Errores comunes en el desarrollo de servicios web

Cada día es más frecuente la creación de nuevos servicios en Internet por parte de las empresas. En un principio, el uso de Internet ha perseguido objetivos publicitarios o el suministro de funcionalidades adicionales que complementaran la actividad principal de la empresa, sin embargo, hoy en día la base de muchos negocios reside en la disponibilidad de los servicios ofrecidos en Internet. La información que se maneja cada día es más delicada y los servicios más complejos. Esto hace necesario dotarlos de medidas de seguridad necesarias que los hagan confiables. La seguridad debe ser un aspecto determinante, presente en todas las fases (antes, durante y después) de la puesta en marcha de un servicio y en todos los elementos que lo forman.

Joaquín Crespo Pérez.

La World Wide Web (WWW) hace mucho tiempo que se ha consolidado como el lugar para la publicación de servicios en Internet por excelencia. A través de la utilización de estándares como el protocolo HTTP o el lenguaje HTML es posible ofrecer servicios con una única plataforma cliente (el concepto de cliente universal), sencillos de implementar y con interfaces amigables. Estas características han contribuido notablemente a su crecimiento exponencial.

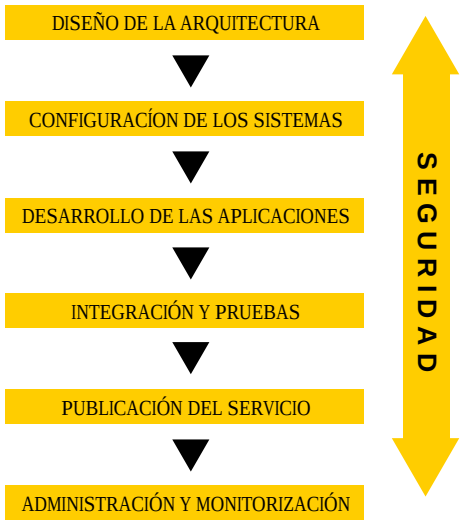
Aunque en un principio la WWW se pensó como una forma de publicación de información estática, hoy en día la interacción con el usuario final es fundamental, y los servicios ofrecidos cada día dependen más de ésta. Este aumento de interactividad con el cliente ha hecho que en la actualidad la WWW se haya convertido en una plataforma donde, además de publicarse contenidos, se ejecutan aplicaciones. Los riesgos que ésta circunstancia presenta son evidentes: antes el control total lo tenía el servidor, ahora cada vez se le da más libertad al cliente. Estas mismas libertades son puertas que ofrecen la posibilidad de ejecutar acciones no deseadas o inesperadas, y eso resulta extremadamente peligroso, ya que esta circunstancia convierte a todo cliente en una fuente de riesgo potencial, es decir, alguien que puede malversar el servicio ofrecido.

A la hora ofrecer un nuevo servicio web, hay que tener en cuenta muchos parámetros que tendrán una relevancia importante en la calidad y fiabilidad del mismo. Las decisiones que deben tomarse son muchas y todas ellas deben ser consensuadas por todas las partes que intervienen en la oferta del servicio, desde el departamento de sistemas encargado de la configuración de los equipos hasta el área de desarrollo de las aplicaciones. Cada uno intervendrá en muchas fases de diseño e implementación, como por ejemplo: el diseño de la arquitectura, los procedimientos de integración, los requisitos de disponibilidad, los métodos de administración y las políticas de seguridad aplicables.

Es muy importante que todos los grupos involucrados en la creación de un nuevo servicio pongan todos los medios para dotar al sistema de las medidas de seguridad necesarias. La seguridad no debe considerarse como algo secundario, sino como parte esencial (y requisito necesario) de todo el proceso de creación del servicio. Con la particular de ser un requisito que, como otros, continua una vez que el servicio se encuentre en producción.

El desarrollo de un servicio en Internet está típicamente organizado en una serie de fases:

- Diseño del servicio.
- Selecciones tecnológicas (arquitectura y plataformas)
- Implementación hardware del servicio.
- Implementación software del servicio.
- Puesta en producción.
- Control y evolución del servicio en producción.



La importancia de la tecnología seleccionada.

La selección de la tecnología es quizás la primera decisión importante que se debe tomar en la fase de diseño de un nuevo servicio. Aquí se incluyen la selección de plataformas (considerando partes dispares como los sistemas de ejecución de las aplicaciones y los sistemas de bases de datos), el diseño de los elementos que interviene en la oferta del servicio y la interacción entre ellos. Estas cuestiones condicionan muchas de las decisiones futuras que se deben tomar en torno al servicio y que afectan al desarrollo de las aplicaciones, la configuración de los equipos, los métodos de administración, etc.

Un parámetro fundamental en la fase de diseño del servicio es la selección de la arquitectura de red que se implementará. Una arquitectura adecuada a los servicios proporcionados resulta de vital importancia para la seguridad del sistema global. El impacto que pueda tener la vulneración de un servicio es muy distinto en función de la arquitectura implantada. Por ejemplo, no es lo mismo la vulneración de un servidor en el que se encuentra el software servidor web y la base de datos en el mismo equipo, que si estuviesen disociados en distintos equipos, separados por equipos cortafuegos y en zonas desmilitarizadas (DeMilitarized Zones, DMZ) distintas. Éste es uno de los aspectos más importantes en el aprovisionamiento del servicio, en el que deben participar todas las partes de la organización implicadas: configuración de los equipos, configuración de los elementos de interconectividad de la infraestructura, desarrollo, monitorización y gestión, etc.

En cuanto a la selección de la plataforma, resulta muy importante tener en cuenta las implicaciones de seguridad que puedan afectar a ésta:

- Vulnerabilidades conocidas, solucionadas y frecuencia de aparición de nuevas vulnerabilidades.
- Método de publicación de problemas de seguridad y sus posteriores soluciones por parte del fabricante, considerando el tiempo de respuesta de éste para la publicación de parches de los sistemas.
- Soporte ofrecido por el fabricante y documentación para la correcta securización de los sistemas.

Sin embargo los criterios no son exclusivamente tecnológicos, sino que además deben valorarse las condiciones de la organización en cuanto a su capacidad de utilización de tecnologías que pueden venir condicionadas por la disponibilidad de personal cualificado de la empresa, políticas de homogenización, y muchas otras características intrínsecas.

La configuración de las plataformas seleccionadas e incluso los propios problemas que pueda presentar el software servidor resulta de vital importancia para proporcionar el servicio con un cierto nivel de seguridad. Las vulnerabilidades que puedan aparecer tendrán un mayor o menor impacto en el sistema, facilitando la tarea de los atacantes o incluso comprometiendo al sistema completo de forma inmediata.

Algunos ejemplos habituales de vulnerabilidades en plataformas de servicios web en Internet (servidores WWW) son:

- Revelación de contenido de directorios: Una mala configuración del servidor o los permisos de acceso puede llevar a la visualización de ficheros que no se han publicado de forma explícita. En muchas ocasiones, esta circunstancia permite conseguir ficheros con información confidencial, dado que es frecuente que en los mismos directorios que se publican las paginas encargadas de servir los contenidos a los clientes se publiquen versiones antiguas de las mismas generalmente renombradas, ficheros de cabecera, ficheros de contraseñas, etc.

- Escalado de directorios: Mediante la introducción de ciertos caracteres en la petición realizada al servidor es posible navegar libremente por algunos de los directorios del servidor, en muchos casos incluso estando fuera de los límites del árbol de contenidos del servidor. Ésta circunstancia puede permitir la revelación de información no accesible directamente desde los contenidos del servidor.

- Visualización de código fuente de páginas dinámicas: En algunos casos algunas vulnerabilidades del servidor, permiten revelar el código fuente de una aplicación en lugar de ejecutarla, facilitando así la tarea de intento de vulneración de las aplicaciones, ya que ésta se reduce a la realización una auditoria del código de las mismas.

- Ejecución directa de código: Algunos errores de diseño pueden llevar a la ejecución de comandos dentro del sistema que ejecuta la aplicación servidora. Estos errores son habitualmente sobrecargas de buffer aunque en algunos casos puede estar relacionados con algunas de las anteriores vulnerabilidades (por ejemplo, con el escalado de directorios en el servidor web Internet Information Services de Microsoft [btq-2002]).

Otras posibles vulnerabilidades se refieren a una mala gestión a la hora de decidir los servicios que serán accesibles desde el exterior. Un ejemplo de este tipo de problemas puede ser la posibilidad de acceder a páginas de administración ocultas: un servidor ofrece sus servicios en <http://www.servidor.com> y la página de administración del mismo se localiza en <http://www.servidor.com/admin>, dirección (Universal Resource Locator, URL) que es posible alcanzar aunque no se publique a partir de los contenidos del servidor. En estos casos los diseñadores suponen que si el acceso al servicio de administración no es público (no está enlazado desde los contenidos del servidor) no va a ser alcanzable y no se restringe necesariamente el acceso al mismo. Esta circunstancia, denominada seguridad por oscuridad, nunca resulta conveniente.

También es importante la selección del método de publicación de contenidos en el servidor. Existen múltiples alternativas como es el uso de transferencias de FTP, o mediante protocolos como WebDAV, ya que una vulnerabilidad en cualquiera de los métodos de publicación podría permitir a un atacante la visualización, modificación o incluso creación no autorizada de contenidos en el servidor. Por tanto, la seguridad de un servicio no reside únicamente en la seguridad de las aplicaciones y las plataformas que lo ofrecen, sino también la de todos los elementos que lo rodean y que forman parte, directa o indirectamente de éste.

Una vez finalizada la implantación del servicio, aparece de nuevo un aspecto muy ligado a las plataformas: los procedimientos de parcheo y actualizaciones. Es necesario un procedimiento que minimice el impacto en el servicio, asegurando su correcto funcionamiento en todo momento. No es recomendable la implantación de un procedimiento de parcheo automático, siendo más fiable hacerlo de forma manual monitorizando que los parches o actualizaciones instaladas no tengan ningún efecto negativo sobre el servicio. Es por tanto imprescindible disponer de un entorno réplica al entorno de producción en el que aplicar los parches para realizar dichas pruebas. Un error común, que sólo se hace evidente cuando surge un problema de seguridad (y su consiguiente parche), es la no existencia de estos sistemas de preproducción.

Desarrollo de las aplicaciones.

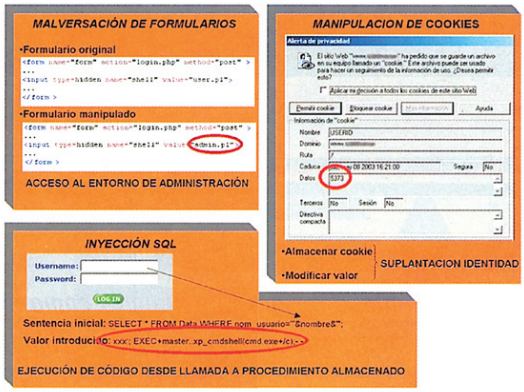
La interacción con el usuario final, el manejo de grandes cantidades de datos (generalmente almacenados en bases de datos), el suministro de servicios en función de la identificación del usuario y demás aspectos que han convertido a la WWW en una plataforma de ejecución de aplicaciones, hacen que el desarrollo de las mismas sea muy complejo, provocando la presencia de un gran número de puntos críticos que deben ser considerados.

La seguridad de las aplicaciones debe ser una premisa que debe estar presente en toda tarea de desarrollo, teniendo siempre en cuenta que cualquier cliente que haga uso del servicio puede ser un potencial atacante para el mismo.

Es fundamental disponer de una metodología de desarrollo estructurada y con una serie de objetivos claros, como son facilitar las tareas de revisión y modificación de código, actualizaciones y ampliaciones de los servicios proporcionados, permitiendo que en todo momento se tenga el control por parte del servidor de lo acontecido en la ejecución de las aplicaciones evitando así que puedan ejecutarse acciones imprevistas que pudieran dar pie a la vulneración del servicio.

Algunos de los problemas comunes en el desarrollo de aplicaciones web son:
- Implementaciones poco robustas de mecanismos de autenticación o de mantenimiento de sesiones.
- Errores en el tratamiento de entrada de datos: el incorrecto procesamiento de éstos puede llevar a una malversación por parte del atacante.
- Ejecución de aplicaciones con máximos privilegios sobre el sistema: como consecuencia de esto un fallo en la aplicación puede comprometer completamente a la plataforma de ejecución.
- Tratamiento incorrecto de los posibles errores y los mensajes generados al producirse éstos.

Las tareas de desarrollo de las aplicaciones deben realizarse en sintonía con la parte de la organización



Debe haber un equilibrio entre la puesta en producción del servicio y la seguridad con la que éste es ofrecido, siendo tareas complementarias que no deben entorpecerse entre sí.

implicada en las tareas de administración y configuración de los sistemas en los que se basará la ejecución de las mismas. La comunicación entre ambos departamentos debe ser fluida ya que hay muchas decisiones que tomar en conjunto: privilegios de los usuarios de ejecución de las aplicaciones, limitaciones de acceso a ciertas llamadas del sistema, comunicaciones entre los distintos elementos de la arquitectura implantada en caso de arquitecturas multinivel, etc.

Incidiendo más en algunos detalles del desarrollo de las aplicaciones, a continuación se enumeran y relatan una serie de puntos típicos en este tipo de aplicaciones y cómo pueden convertirse en potenciales puntos de vulneración por parte de un atacante.

Autenticación.

La implantación de medios de autenticación segura dificulta que un usuario no autorizado acceda a servicios sin autenticación previa. Si los métodos de autenticación resultan débiles un usuario podría conseguir de forma no autorizada las credenciales necesarias para acceder al servicio. El impacto de este acceso no autorizado depende del nivel de criticidad de la información a la que se accede, por ejemplo, no es lo mismo que un usuario consiga falsear la autenticación de un servicio bancario, que en un servicio de votaciones de películas favoritas.

Lo común en este tipo de situaciones es la implantación de un método de autenticación basado en usuario / contraseña, método que normalmente resulta poco robusto, ya que es un método de autenticación que, con tiempo y recursos suficientes, es posible vulnerar.

En muchos casos, los mecanismos para conservar sesiones suelen recaer en cookies alojadas en el cliente. Esta es la única tecnología aplicable para el mantenimiento de sesiones en un protocolo sin estados como es HTTP. Sin embargo, las cookies pueden ser modificadas por un usuario malicioso, si los campos son fácilmente descifrables ésta tarea es sumamente sencilla. Así, un atacante podría manipularlas consiguiendo suplantar la identidad de otro usuario, robar sesiones activas, etc.

Las medidas de seguridad posibles a este nivel son muchas y muy variadas: implantación de números máximos de reintentos de autenticación, obligación de generación de claves robustas, obligación de rotación de claves, generación de cookies con contenidos aleatorios asociadas a las sesiones, etc.

El método más seguro de autenticación extremo a extremo es la utilización de certificados digitales para los clientes. La gran ventaja de estos es que pueden ser almacenados en dispositivos hardware que no puedan ser manipulados y no desvelen la clave privada. Sin embargo, hay que tener en cuenta que la responsabilidad final siempre recae en el cliente que es quien posee la parte privada del certificado, existiendo por tanto la posibilidad de su utilización inadecuada, al igual que una contraseña.

Tratamiento de entrada de datos

Los errores producidos en el diseño de formularios son errores muy típicos y conocidos desde hace tiempo (existen referencias en documentos del WWW Consortium por lo menos desde 1997) en el desarrollo de aplicaciones web, sin embargo, se siguen produciendo frecuentemente en la actualidad. Algunos errores comunes son:

- Introducir contenido sensible en los campos de los formularios que no se muestran directamente al usuario (utilización de campos hidden en ficheros HTML) pero que sin embargo se revelan al analizar el código de la página.
- Pensar que la limitación del tipo de datos y tamaño de cada campo en el formulario HTML aporta seguridad, sin comprobarlo en otros puntos, como puede ser la aplicación que recibe los datos.
- Comprobar los parámetros en el lado del cliente (por ejemplo con programación mediante Javascript o dentro de los applets Java) pero no en el lado del servidor (el de la aplicación).

La malversación de los campos de formularios es sumamente sencilla para un atacante y no son necesarios demasiados conocimientos técnicos. Puede, por ejemplo, almacenar el código de la página en local, modificar los parámetros deseados, sustituir las referencias relativas en los métodos POST por referencias absolutas y abrir el formulario localmente. Además, cuando el procesamiento de los campos incluidos en el formulario se haga mediante un método GET, la manipulación de éstos podría hacerse directamente modificando el URL que muestra el navegador.

El impacto de estos errores de diseño es muy variado en función del desarrollo concreto de la aplicación. En algunos casos, dependiendo del cuidado que se haya tenido en la programación, puede ser incluso posible la ejecución de forma remota de código en la plataforma malversando llamadas al sistema o llamadas a programas en el que los datos proporcionados por el usuario son utilizados como parte de los parámetros.

En algunas ocasiones, los problemas de tratamiento de datos se pueden propagar desde el sitio donde éstos se ejecutan hacia otras plataformas. Por ejemplo,

Guía básica de protección de un servicio web

Se enumeran a continuación algunas guías básicas para ofrecer un servicio con cierto grado de seguridad.

- Configurar los equipos implicados en la dotación del servicio siguiendo las recomendaciones necesarias para evitar problemas de seguridad, actualizando el software a las últimas versiones y con el nivel de parcheo que ofrezca las máximas garantías.
- Dotar al sistema de las medidas de seguridad necesarias para los servicios que no deberían estar disponibles al exterior (acceso restringido y controlado a servicios de publicación de contenidos, administración, etc.)
- Implementar métodos robustos de autenticación y mecanismos que eviten ataques de fuerza bruta contra los parámetros de identificación del usuario. El mecanismo más robusto de autenticación consiste en la utilización de certificados digitales.
- Utilizar para las sesiones cookies con contenido generado aleatoriamente que evite su manipulación.
- Verificar cualquier parámetro introducido por el cliente, considerando que cualquier cliente puede atacar al sistema, de esta forma se evitan las secuencias que puedan provocar resultados inesperados en el servidor.
- Comprobar los parámetros introducidos por el usuario en el servidor aunque también se haga en el cliente. De nada sirve que la única comprobación de parámetros se haga mediante JavaScript (que se ejecuta en el cliente) ya que éste puede saltarse las restricciones impuestas (modificándolo en local o haciendo solicitudes de forma directa).
- Verificar las páginas de error. Es importante que las páginas de error que pudiera provocar cualquiera de las aplicaciones alojadas en el servidor no contengan información sensible que podría ser aprovechable por un atacante. Suele ser útil la redirección a una página de error genérica en caso de que se produzca cualquier problema en las aplicaciones ofrecidas por el servidor.
- Realizar auditorias de seguridad de las aplicaciones antes de su puesta en producción, así como auditorias periódicas para comprobar el estado del servicio de forma continuada.
- Formar adecuadamente a los desarrolladores para que conozcan los riesgos a los que estarán expuestas sus aplicaciones, de esta forma podrán contribuir notablemente a reducir los defectos de seguridad e incrementar la calidad de los desarrollos.

son muy comunes las arquitecturas en los que un componente fundamental del servicio ofrecido son las bases de datos. Éstas permiten almacenar y consultar datos referidos al registro, preferencias o datos del servicio de los usuarios. La interacción entre las aplicaciones Web y las bases de datos se realiza ejecutando de forma remota (a través de interfaces de acceso definidos) sentencias en el lenguaje estructurado de consultas (Structured Query Language, SQL). De esta forma es posible construir dinámicamente páginas que contienen vistas de los datos personalizadas para cada usuario.

La construcción de las sentencias SQL resulta de la unión de la información introducida por el usuario y ciertos parámetros fijos definidos por el programador de la aplicación. La sentencia final es la utilizada para la consulta a la base de datos y en función de su resultado se presenta la vista correspondiente al usuario final.

El problema surge cuando se piensa que los parámetros introducidos por el usuario serán siempre los esperados. Aquí aparece la posibilidad de modificación de la sentencia SQL inicial a partir de caracteres de escape introducidos en los campos solicitados al usuario, para que la sentencia que se ejecute sobre la base de datos proporcione un resultado distinto al esperado. Las implicaciones de este tipo de ataques son muchas y muy diversas: revelación de información confidencial, modificación de datos de la base de datos e incluso ataques de denegación de servicio (DoS) mediante el borrado de los datos. En muchos casos, la ejecución de sentencias SQL puede ir más allá de meras operaciones sobre la base de datos pudiendo llegar incluso a la ejecución de código arbitrario en el sistema a partir de procedimientos almacenados (p.e. xp_cmdshell en MS-SQL).

Tratamiento de errores

Un problema que a menudo no se tiene en cuenta en el desarrollo de las aplicaciones es el de proporcionar mas información de la necesaria al usuario del servicio. Esta circunstancia es muy común en las páginas de error de las aplicaciones, que suelen ser páginas por defecto que ofrecen información detallada acerca del error, información que puede ser aprovechada por el cliente para, por ejemplo, en caso de una sentencia SQL se pueda saber como esta construida dicha sentencia y así malversarla más fácilmente. En muchos casos las página de error también pueden dar información de versiones, paths de las aplicaciones e incluso la localización exacta de ficheros de cabecera (p.e, ficheros .inc) que suelen tener información sensible de la aplicación (usuario / contraseña de acceso a bases de datos, parte del código dinámico de las aplicaciones, etc).

Cross Site Scripting

Otro tipo de ataque basado en el desarrollo de las aplicaciones, es el conocido como Cross Site Scripting. Este tipo de ataques consiste en la inyección de código malicioso por parte de un atacante en una página alojada en un servidor para que cuando otro cliente cargue esa página, se ejecute dicho código en el cliente y realice acciones que pueden ser de provecho para el atacante (envío de cookies, lectura de ficheros, registro, etc.) La naturaleza de este tipo de ataques hace que su ámbito de aplicación se limite a en entornos muy concreto que afecta sobre todo a "comunidades virtuales" en las que se permite a muchos usuarios la publicación de contenidos y terceros clientes puedan consultar dicha información. Algunos ejemplos son webmails, servicios públicos de noticias a través de WWW, etc.

Una visión de futuro.

El entorno Web no es ni mucho menos estático, sino que se incorporan constantemente nuevos conceptos y tecnologías. Tecnologías como SOAP (Simple Object Access Protocol), que favorecen la interactividad entre aplicaciones utilizando como protocolo común HTTP y como lenguaje de intercambio XML, hacen que en un futuro las posibilidades para vulnerar los sistemas a través de las aplicaciones sean cada vez mayores. Al mismo tiempo, se está introduciendo un nuevo concepto de servicios, agrupado en el nombre de moda Web Services. Este tipo de servicios va dirigido a la oferta de mayor funcionalidad al usuario de la forma más personalizada posible, para ello, basándose en estas tecnologías, ofrece todos los servicios disponibles que ni siquiera tienen que estar en el servidor que los ofrece. Además de la introducción de nuevas tecnologías, los nuevos servicios web de Internet se orientan a arquitecturas single sign-on, en las que la identificación en un único punto permite al usuario el acceso a todos los servicios. Esta circunstancia implica que hay que extremar las medidas de seguridad en cuanto autenticación de los usuarios, ya que al atacante se le da la posibilidad de que vulnerando la autenticación del sistema, tenga acceso a toda la funcionalidad ofrecida al usuario y a sus datos sin ningún tipo de restricción. Esto puede ser especialmente atractivo para los posibles atacantes con lo que es posible que los intentos de vulneración aumentarán considerablemente.
La seguridad en estos nuevos servicios es fundamental, ya que de ella dependerá el éxito o fracaso de los mismos. El nivel de seguridad ofrecido por las aplicaciones determinará las funcionalidades que puedan ofertarse al cliente.

Conclusiones

La base de todo servicio seguro en Internet, es tener siempre en cuenta que Internet es un entorno hostil en el que cualquier cliente puede ser peligroso y cualquier servidor el objetivo de un ataque. Con esta premisa inicial, todas las fases de diseño, desarrollo, implantación y mantenimiento del servicio deben girar en torno a la seguridad como aspecto fundamental para proporcionar el servicio.

En este sentido, debe haber un equilibrio entre la puesta en producción del servicio y la seguridad con la que éste es ofrecido, siendo tareas complementarias que no deben entorpecerse entre sí.

Se están haciendo cada vez más frecuentes las soluciones comerciales ofertadas en el mercado de securización de servicios específicos, como es el caso de servicios Web. Además es posible la utilización de tecnologías más tradicionales, como es el caso de cortafuegos de nivel de aplicación. Estas soluciones permiten implementar un amplio abanico de posibilidades de configuración, capturando y analizando las peticiones requeridas por el usuario antes de que éstas lleguen al servidor.

Joaquín Crespo Pérez

División de Seguridad Lógica

Germinus Solutions

jcrespo@germinus.com

Referencias bibliográficas.

[cert, 2002]
CERT Coordination Center, http://www.cert.org/, 2002.
[w3c, 1999]
World Wide Web Consortium Security FAQ, , 2002.
[sans, 2002]
The Twenty Most Critical Internet Security Vulnerabilities, http://www.sans.org/top20.htm, 2002
[owasp, 2002]
Open Web Application Security Project, http://www.owasp.org, 2002.
[sansrr,2002]
Artículos varios de "SANS Reading Room: Web Servers " , 2002
[devx, 2002]
DevX Security Zone, http://www.devx.com/security/, 2002.
[btq, 2002]
BUQTRAQ - Microsoft IIS and PWS Extended Unicode Directory Traversal Vulnerability, http://online.securityfocus.com/bid/1806, 2002.