de una red de perímetro se encuentran los enrutadores[7] de frontera y los servidores de seguridad, entre otros. Los cuales proporcionan la conectividad, seguridad y disponibilidad necesarias en la red.

Los enrutadores de frontera se emplean para conectar la infraestructura de una empresa a un Proveedor de Servicios de Internet (ISP[8]) que ofrece acceso a la red.

Los sistemas Web comerciales deben contar con una redundancia completa para lo que deben incluir al menos dos enrutadores de frontera conectados a dos ISP distintos con el fin de ofrecer tolerancia a los errores y las vías complementarias para el tráfico.

Un servidor de seguridad es un mecanismo para controlar el flujo de datos entre dos partes de una red con niveles de confianza diferentes. El servidor de seguridad de perímetro debe inspeccionar el tráfico de red entre el perímetro y la red DMZ[9], también debe garantizar que únicamente los puertos TCP/IP requeridos atraviesen el Web de aplicaciones para el usuario.

Algunas de las funciones que debe realizar el servidor de seguridad de perímetro son:

- Filtrado del tráfico (por dirección IP, puerto TCP, etiqueta de host, dirección URL completa o tipo de archivo).

- Traducción de direcciones de red (NAT, Network Address Translation).

[7] Enrutador - Dispositivo hardware o software para la interconexión de redes.
[8] ISP, Internet Service Provider – Proveedor de Servicios de Internet
9 DMZ, Demilitarized Zone – Zona desmilitarizada.

- Mecanismos y procedimientos de protección frente a ataques, por ejemplo de denegación de servicio.

DMZ

La zona desmilitarizada (DMZ) se encuentra entre la red de perímetro y la red interna y está separada por servidores de seguridad a ambos lados, en dicha red se proporcionan dos grupos básicos de servicios:

- El primero es el servicio Web de aplicaciones para el usuario, formado por servidores en los que se ejecuta Internet Information Server (IIS). Este grupo ofrece los servicios Web fundamentales y se comunica con los clientes de Internet a través de los protocolos de transporte estándar, como HTTP o HTTPS cuyos puertos son el 80 y 443 respectivamente.

- El segundo grupo proporciona servicios de red como el Sistema de Nombres de Dominio (DNS, Domain Naming System). Todos los servidores de la red DMZ pueden comunicarse con la capa de “Backend” de la zona del “Front End”, y los mismos con la zona interna.

Servidor de seguridad interno

El servidor de seguridad ofrece una protección adicional a los servidores de red internos. El mismo se encuentra inmediatamente detrás de la DMZ y delante de todos los segmentos de red implementados, esta configuración ayuda a impedir el acceso a todas los capas de la red si un intruso logra poner en peligro a un servidor Web.

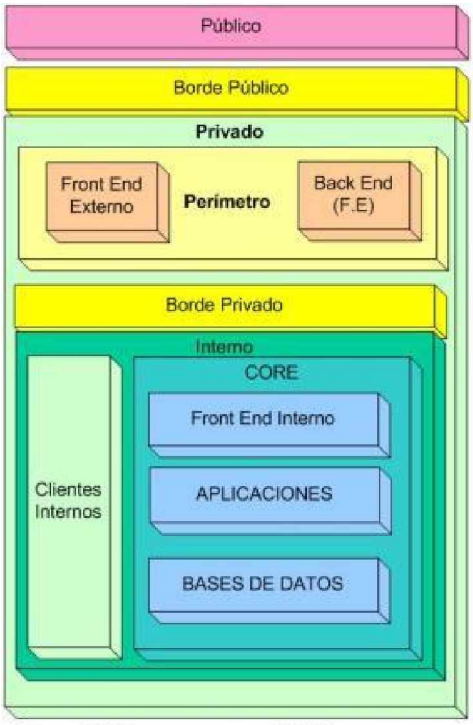

Figura 3. Zonas y capas de infraestructura

Siguiendo las mejores prácticas de la industria descritas en el WSSRA, indican que para facilitar la creación de políticas de seguridad, se deben usar zonas, las cuales son entidades lógicas que contienen una o más capas, dentro de las mismas pueden existir otras zonas o ser miembro de alguna de ellas.

El objetivo de crear una zona es proveer un contenedor lógico, de manera que las capas puedan comunicarse dentro de una o varias zonas diferentes.

Para solventar el tema de la comunicación entre capas se utilizarán las tres tecnologías siguientes:

1. **SSL/TLS (Secure Sockets Layer/Transport Layer Security)**.

Protege el canal entre un explorador y el servidor Web. No obstante, puede utilizarse para proteger mensajes de servicios Web y comunicaciones con un servidor de bases de datos que ejecute Microsoft SQL Server. SSL requiere la instalación de un certificado digital de autenticación en el servidor correspondiente, el mismo deberá ser proveído e instalado por el personal de seguridad informática quien administra la Autoridad Certificadora (CA) de la institución.

Cuando se aplica SSL el cliente utiliza el protocolo HTTPS[10] y el servidor escucha en el puerto TCP[11] 443, por lo que al activar una encripción de datos entre el cliente y el

10 HTTPS , Hyper Text Transfer Protocol Secure – Protocolo Seguro de Transferencia de Hipertexto.
11 TCP, Transmision Control Protocol – Protocolo de Control de Transmisión.

servidor es muy probable que impacte razonablemente el rendimiento de la aplicación, ante esta posibilidad es importante realizar pruebas de tiempos de respuesta antes y después de aplicar el SSL. En el caso que afecte el rendimiento de la aplicación, la recomendación es optimizar el texto en las páginas y usar gráficos más sencillos de manera tal, que el diseño de las páginas sea más "liviano".

Como consideración fundamental, SSL puede funcionar a través de un servidor de seguridad basado en NAT[12], pero en el caso de IPSec[13] esto no es posible.

2. Seguridad del Protocolo Internet (IPSec).

Ofrece una solución para la comunicación segura en el nivel de transporte y puede utilizarse para proteger los datos enviados entre dos equipos, por ejemplo: entre un servidor de aplicaciones y uno de bases de datos, la configuración que se emplea es autenticación mutua. IPSec es transparente para las aplicaciones al implementarse los servicios de cifrado, integridad y autenticación en el nivel de transporte de la red. Las mismas siguen comunicándose entre sí de la forma que habitualmente lo han hecho, mediante puertos TCP.

IPSec proporciona:

- Confidencialidad de mensajes: protege todo el tráfico IP (Internet Protocol) entre dos equipos; SSL es específico de una aplicación concreta.
- Integridad de mensajes entre dos equipos (sin cifrado de datos).
- Autenticación mutua entre dos equipos (no usuarios), restringiendo la comunicación entre sí, también puede limitar la comunicación a protocolos IP y puertos TCP específicos. Por ejemplo, protege a un servidor de base de datos estableciendo un control de acceso para que permita únicamente solicitudes de un servidor aplicativo específico.

[12] NAT, Network Address Translation - Traducción de direcciones de red.
[13] IPSEC, Internet Protocol Security – Seguridad del Protocolo Internet.

La supervisión y configuración de IPSec se realiza utilizando la Consola de Administración de Microsoft conocida por sus siglas en inglés como MMC (Microsoft Management Console). Para validar o probar la configuración se utiliza el monitor propio de la herramienta, el mismo (Ipsecmon.exe) proporciona información acerca de qué directiva IPSec está activa y si se ha establecido un canal seguro entre los servidores.

Cuando se requiere utilizar IPSec para proteger la comunicación entre dos equipos que están separados por un servidor de seguridad (FIREWALL, ISA SERVER) es indispensable asegurarse de que el servidor de seguridad no utiliza la traducción de direcciones de red conocido por sus siglas en inglés como NAT, ya que IPSec no funciona con ningún dispositivo basado en NAT. Recordando además que IPSec requiere que ambos equipos ejecuten Windows 2000 o superior.

3. Cifrado de llamada a procedimiento remoto (RPC).

El protocolo RPC utilizado por COM distribuido conocido por sus siglas en inglés como “DCOM”, proporciona un nivel de autenticación, realiza el cifrado de todos los paquetes de datos enviados entre el cliente y el servidor. Además, ofrece un conjunto de niveles configurables, desde no tener ninguna protección de los datos hasta el cifrado total del estado de los parámetros.

El nivel de cifrado RPC (128 bits por ejemplo) dependerá de la versión del sistema operativo Windows que se ejecute en los equipos cliente y servidor. La configuración del servidor puede definirse en la aplicación, esto se puede realizar mediante atributos de la herramienta de desarrollo, en el caso de Visual Studio .NET se presenta en el momento de crear los componentes o bien mediante la herramienta administrativa durante la implementación.

ASP.NET

Para una aplicación Web basada en ASP.NET v.1.1, el nivel de autenticación utilizado por el cliente se configura mediante el atributo **comAuthenticationLevel** en el elemento **<processModel>** que se encuentra en el archivo **Machine.config**. De esta manera, se proporciona el nivel predeterminado para todas las aplicaciones ASP.NET que se ejecutan en el servidor Web.

Para ejemplificar los tres puntos anteriores (SSL, IPSec y RPC), la figura 4 representa la conexión entre los clientes y los servidores de la compañía. Además se representa como aplicar las tecnologías para solventar el tema de privacidad e Integridad:

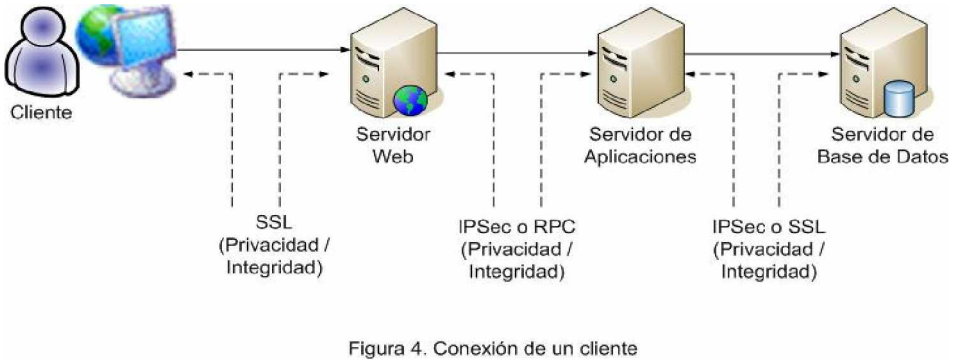

Figura 4. Conexión de un cliente

En el escenario anterior, la seguridad es presentada punto a punto, desglosándolo obtendríamos lo siguientes elementos:

- Explorador del cliente y el servidor Web.
- Servidor Web y servidor de aplicaciones.
- Servidor de aplicaciones y servidor de bases de datos.

Para proteger los datos confidenciales enviados entre los equipos, deberá utilizarse las siguientes recomendaciones para el caso que aplique:

- Autenticación mediante formularios, se requiere proteger las credenciales (código de usuario y contraseña) enviadas a un servidor Web desde un formulario de inicio de sesión. En este caso es necesario proteger el acceso a todas las páginas con SSL (y no sólo la página de inicio de sesión) con el fin de garantizar que la “cookie” de autenticación generada a partir del proceso de autenticación inicial permanezca segura durante toda la sesión de explorador del cliente con la aplicación.

- Si la aplicación transmite datos confidenciales del explorador del cliente al servidor Web (y viceversa), como por ejemplo, números de tarjeta de crédito o información de cuentas bancarias, se debe emplear SSL.

- El canal de transporte entre un servidor Web y un servidor de aplicaciones deberá protegerse con IPSec y/o SSL, en general, se recomienda recurrir SSL porque el servicio Web utiliza el transporte HTTP. SSL permite cifrar solamente los datos enviados hacia y desde el servicio Web (y no todo el tráfico entre los dos equipos), por su parte IPSec realiza el cifrado de todo el tráfico entre los dos equipos.

- Si la aplicación utiliza uno o varios componentes de .NET (mediante .NET Remoting) y los mismos se conectan mediante el canal TCP, se sugiere IPSec para proporcionar un vínculo de comunicación segura. Si tiene componentes de .NET en ASP.NET se puede manejar con SSL (configurado mediante IIS).

- Para proteger los datos enviados entre un servidor de aplicaciones y un servidor de bases de datos, se puede aplicar IPSec. Si el servidor de bases de datos ejecuta SQL Server puede usar SSL. Para esto se requiere que esté instalado un certificado digital de autenticación, como consideración SSL sólo funciona con TCP/IP (el protocolo de comunicación recomendado para SQL Server).

Además de los puntos anteriores, es imprescindible contar con un dispositivo de Firewall entre las diferentes capas, en el mismo se prepara una lista de control de acceso (ACL[14]) para los servidores, con ello se habilitará únicamente las direcciones IP y puertos de comunicación indispensables, los demás quedarán bloqueados.

En una empresa a pesar de que la exposición de sus aplicaciones en la Intranet está restringida a un grupo limitado de usuarios, no se puede despreocupar por desarrollar estrategias para la autenticación y comunicación segura.

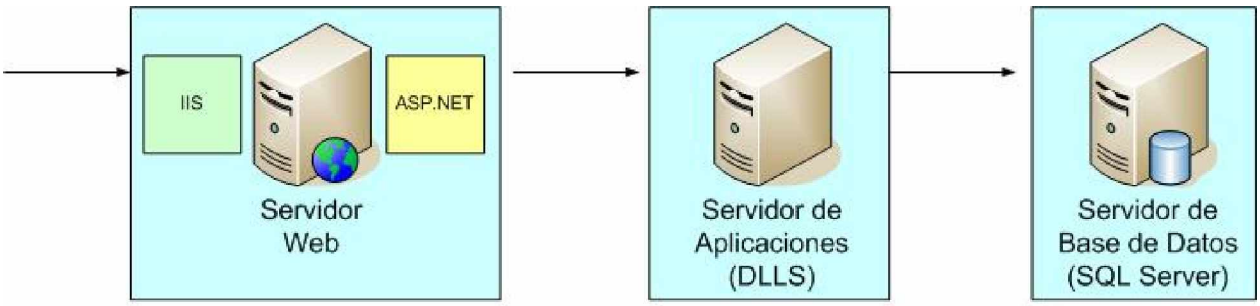

Figura 5. Segmentación de capas

Tal y como se muestra en la figura 5, existe una segmentación de tres capas, por lo que las recomendaciones en el lineamiento son las siguientes:

- Deshabilitar el acceso anónimo para el directorio raíz virtual de la aplicación Web.

- Configurar la aplicación Web ASP.NET para que utilice la autenticación de Windows, para ello se edita el archivo Web.config en la raíz del directorio virtual de la aplicación Web y se establece el elemento <authentication> en: <authentication mode="Windows" />.

[14] ACL, Access Control List – Lista de Control de Acceso.

- Configurar la aplicación Web ASP.NET para la suplantación, se edita el archivo Web.config en el directorio virtual de la aplicación Web, estableciendo el elemento <identity> en: <identity impersonate="true" />.

- Configurar el nivel de suplantación de DCOM utilizado por la aplicación Web ASP.NET para las llamadas salientes. La aplicación Web ASP.NET llama a los componentes remotos a través de DCOM y el nivel predeterminado que se utiliza para las llamadas salientes desde ASP.NET es Impersonate (suplantar). Se recomienda cambiar a Delegate (delegar) en Machine.config, en este caso se edita el archivo “Machine.config”, se busca el elemento <processModel> y se establece el atributo comImpersonateLevel en "Delegate".

- Configurar el nivel de autenticación de DCOM en el cliente, los niveles de autenticación están determinados tanto por el cliente como por el servidor. En este caso, el cliente DCOM es ASP.NET, por lo que se debe editar el archivo “Machine.config”, buscar el elemento <processModel> y establecer el atributo comAuthenitcationLevel en "PktPrivacy".

El objetivo en las empresas es desarrollar productos de calidad y en este caso el software seguro es un subconjunto de la calidad y la confiabilidad del mismo.

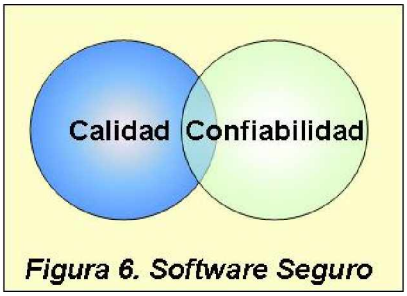

Figura 6. Software Seguro

Si bien es cierto, lo anterior no es tan sencillo como parece, no se trata de un software insuperable o bien de la seguridad perfecta, se habla de un desarrollo de software lo suficientemente seguro y confiable para soportar el ambiente en el cual está operando.

Efectivamente, la seguridad absoluta en cualquier ámbito no existe, pero dado que la misma está basada en las probabilidades, si es posible generar aplicaciones o procesos minimizando a niveles razonables el riesgo.

Metodología

El contar con una metodología adecuada facilitará definitivamente la disminución del riesgo en la creación de las aplicaciones, la misma deberá tener ciertas características que sean de valor agregado para la empresa y no se convierta en un simple requisito que se debe cumplir con la finalidad de implementar las soluciones en un ambiente de producción.

Características de la metodología:

- Alineada a la política de seguridad de la empresa y acompañar los cambios de la misma, considerando que el entorno tecnológico cambia rápidamente.

- Gradual y enfocada en los puntos claves del proceso del desarrollo de sistemas, de forma que minimice el impacto en caso de encontrarse alguna inconsistencia o algún posible riesgo. De esta manera, se trata de contar con varios puntos de chequeo y en el caso de tener que corregir o modificar el código de la aplicación no esperarse hasta el final.

- Completa y de un alcance extenso: no debe limitarse a evaluar solamente la aplicación que está en construcción o adquisición, sino enfocarse de igual manera a los procesos informáticos relacionados, procesos operativos, interfases con sistemas ya existentes e infraestructura.

- Iterativa: al realizarse gradualmente - en diversas oportunidades - deberá tener un alcance y profundidad en cada iteración, dependiendo de la etapa de desarrollo.

- Diversidad: debe considerar y abarcar las diferentes tecnologías en la compañía, por lo que es común la adquisición de nuevos sistemas, cambios o mejoras a los

mismos y no necesariamente utilizan la misma herramienta de desarrollo o la misma infraestructura.

- Procesos de revisión: al ejecutarse cada iteración se debe contemplar la posibilidad de que queden puntos pendientes de resolver, por lo cual se debe establecer un proceso de revisión y seguimiento de los mismos.

La metodología que se recomienda para el cumplimiento de los puntos anteriores se conforma básicamente de una lista de verificación ("Checklist"), la misma se aplicará en las diferentes etapas del proceso de desarrollo y consiste de una serie de evaluaciones que se activan en las diferentes etapas del ciclo de vida de la aplicación y no interfiere con el proceso de desarrollo del software.

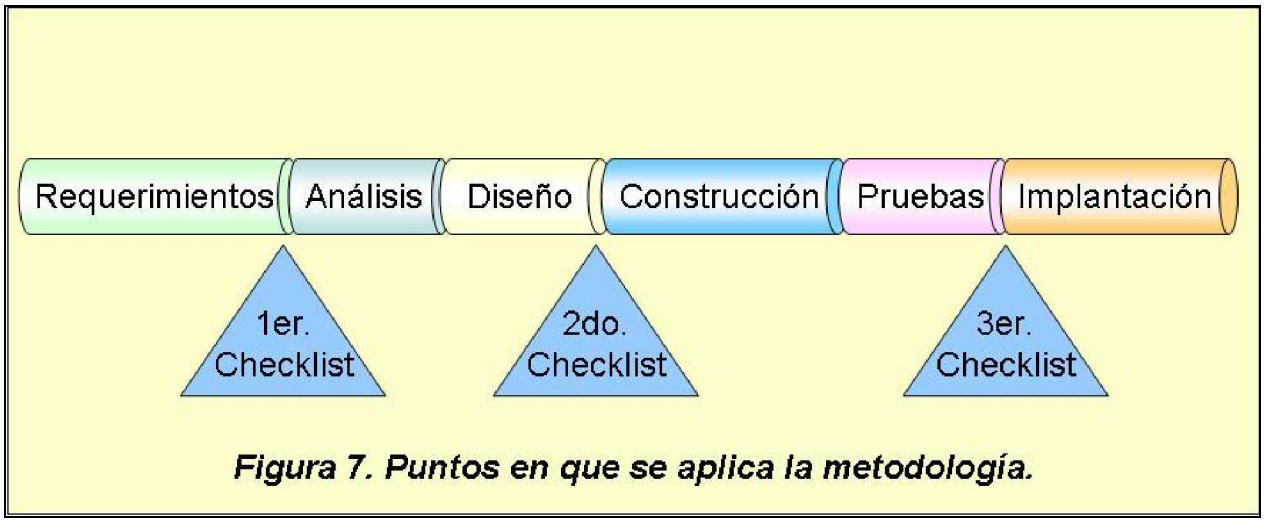

Figura 7. Puntos en que se aplica la metodología.

La lista de verificación permite definir cuáles son los requerimientos de seguridad a implementarse y el impacto que tendrá en la infraestructura. Es importante contar con un modelo de infraestructura global de la empresa, el cual debe validarse y actualizarse conforme crecen las aplicaciones en la compañía.

Es imprescindible definir correctamente los requerimientos de seguridad desde el inicio del proyecto de manera tal, que quede definido el dimensionamiento de la

infraestructura y los costos en recursos (tiempo y dinero). Un estudio realizado por ISACA[15] demostró formalmente que el retorno de la inversión en seguridad es mayor cuanto antes se incluya en el proceso de desarrollo de sistemas.

Lista de verificación

El objetivo es definir un marco de seguridad en el cual se debe desarrollar la implementación de las soluciones. Se refiere a un marco porque en la etapa de análisis quizás haya algunos puntos que no pueden precisarse en detalle, esto dependerá específicamente de la empresa y su entorno.

A continuación se presentarán algunos ejemplos de cómo se compone una lista de verificación:

1. Identificación y autenticación

    - Existencia de una identificación acorde al registro de la información, uso de la misma y riesgo del sistema. ¿Cuáles serán las técnicas de identificación? (contraseñas estáticas, dinámicas, tokens, controles biométricos, etc.)
    - ¿Se requiere autenticación de servidores y/o de usuarios de Internet? En este ejemplo se vislumbraría la utilización de certificados digitales.
    - ¿Se requiere un único login integrado? Esto podría tratarse de utilizar SSO (Single Sing On).

2. Autorización y control de acceso.

    - ¿Existe una definición de todos los requerimientos de autorización y control de acceso para los usuarios del sistema?

[15] Information System Audit and Control Association

- ¿El sistema incluye acceso a datos confidenciales mediante Intranet / Internet? Y de ser así ¿Existe una definición sobre la restricción de acceso a dicha información?
- ¿Se ha identificado la comunidad de usuarios y roles de los mismos? ya que el acceso a la información se basa exclusivamente en la responsabilidad que tengan los usuarios de la aplicación.

3. Integridad y confidencialidad de los datos

- ¿Se ha identificado que la información requiere protección en su confidencialidad e integridad en el almacenamiento y/o en el transito? Ya que de comprobarse se entraría en un tema de encripción, controles compensatorios o firma digital.
- ¿Se ha ofrecido el uso de encripción a proveedores o terceras partes y determinado de quién sería el costo en caso de aceptación de los mismos?

4. No repudio

- ¿Es requerido el no repudio de mensajes o transacciones? Y de ser afirmativo el punto anterior ¿Con cuál técnica?
- Para las aplicaciones de Internet ¿Se requiere certificados digitales o firma digital?
- Si se requieren certificados digitales para los usuarios (clientes) ¿Se ha definido de quién va a ser el costo?
- De necesitarse infraestructura PKI (Public Key Infraestructure) ¿Se dispondrá de una propia o se realizará outsourcing del servicio? Se debe considerar los costos para la empresa y los clientes.

Estándares Internacionales

Varios estándares de seguridad se refieren a los componentes claves de las aplicaciones, incluyendo confidencialidad, integridad y disponibilidad. Los mismos, enfatizan determinadas áreas de seguridad, tales como los controles de acceso, ciclo de vida de las aplicaciones y controles criptográficos, a saber:

- ISO15408
- ISO17799
- COBIT
- SP800-14
- SP800-27

Los anteriores son documentos y estándares ampliamente utilizados, adoptados por muchas organizaciones a nivel internacional.

ISO 15408

El ISO15408 “Evaluation Criteria for IT Security” tiene por objetivo prevenir la divulgación no autorizada, modificación o imposibilidad de uso. Fue publicado en 1999 por un consorcio de organizaciones de estándares de los Estados Unidos y Europa, licenciado a la ISO como ISO/IEC15408. Puede ser usado como aseguramiento de un producto o como garantía dada por el fabricante acerca de las capacidades en cuanto al tema de seguridad se refiere, por una parte el cliente lo puede utilizar para determinar si una aplicación o sistema cumple con sus requerimientos, por la otra, los desarrolladores lo utilizan como guía para la etapa de diseño y construcción de software.

Los requerimientos funcionales de seguridad están agrupados en 11 clases cada uno con un número de “familias” que tiene un objetivo y comportamiento de seguridad específico y deseado. Dichas clases son auditoria de la seguridad, identificación y autenticación, utilización de recursos, soporte criptográfico, manejo de la seguridad, controles de acceso, comunicaciones, privacidad, canales o caminos de confianza

conocido en inglés como “trusted paths/channels”, protección de datos del usuario así como de las funciones de seguridad.

ISO 17799

La Organización Internacional para la Estandarización (ISO) emitió el ISO17799 (“Code of Practice for Information Security Management” – Código de Practicas para la Gestión de la Seguridad de la Información) en el año 2000. Está basado en el estándar británico BS7799, publicado por primera vez en 1995. Este es un amplio conjunto de controles los cuales son considerados las mejores prácticas en lo referente a la seguridad de la información, incluyendo políticas, prácticas, procedimientos, estructura organizacional y funcionalidad del software.

Los controles detallados están organizados en 10 secciones: políticas de seguridad, organización de la seguridad, clasificación control de activos, seguridad del personal, seguridad física y ambiental, gestión de comunicaciones y operaciones, controles de acceso, desarrollo y mantenimiento de sistemas, gestión de la continuidad del negocio y cumplimiento. El objetivo básico de la seguridad de las aplicaciones de acuerdo a la ISO17799 es prevenir las pérdidas y la modificación o mal empleo de la información de la empresa.

CobiT

Publicado por el IT Governance Institute y la Information System Audit and Control Foundation (ISACF). Es un marco de referencia sobre las mejores prácticas de control y gobierno de TI. Esta guía describe un rango de criterios informativos para desarrollar un programa de seguridad comprensivo, en soporte a las necesidades del negocio. Los criterios de información fueron desarrollados a partir de otros modelos de seguridad e integración de conceptos de efectividad, eficiencia, confiabilidad y cumplimiento, así como conceptos tradicionales de confidencialidad, integridad y disponibilidad.

CobiT incluye 34 procesos de Tecnologías de Información (TI) agrupados en cuatro dominios:

- Planificación y organización
- Adquisición y Mantenimiento
- Entrega y soporte
- Monitoreo

Los procesos están divididos en 318 objetivos de control específicos y asociados a guías de auditoria.

SP800-14

Publicado por el Instituto Nacional de Estándares y Tecnología (NIST) en el año 2001, presenta que la seguridad de las aplicaciones se considera en las prácticas específicas de “planeación del ciclo de vida” y “consideraciones de seguridad en el soporte y operaciones de las computadoras”, incluye cinco fases:

- Iniciación: una evaluación primaria debe ser ejecutada.
- Desarrollo / adquisición: los requerimientos de seguridad deben ser documentados e incorporados en las especificaciones.
- Implementación: debe solicitarse la realización de pruebas y la aprobación de la gerencia.
- Operación: las actividades de seguridad deben ser continuas con apropiado monitoreo y auditoria.
- Discontinuación: los datos deben ser removidos y los medios borrados.

SP800-27

Incluye 33 principios de seguridad que aplican a la fase de planificación del ciclo de vida descritos en el SP800-14, su principio se consolida en establecer una política de seguridad como base para diseñar, enfocado en políticas, riesgos y características de mejores prácticas, por ejemplo: simplicidad, controles de acceso, manejo de datos y entrenamiento de desarrolladores en técnicas de seguridad.

Los estándares anteriormente mencionados se enfocan en la seguridad de las aplicaciones desde diferentes puntos de vista, sin embargo, los mismos comparten controles comunes en diversas áreas, tales como desarrollo del ciclo de vida y tratamiento de los datos, entre otros.

Tabla 1, resumen de estándares.

	ISO15408	ISO17799	COBIT	SP800-14+27
Enfoque a la seguridad	Prevenir divulgaciones no autorizadas, modificaciones o pérdidas de información. Ofrecer guías de seguridad para desarrolladores y usuarios.	Prevenir pérdidas, modificaciones o mal uso de datos. Proveer controles de datos, criptográficos y el desarrollo de las aplicaciones.	Proporcionar seguridad y controles sobre procesos de TI, planificación, adquisición, desarrollo, soporte y monitoreo de aplicaciones.	Procurar prácticas y controles sobre cada una de las fases del ciclo de vida del desarrollo de software. Facilitar controles operacionales.
Objetivos	Brindar un estándar para la medición de la seguridad asociada con un producto de TI.	Proporcionar un conjunto de controles que comprenden las mejores prácticas en seguridad de la información.	Proveer al gobierno de TI un enfoque a los requerimientos de negocio, procesos y recursos de TI.	Suministrar prácticas de seguridad para uso de la información, protección y diseño para el ámbito gubernamental.
Componentes claves	Requerimientos de seguridad, funcionales y de aseguramiento, utilizados para evaluar productos de TI. Niveles de evaluación del aseguramiento.	4 dominios cubriendo todos los aspectos de seguridad de TI.	4 Dominios: Planificación y organización, adquisición e implementación, entrega y soporte, y monitoreo.	8 principios y 14 prácticas (sp800-14) y 33 principios de seguridad en las fases del ciclo de vida (sp800-27)
Función de negocio	Desarrollar aplicaciones y productos de TI.	Apoyar la tecnología de información	Apoyar la tecnología de información	Apoyar la tecnología de información
Audiencia	Clientes, desarrolladores y evaluadores	Gerentes y usuarios	Gerentes, usuarios y auditores	Gerentes, usuarios, auditores y desarrolladores

Conclusión

No hay aplicaciones seguras en una infraestructura insegura y viceversa, por lo que la seguridad informática busca la protección contra los riesgos asociados a la misma en función de varios elementos:

1. Proteger de amenazas que pesan sobre los activos (datos) y las vulnerabilidades de los mismos.
2. Resguardar la sensibilidad de la información, la cual se puede dividir en :
   - Confidencialidad.
   - Integridad.
3. Garantizar una infraestructura robusta, escalable, disponible, que brinde continuidad del negocio y sobre todo segura.
4. Crear y divulgar lineamientos con las mejores prácticas de la industria, los cuales vayan alineados acorde con las políticas e infraestructura de la compañía.
5. Utilizar una metodología para la evaluación de la seguridad, la cual debe aplicarse en diferentes etapas del proceso de desarrollo del software.

El activo más importante que poseen las compañías independientemente de su tamaño es la información y por lo tanto deben existir técnicas que la aseguren, más allá de la seguridad física que se establezca sobre los equipos se deben crear políticas y lineamientos para el desarrollo, implementación y administración de las soluciones informáticas.

Es fundamental comprender que los lineamientos tanto para el desarrollo del software como para la seguridad de la infraestructura es algo que no debe establecerse y luego olvidarse. Las técnicas de intrusión están cambiando continuamente, razón por la cual los lineamientos y políticas seguridad se deben revisar constantemente esto con el fin de garantizar que las aplicaciones y la infraestructura estén protegidas de forma adecuada, convirtiéndose de esta manera un proceso evolutivo, iterativo e incremental.

Referencias Bibliográficas

Libros

Howard, Michael & LeBlanc, David. 2003. Writing Secure Code, Second Edition USA: Microsoft Press.

Howard, Michael. 2000. Designing Secure Web-Based Applications for Microsoft Windows 2000. USA: Microsoft Press.

Internet

Information System Audit and Control Association (ISACA). 2002. http://www.isaca.org

- "A survey of application security in current international standars"

Microsoft Corporation. 2004. http://www.microsoft.com/wssra

- Windows Server System Reference Architecture WSSRA.

Microsoft Corporation. 2005. http://msdn.microsoft.com/practices

- Pattern & practices. 2002. "Arquitectura de aplicaciones .Net: Diseño de aplicaciones y servicios".

Microsoft Corporation. 2004. http://support.microsoft.com

- Q257591, "Description of the Secure Sockets Layer (SSL) Handshake"
- Q257587, "Description of the Server Authentication Process During the SSL Handshake"
- Q257586, "Description of the Client Authentication Process During the SSL Handshake"
- Q233256, "HOW TO Enable IPSec Traffic through a Firewall"
- "Windows Server 2003 Security Guide"

  www.microsoft.com/technet/security/prodtech/Windows/Win2003/W2003HG/SGCH00.asp