SEGURIDAD EN BASES DE DATOS

AUTOR: ORLANDO ESPINOZA BARBOZA

DICIEMBRE: 2020

San Marcos

Tabla de Contenido

Introducción a la seguridad

Conceptos básicos

En el libro #1, *Introducción a la seguridad informática y el análisis de vulnerabilidades,* los autores indican lo siguiente: La seguridad consiste en un estado de bienestar, es la ausencia de riesgo por la confianza que existe en alguien o algo. Es una ciencia interdisciplinaria para evaluar y gestionar los riesgos. La seguridad siempre busca la gestión de riesgos, esto quiere decir que se tenga siempre una forma de evitarlo o prevenirlo y que se pueda realizar ciertas acciones para evitar esas situaciones de la mejor forma. Es decir que la seguridad podría ser catalogada como la ausencia de riesgo. Con esto se involucran cuatro acciones que siempre están inmersas en cualquier asunto de seguridad como son:

- Prevención del riesgo
- Transferir el riesgo
- Mitigar el riesgo
- Aceptar el riesgo

De acuerdo a la página https://es.wikipedia.org/wiki/Seguridad_de_la_información se extrae la siguiente información:

La seguridad de la información es el conjunto de medidas preventivas y reactivas delas organizaciones y sistemas tecnológicos que permiten resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e integridad de datos.

El concepto de seguridad de la información no debe ser confundido con el de seguridad informática, ya que este último sólo se encarga de la seguridad en el medio informático, pero la información puede encontrarse en diferentes medios o formas, y no solo en medios informáticos.

En la seguridad de la información es importante señalar que su manejo está basado en la tecnología y debemos de saber que puede ser confidencial: la información está centralizada y puede tener un alto valor. Puede ser divulgada, mal utilizada, ser robada, borrada o saboteada. Esto afecta su disponibilidad y la pone en riesgo. La información es poder, y según las posibilidades estratégicas que ofrece tener acceso a cierta información, ésta se clasifica como:

Crítica: Es indispensable para la operación de la empresa.

Valiosa: Es un activo de la empresa y muy valioso.

Sensible: Debe de ser conocida por las personas autorizadas.

Existen dos palabras muy importantes que son riesgo y seguridad:

Riesgo: Es la materialización de vulnerabilidades identificadas, asociadas con su probabilidad de ocurrencia, amenazas expuestas, así como el impacto negativo que ocasione a las operaciones de negocio.

Seguridad: Es una forma de protección contra los riesgos.

La seguridad de la información tiene como objeto los sistemas el acceso, uso, divulgación, interrupción o destrucción no autorizada de información. La finalidad es proteger la confidencialidad, integridad y disponibilidad de la información.

Confidencialidad

La confidencialidad es la propiedad que impide la divulgación de información a individuos, entidades o procesos no autorizados. Asegura el acceso a la información únicamente a aquellas personas que cuenten con la debida autorización.

Integridad

Es la propiedad que busca mantener los datos libres de modificaciones no autorizadas. La integridad es mantener con exactitud la información tal cual fue generada, sin ser manipulada ni alterada por personas o procesos no autorizados.

Disponibilidad

La disponibilidad es la característica, cualidad o condición de la información de encontrarse a disposición de quienes deben acceder a ella, ya sean personas, procesos o aplicaciones. La disponibilidad es el acceso a la información y a los sistemas por personas autorizadas en el momento que así lo requieran.

Autenticación

Es la propiedad que permite identificar el generador de la información. En un sistema informático se suele conseguir este factor con el uso de cuentas de usuario y contraseñas de acceso u otro tipo de credenciales.

Evolución de las metodologías de ataque

En el libro #1, *Introducción a la seguridad informática y el análisis de vulnerabilidades,* los autores indican lo siguiente:

Ataque informático

Es un intento para provocar daños a una determinada infraestructura tecnológica compuesta por equipos críticos, sistemas informáticos, redes internas, software, etc. Se aprovechan de las debilidades o vulnerabilidades en las redes, sistemas informáticos o las personas que conforman el ambiente informático, malas configuraciones, protocolos débiles, etc. El objetivo es obtener un beneficio, normalmente económico, algunas veces políticos, provocando un daño en la seguridad del sistema y en los activos de la organización.

Hay muchos tipos de ataques informáticos, los más comunes son: Malware, rasomware, espías, virus, troyanos y otros programas sospechosos, ingeniería social, phishing.

Existen muchas amenazas de varias fuentes principalmente de internet que pueden ser: correos electrónicos infectados por virus, firewalls mal configurados, suplantación de contraseñas, contraseñas débiles, robo y destrucción de información, etc.

Metodología de ataque

Normalmente un ataque informático se basa en diferentes fases entre ellas:

- **Reconocimiento**: Se estudia la posible víctima con técnicas que proporcionan información necesaria para el posible ataque.
- **Exploración**: Con la información obtenida, se utiliza para obtener información de accesos como la dirección IP, credenciales, servidores, host entre otros.
- **Obtener el acceso**: Cuando se tiene la información necesaria para el ataque, se analiza las debilidades o vulnerabilidades del sistema ingresando al objetivo identificado.
- **Mantener el acceso**: Los hackers, una vez tenido un ataque exitoso, guardan la información para cuando quieran volver a realizar otro ataque, esto mediante herramientas como puertas traseras, gusanos, etc.
- **Borrar las huellas**: El objetivo de esta fase es eliminar cualquier rastro que se haya dejado.

Principales vectores de ataque

Los vectores de ataque se basan principalmente en las vulnerabilidades o fallas en los sistemas, errores de diseño, configuración o implementación que generan oportunidades de ataque, es decir que hacen viable una amenaza. Estos ataques están orientados hacia el software, hardware y recursos humanos:

Software: compuesto de aplicaciones, servicios, ejecutables, páginas web u otros servicios. Las vulnerabilidades en el software son fallas en la programación o compilación de los programas que ejecutan las computadoras a servidores, los ataques a estas vulnerabilidades pueden derivar en un mal funcionamiento del software, acceso a información restringida, fallos de sistema, etc. Para reducir esta superficie de ataque, hay que reducir al mínimo el software instalado en las computadoras y servidores, mantener actualizado el software y aplicar todos los parches de seguridad publicados por los desarrolladores.

Hardware: Para atacar a un dispositivo hardware, el atacante necesita tener acceso físico al dispositivo. En este caso es fácil analizar que las amenazas naturales como fallos por envejecimiento de equipos o desastres como robos, incendios o inundaciones, afecta específicamente a esta superficie de ataque. Los ataques a hardware también pueden producirse a través de la red o afectando al medio físico de transmisión, por ejemplo, los perturbadores de señal pueden interrumpir las comunicaciones de distinto tipo de tecnología inalámbrica mediante la generación de ruido radioeléctrico en la frecuencia y forma correcta. Este tipo de ataques podría anular sistemas de comunicaciones de los que dependen alarmas, sensores o cualquier otro tipo de comunicaciones, ya sean entre dispositivos o personas.

Recursos humanos: La última superficie de ataque correspondiente a los recursos humanos, que pueden actuar contra los intereses de la organización por descontento, error, engaño o coacción. Además de implementar y exigir el cumplimiento de protocolos de actuación, es aconsejable implementar sistemas de registro y auditoría para verificar quién hace qué y cuándo, de este modo al evitar el anonimato se minimiza la probabilidad de éxito de una amenaza de carácter humano, además se debe invertir esfuerzo y recursos en educar y concienciar a los usuarios de nuestros recursos e infraestructuras para que se impliquen a la hora mantener un alto nivel de seguridad.

Mecanismos de protección generales

La mayoría de los ataques informáticos se pueden evitar o por lo menos disminuir el impacto, si se hiciera utilizando mecanismos protectores o preventivos que consisten en una serie de revisiones periódicas, algunos cambios o mejoras de diferentes aspectos que pueden ser de hardware, software o de cualquier elemento involucrado en los sistemas y procesos.

Entre los mecanismos preventivos se pueden mencionar:

El respaldo de información: Es uno de los procesos más comunes que se pueden realizar en las compañías. Se debe tener un horario de respaldo normalmente en las horas de menos tráfico, control de los medios para que no todas las personas tengan acceso a los respaldos y el tipo de compresión de la información para asegurar que la información sea completa y recuperable.

Otros mecanismos preventivos son los siguientes:

Actualización de sistemas siguiente protocolos seguros y metodologías adecuadas de actualizaciones.

Antivirus para detectar y eliminar virus y software sospechosos. Son programas que se basan en la detección de malware en la fase de pre- ejecución, es decir que analizan archivos y programas antes de que se ejecuten para prevenir que puedan hacer algo malo.

Firewall para la protección de la red privada de ataques o intrusiones no autorizadas, mediante el bloqueo de accesos no autorizados permitiendo las comunicaciones y accesos autorizados.

Navegación por internet principalmente para el control de las páginas seguras que pueden ser accesadas y así controlar el tráfico en la web. El mal uso por parte de los colaboradores de la empresa puede ocasionar perjuicios, de difícil identificación. El desperdicio de tiempo, en actividades improductivas, utilizando Internet, impacta drásticamente en el desempeño de funcionarios, ampliando también la posibilidad de ocurrencia de otros eventos que pueden comprometer la seguridad de la información.

Contraseñas y credenciales seguras: Para proteger los sistemas de información, se deben usar contraseñas seguras y tienen que tomar precauciones para que ésta no sea conocida por nadie más aparte de ellos, de no hacerlo corren un alto riesgo de

suplantación de identidad. Mantener un nivel de seguridad adecuado reduce el riesgo lo cual se traduce en mejores resultados, por otro lado, carecer de esta seguridad hace que la empresa sea mucho más vulnerable con los consiguientes problemas que esto conlleva.

Accesos remotos seguros: Es una tecnología que permite conexiones seguras desde un dispositivo a otro dispositivo terminal ubicado en otro lugar. Esta conexión permite a los usuarios acceder a una red o una computadora de forma remota a través de una conexión a Internet o telecomunicaciones.

Bases de datos

Diseño de bases de datos

En el libro #4, *Fundamentos de bases de datos,* el autor indica lo siguiente:

Una base de datos es un conjunto de elementos de datos interrelacionados, administrados como unidad. Un objeto de base de datos es una estructura de datos con nombre que se guarda en una base de datos

El sistema de administración de bases de datos (DBMS) es el software que ofrece todos los servicios básicos requeridos para organizar y conservar una base de datos: Administración de consultas por parte de usuarios, pueden ser concurrentes o no. Control de transacciones para los cambios en los datos. Contener un lenguaje de consultas y transacciones. Proporción de mecanismos de respaldo y recuperación. Mecanismos de seguridad para evitar la consulta y modificación no autorizadas de los datos

Las bases de datos son estructura administradas por un sistema administrador de bases de datos en diferentes esquemas, pero normalmente se basan en capas físicas, lógicas y externas

La capa física

La capa física incluye los archivos físicos que contienen toda la información de la base de datos.

Casi todos los DBMS modernos permiten que la base de datos se guarde en varios archivos, que se suelen distribuir en varias unidades de disco físicos.

El usuario de una base de datos no necesita comprender cómo se guardan los datos dentro de los archivos ni cuáles archivos contienen los elementos de datos que le interesan. Para esto existe el administrador de base de datos (DBA) el cual maneja los detalles de instalación y configuración del software y los archivos de la base de datos que permiten presentarla a los usuarios.

La capa lógica

La capa lógica (o modelo lógico) está formada por las primeras dos capas de abstracción en la base de datos: la capa física tiene una existencia concreta en los archivos del sistema operativo, mientras que la capa lógica sólo existe como estructuras abstractas de datos integradas en la capa física, según se requiera.

La capa externa

La capa externa (o modelo externo) es la segunda capa de abstracción de la base de datos. Esta capa está formada por las vistas de usuarios analizadas antes, a las que en conjunto se les denomina esquema secundario. En esta capa, los usuarios (los programas de aplicaciones y las personas) que la consultan se conectan y plantean consultas contra la base de datos. Lo ideal es que sólo el DBA administre las capas física y lógica. El DBMS controla la transformación de los elementos seleccionados de una o más estructuras de datos en la capa lógica para formar la vista de cada usuario.

Por su lado, un modelo de bases de datos es la arquitectura que utiliza el DBMS para guardar los objetos dentro de la base de datos y relacionarlos entre sí. El modelo relacional es el modelo moderno más utilizado actualmente y se basa en la agrupación de registros comunes en objetos llamados tablas. El modelo relacional presenta los datos en las familiares tablas bidimensionales y la relación entre los objetos o tablas.

Algunos principios guían el proceso de diseño de la base de datos. El primer principio es que la información duplicada o datos redundantes es perjudicial, porque se pierde espacio y aumenta la probabilidad de errores e incoherencias. El segundo principio es que la corrección y la integridad de información es importante. Si la base de datos contiene información incorrecta, los informes que extraigan la información de la base de datos también contendrán información incorrecta. Como resultado, las decisiones que tome basándose en dichos informes estarán mal informadas.

Por ejemplo, se toma la base de datos clásica de un sistema de pedidos llamada NortWind

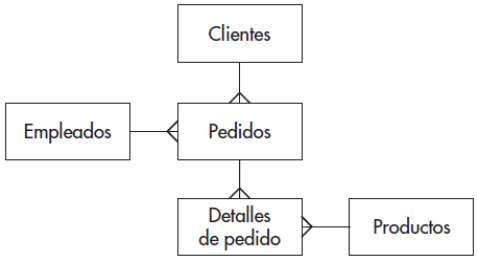


Tabla Clientes

Id de cliente	Compañía	Apellidos	Nombre	Cargo	Ciudad	Estado o provincia
6	Compañía F	Pérez-Olaeta	Francisco	Jefe de compras	Milwaukee	WI
26	Compañía Z	Pinto	Armando	Ayudante de contabilidad	Miami	FL

Tabla Pedidos

Id de pedido	Id de cliente	Id de empleado	Fecha de pedido	Fecha de envío	Gastos de envío
51	26	9	05/04/2006	05/04/2006	$60.00
56	6	2	03/04/2006	03/04/2006	$0.00
79	6	2	23/06/2006	23/06/2006	$0.00

Tabla Empleados

Id	Apellidos	Nombre	Cargo
2	Escolar	Jesús	Vicepresidente de ventas
5	San Juan	Patricia	Jefe de ventas
9	Chaves	Francisco	Representante de ventas

De acuerdo a la página Internet de Microsoft Support/Conceptos Básicos del diseño de una Base de Datos se da la siguiente información:

El proceso de diseño

El proceso de diseño consta de los siguientes pasos:

- **Determinar el propósito de la base de datos**

Es una buena idea anotar el propósito de la base de datos: su propósito, cómo espera usarla y quién la usará. Por ejemplo, para una base de datos pequeña para un negocio familiar, escriba algo como: "La base de datos de clientes es una lista con información de los clientes cuya finalidad es el envío de correo y la creación de informes". Si la base de datos es más compleja o la usan muchas personas, como ocurre normalmente en un entorno corporativo, el propósito podría constar fácilmente de uno o varios párrafos, y debería incluir cuándo y cómo cada persona usará la base de datos. La idea es tener una declaración de objetivos bien desarrollada a la que se pueda hacer referencia en todo el proceso de diseño.

- **Buscar y organizar la información necesaria**

Recopile todos los tipos de información que podría querer registrar en la base de datos, como los nombres de producto y los números de pedido. Recopile dichos documentos y enumere cada tipo de información que se tiene. Si no tiene ningún formulario existente, imagine en su lugar que tiene que diseñar un formulario para registrar la información del cliente. ¿Qué información incluiría en el formulario? ¿Qué cuadros de relleno crearía? Identifique y enumere cada uno de estos elementos. Después, tenga en cuenta los tipos de informes o correspondencia que podría querer crear a partir de la base de datos.

- **Dividir la información en tablas**

Divida los elementos de información en entidades principales o temas, como Productos o Clientes. Después, cada tema se convierte en una tabla. Para dividir la información en tablas, elija las entidades principales, o asuntos. Por ejemplo, después de encontrar y organizar la información para una base de datos de ventas de un producto, la lista preliminar podría ser similar a la siguiente:

Las principales entidades que se muestran aquí son los productos, los proveedores, los clientes y los pedidos. Por tanto, tiene sentido comenzar con estas cuatro tablas: una para los datos sobre productos, otra para datos sobre proveedores, otra para los datos sobre clientes y otra para los datos sobre pedidos. Aunque la lista no está completa con ellas, es un buen punto de partida. Puede seguir ajustando la lista hasta que tenga un diseño que funcione bien.

- **Convertir los elementos de información en columnas**

Decida qué información quiere almacenar en cada tabla. Cada elemento se convierte en un campo y se muestra como una columna en la tabla. Por ejemplo, una tabla de empleados podría incluir campos como Apellidos y Fecha de contratación. Para determinar las columnas de una tabla, decida cuál es la información de la que necesita realizar un seguimiento sobre el tema registrado en la tabla. Por ejemplo, para la tabla Clientes, una buena lista inicial de columnas contendría Nombre, Dirección, Código postal, Dirección de correo electrónico. Algunas de las sugerencias son:

No incluya datos calculados. Almacene la información en sus partes lógicas más pequeñas.

- **Especificar las claves principales**

Elija la clave principal de cada tabla. La clave principal es una columna o varias que se usa para identificar cada fila o registro. Un ejemplo podría ser Id. de producto o Id. de pedido. Una clave principal siempre debe tener un valor. Si en algún momento el valor de una columna puede quedar sin asignar o ser desconocido (un valor que falta), no se puede usar como un componente de una clave principal.

- **Establecer las relaciones de tablas**

Busque en cada tabla y decida cómo se relacionan los datos en una tabla con los datos de otras tablas. Agregue campos a las tablas o cree tablas para aclarar las relaciones, según sea necesario. Es importante determinar el tipo de relación den las tablas, si es de uno a varios, de varios a varios, de uno a uno.

- **Perfeccionar el diseño**

Analice el diseño en busca de errores. Cree las tablas y agregue unos cuantos registros de datos de ejemplo. Compruebe si puede obtener los resultados que quiere de las tablas. Haga algunos ajustes en el diseño, si es necesario. Esto le permitirá resaltar los posibles problemas. Por ejemplo, tal vez deba agregar una columna que olvidó insertar durante la fase de diseño, y es posible que tenga una tabla que debería dividir en dos tablas para eliminar los datos duplicados.

Vea si puede usar la base de datos para obtener las respuestas que quiere. Cree bocetos de los formularios e informes y compruebe si muestran los datos que espera. Busque duplicaciones de datos innecesarias y, si encuentra alguna, modifique el diseño para eliminarla.

Cuando pruebe la base de datos inicial, probablemente descubrirá posibilidades de mejora. Estas son algunas cosas que debería comprobar:

¿Olvidó alguna columna? Si es así, ¿la información pertenece a las tablas existentes? Si se trata de información sobre otra cosa, tal vez necesite crear otra tabla. Cree una columna para cada elemento de información del que necesite realizar un seguimiento. Si no se puede calcular la información de otras columnas, es probable que necesite una nueva columna para ella.

¿Cualquiera de las columnas innecesarias lo son porque se pueden calcular con los campos existentes? Si se puede calcular un elemento de información desde otras

columnas existentes, como, por ejemplo, un precio de descuento calculado a partir del precio de venta, generalmente es mejor simplemente hacerlo y evitar crear una nueva columna.

¿Ha introducido información duplicada en una de las tablas? Si es así, probablemente tenga que dividir la tabla en dos tablas que tengan una relación de uno a varios.

¿Tiene tablas con muchos campos, un número limitado de registros y muchos campos vacíos en registros individuales? Si es así, piense en rediseñar la tabla para que tenga menos campos y más registros.

¿Se ha dividido cada elemento de información en sus partes más pequeñas? Si necesita realizar informes, ordenar, buscar o calcular sobre un elemento de información, coloque dicho elemento en su propia columna.

¿Cada columna contiene datos sobre el tema de la tabla? Si una columna no contiene información sobre el tema de la tabla, pertenece a otra tabla.

¿Las relaciones son entre las tablas representadas, bien sea por los campos comunes o por una tercera tabla? Las relaciones de uno a uno y de uno a varios requieren columnas comunes. Las relaciones de varios a varios requieren una tercera tabla.

- **Aplicar las reglas de normalización**

Aplique las reglas de normalización de datos para ver si las tablas están estructuradas correctamente. Haga algunos ajustes en las tablas, si es necesario. Dichas reglas se usan para ver si las tablas están estructuradas correctamente. El proceso de aplicar las reglas al diseño de la base de datos se denomina normalización de la base de datos, o simplemente, normalización. En cada paso, aplica las reglas consecutivamente, para garantizar que su diseño llegue a tener lo que se denomina "formulario normal" o formula normal.

Primer formulario normal

El primer formulario normal indica que, en cada intersección de fila y columna de la tabla, existe un valor único y nunca una lista de valores. Por ejemplo, no puede tener un campo denominado Precio en el que coloca más de un precio. Si piensa en cada intersección de filas y columnas como en una celda, cada celda puede contener un solo valor.

Segundo formulario normal

El segundo formulario normal requiere que cada columna de clave dependa completamente de toda la clave principal y no solo de parte de la clave. Esta regla se aplica cuando tiene una clave principal que consta de más de una columna.

Tercer formulario normal

El tercer formulario normal requiere que no solo cada columna de clave dependa de toda la clave principal, sino también que las columnas que no son claves sean independientes entre sí.

Otra forma de decirlo es que cada columna que no sea de clave debe depender de la clave principal y nada más que de la clave principal.

Conexión a una base de datos

Una conexión a una base de datos es un archivo de configuración donde se especifica los detalles físicos de las bases de datos, como por ejemplo la dirección IP, el nombre de la base de datos, las credenciales de conexión (que normalmente estas se dan en el momento del acceso), el tipo de conexión que va a realizar (ODBC, ADO, OLE DB, JDBC, etc.).

Por ejemplo, una conexión a una base de datos Oracle desde C# puede ser:

```
// Crea cadena de conexion
 String cadenaConexionOracle = "Data Source=" +
 "(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)" +
 "(HOST=localhost)(PORT=1521))" + "(CONNECT_DATA=(SERVICE_NAME=orcl)));" +
 "User Id = c##usuariotest; Password = lcdusuariotest;";
 // Crea conexion
 OracleConnection conexionOracle = new OracleConnection
                                    (cadenaConexionOracle);
 // Conecta
 conexionOracle.Open();
 // Confirma si la conexion fue abierta
 if (conexionOracle.State == ConnectionState.Open)
 { // Si la conexion está abierta, la cierra
   conexionOracle.Close();
 }
```

El Descriptor de conexión de TNS que proporciona la ubicación de la base de datos y el nombre del servicio de base de datos Oracle Real

Utilice el formato:

```
DESCRIPTION=(ADDRESS=(PROTOCOL=protocolo)(HOST=host)  (PORT=puerto))
(CONNECT_DATA=(SERVICE_NAME=nombre de servicio))
```

Por ejemplo:

```
DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=myhost.company.com)(PORT=1521))(CONNECT_DATA=
(SERVICE NAME=sales.company.com))
```

Lenguaje SQL

En el libro #4, *Fundamentos de bases de datos,* el autor indica lo siguiente:

Los DBMS utilizan, para el mantenimiento, control y acceso a la información un lenguaje estructurado de consultas llamado SQL (Structure Query Language). Este lenguaje tiene diferentes tipos de instrucciones divididas en 4 categorías o tipos:

1. **Lenguaje de consulta de datos (DQL):** por medio de la instrucción SELECT

La instrucción SELECT recupera datos de la base de datos. A continuación, se presentan las cláusulas de la instrucción, que se ejemplifican en las secciones siguientes:

SELECT Muestra una lista de las columnas que van a ser devueltas en los resultados.

FROM Presenta una lista de las tablas o vistas de las que se van a seleccionar los datos.

WHERE Ofrece las condiciones para la selección de filas en los resultados.

ORDER BY Especifica el orden en que se van a devolver las filas. se puede agregar la palabra clave *ASC* después del nombre de la columna para desplegar una secuencia ascendente y *DESC* para una descendente.

GROUP BY Agrupa las filas para diversas funciones de obtención de totales.

Por ejemplo:

```
SELECT LAST_NAME, FIRST_NAME, HIRE_DATE, SALARY
FROM EMPLOYEES
ORDER BY LAST_NAME, FIRST_NAME;
```

La instrucción SELECT emplea la cláusula WHERE para seleccionar las filas que se despliegan. Sin una cláusula WHERE, se muestran todas las filas encontradas en las tablas y vistas de origen.

Cuando se incluye una cláusula WHERE, se aplican las reglas del álgebra booleana para evaluar la cláusula WHERE de cada fila de datos. Sólo se muestran en los resultados de la consulta las filas para las que la cláusula WHERE se evalúa con un verdadero lógico.

2. **Lenguaje de manipulación de datos (DML)**

Los tipos de instrucciones del lenguaje de manipulación de datos son INSERT, UPDATE y DELETE. Estos comandos le permiten agregar, cambiar y eliminar filas de datos en las tablas.

Soporte a transacciones (COMMIT y ROLLBACK)

En el DBMS, una transacción es una serie de una o más instrucciones SQL tratadas como una sola unidad. Una transacción debe funcionar o fracasar por completo, lo que significa que los cambios que hace una transacción en cualquier base de datos deben volverse permanentes cuando la transacción concluye correctamente. Por otra parte, estos cambios deben eliminarse por completo de la base de datos si la transacción falla antes de su conclusión.

Es importante que los demás usuarios de la base de datos no observen fragmentos de un pedido hasta que se haya introducido y confirmado por completo.

SQL permite formalizar las transacciones con las instrucciones COMMIT o ROLLBACK.

3. Instrucciones del lenguaje de definición de datos (DDL)

Las instrucciones del lenguaje de definición de datos definen los objetos de la base de datos, pero no insertan ni actualizan los datos guardados dentro de esos objetos. (Las instrucciones de DML sirven para ese propósito.) En SQL se emplean

tres comandos básicos dentro de DDL:

CREATE Crea un nuevo objeto de la base de datos del tipo mencionado en la instrucción. Con esta instrucción se pueden crear los objetos que interactúan en la base de datos: tablas, vistas, usuarios, procedimientos almacenados, funciones, etc.

Por ejemplo:

```
CREATE TABLE EMPLOYEE_INPUT (
      EMPLOYEE_ID NUMBER(6) NOT NULL,
      FIRST_NAME VARCHAR2(20) NULL,
      LAST_NAME VARCHAR2(25) NOT NULL,
      EMAIL VARCHAR2(25) NOT NULL,
      PHONE_NUMBER VARCHAR2(20) NULL,
      HIRE_DATE DATE NOT NULL,
      JOB_ID VARCHAR2(10) NOT NULL,
      SALARY NUMBER(8,2) NULL,
      COMMISSION_PCT NUMBER(2,2) NULL,
      MANAGER_ID NUMBER(6) NULL,
      DEPARTMENT_ID NUMBER(4) NULL);
```

DROP Descarta (destruye) un objeto existente en la base de datos del tipo indicado en la instrucción. Con esta instrucción se pueden destruir los objetos que no se requieren en la base de datos: tablas, vistas, usuarios, procedimientos almacenados, funciones, etc.

Por ejemplo:

```
DROP TABLE EMPLOYEE_INPUT CASCADE CONSTRAINTS;
```

ALTER Modifica la definición de un objeto existente en la base de datos del tipo señalado en la instrucción. Con esta instrucción se pueden modificar los objetos que interactúan

en la base de datos: tablas, vistas, usuarios, procedimientos almacenados, funciones, etc.

Por ejemplo:

```
ALTER TABLE EMPLOYEE_INPUT
      ADD CONSTRAINT EMP_INPUT_DEPT_FK
      FOREIGN KEY (DEPARTMENT_ID)
      REFERENCES DEPARTMENTS (DEPARTMENT_ID);
```

4. Instrucciones del lenguaje de control de datos (DCL)

Un privilegio de una base de datos es la autorización para hacer algo en ella. Al usuario de la base de datos que concede el privilegio se le llama otorgante y al usuario que recibe el privilegio se le denomina concesionario. Los privilegios caen en dos categorías amplias:

Privilegios del sistema Permiten que el concesionario aplique una función general a la base de datos, como crear nuevas cuentas de usuario o conectarse a la base de datos.

Privilegios de objeto Permiten al concesionario efectuar acciones específicas sobre objetos determinados.

La instrucción GRANT

Se conceden privilegios a los usuarios de SQL mediante la instrucción GRANT. La cuenta del usuario que concede el privilegio debe poseer el privilegio de sistema que le permite otorgar permisos de los objetos.

Por ejemplo:

Suponga que se tiene un usuario HR_USER

```
GRANT SELECT, INSERT, UPDATE ON EMPLOYEES TO HR_USER
GRANT CREATE VIEW TO HR_USER;
```

La instrucción REVOKE

Los privilegios concedidos pueden retirarse mediante la instrucción REVOKE.

```
REVOKE CREATE VIEW FROM HR_USER;
REVOKE SELECT, INSERT, UPDATE ON EMPLOYEES FROM HR_USER;
```

Modelo de almacenamiento cifrado

En el libro #4, *Fundamentos de bases de datos,* el autor indica lo siguiente:

El cifrado es la traducción de datos a un código secreto que no puede ser leído sin el uso de una contraseña o clave secretas. Los datos no cifrados se consideran texto simple, mientras que los cifrados son texto cifrado.

Algunos esquemas para cifrado emplean una clave simétrica, lo que significa que se utiliza una sola clave para cifrar el texto simple y para descifrar el texto cifrado. A esta forma se le considera menos segura en comparación con el uso de claves asimétricas, en donde se aplican

dos claves: una clave pública y una privada. Lo que cifra la clave pública, lo descifra la clave privada, y viceversa. Los nombres provienen del uso esperado de las claves: la clave pública se entrega a todas las personas que hacen negocios con la empresa, y la clave privada es confidencial e interna para la empresa.

Éstos son algunos lineamientos que debe seguir en relación con el cifrado:

Las claves de cifrado deben tener un mínimo de 128 bytes de longitud. Cuanto más extensa sea la clave, más segura se considerará (dentro de lo razonable). Sin embargo, las claves más extensas retardan el proceso de descifrado, de modo que debe buscarse una solución intermedia.

La pérdida de una clave de cifrado debe considerarse con la misma seriedad que la pérdida de los datos que suele cifrar.

Los datos confidenciales deben cifrarse cuando se guardan de manera permanente. El personal de la empresa que posee los datos, no el DBA, debe determinar cuáles datos se consideran confidenciales.

Todos los datos que no son de conocimiento del público deben cifrarse cuando se transporten de manera electrónica a través de conexiones de red que no estén cifradas. Por ejemplo, si una empresa envía un archivo de pedido de compra a un socio comercial mediante FTP, el archivo debe cifrarse. Nada garantiza que personas malintencionadas no vigilen las redes públicas.

El correo electrónico no se considera seguro, de modo que cualquier información delicada que se va enviar por correo electrónico debe estar en un archivo cifrado de datos adjuntos y no en el cuerpo principal del mensaje. Como opción, algunos sistemas de correo electrónico permiten mensajes cifrados y firmados.

El cifrado

- Debe utilizarse para todos los datos confidenciales.
- Siempre deben usar claves de cuando menos 128 bits de longitud.
- Debe aplicarse a los datos confidenciales enviados en una red.
- Puede emplear claves simétricas o asimétricas.
- Nunca debe usarse para credenciales de inicio de sesión.

Inyección de SQL

De acuerdo a la página Internet WIKIPEDIA.ORG/WIKI/INYECCIÓN_SQL se obtiene la siguiente información:

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.

El origen de la vulnerabilidad radica en la incorrecta comprobación o filtrado de las variables utilizadas en un programa que contiene, o bien genera, código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que puede ocurrir en cualquier lenguaje de programación o script que esté embebido dentro de otro.

Se conoce como Inyección SQL, indistintamente, al tipo de vulnerabilidad, al método de infiltración, al hecho de incrustar código SQL intruso y a la porción de código incrustado.

Se dice que existe o se produjo una inyección SQL cuando, de alguna manera, se inserta o "inyecta" código SQL invasor dentro del código SQL programado, a fin de alterar el funcionamiento normal del programa y lograr así que se ejecute la porción de código "invasor" incrustado, en la base de datos.

Este tipo de intrusión normalmente es de carácter malicioso, dañino o espía, por tanto, es un problema de seguridad informática, y debe ser tomado en cuenta por el programador de la aplicación para poder prevenirlo. Un programa elaborado con descuido, displicencia o con ignorancia del problema, podrá resultar ser vulnerable, y la seguridad del sistema (base de datos) podrá quedar eventualmente comprometida.

Seguridad de Bases de datos

En el libro #4, *Fundamentos de bases de datos,* el autor indica lo siguiente:

Conceptos de seguridad en bases de datos

Son las medidas de seguridad que se implementan para la protección de la privacidad y para evitar el acceso no autorizado a las bases de datos, tanto para consultas como para modificaciones de los datos o corrupción de los datos.

Seguridad Física

Son las medidas para limitar el acceso a la infraestructura física, servidores físicos y otros componentes de hardware, así como las copias de respaldos.

Seguridad del sistema operativo

Son las medidas para proteger el sistema operativo, por medio de actualizaciones con mejoras y *Service* Packs

Seguridad en la Base de Datos

Son las medidas para restringir el acceso a los datos, mediante creación de credenciales, roles, niveles de acceso sobre los datos. Algunos de las amenazas que se desean evitar son: privilegios excesivos e inutilizados, abuso de privilegios, inyección por SQL, software sospechoso, auditorías débiles, exposición de los medios de almacenamiento para backup, bases de datos mal configuradas, datos sensibles mal gestionados

Niveles de seguridad

En un DBMS 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. conocimiento, aptitudes y hábitos del usuario • información predefinida

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

La seguridad se logra si se cuenta con un mecanismo que limite a los usuarios. 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 o tabla.
- Vista: Puede permitírsele o impedírsele que el usuario tenga acceso a la información que aparece en una vista.

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.

Limitantes de integridad

Las limitantes de integridad aseguran que las modificaciones realizadas por los usuarios autorizados no provoquen el daño o perdida de datos. Protegen los datos de daños accidentales o malintencionados.

Integridad de una base de datos

La integridad de los datos es el proceso de asegurar que los datos estén protegidos y se mantengan intactos a través de las restricciones definidas que se aplican a los datos. A éstas se les llama restricciones de la base de datos porque evitan cambios en los datos que violarían una o más reglas de negocios. El beneficio principal de imponer reglas de negocios que utilicen restricciones de integridad de datos en la base de datos es que las restricciones no pueden evitarse. A diferencia de las reglas de negocios impuestas por los programas de aplicaciones, las restricciones de una base de datos son impuestas sin importar cómo se conecta alguien. El único modo de evitar las restricciones de una base de datos es que el DBA las elimine o las deshabilite. Por otra parte, los desarrolladores tienen preferencia por ser ellos mismos quienes controlan la imposición de una regla, en lugar de relegarlas a un DBA, y algunas reglas se prueban mejor antes de enviar los datos a la base de datos para procesamiento. En raros casos, por lo general relacionados con las reglas de negocios más importantes, es posible que quiera imponerlas en ambos lugares: en la base de datos porque la regla no puede ser evitada, y en la aplicación, para que el usuario se entere de inmediato cuando viola la regla.

Las reglas de negocios se implementan en la base de datos del modo siguiente:

Restricciones NOT NULL: Las reglas de negocios que declaran cuáles atributos se requieren se traducen como cláusulas NOT NULL en las columnas correspondientes en el diseño de tablas. Las claves principales siempre deben especificarse como NOT NULL. Y las claves externas que participan en una relación obligatoria también deben especificarse como NOT NULL

Restricciones de clave principal: Las restricciones de clave principal requieren que las columnas que forman ésta contengan valores únicos para cada fila de la tabla. Además, las columnas de clave principal deben definirse con restricciones NOT NULL. Una tabla sólo puede tener una restricción de clave principal

Restricciones referenciales (de clave externa): El propósito de la restricción referencial es comprobar que los valores de clave externa en las filas de la tabla secundaria siempre tengan valores de clave principal que coincidan en la tabla primaria.

Restricciones de unicidad: Al igual que las restricciones de clave principal, las de unicidad aseguran que dos filas de una tabla no tengan valores duplicados para las columnas mencionadas en la restricción. Sin embargo, las restricciones de unicidad tienen dos diferencias importantes: Aunque una tabla sólo puede tener una restricción de clave principal, puede tener todas las restricciones de unicidad necesarias. Las columnas que participan en una restricción de unicidad no necesitan tener restricciones NOT NULL.

Restricciones de comprobación: Las restricciones de comprobación (CHECK) se emplean para imponer las reglas de negocios que limitan una columna a una lista, un rango de valores o alguna condición que puede verificarse mediante una comparación única con una constante, un cálculo o un valor de otra columna en la misma fila. Las restricciones de comprobación no pueden usarse para comparar valores de columnas entre filas diferentes, ya sea en la misma tabla o no

Tipos de datos, precisión y escala: El tipo de datos asignado a las columnas de una tabla limita automáticamente los datos a valores que coincidan con el tipo de datos. Para los tipos de datos que permiten la especificación de la precisión (el tamaño máximo) y la escala (las posiciones a la derecha del punto decimal), estas especificaciones también limitan los datos. No es posible incorporar una cadena de caracteres o un número más grande que el tamaño máximo de la columna dentro de la base de datos.

Desencadenadores (triggers) es una unidad de código de programa que se ejecuta en forma automática con base en cierto evento que ocurre en la base de datos, como la inserción, actualización o eliminación de datos en una tabla específica.

Estrategias adicionales para la seguridad

Uso de contraseñas seguras

Para contribuir a la seguridad de los datos, es muy importante utilizar contraseñas seguras difíciles de vulnerar.

Algunas claves para crear contraseñas seguras:

Crear contraseñas largas. Algunos sistemas solicitan un mínimo de 7 u 8 caracteres. Para mayor seguridad, pueden utilizarse 15 caracteres o más, especialmente para proteger aquellos servicios que se consideran críticos

Utilizar mayúsculas, minúsculas, signos de puntuación y caracteres no alfabéticos

Utilizar contraseñas únicas para cada servicio. Por ejemplo, no utilizar una misma clave para proteger datos almacenados en un servidor institucional y en un servicio de almacenamiento en la nube.

Evitar errores comunes, tales como:

- Utilizar palabras de diccionario, aún si están en otros idiomas
- Utilizar datos personales, tales como números de identificación, teléfonos o direcciones
- Utilizar referencias a la cultura popular, tales como nombres de libros, personajes, canciones, bandas musicales, etc.

Y por supuesto, deben tomarse las precauciones que se recomiendan para la protección de cualquier contraseña, como evitar ingresar a cuentas desde conexiones públicas o en espacios inseguros.

Base de datos Oracle

De acuerdo a la página internet https://es.wikipedia.org/wiki/Oracle_Database indica lo siguiente:

Oracle Database es un sistema de gestión de base de datos de tipo objeto-relacional (ORDBMS, por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.

Su dominio en el mercado de servidores empresariales había sido casi total hasta que recientemente tiene la competencia del Microsoft SQL Server y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird.

Las últimas versiones de Oracle han sido certificadas para poder trabajar bajo GNU/Linux.

Arquitectura

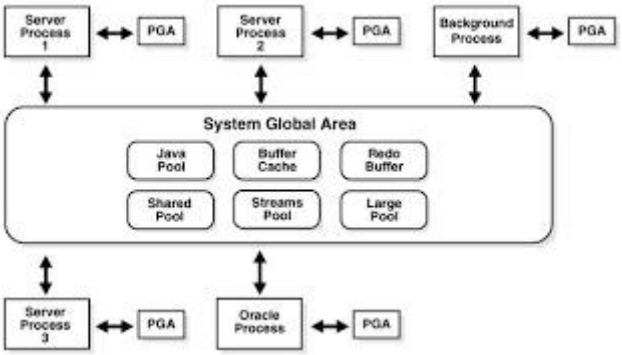
En la página de internet http://adminbd-dcueva.blogspot.com/2009/04/arquitectura-de-base-de-datos-oracle.html se puede encontrar la siguiente información:

La arquitectura Oracle tiene 3 componentes básicos:

- Las estructuras de memoria para almacenar los datos y el código ejecutable
- Los procesos que corren el sistema de bases de datos y las tareas de cada usuario conectado a la base de datos
- Los archivos que sirven para el almacenamiento físico, en disco, de la información de la base de datos.

Estructuras de Memoria

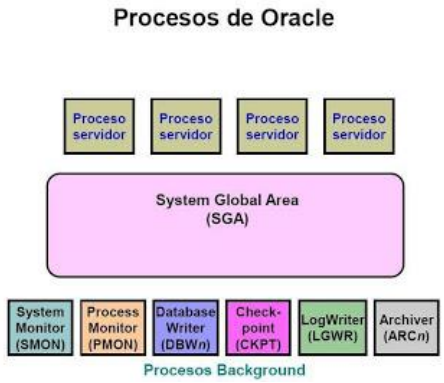
Hay 2 clases de memoria, una compartida por todos los usuarios y otra dedicada al trabajo de cada uno.



Para cada sesión de usuario se crea también un área específica en memoria llamada PGA (program/process global area), la cual no se comparte con las otras sesiones de usuario.

Procesos

Los procesos son programas que se ejecutan para permitir el acceso a los datos. Los procesos se cargan en memoria y son transparentes para los usuarios. Los procesos se clasifican en 3: procesos de base (background), de usuario y de proveedores.



Los procesos background son los que se encargan de traer datos desde y hacia la SGA; mejorando el desempeño al consolidar las tareas que son impartidas por todos los usuarios. Son: SMON, PMON, DBWR, CKPT, LGWR, ARCH, RECO y LCKN.

Los procesos de usuario suceden cuando un usuario se conecta a la base. Se crea un proceso que se encarga de ejecutar el código de aplicación del usuario y manejar el perfil del usuario con sus variables de ambiente. No se pueden comunicar directamente con la BD, sólo lo hacen a través de procesos de servidor.

Los procesos de servidores ejecutan las órdenes SQL de los usuarios y llevan los datos al Database buffer caché, para que los procesos de usuario puedan tener acceso a los datos.

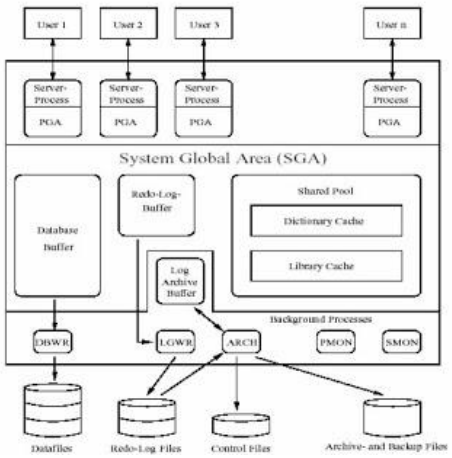
Archivos

La base de datos abarca las estructuras físicas que se encuentran en disco. Estos archivos se dividen en dos: Requeridos y Externos. Entre los archivos requeridos están:

- **Control File**: Almacena el status de las estructuras físicas de la base de datos.
- **Online Redo Log Files**: Almacenan un registro de los cambios realizados a la base de datos mientras estos de van dando
- **Datafiles**: Son el repositorio de la información. Sirven para el almacenamiento físico de las tablas, índices o agrupamientos (clusters) y procedimientos. Las unidades lógicas más grandes manejadas por Oracle son los tablespaces, que le permiten controlar espacios en el disco. Los tablespaces consisten de 1 o más datafiles. El espacio de tablas creado automáticamente es System. Un objeto de BD puede ser una tabla, un índice, un archivo temporal, los cuales se almacenan físicamente en segmentos, que son una colección de segmentos, que son una colección de data blocks, que son mapeados a los bloques del SO.

En cambio, los archivos externos son:

- **Parameter File**: Define la instancia y los parámetros de inicialización. Hay de dos tipos Dinámico (binario, que no se puede ejecutar y se actualiza constantemente) y estático (que se lo puede editar mediante un editor ASCII y que solamente es leído una sola vez cuando la instancia se inicia.)
- **Password File**: Archivo de sistema que almacena los nombres de usuario y contraseña (encriptadas) para poder autenticar a un usuario sin la necesidad del diccionario de datos.
- **Archive Log Files**: Copias de los Online Redo Log Files llenos.
- **Backup Files**: Copias de seguridad.



Conectividad

Para conectar a una base de datos hay varios métodos, la más utilizadas en configurar los TNSNames cuando se instala el cliente Oracle. El archivo de configuración se llama TNSNAMES.ORA y está localizado en la carpeta NETWORK/ADMIN acompañado de otros posibles archivos de configuración como el SQLNET.ORA que es el que permite la conexión con el Oracle Net Services. El LISTENER.ORA que contiene la configuración del listener de la base de datos. NAMES.ORA que contiene el la ubicación y la información de dominio de los parámetros de configuración opcionales. LDAP.ORA Contiene los parámetros necesarios para acceder al servidor de directorios.

El archivo TNSNAMES.ORA contiene los nombres de los servicios de red disponibles para las bases de datos, contiene la siguiente información:

```
nombreTNS =
      (DESCRIPTION =
        (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = < Dirección ip del servidor
           con el cual queremos conect >)(PORT = < Puerto donde escucha
            la base de datos >)))
        (CONNECT_DATA =
        (SERVICE_NAME = < Nombre del servicio de base de datos al que
                          queremos conectar >)
        )
     )
```

Por ejemplo

```
# tnsnames.ora Network Configuration File:
TNSNombre =
(DESCRIPTION =
  (ADDRESS_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1524)))
  (CONNECT_DATA =(SERVER = DEDICATED)
   (SERVICE_NAME = bdpruebas)
   (INSTANCE_NAME = bdpruebas)
)
```

Modos de autenticación

El método por excelencia es el de usuario/contraseña.

El administrador debe crear el usuario y darle los respectivos privilegios de acceso a las diferentes bases de datos

`CREATE USER ElUsuario IDENTIFIED BY ElPassword;` La forma más común de crear usuarios, pero hay otros parámetros que se pueden usar.

Para darle los privilegios de conexión y de acceso se hacen con la siguiente instrucción:

`GRANT CONNECT TO ElUsuario;` Privilegio de conexión
`GRANT CONNECT, RESOURCE, DBA TO ElUsuario;` Privilegio de DBA

`GRANT CREATE SESSION GRANT ANY PRIVILEGE TO ElUsuario;` Privilegio de crear sesión y cualquier otro privilegio

`GRANT UNLIMITED TABLESPACE TO ElUsuario;` Privilegio de espacio

`GRANT SELECT, INSERT, UPDATE, DELETE ON LaBD.LaTabla TO ElUsuario;` Privilegio de accesos a un objeto

Seguridad del sistema operativo

En la página internet https://docs.oracle.com/cd/E24842_01/html/E23286/secov-1.html se puede leer lo siguiente:

Para mantener la seguridad del Sistema operativo el software proporciona las siguientes funciones:

Seguridad del sistema: la capacidad para evitar la intrusión, proteger los recursos y dispositivos del equipo contra el uso inapropiado, y proteger los archivos contra la modificación maliciosa o involuntaria realizada por usuarios o intrusos.

Servicios criptográficos: la capacidad para codificar datos de manera que sólo el remitente y el receptor designado puedan leer el contenido, y para gestionar proveedores criptográficos y objetos de clave pública.

Servicios de autenticación: la capacidad para identificar a un usuario de manera segura, lo que requiere el nombre del usuario y alguna forma de prueba, normalmente, una contraseña.

Autenticación con cifrado: la capacidad para garantizar que las partes autenticadas se puedan comunicar sin intercepción, modificación ni falsificación.

Auditoría: la capacidad para identificar el origen de cambios de seguridad en el sistema, incluidos el acceso a archivos, las llamadas del sistema relacionadas con la seguridad y los errores de autenticación.

Política de seguridad: el diseño y la implementación de directrices de seguridad para un sistema o red de sistemas.

Seguridad post-instalación

En la página https://blogs.oracle.com/oracle-latinoamerica/cinco-buenas-practicas-para-mejorar-la-seguridad-de-su-base-de-datos-v2 se puede leer la siguiente información:

Cinco buenas prácticas para mejorar la seguridad de tu base de datos

La seguridad de la base de datos puede parecer una tarea compleja y lograr la arquitectura de máxima seguridad deseada para proteger los datos confidenciales requiere tiempo, personal y, a menudo, presupuesto. Dicho esto, existen ciertas prácticas fundamentales que toda empresa, de las pequeñas a las de gran porte e intersectorial, debería aplicar. De hecho, estas prácticas básicas de seguridad deberían existir antes de gastar recursos en medidas de seguridad adicionales. Asegúrate de cerrar con llave la puerta principal antes de considerar comprar un costoso sistema de seguridad con cámaras y alarmas.

Considerar cuidadosamente cómo se proporcionan los privilegios administrativos a los usuarios de la base de datos puede ahorrarle grandes dolores de cabeza a una empresa en el futuro, incluida la mitigación del riesgo de una costosa filtración de datos. Si bien es probable que confíes en tus DBA, los ciberdelincuentes suelen utilizar ataques spear-phishing y otros medios para dirigirse a los usuarios privilegiados de una empresa, aprovechando sus cuentas para uso malicioso, incluida la extracción de datos confidenciales. Por ejemplo, si un hacker logra comprometer una cuenta DBA con el privilegio “SELECCIONAR CUALQUIER TABLA”, podría acceder a casi todos los datos de la base de datos, incluidos los números de seguridad social, números de tarjetas de pago y propiedad intelectual.

1. Separación de tareas

El concepto de separación de funciones (SOD) dicta que las tareas de administración deben ser divididas entre varios usuarios y no entre un sólo individuo todopoderoso.

2. Usuarios nombrados

Los administradores nunca deben compartir cuentas por conveniencia (o cualquier razón, para el caso). Las cuentas compartidas eliminan la responsabilidad, aumentan el riesgo y hacen que la auditoría de la actividad del usuario sea esencialmente imposible. Cada usuario de una empresa debe tener una cuenta individual que especifique explícitamente su nombre.

3. Gestión de cuentas SYSDBA

La cuenta del propietario de la base de datos SYS (SYSDBA) es un privilegio administrativo que proporciona acceso sin restricciones a la base de datos, como una cuenta ROOT para la gestión del sistema operativo. Esto es simplemente demasiado poder para que cualquier usuario tenga a perpetuidad. De hecho, muchos

administradores de bases de datos consideran que la concesión de privilegios SYSDBA los coloca en una posición indeseable de responsabilidad potencial, en caso de que algo salga mal. Como tal, el uso de esta cuenta y privilegio debe ser gestionado y supervisado de cerca, y limitado a sólo cuando sea absolutamente necesario, como durante las actualizaciones y parches de la base de datos.

4. Privilegio mínimo

La separación de tareas (SOD) separa a las personas, los procesos y las cuentas, pero no se puede hacer cumplir si todos los usuarios y cuentas tienen todos los privilegios. Una vez que haya implementado SOD, la aplicación del principio de privilegio mínimo limita cada usuario y cuenta a solo los privilegios que necesita para las operaciones diarias.

5. Protección de auditoría

Los registros de auditoría son necesarios para la presentación de informes de cumplimiento y para los análisis forenses en caso de incumplimiento u otro evento adverso. Capturar un registro irrefutable de las acciones tomadas por las cuentas nombradas, incluyendo “CREAR USUARIO”, “CREAR CUALQUIER TABLA”, “SISTEMA ALTERNO” y “SESIÓN ALTERNA”, junto con información contextual como dirección IP y hora del evento. Los registros de auditoría ayudarán a una empresa a identificar a los usuarios de riesgos, agilizar las auditorías y simplificar el cumplimiento.

Aplicación de parches

En la página https://docs.oracle.com/cd/E24842_01/html/E23289/gksod.html se encuentra la siguiente información:

Planificación y preparación

La aplicación de parches a un sistema de base de datos requiere un reinicio, que puede tardar varios minutos. Para minimizar el impacto en los usuarios, ejecute el parche cuando el sistema tenga la menor cantidad de usuarios. Para evitar la interrupción del sistema, considere la posibilidad de implementar una estrategia de alta disponibilidad.

Oracle recomienda realizar una copia de seguridad de la base de datos y probar el parche en un sistema de prueba antes de aplicarlo.

Aplique siempre un parche a un sistema de base de datos antes de aplicar el parche a las bases de datos de ese sistema. La consola muestra la última versión del parche del sistema de base de datos y la versión anterior del parche.

Para la aplicación de parches según los requisitos y tiempo disponible, se debe hacer en una ventana de mantenimiento mediante la aplicación de alguna de las estrategias de aplicación de parches definidas, entre ellas:

Actualización automática: método para actualizar un sistema mientras éste sigue en funcionamiento

Aplicación de un clúster de parches recomendados: El clúster de parches recomendados contiene todos los parches del sistema operativo disponibles. Estos parches incluyen:

- Correcciones relacionadas con la seguridad
- Correcciones relacionadas con el deterioro de datos
- Correcciones relacionadas con problemas de disponibilidad del sistema
- Parches recomendados
- Últimos parches de utilidad de parches
- Cualquier otro parche requerido

Aplicación de una línea base de parches de estándares de instalación empresarial: El conjunto de parches de estándares de instalación empresarial (EIS) se basa en el clúster de parches recomendados para el sistema operativo. La línea base de parches de EIS tiene parches adicionales que incorporaron los ingenieros de campo de Oracle para productos adicionales y para abordar problemas de incumplimiento de los criterios de inclusión en el clúster de parches recomendados.

Acceso y administración de usuarios

En la página internet https://www.monografias.com/trabajos908/oracle/oracle2.shtml se obtiene la siguiente información.

Mecanismos de seguridad

El sistema de base de datos ORACLE provee control de acceso discrecional, el cual es un medio de restricción de acceso basado en privilegios. Para que un usuario pueda acceder a un objeto, se le deben otorgar los privilegios apropiados. Los usuarios con los

privilegios adecuados pueden otorgar privilegios a otros usuarios a su criterio. Por esta razón, este tipo de seguridad es llamada "discrecional".

Oracle administra la seguridad de la base de datos utilizando diferentes servicios o facilidades:

- Usuarios y esquemas
- Privilegios
- Roles
- Configuración del almacenamiento y quotas
- Límites a los recursos
- Monitoreo

Usuarios y esquemas

Cada base de datos tiene una lista de nombres de usuarios. Para acceder a la base de datos, un usuario debe usar una aplicación e intentar entablar una conexión con un nombre de usuario válido de la base de datos. Cada usuario tiene una clave de acceso asociada para prevenir el uso no autorizado.

Privilegios

Un privilegio es un permiso para ejecutar un tipo particular se sentencias SQL.

Roles

ORACLE provee los roles para una administración más fácil y controlada de los privilegios. Los roles son un grupo con nombre de privilegios, que son asignados a usuarios o a otros roles. Las siguientes propiedades de los roles permiten administrar los privilegios de una manera más fácil:

Reducida asignación de privilegios: En lugar de otorgar explícitamente el mismo conjunto de privilegios a muchos usuarios el administrador de la base de datos puede asignar los privilegios a un rol y éste a un grupo de usuarios.

Monitoreo

ORACLE permite realizar un monitoreo selectivo de las acciones de los usuarios para ayudar en la investigación de usos maliciosos de la base de datos. El monitoreo puede realizarse a tres niveles distintos:

Monitoreo de sentencias: Es el monitoreo de sentencias SQL específicas sin atender concretamente a los objetos. Este tipo de monitoreo puede hacerse para todos los usuarios del sistema o se puede enfocar sólo a algunos usuarios seleccionados.

Monitoreo de privilegios: Es el monitoreo de los privilegios del sistema sin atender concretamente a los objetos. Este tipo de monitoreo puede hacerse para todos los usuarios del sistema o se puede enfocar sólo a algunos usuarios seleccionados.

Monitoreo de objetos: Es el monitoreo de los accesos a esquemas específicos sin considerar el usuario. Monitorea las sentencias permitidas por los privilegios.

Para todos los tipos de monitoreo, ORACLE permite el monitoreo selectivo de sentencias ejecutadas con éxito, sentencias ejecutadas sin éxito o ambas.

Administración de usuarios de base de datos

Cada base de datos tiene uno o más administradores de seguridad quienes son responsables del mantenimiento de todos los aspectos de la política de seguridad. Si el sistema de bases de datos es pequeño, el administrador de bases de datos podría tener las responsabilidades del administrador de seguridad. Sin embargo, si el sistema de bases de datos es grande, una persona o grupos de personas especiales podrían tener responsabilidades limitadas de aquellas que corresponden a un administrador de seguridad.

Autentificación de usuarios

Los usuarios de bases de datos pueden ser autentificados (verificados como una persona correcta) por Oracle usando el sistema operativo de host, los servicios de red, o la base de datos.

Auditoría de roles, usuarios y passwords

Los administradores de Seguridad deben establecer una política para el procedimiento de auditoria de cada base de datos. Cuando es necesaria una auditoria el administrador de seguridad debe decidir a qué nivel de detalle se realizará la auditoría de la base de datos.

Una vez detectado alguna actividad de origen sospechosa a través del sistema general de auditoria, se realizará una auditoría más específica.

Trusted ORACLE

Trusted Oracle es un sistema de administración de la seguridad de bases de datos. Fue diseñado para proveer el más alto nivel de capacidades para la administración de la seguridad requerido por organizaciones que procesan información sensitiva. Trusted Oracle es compatible con todos los productos Oracle.

Además, Trusted Oracle implementa Mandatory Access Control (MAC) (control de acceso obligatorio). Mandatory Access Control es un medio para restringir el acceso a la información basado en etiquetas o rótulos. La etiqueta de un usuario indica a qué información le está permitido acceder a un usuario y con qué tipo de acceso (lectura o escritura). La etiqueta de un objeto indica la sensibilidad de la información que el objeto contiene.

Propietarios de esquemas y aplicaciones

El esquema de Oracle es básicamente el conjunto de todas las tablas y otros objetos que conforman la base de datos para un sistema o aplicacion propiedad de una cuenta de usuario, por lo que equivale aproximadamente a una cuenta de usuario. Es decir, un esquema de base de datos se crea exactamente igual como se crea un usuario, solo que en el esquema se crean los objetos que componen la base de datos y el usuario se le define los privilegios de acceso.

Administración de contraseñas

La seguridad de los sistemas de base de datos depende de no divulgar las passwords en ningún momento. No obstante, son vulnerables al robo, falsificación y abuso. Para permitir un mayor control en la seguridad sobre las bases de datos, la política administrativa de passwords de Oracle es controlada por DBAs.

Cuando un usuario excede un determinado número de intentos de accesos fallidos, el server automáticamente cierra la cuenta del usuario. DBA especifica el número permitido de accesos fallidos usando la sentencia CREATE PROFILE. También especifica el período de tiempo que la cuenta quedará cerrada.

Si el DBA no especifica el tiempo que tardará en habilitarse la cuenta nuevamente, el sistema toma un tiempo por default; y si el tiempo estipulado es ilimitado, el encargado del sistema de seguridad deberá directamente habilitar la cuenta. De esta manera, el período de tiempo que una cuenta permanece cerrada depende de cómo la DBA configura los recursos asignados al usuario.

El encargado de seguridad también puede directamente cerrar cuentas de usuarios. Cuando esto ocurre sólo el encargado de seguridad puede habilitarla nuevamente.

Expiración de las passwords

El DBA usa la sentencia CREATE PROFILE para especificar un máximo de tiempo de vida de las passwords. Pasado este tiempo la password expira y el usuario deberá cambiarla. También puede especificar un período de gracia usando la misma sentencia.

Controles de acceso

En la página internet https://magicplsql.blogspot.com/2016/07/control-de-acceso-usuarios-de-la-base.html indica lo siguiente:

Control de Acceso a Usuarios.

En un entorno de varios usuarios, necesita mantener la seguridad del acceso y el uso de la base de datos. Con la seguridad de base de datos de Oracle Server, puede:

- •Controlar el acceso a la base de datos.
- •Otorgar acceso a objetos específicos de la base de datos.
- •Confirmar los privilegios otorgados y recibidos con el diccionario de datos Oracle.
- •Crear sinónimos para objetos de base de datos.

La seguridad de base de datos se puede clasificar en dos categorías: **Seguridad del Sistema** y **Seguridad de los Datos**. La seguridad del sistema cubre el acceso y el uso de la base de datos en el nivel del sistema como, por ejemplo, nombre de usuario y contraseña, el espacio en disco asignado a los usuarios y las operaciones del sistema que pueden realizar los usuarios. La seguridad de datos cubre el acceso y el uso de los objetos de base de datos y las acciones que esos usuarios pueden llevar a cabo en los objetos.

Privilegios.

Privilegios del sistema: Obtención de acceso a la base de datos.

Privilegios de objeto: Manipulación del contenido de los objetos de base de datos.

Los privilegios son el derecho a ejecutar sentencias SQL en particular. El DBA es un usuario de alto nivel con la capacidad de crear usuarios y de otorgarles acceso a la base de datos y a sus objetos. Los usuarios necesitan privilegios del sistema para obtener acceso a la base de datos y privilegios de objeto para manipular el contenido de los

objetos de la base de datos. A los usuarios también se les puede otorgar el privilegio de otorgar privilegios adicionales a otros usuarios o a roles, que son grupos especificados de privilegios relacionados.

Privilegios de DBA.

Privilegio del Sistema.	***Operaciones Autorizadas***
CREATE USER	La persona a la que se otorga el privilegio puede crear otros usuarios de Oracle.
DROP USER	La persona a la que se otorga el privilegio puede borrar otro usuario.
DROP ANY TABLE	La persona a la que se otorga el privilegio puede borrar una tabla de cualquier esquema.
BACKUP ANY TABLE	La persona a la que se otorga el privilegio puede realizar copias de seguridad de cualquier esquema con la utilidad de exportación.
SELECT ANY TABLE	La persona a la que se otorga el privilegio puede consultar tablas, vistas o instantáneas en cualquier esquema.
CREATE ANY TABLE	La persona a la que se otorga el privilegio puede crear tablas en cualquier esquema.

Privilegios de Usuario Típicos.

Una vez creado el usuario, el DBA le puede otorgar privilegios del sistema específicos. Un desarrollador de aplicaciones, por ejemplo, puede tener los siguientes privilegios del sistema:

Privilegio del Sistema	**Operaciones Autorizadas**
CREATE SESSION	Conectarse a la base de datos
CREATE TABLE	Crear tablas en el esquema del usuario
CREATE SEQUENCE	Crear una secuencia en el esquema del usuario
CREATE VIEW	Crear una vista en el esquema del usuario
CREATE PROCEDURE	Crear un procedimiento, una función o un paquete en el esquema del usuario

Algunas de las sentencias utilizadas son:

```
CREATE USER IDENTIFIED BY password;
GRANT privilege [, privilege...] TO user [, user| role, PUBLIC...];
CREATE ROLE role;
GRANT select on table TO role;
REVOKE {privilege [, privilege...]|ALL} ON object FROM {user[, user...]|role|PUBLIC}
        [CASCADE CONSTRAINTS];
```

Base de datos MS SQL Server

En la página internet https://es.wikipedia.org/wiki/Microsoft_SQL_Server indica lo siguiente:

Microsoft SQL Server es un sistema de gestión de base de datos relacional, desarrollado por la empresa Microsoft.

El lenguaje de desarrollo utilizado es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

Librerías de Red

En la página de internet de Microsoft, https://docs.microsoft.com/es-es/sql/sql-server/install/network-protocols-and-network-libraries?view=sql-server-ver15 indica lo siguiente:

Protocolos de red y bibliotecas de red

Un servidor puede escuchar en, o supervisar, varios protocolos de red al mismo tiempo. Sin embargo, cada protocolo debe estar configurado. Si un protocolo concreto no está configurado, el servidor no podrá escuchar en dicho protocolo. Después de la instalación, podrá cambiar las configuraciones de protocolo mediante el Administrador de configuración de SQL Server.

Configuración de red de SQL Server predeterminada

Se configura una instancia predeterminada de SQL Server para el puerto TCP/IP 1433 y la canalización con nombre \\.\pipe\sql\query. SQL Server se configuran para puertos dinámicos TCP, con un número de puerto asignado por el sistema operativo.

Si no puede utilizar direcciones de puerto dinámicas (por ejemplo, cuando las conexiones de SQL Server deben pasar por un servidor de firewall configurado pasar a través de direcciones de puerto específicas). Seleccione un número de puerto sin asignar.

Para mejorar la seguridad, la conectividad de red no se habilita totalmente al instalar SQL Server. Para habilitar, deshabilitar y configurar protocolos de red después de completar la instalación, utilice el área Configuración de red de SQL Server del Administrador de configuración de SQL Server.

Logins

En la página de internet de Microsoft, https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/logging-in-to-sql-server?view=sql-server-ver15 indica lo siguiente:

Iniciar sesión en SQL Server

Puede iniciar sesión en una instancia de Microsoft SQL Server utilizando cualquiera de las herramientas de administración gráfica o desde un símbolo del sistema.

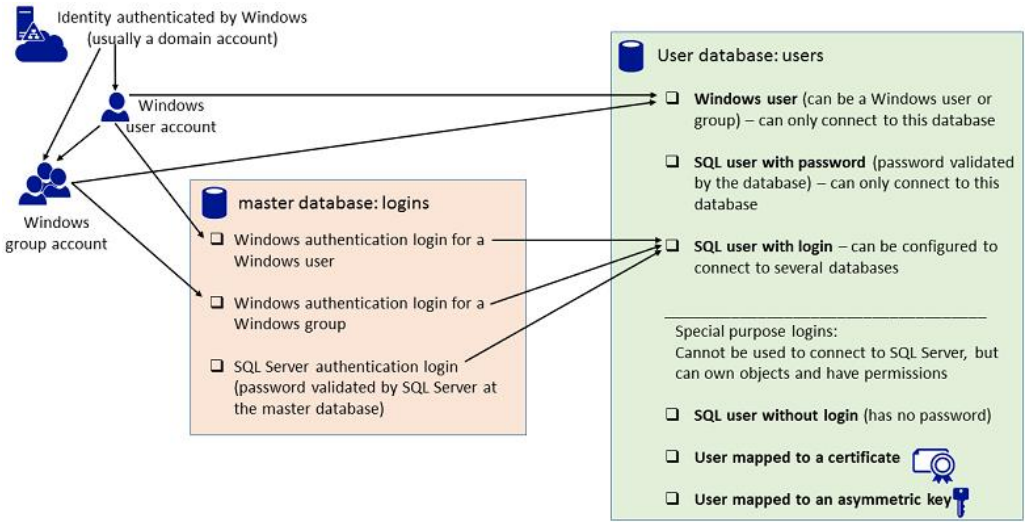
Cuando inicia sesión en una instancia de SQL Server mediante una herramienta de administración gráfica como SQL Server Management Studio, se le solicita que proporcione el nombre del servidor, un inicio de sesión de SQL Server y una contraseña, si es necesario. Si inicia sesión en SQL Server mediante la autenticación de Windows, no es necesario que proporcione un inicio de sesión de SQL Server cada vez que acceda a una instancia de SQL Server. En su lugar, SQL Server usa su cuenta de Microsoft Windows para iniciar sesión automáticamente. Si SQL Server se está ejecutando en autenticación de modo mixto (SQL Server y modo de autenticación de Windows) y elige iniciar sesión utilizando la autenticación de SQL Server, debe proporcionar un nombre de usuario y contraseña de SQL Server. Cuando sea posible, utilice la autenticación de Windows.

Formato para especificar el nombre de SQL Server

Al conectarse a una instancia del Motor de base de datos, debe especificar el nombre de la instancia de SQL Server. Si la instancia de SQL Server es la instancia predeterminada (una instancia sin nombre), especifique el nombre de la computadora donde está instalado SQL Server o la dirección IP de la computadora. Si la instancia de SQL Server es una instancia con nombre (como SQLEXPRESS), especifique el nombre de la computadora donde está instalado SQL Server, o la dirección IP de la computadora, y agregue una barra y el nombre de la instancia.

Usuarios

En la página de internet de Microsoft, https://docs.microsoft.com/en-us/sql/relational-databases/security/authentication-access/create-a-database-user?view=sql-server-ver15 indica lo siguiente:



Crear un usuario de base de datos

El usuario de la base de datos es la identidad del inicio de sesión cuando está conectado a una base de datos. El usuario de la base de datos puede utilizar el mismo nombre que el inicio de sesión, pero no es necesario.

- Los Logins son asignados a los usuarios
- Los grants se les asignan a los usuarios.
- A los usuarios se le asignan sus propios Esquemas(schemas)

Para crear un usuario en MsSqlServer hay muchas herramientas gráficas, pero si lo hace mediante SQL la instrucción es la siguiente:

```
CREATE LOGIN elLogin WITH PASSWORD = 'password'
GO
CREATE USER usuario FOR LOGIN elLogin
GO
```

Roles

Los Roles pueden existir a nivel de instancia o base de datos.

A nivel de Instancia:

- Los logins pueden ser otorgados roles llamados “server roles”.
- No se pueden crear roles nuevos

A nivel de Base de Datos

- Los usuarios de base de datos pueden ser otorgados roles.
- Se pueden crear roles nuevos.

Roles de una Aplicación

Un rol de aplicación sirve para asignarle permisos a una aplicación:

- Tiene un password
- No contiene usuarios

Auditoría

En La página de Microsoft, https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine?view=sql-server-ver15 se puede encontrar la siguiente información:

La auditoría de una instancia del Motor de base de datos de SQL Server o de una base de datos individual implica el seguimiento y registro de eventos que ocurren en el Motor de base de datos. La auditoría de SQL Server le permite crear auditorías de servidor, que pueden contener especificaciones de auditoría de servidor para eventos de nivel de servidor y especificaciones de auditoría de base de datos para eventos de nivel de base de datos. Los eventos auditados se pueden escribir en los registros de eventos o en archivos de auditoría.

Existen varios niveles de auditoría para SQL Server, según los requisitos gubernamentales o de estándares para su instalación. SQL Server Audit proporciona las herramientas y procesos que debe tener para habilitar, almacenar y ver auditorías en varios objetos de servidor y base de datos.

Puede registrar los grupos de acciones de auditoría del servidor por instancia y los grupos de acciones de auditoría de la base de datos o las acciones de auditoría de la base de datos por base de datos. El evento de auditoría ocurrirá cada vez que se encuentre la acción auditable.

Todas las ediciones de SQL Server admiten auditorías a nivel de servidor. Todas las ediciones admiten auditorías de nivel de base de datos a partir de SQL Server 2016 SP1. Antes de eso, la auditoría a nivel de base de datos se limitaba a las ediciones Enterprise, Developer y Evaluation.

Componentes de auditoría de SQL Server

Una auditoría es la combinación de varios elementos en un solo paquete para un grupo específico de acciones del servidor o acciones de la base de datos. Los componentes de la auditoría de SQL Server se combinan para producir un resultado que se denomina

auditoría, al igual que una definición de informe combinada con gráficos y elementos de datos produce un informe.

Descripción general del uso de auditoría de SQL Server

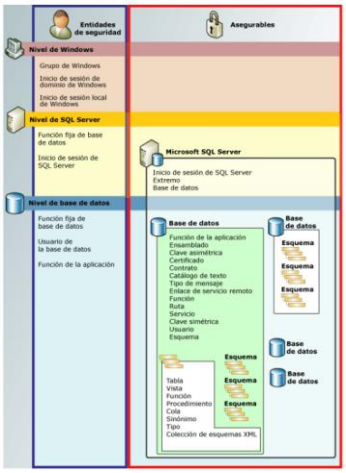
Puede utilizar SQL Server Management Studio o Transact-SQL para definir una auditoría. Una vez creada y habilitada la auditoría, el destino recibirá entradas.

Puede leer los registros de eventos de Windows mediante la utilidad Visor de eventos en Windows. Para los destinos de archivos, puede usar el Visor de archivos de registro en SQL Server Management Studio o la función fn_get_audit_file para leer el archivo de destino.

El proceso general para crear y utilizar una auditoría es el siguiente.

- Cree una auditoría y defina el objetivo.
- Cree una especificación de auditoría de servidor o una especificación de auditoría de base de datos que se corresponda con la auditoría. Habilite la especificación de auditoría.
- Habilite la auditoría.
- Leer los eventos de auditoría mediante el uso de Windows el Visor de eventos , archivo de registro Viewe r, o la función fn_get_audit_file.

Modelo de Seguridad

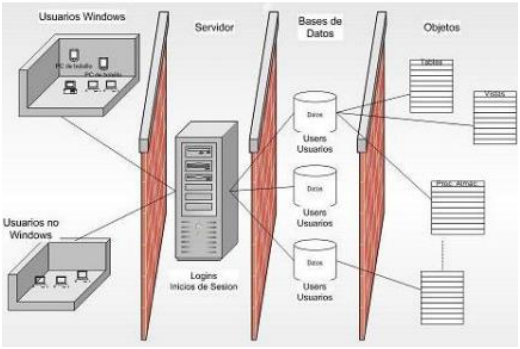


En SQL Server nos encontramos con tres niveles o capas en los cuales podemos gestionar la seguridad. El primero de ellos se encuentra a nivel de servidor, en él podemos gestionar quién tiene acceso al servidor y quién no, y además gestionamos que roles va a desempeñar. Para que alguien pueda acceder al servidor debe tener un

inicio de sesión (login) asignado, y a éste se asigna los roles o funciones que puede realizar sobre el servidor.

El que alguien tenga acceso al servidor no quiere decir que pueda acceder a las bases de datos que se encuentran en él. Para ello hay que tener acceso a la siguiente barrera de seguridad, que es a nivel de base de datos. Para que un login tenga acceso a una base de datos, tenemos que crear en ella un usuario (user).

Se debe crear un usuario en cada una de las bases de datos a las que queramos que acceda un login. Análogamente, el que un usuario tenga acceso a una base de datos no quiere decir que tenga acceso a todo su contenido, ni a cada uno de los objetos que la componen. Para que esto ocurra tendremos que irle concediendo o denegando permisos sobre cada uno de los objetos que la componen.



Autenticación

En la página de Microsoft https://docs.microsoft.com/es-es/dotnet/framework/data/adonet/sql/authentication-in-sql-server podemos encontrar la siguiente información relacionada:

SQL Server admite dos modos de autenticación, el modo de autenticación de Windows y el modo mixto.

- La autenticación de Windows es el modo predeterminado y a veces se le conoce como seguridad integrada porque este modelo de seguridad de SQL Server está estrechamente integrado en Windows. Se confía en las cuentas de usuario y grupo específicas de Windows para iniciar sesión en SQL Server. Los usuarios de Windows que ya se han autenticado no tienen que presentar credenciales adicionales.
- El modo mixto admite la autenticación mediante Windows y SQL Server. Los pares de nombre de usuario y contraseña se mantienen en SQL Server.

Con la autenticación de Windows, los usuarios ya están registrados en Windows y no es necesario que inicien sesión por separado en SQL Server. La siguiente SqlConnection.ConnectionString especifica autenticación de Windows sin que los usuarios tengan que proporcionar un nombre de usuario ni una contraseña.

```
C#: "Server=MSSQL1;Database=AdventureWorks;Integrated Security=true;"
```

Tipos de inicios de sesión

SQL Server admite tres tipos de inicios de sesión:

- Una cuenta de usuario local de Windows o una cuenta de dominio de confianza. SQL Server se basa en Windows para autenticar las cuentas de usuario de Windows.
- Grupo de Windows. La concesión de acceso a un grupo de Windows otorga acceso a todos los inicios de sesión de usuario de Windows que son miembros del grupo.
- Inicio de sesión de SQL Server. SQL Server almacena el nombre de usuario y un hash de la contraseña en la base de datos master mediante métodos de autenticación internos para comprobar los intentos de inicio de sesión.

Fortaleza de contraseñas

En la documentación Microsoft https://docs.microsoft.com/en-us/sql/relational-databases/security/strong-passwords?view=sql-server-ver15 podemos encontrar la siguiente información:

Las contraseñas pueden ser el vínculo más débil en una implementación de seguridad de servidor. Tenga mucho cuidado al seleccionar una contraseña. Una contraseña segura tiene las siguientes características:

- Tiene al menos ocho caracteres.
- Combina letras, números y símbolos dentro de la contraseña.
- No se encuentra en un diccionario.
- No es el nombre de un comando.
- No es el nombre de una persona.
- No es el nombre de un usuario.
- No es el nombre de una computadora.
- Se cambia con regularidad.

- Es diferente a las contraseñas anteriores.

Las contraseñas de Microsoft SQL Server pueden contener hasta 128 caracteres, incluidas letras, símbolos y dígitos. Debido a que los inicios de sesión, los nombres de usuario, los roles y las contraseñas se utilizan con frecuencia en las instrucciones Transact-SQL, ciertos símbolos deben ir entre comillas dobles (") o corchetes ([]). Utilice estos delimitadores en las instrucciones Transact-SQL cuando El inicio de sesión, usuario, rol o contraseña del servidor tiene las siguientes características:

- Contiene o comienza con un carácter de espacio.
- Comienza con el carácter $ o @.

Restricción de privilegios

La asignación o eliminación de los privilegios se puede hacer con las instrucciones GRAN/REVOKE/DENY. Por medio de estas instrucciones el DBA brinda la restricción de privilegios.

Aplicación de parches de seguridad

Microsoft cuenta con una serie de guías para proteger de vulnerabilidades mediante parches de seguridad. Todos estos parches están registrados y pueden descargarse de la cuenta https://support.microsoft.com/. Allí se indica cómo aplicar estos parches respectivos.

Seguridad del Servidor MS SQL

En la documentación Microsoft https://docs.microsoft.com/es-es/sql/relational-databases/security/securing-sql-server?view=sql-server-ver15 se encuentra la siguiente información:

La protección de la seguridad en SQL Server conlleva una serie de pasos que afectan a cuatro áreas: la plataforma, la autenticación, los objetos (incluidos los datos) y las aplicaciones que tienen acceso al sistema. Los siguientes temas le guiarán durante el proceso de creación e implementación de un plan de seguridad eficaz.

Puede buscar más información sobre la seguridad de SQL Server en el sitio web de SQL Server. En él encontrará una guía de prácticas recomendadas y una lista de

comprobación de seguridad. Este sitio también incluye las últimas descargas e información sobre los Service Packs

Seguridad de la plataforma y de la red

La plataforma de SQL Server incluye el hardware físico y los sistemas de redes que conectan los clientes con los servidores de bases de datos, así como los archivos binarios que se utilizan para procesar solicitudes de base de datos.

Seguridad física

Las recomendaciones de seguridad física limitan de forma estricta el acceso al servidor físico y a los componentes de hardware. Por ejemplo, use salas cerradas de acceso restringido para el hardware de servidor de base de datos y los dispositivos de red. Además, limite el acceso a los medios de copia de seguridad almacenándolos en una ubicación segura fuera de las instalaciones.

La implementación de la seguridad de la red física comienza por mantener a los usuarios no autorizados fuera de la red.

Seguridad del sistema operativo

Los Service Packs y las actualizaciones del sistema operativo incluyen mejoras de seguridad importantes. Aplique todas las revisiones y actualizaciones al sistema operativo después de probarlas con las aplicaciones de base de datos.

Los firewalls también proporcionan formas eficaces de implementar la seguridad. Lógicamente, un firewall es un separador o limitador del tráfico de red, que puede configurarse para aplicar la directiva de seguridad de datos de su organización. Si utiliza un firewall, aumentará la seguridad del sistema operativo, ya que proporciona un cuello de botella en el que pueden concentrarse las medidas de seguridad.

Seguridad de los archivos del sistema operativo de SQL Server

SQL Server usa archivos del sistema operativo para el funcionamiento y el almacenamiento de datos. Las recomendaciones de seguridad de archivos indican que se restrinja el acceso a estos archivos.

Entidades de seguridad y seguridad de objetos de base de datos

Las entidades de seguridad son los individuos, grupos y procesos que tienen acceso a SQL Server. Los "elementos protegibles" son el servidor, la base de datos y los objetos incluidos en la base de datos. Cada uno de estos elementos dispone de un conjunto de permisos que pueden configurarse para reducir el área expuesta de SQL Server

Cifrado y certificados

El cifrado no resuelve los problemas de control de acceso. Sin embargo, mejora la seguridad debido a que limita la pérdida de datos, incluso en el caso poco probable de que se superen los controles de acceso. Por ejemplo, si el equipo host de base de datos no está configurado correctamente y un usuario malintencionado obtiene datos confidenciales, como números de tarjetas de crédito, esa información robada podría resultar inservible si está cifrada

Base de datos PostgreSQL

En la página de internet https://es.wikipedia.org/wiki/PostgreSQL se encuentra la siguiente información:

PostgreSQL, también llamado Postgres, es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

PostgreSQL no tiene un gestor de defectos, haciendo muy difícil conocer el estado de sus defectos.

"Seguro por defecto"

Por defecto, sólo puedes conectarte desde localhost:

- ➔ A través de sockets UNIX, como el mismo usuario del S.O.
- ➔ A través de TCP/IP, típicamente con password

Si quieres conectarte desde otras Ips:

- ➔ Añade la(s) IP(s) en postgresql.conf, parámetro listen_addresses (formato CSV)
- ➔ Edita pg_hba.conf para añadir la red (CIDR)desde donde se permita conectarse

Conectividad

pgAdmin es la herramienta oficial para administrar nuestras bases de datos en PostgreSQL. Nos permite desde hacer búsquedas SQL hasta desarrollar toda nuestra base de datos de forma muy fácil e intuitiva; directamente desde la interfaz gráfica.

Conexión a la base de datos

Para conectarse a ella, es necesario disponer de la siguiente información relativa a la base de datos:

- la dirección de la instancia en la que está alojada;

- el puerto;
- el nombre de usuario;
- la contraseña;

Para conectar desde una aplicación realizada en C# debe tener la siguiente programación:

```
using npgsql;
private void btnaceptar_Click(object sender, RoutedEventArgs e)
{
    bool blnfound = false;

    NpgsqlConnection conn = new NpgsqlConnection("Server=localhost;Port=5432;
                                User Id=postgres;Password=1234;Database = systemBD");
    conn.Open();
    NpgsqlCommand cmd = new NpgsqlCommand("Select * from usuario where cod_usu = '" +
txt1 + "' and con_usu = '" + txt2 + "' ", conn);
    NpgsqlDataReader dr = cmd.ExecuteReader();
    if (dr.Read())
    {
        blnfound = true;
        modulos form = new modulos();
        form.Show();
        this.Hide();
    }
    if (blnfound == false)
        MessageBox.Show("Usuario o Contraseña Incorrecta", "Mensaje de Alerta",
MessageBoxButton.OK);
    dr.Close();
    conn.Close();
}
```

Modos de autenticación

En la pagina https://wiki.postgresql.org/wiki/Autenticaci%C3%B3n_del_Cliente se encuentra la siguiente información:

Autenticación del Cliente

Los usuarios permitidos para el establecimiento de la conexión a la base de datos son controlados por un archivo sencillo en el directorio raíz de su base de datos, llamado **pg_hba.conf**. En cuanto corre initdb para la creación del clúster de la base de datos, es creado un fichero pg_hba.conf con los valores por omisión.

Los permisos que existen por omisión dependen de cómo fue invocado initdb. Por omisión, los nuevos clústers son creados con el tipo trust con el cual, cualquier usuario local puede conectarse a la base de datos. Sin embargo, algunos empaquetadores de PostgreSQL pueden cambiar esto. Por ejemplo, si usa 'service initdb' en Red Hat para crear el clúster, éste llama a initdb de la siguiente manera:

```
initdb --pgdata='$PGDATA' --auth= 'ident sameuser'
```

El cual usa el no particularmente popular método ident para averiguar si un usuario está permitido conectarse, lo cual suele ser frustrante para quienes no son conscientes de ello.

Una instalación típica recomendada para el acceso desde la red de la base de datos toma la dirección local LAN y sólo permite los clientes que se autentiquen usando una contraseña encriptada a través de MD5. La siguiente entrada en en el pg_hba.conf ilustrará algo como esto:

```
# TIPO BASE_DE_DATOS USUARIO CIDR-DIRECCIÓN   MÉTODO
host    all          all     192.168.1.0/24   md5
```

Esto sólo permite conectarse a los clientes con las direcciones IP de la red 192.168.1.0 -- 192.168.1.255 y sólo si proveen la contraseña correcta para el usuario. Observe que un acceso de la red como éste puede establecerse sólo si está permitido en el parámetro listen_addresses de postgresql.conf.

Las contraseñas de los usuarios de la base de datos son aplicadas cuando se crea el usuario con CREATE ROLE y pueden ser modificadas con la orden ALTER ROLE. createuser puede ser una herramienta muy útil para ayudar en este sentido.

Hay una pequeña trampa aquí: puesto que para la creación de un nuevo rol se requiere que la conexión a la base de datos sea como un superusario, ¿cómo se pueden empezar a crear usuarios? Un enfoque común es comenzar con un pg_hba.conf que confía sólo a los usuarios que se conectan localmente:

```
# Sólo conexiones locales para el socket de Unix
local   all         all                               trust
# IPv4 local connections:
host    all         all         127.0.0.1/32          trust
```

Si la base de datos está configurada de esta forma, cualquiera logueado en el sistema puede ahora conectarse al servidor a su voluntad, así que no desea conservar esta configuración por mucho tiempo. Lo que puede hacer ahora es usar la orden ALTER ROLE para asignar una contraseña fuerte al superusuario postgres. Una vez que lo haya hecho, puede detener el servicio (pg_ctl stop), cambiar todos los "trust" por "md5", agregar su rango o bloque de red completo a pg_hba.conf, y cambiar el parámetro listen_address de postgresql.conf. Cuando lo inicie nuevamente, ya tendrá una cuenta de superusuario de la cual conoce la contraseña, con la cual puede crear las cuentas para sus usuarios regulares.

Roles

En la página Postgresql.Org/Docs/Crear_Roles se encuentra la siguiente información.

CREATE ROLE agrega un nuevo rol a un clúster de base de datos PostgreSQL. Un rol es una entidad que puede poseer objetos de base de datos y tener privilegios de base de datos; un rol puede considerarse un "usuario", un "grupo" o ambos, dependiendo de cómo se utilice. Debe tener el privilegio CREATEROLE o ser un superusuario de la base de datos para usar este comando.

Tenga en cuenta que los roles se definen en el nivel del clúster de la base de datos, por lo que son válidos en todas las bases de datos del clúster.

Auditoría

La auditoría se debe programar por medio de disparadores (triggers) en las tablas que se desea hacer una auditoria de datos, el disparador deja la información recolectada en una tabla que normalmente se llama AUDITORIA y se crea de la siguiente forma:

```
CREATE schema audit;
REVOKE CREATE ON schema audit FROM public;
create table audit.auditoria
( fechahora timestamp not null,
usuario text not null,
operacion text not null,
nombre_esquema text not null,
nombre_tabla text not null,
antes hstore,
despues hstore );
REVOKE ALL ON audit.auditoria FROM public;
```

Catálogo y tablas principales del sistema

En la página internet https://www.ibiblio.org/pub/linux/docs/LuCaS/Postgresql-es/web/navegable/todopostgresql/x16698.html encontramos la siguiente informacion:

Nombre del Catalogo	Descripción
pg_database	base de datos
pg_class	clases
pg_attribute	atributos de clases
pg_index	índices secundarios
pg_proc	procedimientos (ambos C y SQL)
pg_type	tipos (ambos base y complejo)
pg_operator	operadores
pg_aggregate	conjunto de funciones
pg_am	método de acceso
pg_amop	operador de método de acceso

pg_amproc	soporte de operador de método de acceso
pg_opclass	operador de clases de método de acceso

Consideraciones a nivel de sistema operativo

En la página oficial de PostgreSQL https://todopostgresql.com/tipos-de-instalaciones-de-postgresql/ se tiene la siguiente informacion:

Si necesitas instalar PostgreSQL, lo primero que debes de hacer es recopilar la siguiente información:

- Sistema Operativo
- Arquitectura (32/64 bits, x86, etc)
- Tipo de instalación

Sistemas Operativos soportados por PostgreSQL

Se puede instalar en multitud de Sistemas Operativos, éstas son algunas de las plataformas soportadas:

- Distribuciones BSD, como puede ser FreeBSD, OpenBSD. Estos SO. son de código abierto.
- Distribuciones GNU / Linux, tenemos todas las distribuciones recientes. Entre ellas tenemos sistemas Red Hat, CentOS, Fedora, Oracle Linux, Debian, Ubuntu, Suse, OpenSuse y análogas.
- Mac OS X.
- Microsoft Windows, desde Windows 2000 hasta el más reciente
- Solaris.

En la web oficial de postgreSQL, se tienen las notas para instalar PostgreSQL en plataformas específicas: AIX, Cygwin, HP-UX, MingGW/Native Windows, SCO OpenServer y SC UnixWare, Solaris.

Además, PostgreSQL puede instalarse en muchas arquitecturas de CPU, entre ellas, 32 bits (X86) y 64 bits (X64).

Dado que cada plataforma dispone de una forma diferente de instalar Postgres, puedes ver como se instala en los principales sistemas operativos donde se utiliza PostgreSQL.

Bibliografía

1. Romero M., Figueroa G., Vera D., Álava J., Parrales G., Álava C., Murillo Á., Castillo M. (2018). Introducción a la seguridad informática y el análisis de vulnerabilidades. Universidad Estatal del Sur de Manabí: Editorial Área de Innovación y Desarrollo, S.L. Recuperado de https://www.3ciencias.com/wp-content/uploads/2018/10/Seguridad-inform%C3%A1tica.pdf
2. Camps R., Casillas L., Costal D., Gibert M. (2005). Bases de datos. Barcelona, España, ISBN: 8497882695
3. Silberschatz, Korth, Sudarshan. (2002). FUNDAMENTOS DE BASES DE DATOS. Madrid, España: McGRAW-HILL/INTERAMERICANA DE ESPAÑA.
4. Oppel, A. (2010). Fundamentos de bases de datos. México: McGraw-Hill Interamericana. ISBN: 9786071502544, recuperado de https://elibro.net/es/ereader/usanmarcos/37322.

San Marcos

www.usanmarcos.ac.cr

San José, Costa Rica