PROGRAMA EDUCATIVO
INFORMATICA ADMINISTRATIVA

UNIDAD DE APRENDIZAJE
ADMINISTRACION DE BASES DE DATOS

Unidad de competencia IV

Identificar los conceptos sobre administración y seguridad en las bases de datos

Seguridad y Administración

ELABORACION
ADRIAN TRUEBA ESPINOSA

PRESENTACIÓN DEL CURSO

Una de las principales actividades del Licenciado en Informática Administrativa es la elaboración de bases de datos, desde su diseño hasta la administración por lo que, cuyas bases deben ser adquiridas en su formación. La administración de bases de datos como una parte de la informática, evoluciona continuamente, sin embargo la administración de las bases de datos es uno de los conocimientos base en la construcción de aplicaciones de mediana y alta complejidad. Esta unidad de aprendizaje pretende introducir al alumno en el manejo y almacenamiento de datos por medios electrónicos

CONTENIDO DEL CURSO

Unidad 1: CONCEPTOS FUNDAMENTALES DE BASES DE DATOS

Unidad 2: MODELO DE DATOS

Unidad 3: METODOLOGÍA DE DISEÑO DE BASE DE DATOS

Unidad 4: CONCEPTOS DE DISEÑO DE APLICACIONES DE BASES DE DATOS

Unidad 5: DISEÑO E IMPLEMENTACIÓN DE BASES DE DATOS

Unidad 6: ADMINISTRACIÓN Y SEGURIDAD EN BASE DE DATOS

Unidad 7. TENDENCIAS ACTUALES EN BASES DE DATOS

METAS A ALCANZAR

Que el alumno conozca los elementos teóricos y prácticos de la seguridad y administración de una bases de datos como

- Concepto de seguridad, tipos de seguridad
- La auditoria de las bases de datos y la administración

OBJETIVO DEL MATERIAL DIDÁCTICO

Identificar los conceptos sobre administración y seguridad en las bases de datos.

METODOLOGÍA DEL CURSO

El curso se desarrollará bajo el siguiente proceso de estudio:

1. Exposición de parte del profesor mediante la utilización de este material en diapositivas.
2. Control de lecturas selectas que el profesor asignará para complementar la clase.
3. Investigación de temas, conceptos, procesos y métodos de la seguridad.
4. Participación en clases
5. Prácticas de laboratorio

UTILIZACIÓN DEL MATERIAL DE DIAPOSITIVAS

El material didáctico visual es una herramienta de estudio que sirve como una guía para que el alumno repase los temas más significativos de “Administración y seguridad de las bases de datos”,.

UNIDAD DE COMPETENCIA VI

Identificar los conceptos sobre administración y seguridad en las bases de datos

Seguridad y administración

INTRODUCCIÓN

En los SGBD, seguridad se refiere a la protección de los datos ante usuarios no autorizados, es decir, definir el nivel que permita hasta donde el usuarios pueden acceder a que datos

Al tratar el tema de la seguridad en Base de Datos, es importante considerar la necesidad de proteger totalmente la máquina completa contra todos los tipos de ataques posibles: intercepción pasiva de paquetes, reproducción de comandos, y denegación de servicio.

LAS 3 PRINCIPALES CARÁCTERÍSTICAS DE LA SEGURIDAD EN UNA BASE DE DATOS

1. La Confidencialidad de la información
2. La Integridad de la información
3. La Disponibilidad de la información

EN SEGURIDAD SE MIDE:

La protección del sistema frente a ataques externos.
La protección frente a caídas o fallos en el software o en el equipo.
La protección frente a manipulación por parte de usuarios no autorizados.

La seguridad en las bases de datos es un área de muy importante. Esto considerando que los DBMS (Sistema manejador de bases de datos) son muy usados y están accesibles a través de todas las redes en todo el mundo

Utilizando la WEB para el acceso y consulta.

El uso de DBMS en aplicaciones web o de e-commerce ha aumentado considerablemente el riesgo a ataques directos e indirectos a través de Internet

• Los riesgos típicos que afectan a los DBMS incluyen:

– Acceso no autorizado a información confidencial.

– Modificación no autorizada a la información.

– Pérdida de disponibilidad del servicio.

– Habilidad para comprometer el sistema operativo de un DBMS para usarlo como base de futuros ataques a otros equipos

REQUISITOS PARA LA SEGURIDAD DE LAS BD

1. La base de datos debe ser protegida contra el fuego, el robo y otras formas de destrucción.
2. Los datos deben ser re construibles, ya que siempre pueden ocurrir accidentes.
3. Los datos deben poder ser sometidos a procesos de auditoria.
4. El sistema debe diseñarse a prueba de intromisiones, no deben poder pasar por alto los controles.
5. Ningún sistema puede evitar las intromisiones malintencionadas, pero es posible hacer que resulte muy difícil eludir los controles.
6. El sistema debe tener capacidad para verificar que sus acciones han sido autorizadas.
7. Las acciones de los usuarios deben ser supervisadas, de modo tal que pueda descubrirse cualquier acción indebida o errónea.

Elementos que hay que considerar para una base de datos segura

Privilegios excesivos e inutilizados. Cuando a alguien se le otorgan privilegios de base de datos que exceden los requerimientos de su puesto de trabajo se crea un riesgo innecesario. Los mecanismos de control de privilegios de los roles de trabajo han de ser bien definidos o mantenidos.

Abuso de Privilegios. Los usuarios pueden llegar a abusar de los privilegios legítimos de bases de datos para fines no autorizados, por ejemplo, sustraer información confidencial. Una vez que los registros de información alcanzan una máquina cliente, los datos se exponen a diversos escenarios de violación.

Inyección por SQL. Un ataque de este tipo puede dar acceso a alguien y sin ningún tipo de restricción a una base de datos completa e incluso copiar o modificar la información.

Malware* y *spear phising. Se trata de una técnica combinada que usan los cibercriminales, *hackers* patrocinados por estados o espías para penetrar en las organizaciones y robar sus datos confidenciales.

Auditorías débiles. No recopilar registros de auditoría detallados puede llegar a representar un riesgo muy serio para la organización en muchos niveles.

Exposición de los medios de almacenamiento para backup. Éstos están a menudo desprotegidos, por lo que numerosas violaciones de seguridad han conllevado el robo de discos y de cintas. Además, el no auditar y monitorizar las actividades de acceso de bajo nivel por parte de los administradores sobre la información confidencial puede poner en riesgo los datos.

Explotación de vulnerabilidades y bases de datos mal configuradas. Los atacantes saben cómo explotar estas vulnerabilidades para lanzar ataques contra las empresas.

Datos sensibles mal gestionados. Los datos sensibles en las bases de datos estarán expuestos a amenazas si no se aplican los controles y permisos necesarios.

Denegación de servicio (DoS). En este tipo de ataque se le niega el acceso a las aplicaciones de red o datos a los usuarios previstos. Las motivaciones suelen ser fraudes de extorsión en el que un atacante remoto repetidamente atacará los servidores hasta que la víctima cumpla con sus exigencias.

Limitado conocimiento y experiencia en seguridad y educación. Muchas firmas están mal equipadas para lidiar con una brecha de seguridad por la falta de conocimientos técnicos para poner en práctica controles de seguridad, políticas y capacitación.

MEDIDAS DE SEGURIDAD

FÍSICAS: Controlar el acceso al equipo, mediante tarjetas de acceso…
PERSONAL: Acceso solo de personal autorizado, identificación directa de personal…
SGBD: Uso de herramientas que proporcione el SGBD perfiles de usuario, vistas, restricciones de uso de vistas…

HAY DOS TIPOS DE SEGURIDAD:

- ***DIRECCIONAL***

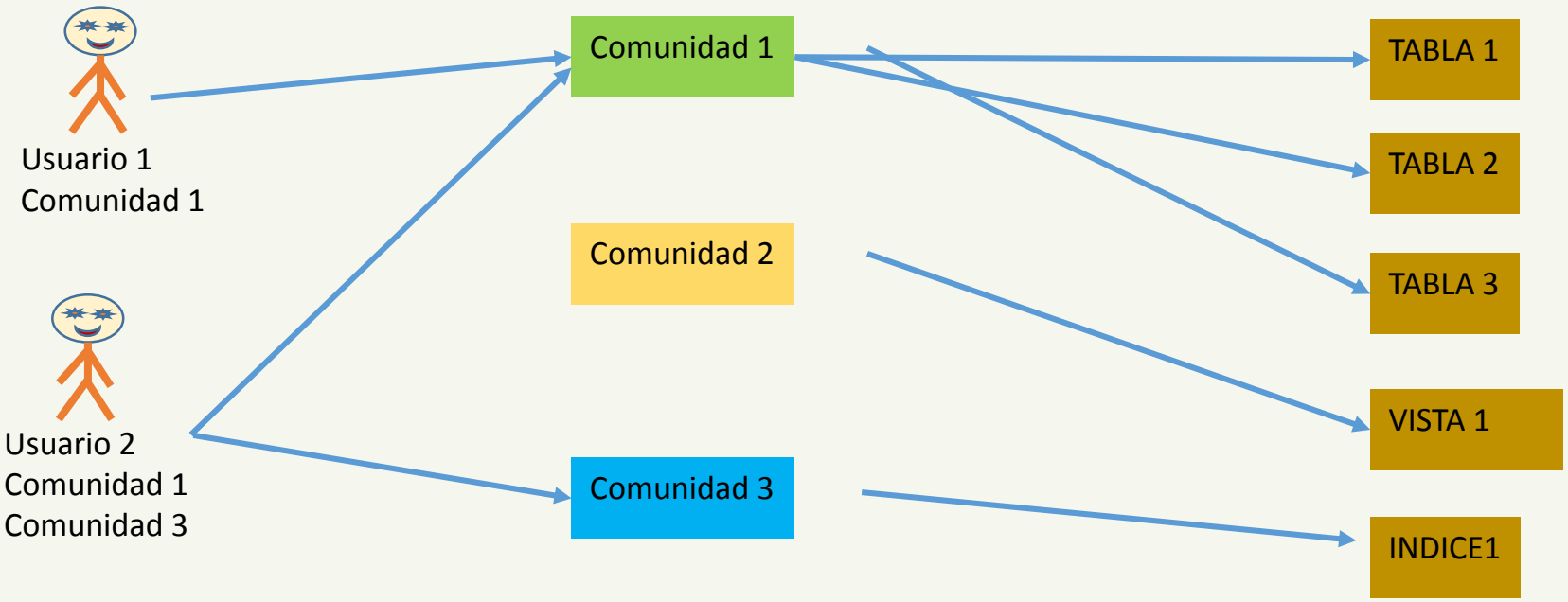
Se usa para otorgar y revocar privilegios a los usuarios a nivel de archivos, registros o campos en un modo determinado (consulta o modificación).

- ***OBLIGATORIA***
- Sirve para imponer seguridad de varios niveles tanto para los usuarios como para los datos.
- Para eso se utilizan mecanismos de protección.

SEGURIDAD DISCRECIONAL

Se basa en otorgar privilegios a usuarios (o grupos de usuarios), en los que se incluye la capacidad de tener acceso a tablas, registros o campos específicos con un determinado modo (para leer, insertar o actualizar)

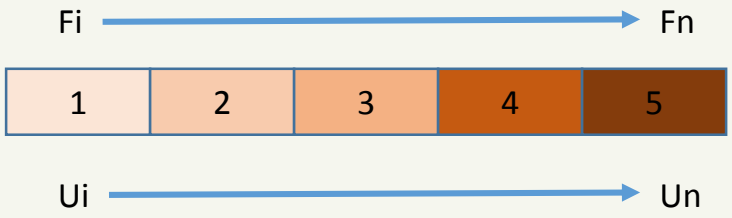
- Autorizar al usuario X a realizar consultas en filas de la tabla A
- Autorizar al usuario X a utilizar un procedimiento almacenado B



SEGURIDAD OBLIGATORIA

Consiste en asignar seguridad de varios niveles, clasificando los datos y los usuarios en varias clases (o niveles) de seguridad, de manera que los usuarios puedan acceder a los datos según el nivel, para el dato que desean acceder

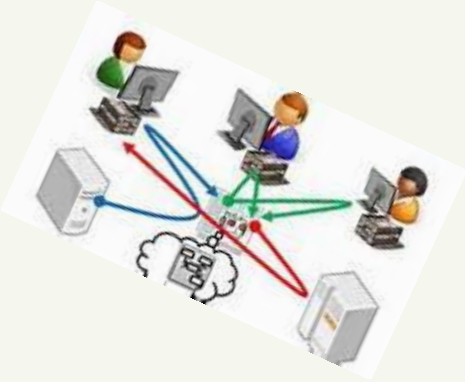
- Los atributos tienen un nivel Fi de seguridad, que solo se pueden leer si el usuario tiene un nivel Ui >= Fi de seguridad...



HAY DOS TIPOS DE USUARIOS:

- Usuario con derecho a crear, borrar y modificar objetos y que además puede conceder privilegios a otros usuarios sobre los objetos que ha creado.
- Usuario con derecho a consultar, o actualizar, y sin derecho a crear o borrar objetos. Privilegios sobre los objetos, añadir nuevos campos, indexar, alterar la estructura de los objetos, etc.

– **Usuario de Aplicación** – accesible por un usuario interactivo. Requiere acceso de sólo lectura (select), insert, update o delete en tablas existentes.



– **Administrador de Base de Datos (DBA)** – para los propósitos de este documento, el DBA se define como la cuenta responsable para configurar y operar la base. El DBA tiene privilegios "full" sobre todos los objetos, recursos y usuarios de la base de datos.

– **Dueño de Aplicación (Application Owner)** – Un dueño de aplicación es dueño de todos los objetos definidos y utilizados por una aplicación. Define roles y asigna permisos a los objetos usuarios sobre los objetos de la aplicación sobre los roles de la aplicación. Los privilegios del dueño de aplicación están limitados a creación (create), elminiación (drop), o alteración (alter) de los objetos de la base.

– **Usuario Administrativo de Aplicación** (Application User Manager) – un usuario administrativo tiene privilegios de creación y administración sobre usuarios de la aplicación para con la base de datos y define sus roles.

– **Cuenta de Aplicación** – es una cuenta de usuario especializada. Es utilizada por una aplicación, un servicio o un batch. Puede tener restricciones de acceso especiales debido a los elevador privilegios asignados.

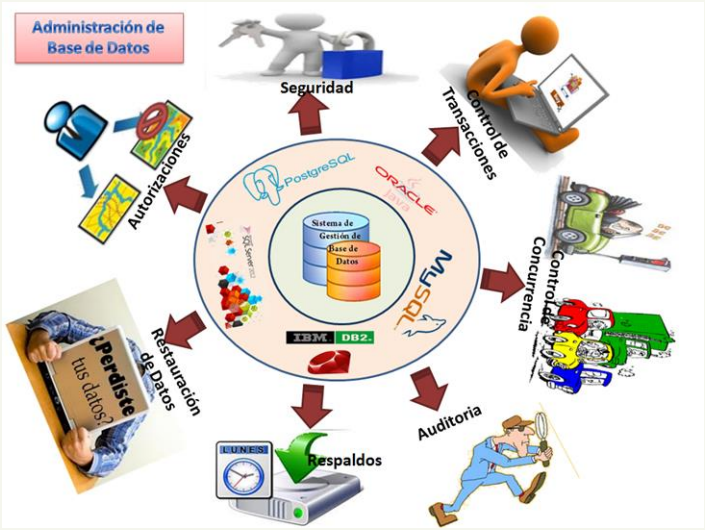
– **Auditor de Base de Datos** – maneja los registros de auditoría. La utilización de esta cuenta permite segregación de funciones. Sin esta segregación de funciones, las acciones del DBA no podrían ser monitoreadas de manera confiable.

– **Operador de Base de Datos** – a esta cuenta de la base se le pueden asignar privilegios limitados a funciones de soporte, como backup o reseteo de la base.

– Las categorías asignadas a las cuentas son importantes para la asignación de roles en la base de datos.

– La administración de cuentas de usuario en las bases de datos, así como lo es en sistemas operativos, es crítica, por lo que las cuentas de usuario deberían ser permanentemente mantenidas y administradas.

– Cuentas en desuso o expiradas activas en la base, presentan oportunidades de acceso indetectables. El uso de controles automáticos, como el bloqueo por inactividad, sería recomendable.



IDENTIFICACIÓN Y AUTENTIFICACIÓN

En un SGBD existen diversos elementos que ayudan a controlar el acceso a los datos.
En primer lugar el sistema debe identificar y autentificar a los usuarios utilizando alguno de las siguientes formas:

- Código y contraseña
- Identificación por hardware
- Del usuario, conocimiento, aptitudes y hábitos
- Información predefinida (Aficiones, cultura…)

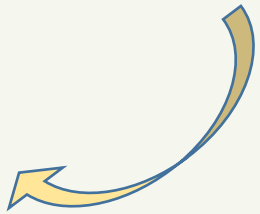
Además, el administrador deberá especificar los privilegios que un usuario tiene sobre los objetos:

- Usar una B.D.
- Consultar ciertos datos
- Actualizar datos
- Crear o actualizar objetos
- Ejecutar procedimientos almacenados
- Referenciar objetos
- Indexar objetos
- Crear identificadores

MATRIZ DE AUTORIZACIÓN

La seguridad se logra si se cuenta con un mecanismo que limite a los usuarios a su vista o vistas personales.
La norma es que la base de datos relacionales cuente con dos niveles de seguridad:

- Relación: Puede permitírsele o impedírsele que el usuario tenga acceso directo a una relación.
- Vista: Puede permitírsele o impedírsele que el usuario tenga acceso a la información que aparece en un vista.

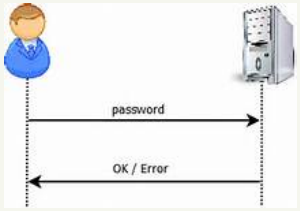


AUTENTICACIÓN

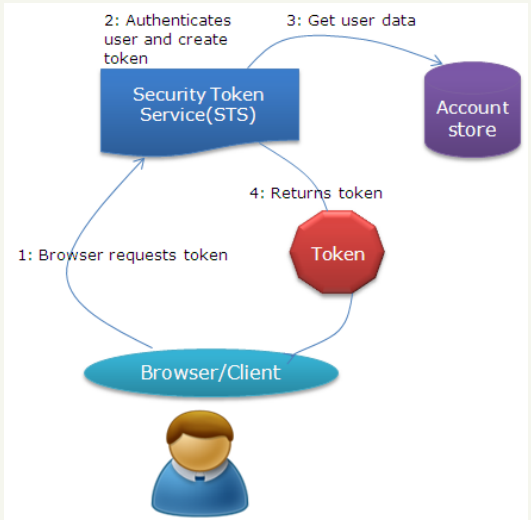
– El control de acceso está basado en la autenticación y la identificación.

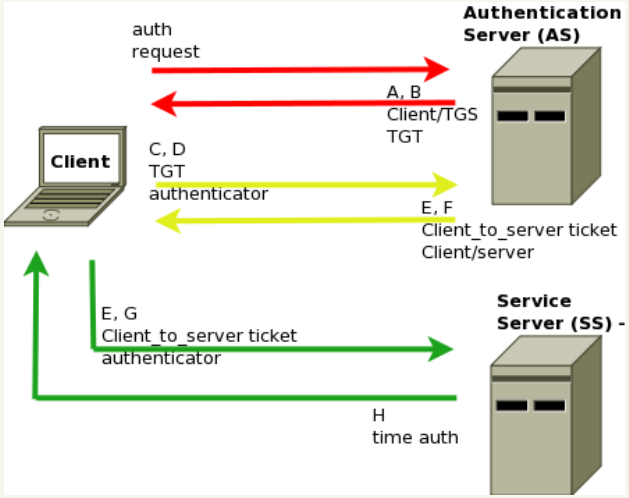
DBMS provee mecanismos estándar de identificación y autenticación de usuarios.

Los usuarios se identifican frente a la base de datos como usuarios de la misma mediante el sistema operativo del host, un servicio de directorios, un servicio de autenticación de red o por la misma base de datos. DBMS soporta varios métodos de autenticación, los cuales no siempre están limitados a contraseñas, certificados o tokens.



Cuentas de Usuario de las Bases de Datos



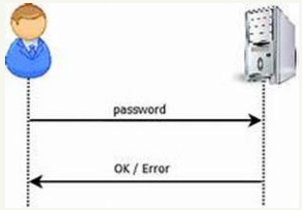


• **Autenticación por Contraseña**

– La mayoría de las bases de datos soportan el mecanismo de autenticación usuario/password.

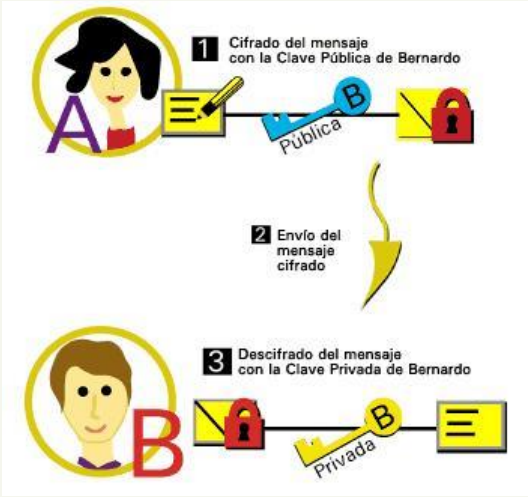
– Es imprescindible una correcta administración de las contraseñas para mantener adecuados niveles de seguridad. Deberán tenerse en cuenta, principalmente, la complejidad, reutilización, expiración y encripción.

– Cuando las credenciales de usuario son administrados por una aplicación, la responsabilidad de la administración de contraseñas recae sobre la aplicación. La protección de las credenciales de usuario será revisada durante la revisión de la aplicación.



• **Autenticación por Certificados**

– Algunos DMBS proveen la posibilidad de autenticación mediante certificados. Otras dependen del sistema operativo del host o del servicio de directorios para proveer dicho tipo de autenticación. Los certificados “Self-Signed” (propios) son inseguros y deberán tratar de evitarse.

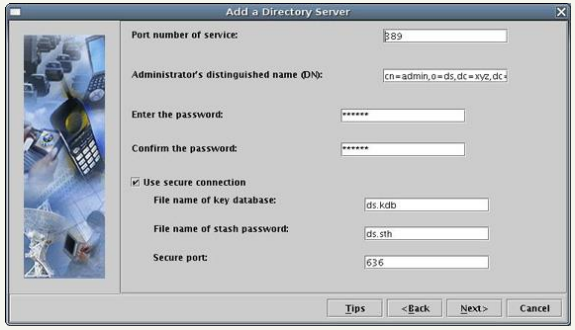


• **Autenticación Externa**

– La autenticación con la base de datos puede darse entre el DBMS y un servicio de autenticación externa.

– Cuando se utiliza autenticación basada en el sistema operativo, los usuarios se autentican con el SO del host y el DBMS confirma la autenticación con el SO.

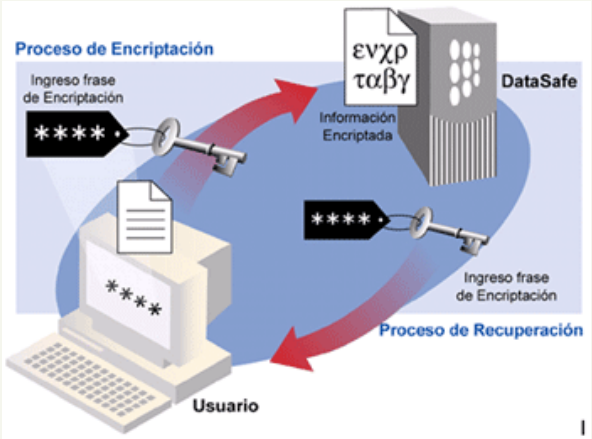
– Cuando se utiliza un servicio de directorios externo para autenticar usuarios con la base, la autenticación corre por parte del servicio de directorios. Generalmente se requiere del uso de protocolos como SSL o TLS para la comunicación.



• **Almacenamiento de Credenciales**

– El acceso a recursos externos, incluidas otras bases de datos, pueden resultar con el almacenamiento de contraseñas utilizadas para conectarse al recurso. Estas credenciales almacenadas requieren un estricto control.

– Las credenciales utilizadas para mantenimiento, administración de tareas u otras aplicaciones cuyas credenciales queden almacenadas en la base, requieren de protección y encriptación. La encriptación debe presentar fuertes mecanismos y el acceso a estas credenciales únicamente otorgado a cuentas autorizadas.



Recomendaciones en MySQL para hacer mas segura la base de datos

• No dé nunca a nadie (excepto a la cuenta root de MySQL acceso a la tabla User en la base de datos mysql) Esto es crítico. La clave cifrada es la verdadera clave en MySQL. • Estudie el sistema de privilegios de acceso de MySQL. Las sentencias GRANT y REVOKE se utilizan para controlar el acceso a MySQL. No otorgue más privilegios de los necesarios.

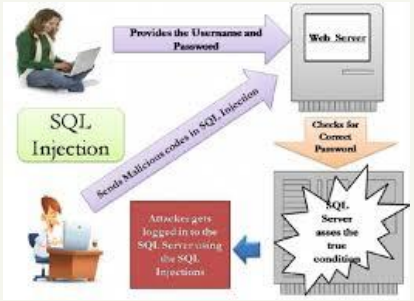
• No elija claves que puedan aparecer en un diccionario. Existen programas especiales para romperlas. Incluso claves como ``perro98'' son muy malas. Es mucho mejor ``oweei98''. • Invierta en un firewall. Le protegerá de al menos el 50% de todos los tipos de vulnerabilidades de cualquier software. Ponga MySQL tras el firewall o en una zona desmilitarizada (DMZ).

• Intente escanear sus puertos desde Internet utilizando una herramienta como nmap. MySQL utiliza el puerto 3306 por defecto. • Probar si el puerte MySQL está abierto, intente el siguiente comando desde alguna máquina remota, donde su servidor MySQL se estÃ¡ ejecutando: shell> telnet server_host 3306

INYECCIÓN SQL
¿QUÉ ES LA INYECCIÓN SQL?

Inyección SQL es un método de infiltración de código intruso que se vale de una vulnerabilidad informática presente en una aplicación en el nivel de validación de las entradas para realizar operaciones sobre una **base de datos**

•Ejemplo de inyección SQL
SELECT * FROM usuarios WHERE usuario = ‘ " + Usuario + “ ‘ and password=‘ "+ pass+ “ ‘;



•Un usuario cualquiera colocaría su nombre y su password de la siguiente manera:
SELECT * FROM usuariosWHERE usuario= ' pepe' and password =' 020304'

•Hasta aquí todo normal, pero un usuario podría modificar el campo password:
SELECT * FROM usuarios WHERE usuario = ' pepe' and password=' 020304' OR passwordLIKE '%'

•Como hemos visto la inyección SQL se ha hecho con el fin de burlar la restricción de acceso, pero se pueden realizar cosas más desastrosas en la BD, como por ejemplo:
DROP TABLE usuarios

PROTEGERSE DE INYECCIÓN SQL
•ASIGNACION DE MÍNIMOS PRIVILEGIOS
–Debe tener los privilegios necesarios, ni mas ni menos.

•VALIDARTODAS LAS ENTRADAS
–Especifique el tipo de dato de entrada, si son números, asegúrese de que son solo números.

•EMPLEO DE PROCEDIMIENTOS ALMACENADOS
–Utilizar procedimientos almacenados y aceptar los datos del usuario como parámetros en lugar de comandos sql.

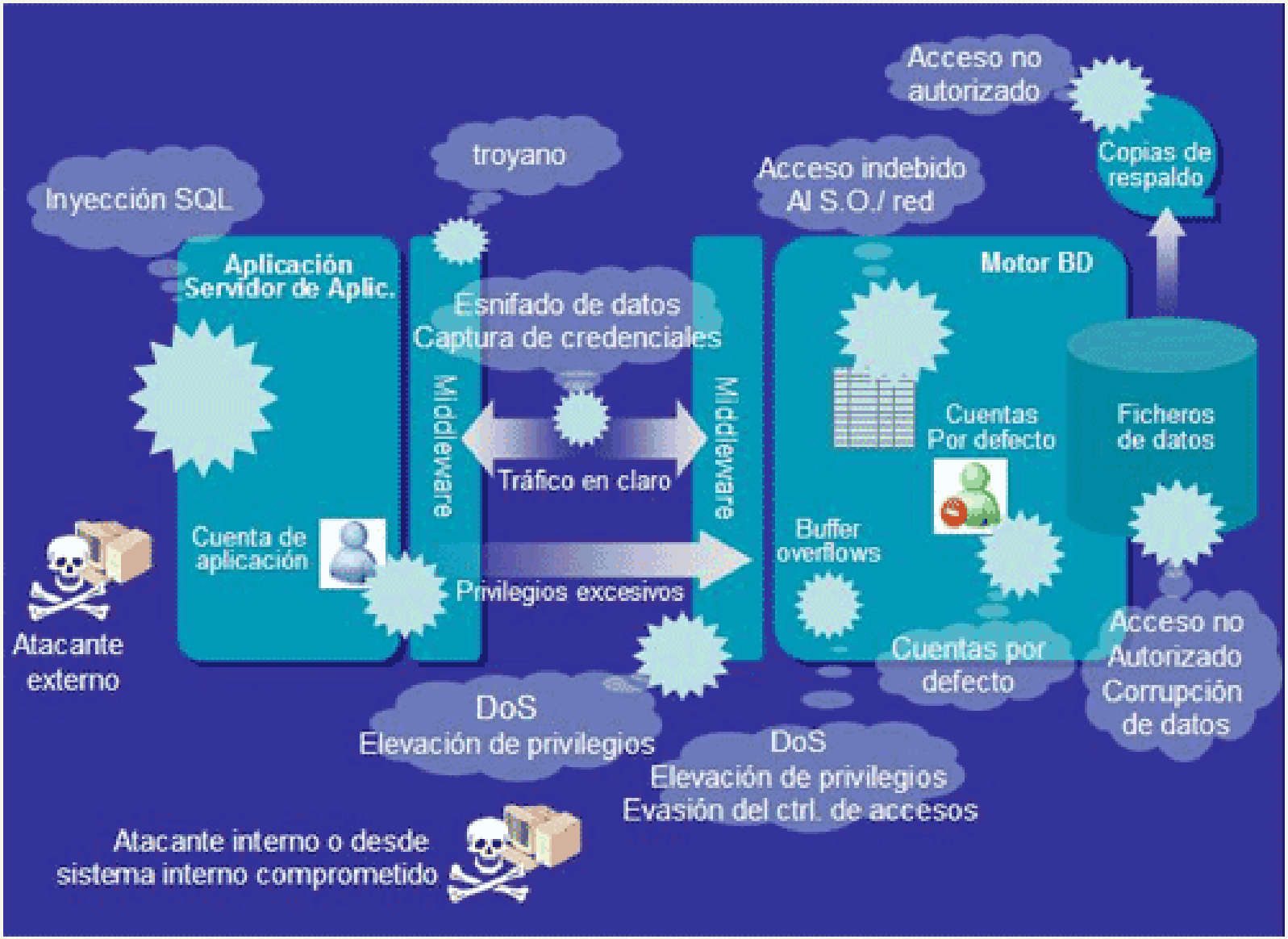
•UTILIZAR COMILLAS DOBLES EN LUGAR DE SIMPLES
–Puesto que las comillas simples finalizan las expresiones SQL, y posibilitan la entrada de expresiones de más potencia.

INYECCIÓN SQL EN

•EN PHP
–Para MySQL, la función a usar es mysql_real_escape_string:
Ejemplo:
$query_result= mysql_query("SELECT * FROM usuarios WHERE nombre= \"" . mysql_real_escape_string($nombre_usuario) . "\"");

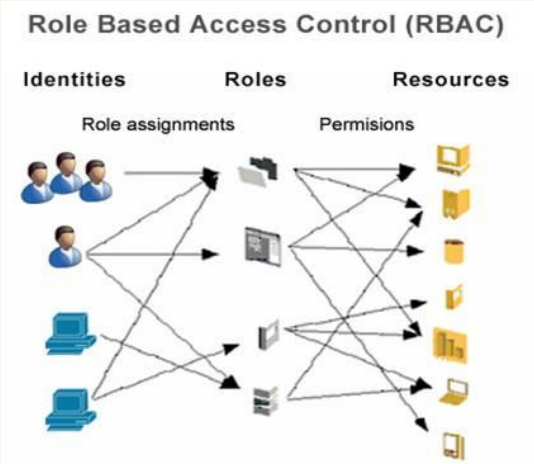
•EN JAVA
–En Java, tenemos que usar la clase PreparedStatement
En vez de:
Connectioncon = (acquireConnection) Statementstmt= con.createStatement(); ResultSetrset= stmt.executeQuery("SELECT * FROM usuarios WHERE nombre = '" + nombreUsuario+ "';");
Habría que poner:
Connectioncon = (acquireConnection) PreparedStatement pstmt= con.prepareStatement("SELECT * FROM usuarios WHERE nombre = ?"); pstmt.setString(1, nombreUsuario); ResultSetrset= pstmt.executeQuery();

•EN C#
–El siguiente ejemplo muestra cómo prevenir los ataques de inyección de código usando el objeto SqlCommand


Acceso no autorizado
troyano
Copias de respaldo
Inyección SQL
Acceso indebido Al S.O./ red
Aplicación Servidor de Aplic.
Esnifado de datos Captura de credenciales
Middleware
Middleware
Motor BD
Tráfico en claro
Cuentas Por defecto
Ficheros de datos
Cuenta de aplicación
Buffer overflows
Privilegios excesivos
Atacante externo
Cuentas por defecto
Acceso no Autorizado Corrupción de datos
DoS
Elevación de privilegios
DoS
Elevación de privilegios
Evasión del ctrl. de accesos
Atacante interno o desde sistema interno comprometido

Algunas bases de datos pueden permitir controles de acceso granulares, asignando controles sobre celdas específicas
– Las mejores prácticas sugieren segregación de funciones y asignación de los mínimos privilegios necesarios.

El método más aceptado es el RBAC (Role Based Access Control). Los roles son asignados a usuarios de la base de datos, basándose en las funciones que deben cumplir en la aplicación.



– A cada rol se le definen mínimos privilegios sobre ciertas tablas necesarias para el mencionado rol, luego a cada usuario se le asigna uno o varios roles de acuerdo a sus necesidades. De esta manera la seguridad es óptima, ya que cada función está llevada a un nivel de mínimos privilegios.

– Algunas aplicaciones multinivel utilizan un único usuario de base de datos para los accesos a la misma. Otras utilizan múltiples cuentas a fin de asistir a la base con las funciones de auditoría. La solución dependerá específicamente de cada aplicación.

– Según las mejores prácticas de segregación de funciones, la posibilidad de asignar privilegios a otros usuarios debería estar restringida a administradores autorizados únicamente

• Cuentas de Usuario por Defecto

– La mayoría de las bases de datos crean uno o más usuarios por defecto durante la instalación. El usuario DBA es el más común. Otras cuentas por defecto suelen encontrarse. Luego de la instalación, se recomienda revisar todas las cuentas creadas por defecto a fin de establecer las necesidades de cada una. Una buena práctica es renombrar las cuentas creadas por defecto para, de esta manera, limitar ataques sobre cuentas conocidas.

• Confidencialidad

– La información sensible es información no clasificada que, de ser expuesta, puede comprometer la seguridad o la privacidad de una persona. El Dueño de la Información es la persona responsable de la información y el encargado de definir controles de acceso y protección de información sensible almacenada en la base de datos de
manera tal que no sea visible por personas no autorizadas.

– Generalmente, esto incluye al DBA y a otros usuarios de la base de datos, por lo que la información deberá estar encriptada bajo fuertes mecanismos.

– Cuando la información se transmite desde la base de datos a un cliente, la información puede ser transmitida por canales no seguros. A fin de proteger la información enviada, el canal de comunicación podrá estar encriptado.

– La información sensible almacenada en las tablas no es la única que requiere encriptación. Los códigos fuentes de las aplicaciones pueden ofrecer a un atacante información sensible del manejo de información por parte de la misma (las credenciales de usuarios o llaves de encripción NUNCA deberían estar incluidas en elcódigo), por lo que los códigos fuentes deberán estar encriptados o encodeados si contiene información sensible.

– Dado que la información de las bases de datos está almacenada en archivos, dichos archivos deberán estar adecuadamente protegidos en el file system del host. En algunos casos estos archivos podrían estar también encriptados.

– La configuración de la base, los logs transaccionales, rastros de auditoría y demás archivos utilizados o creados por la base de datos podrían también estar encriptados.

Figura 2

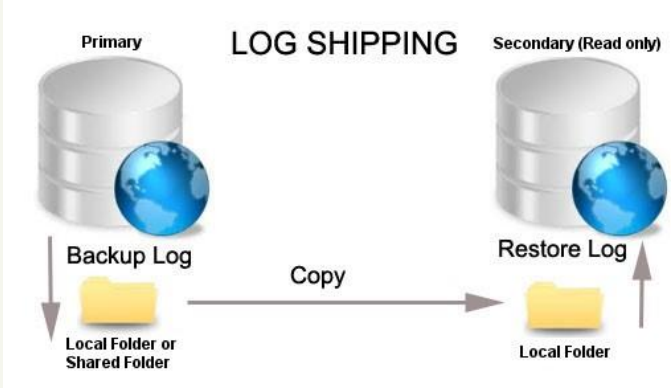
• Integridad de la Información

– Muchos DBMS proveen capacidades de logeo de transacciones o “redo logs” para proteger la integridad de la información y evitar inconsistencias. En caso de interrupciones previas a que la transacción sea adecuadamente completada, estos logs permiten volver a la tabla al estado conocido y reprocesar la transacción para corregir el error.

– Esta funcionalidad requiere un cuidadoso manejo de los logs de transacciones, de forma tal de brindarles capacidades de seguridad adecuadas ya que normalmente contendrán información sensible (lo cual pueden requerir encripción).

– Dado que también son necesarios para mantener la disponibilidad, deberán ser cuidados contra pérdida o daño de archivos (ej: mediante backups y redundancia).

– La integridad de los datos también puede ser protegida mediante controles previos a al almacenamiento del dato en las tablas, de forma tal de impedir que se asignen valores inválidos, inconsistentes o que no respeten relaciones establecidas. El uso de controles de tipo, rango u otros controles más inteligentes son recomendables en la aplicación para facilitar la tarea.



• Auditoría

– A la hora de definir capacidades de auditoría en una base de datos es importante establecer qué acciones pueden indicar tareas maliciosas o no autorizadas.

– La auditoría debe ser suficientemente detallada como para permitir el seguimiento de los eventos.

– La primera actividad a considerar es la conexión a la base. La decisión de registrar todas las conexiones a la base o sólo las fallidas depende del uso que se le va a dar a la base. Algunas bases tienen una cantidad muy grandes de conexiones constantemente, lo que hace imposible registrar todas las conexiones.

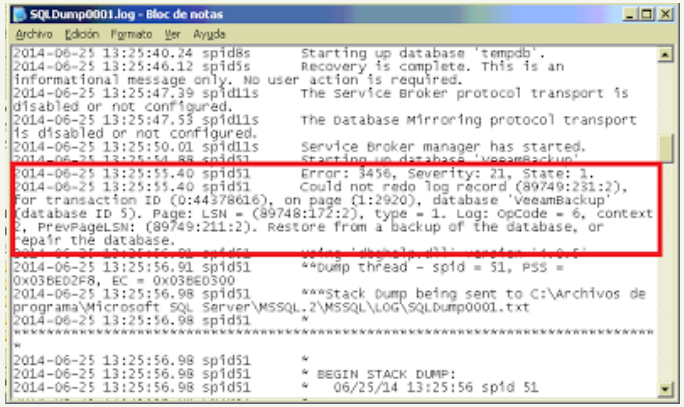
– Asimismo deben registrase actividades como: asignación de privilegios, actividades administrativas, agregado, actualización o borrado de objetos de la base. No debemos olvidar registrar todos los accesos a los logs de auditoría de la base.

– Por su parte, deben identificarse las tablas correspondientes a las aplicaciones más críticas que requieran una auditoría más extensiva. En algunos casos debería considerarse el monitoreo de modificaciones en tablas críticas y el uso de alarmas.

– Los logs de auditoría siempre deben considerarse críticos. Por esta razón su acceso de lectura debe estar restringido (pensemos en la información que pueden contenter) y deben eliminarse todos los permisos de modificación de los mismos (un usuario malicioso podría querer modificarlos para borrar sus huellas), sólo permitiendo el borrado a personal autorizado para casos de depuración, tarea que debe quedar registrada.

– Dependiendo de las regulaciones vigentes, el período de retención de los logs de auditoría puede variar de algunos meses a varios años (ej: SOX, banco central, ley dehabeas data, etc).

– No debemos olvidar que no sólo es necesario generar los logs sino también definir procesos de revisión, en busca de actividades anormales. Estos procesos pueden automatizarse e incluso pueden generar reportes que contengan información estadística e indicadores.



– Dado que los logs de auditoría pueden crecer demasiado, es recomendable almacenar offline aquellos registros mayores a 1 mes mientras que estén online los más nuevos (rotación de logs).

– En muchos casos es importante considerar que el log de la base de datos puede no contener el verdadero nombre del usuario sino sólo el del servicio de la aplicación que accede la base. Algunos DBMS ya consideran esta limitación y proveen herramientas para transferir el nombre del usuario hasta el registro de auditoría.

– En las bases de datos de desarrollo la protección de los datos contenidos no es importante (siempre y cuando no se haya copiado de una base de producción). Sin embargo es importante que el código fuente de desarrollo no sea modificado, razón por la cual es recomendable auditar los accesos al mismo.

• Replicación y Federación

– Muchos DBMS proveen la capacidad de que una conexión iniciada a las mismas sea remitida a otros DBMS. Estas conexiones están definidas dentro de la misma base y muchas veces se almacenan las credenciales dentro de la misma base. Esta situación se conoce como “Database Links”.

– Si la seguridad de un DBMS en la cadena estuviera compometida, se podría comprometer la seguridad del resto de los DBMS de la cadena. Por esta razón debería tratar de evitarse su uso. Sin embargo, en los casos en que fuera requerido, debería considerarse las siguientes medidas:

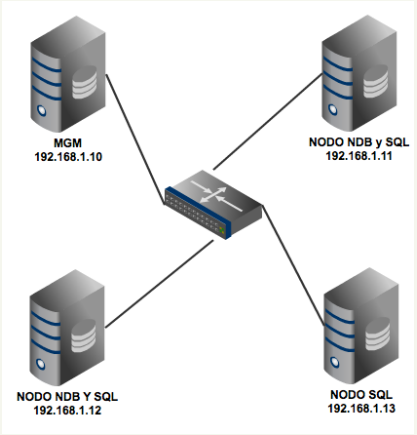
1. Utilizar diferentes credenciales para diferentes links.
2. Utilizar credenciales de un servicio de directorio en lugar de links.
3. Auditar el uso de los links.
4. Proteger adecuadamente los links.
5. Securizar los canales de comunicación entre los DBMS.
6. Sensibilidad de los datos que pasarán por estos links.
7. Privilegios asignados a las cuentas de los DBMS remotos

– Los “database links” normalmente son usados para proveer servicios de backup.
También pueden ser usados para replicación de las bases de datos.

– Federación de bases de datos es la situación que se presenta cuando los datos están distribuidos en dos o más bases. De esta manera se distribuye la carga y en algunos casos también pueden proveer alta disponibilidad. En estos casos también se utilizan “database links”.

• Clustering de bases de datos

El clustering provee alta disponibilidad de las bases de datos. Esta configuración permite a dos DBMS funcionar en forma independiente pero brindando el mismo servicio y virtualizado como si fueran un sólo DBMS. De esta forma se permite una recuperación transparente en caso de caída de un DBMS.

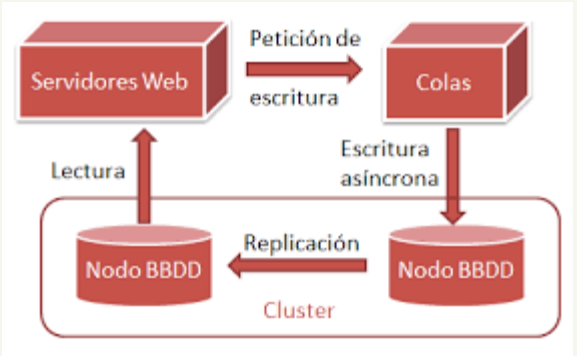


– El clustering debiera considerar los mismos requerimientos de seguridad que otras actividades que incluyan accesos remotos a bases de datos, incluyendo:

– La seguridad de las cuentas de usuario para los servicios de clustering, considerando que estas cuentas sólo sean utilizadas para estos fines.

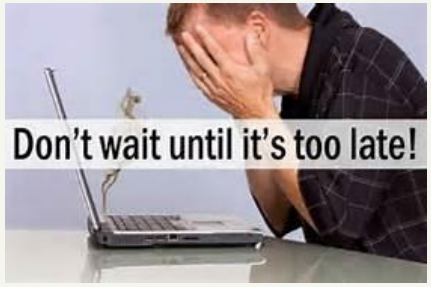
– La sincronización entre ambos DBMS.

– La seguridad de los vínculos de comunicaciones.



• Backup y recuperación

– Los procesos de backup y recuperación de bases de datos son una parte importante de la disponibilidad e integridad de los DBMS. Después de instalarlos, la mayor parte de los DBMS no contemplan tareas de backup automáticamente. Adicionalmente, dado que mientras el DBMS esté operando existen archivos abiertos, muchos productos de backup no contemplan nativamente el soporte para motores de bases de datos. Por esta razón muchos DBMS están empezando a incluir capacidades de backup en forma nativa.



– Actualmente la "ventana de backup" es uno de los aspectos más importantes a contemplar ya que en la misma el DBMS generalmente está detenido.

– Dado que los backups contienen la misma información (y a que es más fácil robar una cinta que un servidor), es especialmente importante la seguridad física que se asigne a la protección de las cintas de backup. En muchos casos inclusive se recomienda encriptar la información de los backups.

• **Protecciones al sistema operativo**

– Como cualquier otra aplicación, los DBMS dependen de la seguridad del host. Una configuración débil de la seguridad del host puede comprometer la seguridad del DBMS y viceversa.

– Es recomendable que el host que soporte al DBMS no brinde otros servicios adicionales (ej.: web server, active directory, file server), ya que a más servicios, es más difícil lograr niveles aceptables de seguridad y si un servicio fuera comprometido, el resto también lo estará.

– Lo más importante a tener en cuenta respecto al sistema operativo es la seguridad del file system que soportará la base de datos, considerando los permisos en archivos y directorios y la auditoría de los mismos. También es importante que los archivos de la base de datos estén almacenados en particiones distintas al sistema operativo. Esto
simplifica el uso de distintos esquemas de hardware para cada caso (ej: RAID, tecnología de discos).

– Las cuentas de usuario del sistema utilizadas por el servicio del DBMS deberán ser únicas. Los permisos para esta cuenta deberán ser cuidadosamente definidos. Normalmente los DBMS utilizan diferentes cuentas de servicio para diferentes componentes (ej.: motor de BD, componente de backup, componente de logs)

– En los casos donde se instalen múltiples DBMS en múltiples servidores es recomendable utilizar distintas cuentas de servicio en cada caso para limitar el riesgo en caso que la misma sea comprometida.

– Dependiendo de la estructura de la organización, es recomendable que el administrador del host no posea los privilegios del DBMS, para delimitar correctamente las responsabilidades. De esta forma también se limita el riesgo en caso que alguna de las cuentas haya sido comprometida.

– Dependiendo de la tecnología de backup utilizada, este proceso puede ser gestionado por el sistema operativo del host directamente. En este caso el administrador del host será responsable de la ejecución de los backups, sin embargo, el administrador de la base de datos deberá dar seguimiento al proceso para asegurarse que esté completamente realizado y definido.

– Es importante considerar las actualizaciones al DBMS que son provistas por el fabricante periódicamente, teniendo en cuenta que el fabricante provee parches para vulnerabilidades conocidas para las versiones más nuevas únicamente. Por esta razón es importante mantener el DBMS en una versión que aún sea soportada por el fabricante (no descuidar los parches del host también).

– La aplicación de estos parches debe estar cuidadosamente planificada, probando exhaustivamente en desarrollo previo a aplicarlos en el ambiente de producción.

• Protección de las aplicaciones
– Las aplicaciones introducen la mayor cantidad de vulnerabilidades y vías de ataques a las bases de datos. Peor es el caso de aquellas aplicaciones web publicadas a Internet. En la gran mayoría de los casos, el usuario utilizado por la aplicación para acceder al DBMS cuenta con muchos más privilegios de los realmente requeridos, esto provee una vía para un atacante que quiere tomar el control de la aplicación.

– Adicionalmente, si la aplicación no genera logs de auditoría es muchas veces difícil o incluso imposible conocer el verdadero autor de un ataque.
– Asimismo, cuando la aplicación no controla los parámetros de entrada, se convierte en la principal vía de ataque a la base de datos (SQL injection)

– Todas estas razones son más que suficientes para prestar principal atención al diseño de las aplicaciones que se ponen delante de las bases de datos.

– Es importante considerar también el esquema de autenticación que utilizará la aplicación y cómo interactuará con la base de datos, validando las contraseñas, su complejidad, los canales de comunicaciones entra las mismas y cómo almacenará las credenciales.

– La cuenta de usuario utilizada por la aplicación para acceder al DBMS deberá tener sólo los privilegios necesarios para utilizar las tablas asignadas, eliminando aquellos privilegios excesivos (ej: versión y modelo del DBMS, tablas de usuarios, objetos y store procedures).

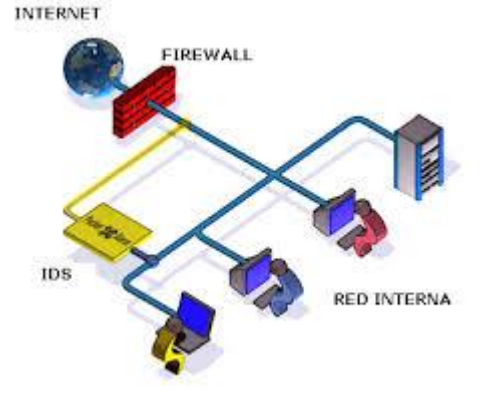
• **Protección** de la red

– Los accesos a la base de datos pueden ser controlados por elementos de red (firewalls, router, etc). Esto se vuelve más crítico con el uso de aplicaciones web publicadas a Internet (aplicaciones multicapa). Incluso algunos DBMS proveen acceso web directamente vía HTTP.

– Lo primero a considerar es la necesidad de accesos de los clientes del DBMS. En muchos casos existe un componente llamado “listener” que controla los accesos de los clientes al DBMS, permitiendo aplicar algunos filtros adicionales.

– Adicionalmente a filtrar los puertos, es posible definir controles adicionales, ej: por tiempo o por cantidad de conexiones. Esto puede proteger al DBMS de ataques de denegación de servicio.

– Finalmente, no debemos olvidar la arquitectura de red que dará soporte al sistema, considerando por qué vínculos circulará la información sensible y si los mismos estarán encriptados o no. En los casos en los que los componentes no soporten encripción se puede utilizar VPN.



Lecturas recomendadas

[1] ISO/IEC 27001:2005 - Information technology -- Security techniques [en] http://www.iso.org/iso/catalogue_detail?Csnumber=42103

[2] ISO/IEC 17799:2005 - Information technology -- Security techniques [en] http://www.iso.org/iso/catalogue_detailcsnumber=39612

[3] Malware - Ataque a la Base de Datos [en] http://ataquebd.blogspot.mx/

[4] Inyección de código SQL - MSDN – Microsoft [en] http://msdn.microsoft.com/es-es/library/ms161953.aspx

[5] Escolano F. "Inteligencia Artificial", Editorial Paraninfo, 2003

[6] Aguilera L "Seguridad Informática" 2010, Madrid, Editorial Editex, S.A.

- El Reporte X-Force de IBM revela que el phishing y las amenazas relacionadas a documentos se incrementan [en] http://www.lawebdelprogramador.com/noticias/mostrar.php?id=2460
- http://sox.sourceforge.net/
- Daniel Camargo Montero, Sistema de selección de personal inspirado en agentes inteligentes, [en] http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/camargo_m_d/

Bibliografía

R. Elmasri y S. Navathe. Fundamentos de los Sistemas de Bases de Datos (3ª edición). Addison-Wesley, 2002.

A. Silberschatz, H. F. Korth y S. Sudarshan. Fundamentos de Bases de Datos (4ª edición). McGraw Hill, 2002

B. ROBERT Fundamentos de SQL , 3ra Edición- Andy Oppel 2016