SEGURIDAD EN REDES Y SISTEMAS INFORMÁTICOS

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

UNIDAD 6. Seguridad en bases de datos

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

ÍNDICE UNIDAD 6

UNIDAD 6. SEGURIDAD EN BASES DE DATOS

1. Que es una base de datos. Conceptos generales.
2. Tipos de bases de datos.
3. Amenazas en las bases de datos.
4. Securización de una base de datos.
5. Inferencia.
6. Inyección de código SQL.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

1. ¿Qué es una base de datos? Conceptos generales

Una base de datos o banco de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su posterior uso. En este sentido, una biblioteca puede considerarse una base de datos compuesta en su mayoría por documentos y textos impresos en papel e indexados para su consulta.

En la actualidad, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos tienen formato electrónico, que ofrece un amplio rango de soluciones al problema de almacenar datos.

En informática existen los sistemas gestores de bases de datos (SGBD), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Las propiedades de los sistemas gestores de bases de datos se estudian en informática.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas. También son ampliamente utilizadas en entornos científicos con el objeto de almacenar la información experimental.

Aunque las bases de datos pueden contener muchos tipos de datos, algunos de ellos se encuentran protegidos por las leyes de varios países. Por ejemplo en España, los datos personales se encuentran protegidos por la Ley Orgánica de Protección de Datos de Carácter Personal (LOPD).

Preguntas comunes:

¿Qué es una DB?
Data Base = Base de datos en inglés
¿Qué es una BBDD o BD?
Base de datos
¿Para qué sirve una DB?
Almacenar datos
¿Cómo se aprende a programar en mySQL?
No se aprende, mySQL es una base de datos, no un lenguaje
¿SQL es una base de datos?

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

SQL Server sí, pero SQL es un lenguaje

¿Cuántos tipos de bases de datos hay?

Muchos, pero los habituales en Web son Access, mySQL y SQL Server, Mariadb, Mongodb, Sql little

¿Necesito algún lenguaje especial para conectarme?

Sí, SQL es el lenguaje para interactuar con las bases de datos

¿Qué es un Query?

Una consulta a la base de datos

¿Qué es ABM?

Alta, Baja y Modificación de cualquier cosa. Usualmente de una base de datos que contiene datos.

¿Qué es CMS?

Control Managment System: Un panel de control que administra un Web site, y a menudo una base de datos.

Ahora, aclaradas estas dudas básicas, vamos al grano. Separaré en varios items la explicación a fin de un mejor orden.

Funcionalidad de un DB

Una base de datos (sea cual sea) es un soporte digital que tiene como fin el almacenamiento masivo de información en formato texto plano.

No es capaz de almacenar imágenes como se cree, sino que almacena las rutas (path) de dichas fotos; ni almacena otro tipo de datos; sino que almacena sus rutas de acceso de ser necesario.

Las bases de datos, son utilizadas en sistemas que requieren una interacción fluida con la aplicación; estas se encargan muchas veces de administrar, editar, y dar de alta. Usualmente la base de datos, esta ligada a la programación directa del site, causando que una edición en ella cause una modificación directa en lo que ve el usuario.

Ejemplos de aplicación de una base de datos (entiéndase que están ligadas a un lenguaje dinámico como PHP o ASP):
E – comerce, Agendas, Libros de visitas, foros, portales, etc.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Las bases de datos pueden clasificarse de varias maneras, de acuerdo al criterio elegido para su clasificación:

- Bases de datos estáticas

Éstas son bases de datos de sólo lectura, utilizadas primordialmente para almacenar datos históricos que posteriormente se pueden utilizar para estudiar el comportamiento de un conjunto de datos a través del tiempo, realizar proyecciones y tomar decisiones.

- Bases de datos dinámicas

Éstas son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización y adición de datos, además de las operaciones fundamentales de consulta. Un ejemplo de esto puede ser la base de datos utilizada en un sistema de información de una tienda de abarrotes, una farmacia, un videoclub, etc.

Según el contenido:

- Bases de datos bibliográficas:

Solo contienen un su rogante (representante) de la fuente primaria, que permite localizarla. Un registro típico de una base de datos bibliográfica contiene información sobre el autor, fecha de publicación, editorial, título, edición, de una determinada publicación, etc. Puede contener un resumen o extracto de la publicación original, pero nunca el texto completo, porque sino estaríamos en presencia de una base de datos a texto completo (o de fuentes primarias—ver más abajo). Como su nombre lo indica, el contenido son cifras o números. Por ejemplo, una colección de resultados de análisis de laboratorio, entre otras.

Almacenan las fuentes primarias, como por ejemplo, todo el contenido de todas las ediciones de una colección de revistas científicas.

- Bases de datos de texto completo

Un ejemplo son las guías telefónicas en formato electrónico.

- Bases de datos o "bibliotecas" de información Biológica

Son bases de datos que almacenan diferentes tipos de información proveniente de las ciencias de la vida o médicas. Se pueden considerar en varios subtipos:

Aquellas que almacenan secuencias de nucleótidos o proteínas.

Las bases de datos de rutas metabólicas

Bases de datos de estructura, comprende los registros de datos experimentales sobre estructuras 3D de biomoléculas

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

- Bases de datos clínicas
- Bases de datos bibliográficas (biológicas)

Modelos de bases de datos

Además de la clasificación por la función de las bases de datos, éstas también se pueden clasificar de acuerdo a su modelo de administración de datos.

Un modelo de datos es básicamente una "descripción" de algo conocido como *contenedor de datos* (algo en donde se guarda la información), así como de los métodos para almacenar y recuperar información de esos contenedores. Los modelos de datos no son cosas físicas: son abstracciones que permiten la implementación de un sistema eficiente de *base de datos*; por lo general se refieren a algoritmos, y conceptos matemáticos.

Algunos modelos con frecuencia utilizados en las bases de datos:

- Bases de datos jerárquicas

Éstas son bases de datos que, como su nombre indica, almacenan su información en una estructura jerárquica. En este modelo los datos se organizan en una forma similar a un árbol (visto al revés), en donde un *nodo padre* de información puede tener varios *hijos*. El nodo que no tiene padres es llamado *raíz*, y a los nodos que no tienen hijos se los conoce como *hojas*.

Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento.

Una de las principales limitaciones de este modelo es su incapacidad de representar eficientemente la redundancia de datos.

- Base de datos de red

Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de *nodo*: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico).

Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que significa administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

- Base de datos relacional

Artículo principal: Modelo relacional

Éste es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postulados sus fundamentos en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos. Su idea fundamental es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados "tuplas". Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar. Esto es pensando en cada relación como si fuese una tabla que está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

En este modelo, el lugar y la forma en que se almacenen los datos no tienen relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar para un usuario esporádico de la base de datos. La información puede ser recuperada o almacenada mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales.

Durante su diseño, una base de datos relacional pasa por un proceso al que se le conoce como normalización de una base de datos.

Durante los años '80 (1980-1989) la aparición de base produjo una revolución en los lenguajes de programación y sistemas de administración de datos. Aunque nunca debe olvidarse que dBase no utilizaba SQL como lenguaje base para su gestión.

- Bases de datos orientadas a objetos

Este modelo, bastante reciente, y propio de los modelos informáticos orientados a objetos, trata de almacenar en la base de datos los *objetos* completos (estado y comportamiento).

Una base de datos orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

Encapsulación - Propiedad que permite ocultar la información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.

Herencia - Propiedad a través de la cual los objetos heredan comportamiento dentro de una jerarquía de clases.

Polimorfismo - Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

En bases de datos orientadas a objetos, los usuarios pueden definir operaciones sobre los datos como parte de la definición de la base de datos. Una operación (llamada función) se especifica en dos partes. La interfaz (o signatura) de una operación incluye el nombre de la operación y los tipos de datos de sus argumentos (o parámetros). La implementación (o método) de la operación se especifica separadamente y puede modificarse sin afectar la interfaz. Los programas de aplicación de los usuarios pueden operar sobre los datos invocando a dichas operaciones a través de sus nombres y argumentos, sea cual sea la forma en la que se han implementado. Esto podría denominarse independencia entre programas y operaciones.

- Bases de datos documentales

Permiten la indexación a texto completo, y en líneas generales realizar búsquedas más potentes. Tesaurus es un sistema de índices optimizado para este tipo de bases de datos.

- Base de datos deductivos

Un sistema de base de datos deductivos, es un sistema de base de datos pero con la diferencia de que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

Gestión de bases de datos distribuida

La base de datos está almacenada en varias computadoras conectadas en red. Surgen debido a la existencia física de organismos descentralizados. Esto les da la capacidad de unir las bases de datos de cada localidad y acceder así a distintas universidades, sucursales de tiendas, etcétera.

Principales elementos constitutivos de una base de datos:

• Tablas:

– Las tablas forman el repositorio lógico donde se almacenan los datos de las aplicaciones.

Por ejemplo nombres, direcciones, teléfonos, estados de cuentas bancarios, entre otros.

• Vistas:

– Una vista es una tabla virtual compuesta por el resultado de la ejecución de una consulta, la cual puede contener a varias tablas. En ocasiones, las vistas pueden ser modificadas como una tabla más, sin embargo NO componen la estructura física de la base de datos.

• Stored Procedures:

– Un Stored Procedure es un conjunto de operaciones agrupadas con un fin común que realizan una cierta tarea dentro de la base de datos. Por ejemplo insertar registros en una tabla aplicando controles de validación de datos

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

- Usuarios:

– Un usuario es aquel individuo autorizado a utilizar los datos y recursos provistos por la base de datos. Por lo general dicho usuario puede ser personalizado (un usuario asociado a una persona física) o no personalizado (un usuario asignado a una aplicación o grupo de usuarios).

- Rol:

– Un rol permite asignar permisos a un grupo de usuarios mediante la implementación de una estructura que los contiene. Es similar al concepto de grupo a nivel de sistema operativo.

- Perfil:

– Un perfil permite agrupar usuarios con características similares y mejorar así su administración (por ejemplo implementación de políticas de contraseñas, asignación de recursos, etc.).

Manejo básico de una base de datos:

Consulta SQL:

– Es el lenguaje utilizado para comunicarse con la base de datos. Traduce nuestras peticiones en sentencias comprensibles para el motor.

- Comandos SELECT, INSERT, UPDATE, DELETE:

– Son los comandos estándares para la mayoría de los usuarios. Permiten agregar, modificar, borrar y buscar datos en la base.

- Comandos DROP, CREATE:

– Son comandos adicionales utilizados generalmente por los DBA o usuarios con privilegios elevados para crear, modificar o borrar tablas, bases de datos, esquemas y permisos.

- SELECT * FROM usuarios;

– Selecciona todos los usuarios.

- DELETE FROM usuarios WHERE nombre='Juan';

– Borra todos los usuarios que se llamen Juan.

- CREATE TABLE prueba (ID integer, descripción varchar (50));

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

– Crea una tabla llamada prueba con los atributos ID y descripción

2. Seguridad de Bases de Datos

La gran mayoría de los datos sensibles del mundo están almacenados en sistemas gestores de bases de datos comerciales tales como Oracle, Microsoft SQL Server entre otros, y atacar una bases de datos es uno de los objetivos favoritos para los criminales.

Mientras que la atención generalmente se ha centrado en asegurar los perímetros de las redes por medio de, firewalls, IDS / IPS y antivirus, cada vez más las organizaciones se están enfocando en la seguridad de las bases de datos con datos críticos, protegiéndolos de intrusiones y cambios no autorizados.

En esta sección daremos siete recomendaciones sobre seguridad en bases de datos, instaladas en servidores propios de la organización.

1 Identifique su sensibilidad

No se puede asegurar lo que no se conoce.

Confeccione un buen catálogo de tablas o datos sensibles de sus instancias de base de datos. Además, automatice el proceso de identificación, ya que estos datos y su correspondiente ubicación pueden estar en constante cambio debido a nuevas aplicaciones o cambios producto de fusiones y adquisiciones.

Desarrolle o adquiera herramientas de identificación, asegurando éstas contra el malware, colocado en su base de datos el resultado de los ataques de inyección SQL; pues aparte de exponer información confidencial debido a vulnerabilidades, como la inyección SQL, también facilita a los atacantes incorporar otros ataques en el interior de la base de datos.

2. Evaluación de la vulnerabilidad y la configuración

Evalúe su configuración de bases de datos, para asegurarse que no tiene huecos de seguridad.

Esto incluye la verificación de la forma en que se instaló la base de datos y su sistema operativo (por ejemplo, la comprobación privilegios de grupos de archivo -lectura, escritura y ejecución- de base de datos y bitácoras de transacciones).

Asimismo con archivos con parámetros de configuración y programas ejecutables.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

Además, es necesario verificar que no se está ejecutando la base de datos con versiones que incluyen vulnerabilidades conocidas; así como impedir consultas SQL desde las aplicaciones o capa de usuarios. Para ello se pueden considerar (como administrador):

- Limitar el acceso a los procedimientos a ciertos usuarios.
- Delimitar el acceso a los datos para ciertos usuarios, procedimientos y/o datos.
- Declinar la coincidencia de horarios entre usuarios que coincidan.

3. Endurecimiento

Como resultado de una evaluación de la vulnerabilidad a menudo se dan una serie de recomendaciones específicas. Este es el primer paso en el endurecimiento de la base de datos. Otros elementos de endurecimiento implican la eliminación de todas las funciones y opciones que se no utilicen. Aplique una política estricta sobre que se puede y que no se puede hacer, pero asegúrese de desactivar lo que no necesita.

4. Audite

Una vez que haya creado una configuración y controles de endurecimiento, realice auto evaluaciones y seguimiento a las recomendaciones de auditoría para asegurar que no se desvíe de su objetivo (la seguridad).

Automatice el control de la configuración de tal forma que se registre cualquier cambio en la misma. Implemente alertas sobre cambios en la configuración. Cada vez que un cambio se realice, este podría afectar a la seguridad de la base de datos.

5. Monitoreo

Monitoreo en tiempo real de la actividad de base de datos es clave para limitar su exposición, aplique o adquiera agentes inteligentes de monitoreo, detección de intrusiones y uso indebido.

Por ejemplo, alertas sobre patrones inusuales de acceso, que podrían indicar la presencia de un ataque de inyección SQL, cambios no autorizados a los datos, cambios en privilegios de las cuentas, y los cambios de configuración que se ejecutan a mediante de comandos de SQL.

Recuerde que el monitoreo usuarios privilegiados, es requisito para la gobernabilidad de datos y cumplimiento de regulaciones como SOX y regulaciones de privacidad. También, ayuda a detectar intrusiones, ya que muchos de los ataques más comunes se hacen con privilegios de usuario de alto nivel.

El monitoreo dinámico es también un elemento esencial de la evaluación de vulnerabilidad, le permite ir más allá de evaluaciones estáticas o forenses.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

Un ejemplo clásico lo vemos cuando múltiples usuarios comparten credenciales con privilegios o un número excesivo de inicios de sesión de base de datos.

6. Pistas de Auditoría

Aplique pistas de auditoría y genere trazabilidad de las actividades que afectan la integridad de los datos, o la visualización los datos sensibles.

Recuerde que es un requisito de auditoría, y también es importante para las investigaciones forenses.

La mayoría de las organizaciones en la actualidad emplean alguna forma de manual de auditoría de transacciones o aplicaciones nativas de los sistemas gestores de bases de datos. Sin embargo, estas aplicaciones son a menudo desactivadas, debido a:

- su complejidad
- altos costos operativos
- problemas de rendimiento
- la falta de segregación de funciones y
- la necesidad mayor capacidad de almacenamiento.

Afortunadamente, se han desarrollado soluciones con un mínimo de impacto en el rendimiento y poco costo operativo, basado en tecnologías de agente inteligentes.

7. Autenticación, control de acceso, y Gestión de derechos

No todos los datos y no todos los usuarios son creados iguales. Usted debe autenticar a los usuarios, garantizar la rendición de cuentas por usuario, y administrar los privilegios para de limitar el acceso a los datos.

Implemente y revise periódicamente los informes sobre de derechos de usuarios, como parte de un proceso de formal de auditoría.

Utilice el cifrado para hacer ilegibles los datos confidenciales, complique el trabajo a los atacantes, esto incluye el cifrado de los datos en tránsito, de modo que un atacante no puede escuchar en la capa de red y tener acceso a los datos cuando se envía al cliente de base de datos.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

Principales riesgos de seguridad en una base de datos:

1.- Nombre de usuario/password en blanco, por defecto o débil.

No es nada raro conseguir en el día a día pares de usuario/password como sa/1234, esta el la primera línea de defensa y un punto fundamental de la armadura de nuestras bases de datos. Es importante hacer revisiones periódicas de credenciales.

2.- Inyecciones SQL.

Cuando la plataforma de base de datos falla para desinfectar las entradas, los atacantes son capaces de ejecutar las inyecciones SQL de forma similar a como lo hacen en los ataques basados en Web, lo que les permite elevar sus privilegios y obtener acceso a una amplia gama de funcionalidades. Muchos de los proveedores han dado a conocer soluciones para evitar estos problemas, pero no servirá de mucho si los parches no se aplican o no se toman los correctivos correspondientes.

3.- Preferencia de privilegios de usuario por privilegios de grupo.

Las organizaciones necesitan garantizar que los privilegios no se les den a los usuarios por asignación directa quien finalmente los recogerá como conserjes recogen las llaves en sus llaveros. En cambio, Rothacker recomienda que los usuarios sólo reciban privilegios por parte de grupos o funciones y sean manejados colectivamente. De esta forma será más fácil eliminar derechos a un usuario con simplemente eliminarlo del grupo, sin que queden derechos ocultos u olvidados asignados a dicho usuario.

4.- Características de base de datos innecesariamente habilitadas.

Cada instalación de base de datos viene con paquetes adicionales de todas las formas y tamaños que en su mayoría rara vez son utilizados por una sola organización. Dado que el nombre del juego en materia de seguridad de base de datos es el de reducir las superficies de ataque, las empresas necesitan buscar los paquetes que no utilizan y desactivarlos. Esto no sólo reduce los riesgos de ataques (0)day a través de estos vectores, sino que también simplifica la gestión de parches.

5.- Configuración de seguridad ineficiente.

Del mismo modo, las bases de datos tienen una gran cantidad opciones de configuración y consideraciones diferentes a disposición de los administradores para ajustar el rendimiento y funcionalidades mejoradas. Las organizaciones necesitan conseguir y desactivar aquellas configuraciones inseguras que podrían estar activadas por defecto para mayor comodidad de los DBA o desarrolladores de aplicaciones. Las configuraciones de bases de datos en producción y desarrollo deben ser radicalmente diferentes.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

6.- Desbordamientos de búfer.

Otro favorito de los delicuentes cibernéticos, las vulnerabilidades de desbordamiento de búfer, son explotadas por las inundaciones de las fuentes de entrada con valores diferentes o muy superiores a los que aplicación espera - por ejemplo, mediante la adición de 100 caracteres en un cuadro de entrada pidiendo un número de Seguro Social. Los proveedores de bases de datos han trabajado duro para solucionar los problemas técnicos que permiten estos ataques se produzcan. Esta es otra razón por la cual los parches son tan importantes.

7.- Escalada de privilegios

Del mismo modo, las bases de datos con frecuencia exponen vulnerabilidades comunes que permiten a un atacante escalar privilegios en una cuenta de privilegios bajos hasta tener acceso a los derechos de un administrador. A medida que estas vulnerabilidades son descubiertas, los proveedores las corrigen y los administradores deben mantener las actualizaciones y parches actualizados.

8.- Ataque de denegación de servicio

El caso del SQL Slammer es siempre un ejemplo muy esclarecedor de cómo los atacantes pueden utilizar las vulnerabilidades de los DBMS para derribar los servidores de base de datos a través de un alto flujo de tráfico. Aún más ilustrativo es el hecho de que cuando el Slammer atacó en 2003, un parche ya estaba por ahí que se dirigió a corregir la vulnerabilidad por la que se generó su ataque. Hoy en día siete años más tarde, SQL Slammer todavía está dando dolores de cabeza en los servidores no actualizados.

9.- Bases de datos sin actualizar.

Esto podría sonar repetitivo, pero vale la pena repetirlo. Los administradores de base de datos a veces no aplican un parche en el momento oportuno porque tienen miedo de este dañe sus bases de datos. Pero el riesgo de ser hackeado hoy es mucho más alto que el riesgo de aplicar un parche que descomponga la base de datos. Además exixten ante esos temores los backups y las réplicas. Quizás este punto pudo haber sido válido hace cinco años, pero los proveedores ahora

Sin encriptar los datos sensibles en reposo y en movimiento

10.- Datos sensibles sin cifrar, tanto en reposo como en movimiento.

Tal vez sea una obviedad, pero las organizaciones no deben almacenar los datos sensibles en texto plano en una tabla. Y todas las conexiones a la base de datos siempre que manejen datos sensibles deben utilizar el cifrado.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

Requisitos esenciales para la seguridad de la base de datos:

- La base de datos debe ser protegida contra el fuego, el robo y otras formas de destrucción.
- Los datos deben ser reconstruibles, porque por muchas precauciones que se tomen, siempre ocurren accidentes.
- Los datos deben poder ser sometidos a procesos de auditoria. La falta de auditoria en los sistemas de computación ha permitido la comisión de grandes delitos.
- El sistema debe diseñarse a prueba de intromisiones. Los programadores, por ingeniosos que sean, no deben poder pasar por alto los controles.

Ningún sistema puede evitar de manera absoluta las intromisiones malintencionadas, pero es posible hacer que resulte muy difícil eludir los controles. El sistema debe tener capacidad para verificar que sus acciones han sido autorizadas. Las acciones de los usuarios deben ser supervisadas, de modo tal que pueda descubrirse cualquier acción indebida o errónea.

La seguridad de las Bases de Datos se concreta mediante mecanismos, tanto "hardware" como "software".

Así estos mecanismos son:

- El primero se denomina identificación, que procede a identificar a los sujetos (procesos, normalmente transacciones que actúan en su nombre o usuarios) que pretenden acceder a la base de datos.

- El siguiente mecanismo que actúa es el de autenticación. El proceso usual es mediante contraseñas, constituidas por un conjunto de caracteres alfanuméricos y especiales que sólo el sujeto conoce. También se puede realizar mediante algún dispositivo en poder del mismo o alguna de sus características bioantropométricas.

- En caso de que el sujeto sea positivamente identificado y autenticado, se debe controlar el acceso que pretende a los objetos (datos y recursos accedidos por los sujetos. Por ejemplo, si se considera un SGBD relacional los recursos que deben protegerse son las relaciones, vistas y atributos).

El mecanismo involucrado se denomina de control de accesos y se encarga de denegar o conceder dichos accesos en base a unas reglas, que establecen en qué condiciones el sujeto puede acceder y realizar ciertas operaciones sobre el objeto especificado.

Estas reglas son dictadas por una persona con autoridad suficiente, que normalmente es el propietario de los datos o, en el caso de una organización, el administrador de la base de datos, de acuerdo con unas políticas de seguridad.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

Una regla de autorización se suele representar mediante una tripleta (s,o,p), que especifica que el sujeto esta autorizado para ejercer un privilegio sobre un objeto.

Los sujetos de autorización son las entidades del sistema a las que se les asignan las autorizaciones sobre los objetos.

Los sujetos se pueden clasificar en las siguientes categorías:

-Usuarios, es decir, individuos simples conectados al sistema. A veces seria más útil especificar los criterios de acceso basándose en sus calificaciones y características, más que en la identidad del usuario.

-Grupos, es decir, conjuntos de usuarios.

-Roles, o lo que es igual, conjuntos de privilegios necesarios para realizar actividades especificas dentro del sistema.

-Procesos, que ejecutan programas en nombre de los usuarios. Necesitan recursos del sistema para llevar a cabo sus actividades, y normalmente tienen acceso sólo a los recursos necesarios para que se puedan realizar las tareas del proceso. Esto limita el posible daño derivado de fallos del mecanismo de protección.

Los privilegios de autorización establecen los tipos de operaciones que un sujeto puede ejercer sobre los objetos del sistema.

El conjunto de privilegios depende de los recursos a proteger. Por ejemplo, los privilegios típicos de un SGBD relacional son seleccionar, insertar, actualizar y eliminar.

Normalmente, los privilegios están organizados jerárquicamente y la jerarquía representa una relación de asunción entre privilegios. Si la transacción invocada trata de modificar el contenido de la base de datos, los cambios propuestos son chequeados por el sistema de gestión de la misma, para garantizar su integridad semántica o elemental.

Así mismo, el sistema de gestión se responsabiliza de evitar accesos concurrentes a dicha base. Finaliza la transacción, con éxito o no, el citado sistema de gestión graba en un registro de auditoria todas las características de aquella.

Este registro también contiene la información pertinente para la recuperación de la base de datos, caso de un fallo de ésta o una caída del sistema. Aunque este mecanismo no impide los accesos no autorizados, tiene efectos disuasorios sobre potenciales atacantes, permitiendo además encontrar puntos débiles en los mecanismos de seguridad. Adicionalmente a todos estos mecanismos, el medio físico sobre el que se almacena la base de datos puede estar protegido criptográficamente. Igualmente las copias de seguridad pueden estas así defendidas frente a ataques.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Los tipos de elementos se combinan para formar el sistema que se utiliza para analizar los métodos de protección:

- Los usuarios con acceso a la base de datos, a los que por brevedad denominaremos accesores
- El tipo de acceso deseado
- Los elementos a los que se realizará el acceso

Cada uno de estos elementos debe estar adecuadamente identificado a fin de lograr el control del acceso a los datos. También es necesario considerar el entorno o frontera del área dentro de la cual es válido es sistema de protección.

Definiciones.

Se definirá cierto número de términos a fin de que el análisis subsecuente de los mecanismos resulte claro:

Entorno: Existe un área con perímetro bien definido, conocido como sistema de la base de datos.

Usuarios e intrusos: Dentro de esta área puede haber individuos autenticados adecuadamente identificados; individuos disfrazados de usuarios validos, e intrusos.

Alcance limitado: El sistema desconoce la identidad de los individuos en el mundo exterior.

Privilegios: Existen varios privilegios de acceso a los datos, relacionados con la identificación de un individuo. La descripción de estos privilegios se mantiene como parte del sistema de la base de datos.

Protección: Todos los elementos dato están protegidos hasta cierto punto mientras se encuentren dentro del área del sistema de la base de datos, y perderán toda la protección que proporciona el sistema al sacarse del área.

Confiabilidad: Un prerrequisito para lograr la protección de la base de datos es un alto nivel de confiabilidad del sistema.

La identificación externa de los usuarios con acceso a la base de datos es en primer lugar el nombre, en la forma en que lo introduzcan al sistema. Un usuario con derecho de acceso también puede identificarse mediante una clave de acceso (password), darse al ser solicitada, o tal vez por una llave o identificación que la maquina pueda aceptar.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ana González (Kao)

Inferencia

Entendiendo la inferencia desde el punto de vista de la lógica

Definición de inferencia

El proceso por el cual se infiere una conclusión de varias observaciones se llama razonamiento inductivo. La conclusión puede ser correcto o incorrecto, o correcto dentro de un cierto grado de precisión o correcto en determinadas situaciones. Conclusiones inferidas a partir de observaciones múltiples pueden analizarse por observaciones adicionales.

Esta definición es discutible (debido a su falta de claridad. Ref: Diccionario de inglés Oxford:

> "inducción… 3. Lógica la inferencia de una ley general de casos particulares.") La definición dada por lo tanto aplica sólo cuando la "conclusión" es general.
>
> 1. Una conclusión sobre la base de evidencia y razonamiento. 2. El proceso de llegar a esa conclusión: "orden, salud y por limpieza de inferencia".

Ejemplos de inferencia

Filósofos griegos define una serie de silogismos, corregir tres de parte de inferencias, que pueden utilizarse como bloques de construcción para razonamiento más complejo.

Comenzamos con el más famoso de todos ellos:

1. Todos los hombres son mortales
2. Sócrates es un hombre
3. Por lo tanto, Sócrates es mortal.

¿El lector puede comprobar que las premisas y conclusión son verdaderas, pero la lógica se ocupa de inferencia: sigue la verdad de la conclusión de que las instalaciones?

La validez de una inferencia depende de la forma de la inferencia. Es decir, la palabra "válido" no se refiere a la verdad de las premisas o la conclusión, sino a la forma de la inferencia. Una inferencia puede ser válida incluso si las piezas son falsas y pueden ser válidas incluso si las

piezas son verdaderas. Pero un formulario válido con premisas verdaderas siempre tendrá una conclusión verdadera.

Por ejemplo, considere la forma de la siguiente symbological pista:

1. Todas las frutas son dulces.
2. Una banana es una fruta.
3. Por lo tanto, una banana es dulce.

Para que la conclusión que ser necesariamente cierto, las premisas deben ser verdaderas.

Ahora pasamos a un formulario no es válido.

1. Todos los a son B.
2. C es un B.
3. Por lo tanto, C es un A.

Para mostrar que este formulario no es válido, demostramos cómo puede conducir a una conclusión falsa de premisas verdaderas.

1. Las manzanas son frutas. (Corregir)
2. Los plátanos son frutas. (Corregir)
3. Por lo tanto, plátanos son manzanas. (Mal)

Un argumento válido con premisas falsas puede llevar a una conclusión falsa:

1. Todas las personas altas son griegos.
2. John Lennon era alto.
3. Por lo tanto, John Lennon era griego.

Cuando se utiliza un argumento válido para derivar una conclusión falsa de premisas falsas, la inferencia es válida porque sigue la forma de una inferencia correcta.

También puede utilizarse un argumento válido para derivar una conclusión verdadera de premisas falsas:

1. Todas las personas altas son músicos
2. John Lennon era alto

3. Por lo tanto, John Lennon fue un músico

En este caso tenemos dos premisas falsas que implican una conclusión verdadera.

Una vez entendido el concepto de inferencia pensemos en su resultado aplicado a las bases de datos:

Mecanismos de inferencia en las bases de datos deductivas

Control de Inferencia:

Trata de proteger los datos de descubrimientos indirectos de información. Ocurre cuando un conjunto de items de datos X puede ser leído por un usuario, y puede ser usado para obtener el conjunto de datos Y de la forma Y=f(X).

Un 'canal de inferencia' es un canal donde los usuarios pueden encontrar un item X y entonces usarlo para derivar Y.15

Ejemplo de Control de Inferencia:

El problema que provoca la poli-instanciación.

Inferencia de información en base a los datos que son accesibles:

-Saber el salario de los individuos teniendo acceso a la categoría profesional y a una tabla de equivalencias

Los principales canales de inferencia son:

• Acceso Indirecto. Ejemplo: insertar una tupla cuya clave primaria es la misma de una tupla no visible para el usuario. Cuando el sistema avisa del problema, inferimos la existencia de una tupla a la que no podemos acceder (poli-instanciación)

• Datos correlacionados. Ejemplo: z=t*k. Si t y k son visibles, y z no es visible, el valor de z puede ser inferido fácilmente.

Saber el salario de los individuos teniendo acceso a la categoría profesional y a una tabla de equivalencias

• Valores ocultos. A veces, la existencia de valores nulos o no visibles, puede dar sospechas de que se esconde información confidencial:

Clasificar datos en una base de datos de acuerdo a rangos concretos. Los usuarios, analizando la información pueden inferir esos rangos de acuerdo a la información que pueden ver.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

El problema de la inferencia estadística

Implica la deducción de datos. Las bases de datos estadísticas permiten la consulta de datos estadísticos sobre grupos de individuos. En esas bases de datos, el acceso a datos sobre individuos individuales no está permitido, puesto que los datos son sólo accesibles a través de funciones estadísticas.

Sin embargo, analizando datos estadísticos se pueden obtener datos sobre individuos concretos. Para hacer frente a este tipo de problemas se suelen utilizar dos tipos de controles:

–Control de consultas: Restringen las consultas estadísticas que podrían revelar al usuario información confidencial

–Perturbaciones de datos. Introducen algunos tipos de modificaciones durante el procesamiento de la consulta

Inyección de SQL

El uso de lenguajes interpretados en las aplicaciones web ha posibilitado la aparición de nuevos tipos de ataques con gran éxito.

La inyección de código no compilado (scripts) permite una mayor “flexibilidad" a los atacantes, ya que pueden crear ataques independientes de la plataforma.

Una de las formas más utilizadas de este tipo de ataque es el conocido como SQL-injection.

Muchas aplicaciones web (modelo de tres capas) se componen de una interfaz que genera código para el navegador del cliente (HTML, CSS, bash, javascript, etc.) a partir de una lógica de negocio y unas bases de datos que almacenan la información.

Generalmente, las consultas a las bases de datos se realizan utilizando el lenguaje SQL.

Las consultas a la base de datos la realiza la capa de negocio utilizando datos obtenidos del cliente (nombre de usuario, producto que busca, etc.) a través de un formulario web o cookies.

La inyección de código se produce cuando el cliente es capaz introducir comandos SQL camuflados en los datos que manda al servidor y este los confunde como comandos validos.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com

Ejemplo de código incorrecto:

```
import Mysqldb;

import cgi;

# Procesa los campos de un formulario web.

Formulary_dict = cgi.FieldStorage();

user_name = Formulary_dict["user"].value;

# Connecta con la DB

connection = Mysqldb.mysqldb("web","pass");

cursor = connection.cursor();

myquery = "SELECT * FROM users WHERE name='" + user
name + "';";

cursor.execute(myquery);

row = cursor.fetchone();

if len(row) > 0:

print "Allow access."
```

Si el usuario introduce un nombre de usuario valido, por ejemplo “Juan", entonces la consulta sql que se realiza es:

```
SELECT * FROM users WHERE name='Juan';
```

Pero si un como nombre de usuario se introduce la cadena:

```
Alicia'; SELECT * FROM users where name LIKE '%
```

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com
Ana González (Kao)

Entonces se realizan tres operaciones:

```
SELECT * FROM users WHERE name = 'Alicia';

SELECT * FROM users WHERE name LIKE '%';
```

Se puede obtener información reservada e incluso modificar o borrar la información de las bases de datos.

Prevención de inyección de SQL:

El fallo se produce porque los datos que introduce el usuario no se separan claramente del código SQL.

Es necesario eliminar de los datos de usuarios todos aquellos símbolos que puedan romper el comando SQL.

La mayoría de los lenguajes web ofrecen funciones que escapan los símbolos problemáticos

Estas funciones retornan una cadena que se puede utilizar para construir sentencias SQL. (En python se puede utilizar: MySQLdb.escape string(user name))

Otro método que ofrecen los lenguajes consiste en transformar las sentencias SQL en llamadas a funciones, donde los parámetros se traten como argumentos de funciones.

El contenido de este tema es de mi Autoría y/o recopilación de diversas
Fuentes www.informatico-madrid.com