ASEGURAR SISTEMA DE GESTION DE BASES DE DATOS ORACLE

Julian Camilo Alvarado Carlos

Universidad Libre

Facultad de Ingeniería

Ingeniería de Sistemas

Bogotá D.C

2017

Contenido

Resumen

Oracle nos permite asegurar nuestras bases de datos desde diferentes aspectos como lo son proteger las cuentas de usuario asignando una política de contraseñas fuertes y limitando los recursos de sistema para cada usuario mediante el uso de perfiles, administrando los controles de acceso a la base de datos mediante el uso de privilegios y roles.

Se puede proteger la información de la base de datos mediante el uso de herramientas como lo son Oracle Virtual Data Base y Oracle Data Redaction permitiendo tener un control sobre lo que un usuario puede consultar, modificar y eliminar de una tabla. Permitiendo ocultar la información, cifrando la información de la base de datos hasta el tráfico de la red.

Oracle nos permite tener un control sobre los usuarios mediante su función de auditoria monitoreando las actividades de cualquier usuario sobre la base de datos lo que permite encontrar vulnerabilidades en el control de acceso.

Introducción

En la actualidad dentro de las organizaciones se está poniendo en marcha distintas normas, procedimientos, métodos y técnicas orientados en prevenir, proteger y resguardar lo que es considerado como información sensible que puede ser susceptible de robo, perdida o daño.

Se deben identificar las necesidades de seguridad y los riesgos que pueden amenazar al sistema, relacionando las acciones o medidas de seguridad que deben implementarse para afrontar los riesgos anteriormente identificados.

La presente monografía tiene como objetivo exponer los aspectos y herramientas a tener en cuenta para proteger una base datos Oracle, Controlando el uso de los recursos de sistema por usuario, monitoreando las acciones de los usuarios lo cual permitirá hallar posibles vulnerabilidades en el sistema, cifrando los datos de la base de datos y en el tráfico de red evitando que los paquetes que viajan sean ilegibles para intrusos, administrando el acceso y el uso de los datos, previniendo accesos no autorizados a los distintos objetos de la base de datos como los son tablas, vistas, índices, procedimientos etc.

Desarrollo

"La seguridad de la información está relacionada con las medidas preventivas aplicadas con el fin de salvaguardar y proteger la información bajo la confidencialidad, disponibilidad e integridad." (Solarte, Rosero, del Carmen Benavides, 2015, p. 497).

Oracle dispone de una gran variedad de herramientas y funciones enfocadas al aseguramiento de las bases de datos controlando desde el acceso de los usuarios mediante un método de autenticación, la implementación de políticas de contraseñas fuertes, protección y control de acceso de los datos, hasta el monitoreo de las acciones de los usuarios mediante la función de auditoria.

1. Usuarios

Las bases de datos Oracle tienen una lista de usuarios válidos para acceder a la base de datos, cada usuario debe ser único, debe tener un método de autenticación que puede ser por medio de contraseña, global o de forma externa además debe tener asignado un tablespace predeterminado y temporal, un perfil y un estado que puede ser:

- Abierto.
- Bloqueado.
- Expirado.

1.1.Usuarios Comunes

Es un usuario que existe con la misma identidad en cada uno de los PDB (Pluggable database) en un ambiente CDB (container database), Este tipo de usuario puede conectarse a la raíz del CDB, y puede ejecutar tareas sobre los PDB's como lo son:

- Asignar permisos a los usuarios comunes y locales.

- Ejecutar la sentencia ALTER DATABASE para el CDB.
- Ejecutar la sentencia ALTER PLUGGABLE DATABASE para cambiar el estado de algún PDB mientras esté conectado al CDB.

1.2.Usuarios Locales

Son usuarios que solo existen para solo un PDB (Pluggable database), solo pueden ejecutar tareas en su PDB, se caracterizan por:

- No pueden crear usuarios comunes, ni asignarles privilegios.
- Se le puede asignar roles comunes, pero solo aplicaría para el PDB al cual pertenezca el usuario local.
- Una cuenta de usuario local debe ser única solo en su PDB.

1.3.Autenticación de Usuarios

Autenticación significa verificar la identidad de un usuario, dispositivo u otra entidad que desee utilizar la datos, recursos o aplicaciones, la validación de esta entidad establece una relación de confianza para futuras interacciones con la base de datos.

La autenticación también permite la rendición de cuentas haciendo posible vincular el acceso y las acciones a identidades específicas, después de la autenticación los procesos de autorización pueden permitir o limitar los niveles de acceso y acción permitidos a esta entidad.

Oracle nos ofrece tres tipos de autenticación por medio de contraseña donde el usuario tiene asociada una contraseña la cual debe ser ingresada para establecer una conexión con la base de datos, autenticarse de forma externa mediante el sistema operativo como lo es mediante el directorio activo y por último la autenticación de forma global mediante la herramienta Oracle Internet Directory.

2. Perfiles

Un perfil es una colección de atributos que se aplican a un usuario, permitiendo un único punto de referencia para cualquiera de los usuarios que compartan dichos atributos.

Cada usuario puede tener sólo un perfil y asignar uno nuevo reemplaza al anterior, si se realiza un cambio a un perfil y el usuario se encuentra logueado, el cambio tomara efecto hasta la siguiente autenticación; Sólo se deben crear y administrar perfiles de usuario en caso de aplicar una política de contraseñas fuertes o limitar los recursos del sistema a determinados usuarios.

2.1.Protección de Contraseñas

Se pueden asegurar las contraseñas de los usuarios en una varias de formas, controlando los requisitos mínimos para asignar una contraseña o utilizar directicas de administración de contraseñas.

Las bases de datos Oracle nos proporcionan las siguientes formas de proteger las contraseñas:

- Encriptación de contraseñas: Oracle cifra automáticamente y de forma transparente las contraseñas durante las conexiones de red (Cliente-Servidor y Servidor-Servidor) utilizando AES (Advanced Encryption Standard) antes de enviar los paquetes por la red, se aconseja tener activada la encriptación de red nativa de la opción de seguridad avanzada o configurar Secure Sockets Layer (SSL).
- Comprobación de complejidad: Viene por defecto en la instalación de la base de datos, proporciónalas funciones de verificación de la contraseña the ora12c_verify_function y ora12c_strong_verify_function para garantizar que las contraseñas nuevas o modificadas sean lo suficientemente complejas para evitar que sean adivinadas por intrusos en el sistema.

- Sensibilidad de mayúsculas y minúsculas: En las bases de datos Oracle, las contraseñas son sensibles a las mayúsculas por ejemplo la contraseña ULIBRE21 fallaría si se introduce Ulibre21.
- Protección SHA-512: Para verificar la contraseña de los usuarios y aplicar la sensibilidad de contraseñas Oracle Database usa una versión de contraseña 12C la cual se basa en un algoritmo optimizado que involucra la función Password-Based Key Derivation Function y la función SHA-512.
- Retraso de inicio de sesión: Si un usuario intenta iniciar sesión varias veces utilizando una contraseña incorrecta, la base de datos retrasará cada inicio de sesión por un segundo, esta protección se aplica a los intentos que se realizan desde diferentes direcciones IP o múltiples conexiones de cliente, esta característica ofrece una reducción significativa del número de contraseñas que un intruso podría intentar dentro de un periodo de tiempo fijo.

Se recomienda buscar las cuentas de usuario con contraseñas predeterminadas mediante la vista DBA_USERS_WITH_DEFPWD, ya que al instalar una base de datos se crean cuentas de usuario como lo son HR, OE y SCOTT cuyas contraseñas son conocidas por ende vulnerables, para mayor seguridad se deben cambiar las contraseñas de estas cuentas o desactivar dichas cuentas de usuario, ya que el uso de una contraseña predeterminada hace que la base de datos sea vulnerable a ataques de intrusos.

2.2.Administrar Contraseñas Mediante Perfiles

Se recomienda como mínimo controlar el número máximo de veces en las cuales se puede intentar iniciar sesión y fallar hasta bloquear la cuenta de usuario, asignar el número de días en los cuales un usuario tiene que cambiar su contraseña antes de que expire, definir el número de días en los cuales el usuario puede usar su contraseña actual hasta que le solicite cambiarla y por ultimo asignar el número de días en los cuales una cuenta estará bloqueada después de sobrepasar número de intentos fallidos de inicio de sesión.

Se puede crear un perfil que quede por defecto controlando las anteriores recomendaciones asignando al perfil las sentencias:

- FAILED_LOGIN_ATTEMPTS
- PASSWORD_GRACE_TIME
- PASSWORD_LIFE_TIME
- PASSWORD_LOCK_TIME.

2.3. Complejidad de Contraseñas

El uso de una función de verificación de complejidad obliga a los usuarios a crear contraseñas fuertes y seguras brindando una protección moderada ante intrusos que intenten ingresar al sistema adivinando contraseñas.

Oracle database ofrece cuatro funciones de verificación de contraseñas, estas funciones se encuentran en el script ultpdmg.sql ubicado en $ORACLE_HOME/rdbms/admin, cuando se aplica la comprobación de contraseñas aplicara para todos los usuarios exceptuando el usuario SYS.

2.3.1. Función verify_function_11G

- La contraseña no debe ser igual al nombre de usuario ni deletreado de forma inversa o con números.
- La contraseña no debe ser la misma que el nombre del servidor o el nombre del servidor con números.
- La contraseña no debe contener (oracle, oracle123, welcome1, database1, account1, user1234, password1, oracle123, computer1, abcdefg1).
- La contraseña debe contener al menos un número y un carácter.
- La contraseña debe ser diferente a la contraseña anterior al menos por 3 caracteres.
- La contraseña debe tener como mínimo 8 caracteres.

2.3.2. Función ora12c_verify_function

- La contraseña como mínimo debe tener 8 caracteres y al menos un número.
- La contraseña no debe ser igual al nombre de usuario o el mismo invertido.
- No puede ser igual al nombre de la base de datos.
- La contraseña no debe contener (Oracle, oracle123, welcome1, database1, account1, user1234, password1, oracle123, computer1, abcdefg1).
- La contraseña debe ser diferente a la contraseña anterior al menos por 3 caracteres.
- La contraseña debe tener al menos un carácter especial.
- La contraseña no debe sobrepasar los 30 caracteres.

2.3.3. Función ora12c_strong_verify_function

- La contraseña debe contener al menos 2 mayúsculas, 2 minúsculas, 2 números y 2 caracteres especiales.
- La contraseña debe ser diferente a la anterior con diferencia de 4 caracteres.
- Como mínimo debe tener 9 caracteres y como máximo 30.

2.3.4. Función ora12c_stig_verify_function

- la contraseña debe tener al menos 15 caracteres.
- debe tener 1 mayúscula y 1 minúscula.
- debe tener al menos 1 numero.
- debe tener al menos un carácter especial.
- tiene que ser diferente a la contraseña anterior por 8 caracteres.
- no debe exceder los 30 caracteres.

2.4.Límite de recursos

Se pueden establecer límites a la cantidad de recursos para cada usuario como parte del dominio de seguridad, estos límites deben ser asignados a un perfil y este a un usuario; Oracle clasifica el uso de los recursos del sistema en las siguientes categorías:

- A nivel de sesión.
- A nivel de llamada.
- A nivel de tiempos de CPU.
- A nivel de lectura lógicas.
- Otros.

2.4.1. A nivel de sesión

Al conectarse un usuario a la base de datos se crea una sesión a la cual se le pueden asignar limites, si el usuario llega a sobrepasar el límite de recursos asignados, la base de datos realiza un rollback y le indica al usuario que alcanzó el límite de recursos que puede usar.

2.4.2. A nivel de llamada

Cada vez que un usuario ejecute una sentencia SQL la base de datos Oracle realiza diferentes pasos para ejecutar la sentencia, durante este proceso se realizan varias llamadas a la base de datos, para evitar que una sola llamada utilice los recursos del sistema de forma excesiva, Oracle permite asignar limites a cada llamada.

Si un usuario excede los límites de recursos a nivel de llamada, la base de datos detiene el procesamiento de la sentencia, ejecuta un roll back y retorna un error.

2.4.3. A nivel de tiempos de la CPU

Cuando se ejecutan sentencias SQL y llamadas a la base de datos el tiempo de CPU es necesario para procesarlas.

Las llamadas requieren una pequeña cantidad de tiempo de CPU, pero una sentencia SQL que implique una gran cantidad de recursos puede utilizar una gran cantidad de tiempo de CPU, reduciendo el tiempo de CPU disponible para otras sentencias.

Para evitar el uso excesivo de tiempo de la CPU se pueden establecer límites fijos o dinámicos en los tiempos de la CPU para cada llamada, estos tiempos se miden en centésimas de segundo (0.01 segundos).

2.4.4. A nivel de Lecturas Lógicas

Input/output es una de las operaciones que más recursos de sistema usa, este tipo de operaciones pueden consumir la mayoría de recursos y hacen que las otras operaciones de base de datos compitan por estos recursos.

Para prevenir lo anterior se puede limitar las lecturas de los bloques de datos lógicos para cada llamada y para cada sesión, estas lecturas a los bloques de datos hacen uso de memoria como de disco, los limites se estableces y se miden en número de lecturas de bloques realizadas por una llamada o durante una sesión.

2.4.5. Otros

- Limitar el número de sesiones concurrentes de cada usuario.
- Limitar el tiempo de inactividad para una sesión.
- Limitar el tiempo de conexión para cada sesión.

3. Privilegios

Un privilegio de usuario es el derecho a ejecutar un tipo particular de sentencias SQL o el derecho a acceder a un objeto que pertenece a otro usuario o ejecutar un paquete PL/SQL permitiendo controlar las acciones que pueden ejecutar los usuarios en la base de datos por lo cual se deben asignar cuidadosamente.

"Un privilegio mal asignado o la incapacidad de retirar el permiso en el momento adecuado conllevan a accesos no autorizados a la información y los recursos protegidos ocasionando incidentes en donde se comprometa la confidencialidad e integridad de la información y los recursos". (Montoya, Restrepo, 2012, p.25).

los privilegios se clasifican en dos categorías:

- Privilegios de sistema
- Privilegios de Objeto

3.1.Privilegios de Sistema

Son los privilegios más potentes del sistema por lo que se deben asignar con precaución a solo usuarios de confianza. Un privilegio de sistema es el derecho de realizar acciones sobre cualquier objeto de cualquier esquema en la base de datos como lo es crear un tablespace o borrar una tabla de la base de datos.

3.2.Privilegios de Objeto

Permiten realizar acciones en objetos de un schema determinado, por ejemplo:

- Actualizar los registros de una tabla.
- Consultar una tabla de otro usuario.
- Ejecutar un procedimiento almacenado de otro usuario.

3.2.1. Privilegios de Tablas y Vistas

Permiten asegurar las tablas a nivel de sentencias DML o DDL asignando privilegios para el uso de DELETE, INSERT, SELECT y UPDATE en tablas y vistas.

3.2.2. Privilegios de Procedimientos

Privilegios para ejecutar procedimientos almacenados o funciones, en caso de necesitar crear o modificar procedimiento almacenados se deben asignar privilegios de sistema.

4. Roles

Un rol es un grupo de privilegios relacionados, lo que permite administrar y controlar los privilegios de forma más fácil, estos son creados por usuarios generalmente administradores para agrupar privilegios u otros roles, lo cual es una manera de facilitar la concesión de múltiples privilegios a los usuarios.

4.1.Propiedades de los roles

- A un rol se le pueden otorgar privilegios de sistema o de objeto.
- Cualquier rol se puede conceder a cualquier usuario de la base de datos.
- Se puede otorgar un rol a otros roles. Sin embargo, un rol no puede ser concedido a sí mismo y no puede ser concedido circularmente. Por ejemplo, el rol-1 no se puede conceder al rol-2 si el rol-2 se ha concedido previamente a rol-1.
- Opcionalmente, se puede hacer configurar para que un rol sea un rol predeterminado y aplique automáticamente cuando un usuario es creado.

4.2.Ventajas del uso de roles

- Reduce la administración de privilegios ya que en lugar de conceder los privilegios uno a uno a determinado usuario, se puede asignar los privilegios que sean necesarios a un rol y este ser asignado a un grupo de cuentas de usuario.
- Gestión de privilegios dinámicos, al modificar los privilegios de un rol automáticamente se aplican los cambios a los usuarios que tengan asignado dicho rol.
- Disponibilidad selectiva de privilegios, se puede activar o desactivar selectivamente roles que hayan sido asignados a un usuario.
- Protección de roles con contraseña, se puede activar un rol cuando se ingresa la contraseña asociada a su rol.

Se recomienda solo asignar los privilegios necesarios según los permisos que necesite cada rol, como buena práctica se deben asignar los privilegios a los roles no a los usuarios, para asegurar el control de acceso se deben restringir al máximo los privilegios de tipo Sistema y Objeto, así mismo limitar el número de usuarios que se pueden conectar como SYS y tener especial cuidado cuando se esté asignando un privilegio con la sentencia ANY ya que le permite ejecutar acciones sobre cualquier objeto de la base de datos por ejemplo DROP ANY TABLE, se deben asignar los privilegios que puedan crear, modificar y/o borrar objetos de la base de datos solo a usuarios de confianza.

5. Protección Transparente de información sensible

Transparent Sensitive Data Protection (TSDP) es una herramienta de seguridad solo disponible para versiones Enterprise que permite encontrar rápidamente columnas de forma centralizada para cualquier tabla la cual contenga información confidencial como lo pueden ser números de tarjetas de crédito, números telefónicos, direcciones etc; De esta forma nos permite crear políticas de seguridad para proteger dichas columnas mediante el uso de Oracle Data Redaction y Oracle Virtual Private Database.

5.1.Beneficios de TSDP

- Se puede configurar la política de seguridad y designar que clase de datos deben ser protegidos, en pocas palabras no se debe relacionar a que columnas se les debe aplicar dicha política, TSDP busca las columnas objetivo definidas en la directiva configurada.
- Se puede administrar la protección de varias columnas con información sensible, se puede habilitar o deshabilitar por medio de un solo parámetro a nivel de toda la base de datos, de una clase de dato sensible, un esquema, una tabla o columna específica, esta granularidad permite un alto nivel de control sobre la seguridad de los datos.

6. Oracle Virtual Private Database

OVPD (Oracle Virtual Private Database) permite crear una política de seguridad para controlar el acceso a nivel de columnas y filas, Esencialmente OVPD agrega una clausula dinámica WHERE a una sentencia SQL ejecutada en una tabla o vista con una directiva OVPD asignada.

OVPD refuerza la seguridad a un nivel muy granular, ya que añade las políticas de seguridad directamente a dichos objetos (tablas, vistas y sinónimos) y se aplican automáticamente cada vez que un usuario acceda a la tabla o vista por lo que no hay forma de burlar la seguridad, Cuando un usuario accede directa o indirectamente a una tabla, vista o sinónimo que esté protegido por una política de OVPD, se modifica la sentencia SQL del usuario añadiendo la condición WHERE; Oracle modifica la sentencia de forma dinámica y transparente para el usuario utilizando cualquier condición que pueda se expresada o devuelta en una función, estas políticas se pueden aplicar a las sentencias SELECT, INSERT, UPDATE, INDEX y DELETE.

Por ejemplo, si un usuario ejecuta la sentencia SELECT * FROM EMPLEADOS la política de seguridad configurada solo le permitiría al usuario consultar los empleados del área de "Finanzas" lo que automáticamente agregaría a la sentencia la condición WHERE con esa área. SELECT * FROM EMPLEADOS WHERE DEPARTAMENTO = 'FINANZAS'.

6.1.Beneficios de OVPD

- Seguridad, ya que brinda un control granular a las acciones que puede ejecutar un usuario en una tabla, vista o sinónimo.
- Sencillez, agregar una política a una tabla vista o sinónimo solo una vez, en lugar de asignar una política para una tabla otra para una vista y otra para un sinónimo.

- Flexibilidad, puede tener una directiva de seguridad para sentencias SELECT otra para INSERT y otras para sentencias UPDATE y DELETE, por ejemplo, es posible que se desee habilitar a los empleados de recursos humanos para que tengan privilegios de SELECT para todos los registros de empleados de su área, o para actualizar únicamente los salarios de los empleados del área de mercadeo cuyos apellidos empiecen por la B.

7. Oracle Data Redaction

Esta funcionalidad enmascara los datos de las columnas en tiempo real al momento en el que un usuario trata de ver los datos, es ideal para sistemas dinámicos en los cuales los datos cambian contestemente

Se pueden enmascarar las columnas con los siguientes métodos:

- Redacción completa: Se enmascara todo el contenido de los datos de la columna, el valor que enmascarado al ser consultado retornara según el tipo de dato de la columna en caso de ser un NUMBER retornara con un cero (0) y los datos CHAR se enmascaran como un espacio en blanco.
- Redacción parcial: Se enmascara una parte de los datos de la columna, se puede usar en las columnas que contengan los números de tarjetas de crédito enmascarando con asteriscos (*) los primeros 8 dígitos.
- Expresiones Regulares: Pueden ser usadas mediante redacción completa y parcial basados en un patrón de búsqueda para los datos, puede ser usado para enmascarar números de teléfono o correos electrónicos específicos.
- Redacción de valores NULL: Esta característica nos permite ocultar todos los datos confidenciales en una tabla o columna y remplazarla con valores NULL.

- Redacción aleatoria: Los datos que son protegidos son retornados al usuario como valores generados aleatoriamente cada vez que se visualice.

8. Encriptación de datos

La encriptación de datos consiste en un procedimiento que mediante un algoritmo transforma o enmascara la información de texto claro a uno que sea incomprensible, Se debe tener en cuenta que al encriptar datos de campos que sean índices o llaves ya que automáticamente quedarían inhabilitados.

Es un error común pensar que el cifrado controlara los problemas de control de acceso, por lo que es importante que el cifrado no interfiera con la asignación de privilegios y roles.

No protege ante un Administrador malicioso, si un usuario llega a obtener privilegios elevados cifrar los datos no protegerá la base de datos ante todas las acciones destructivas que pueda ejecutar dicho intruso como puede ser borrar información, la solución correcta seria proteger las cuentas con grandes privilegios y cambiar las contraseñas predeterminadas para tener dicho riesgo, Se debe tener en cuenta la disponibilidad de los datos ya que el cifrado de datos puede hacer que los datos no estén habilitados debido a una reducción de rendimiento de la base de datos, la disponibilidad también se ve afectada ya que por buenas prácticas se deben cambiar las llaves de cifrado cada cierto tiempo, al realizar este proceso deriva en que la base de datos estará inaccesible mientras los datos son descifrados con la llave anterior y vueltos a cifrar con la nueva.

8.1.Generación de llaves

Los datos encriptados son tan seguros como la llave utilizada para cifrarlos, una llave de cifrado debe generarse de forma segura, Oracle proporciona la función RANDOMBYTES que permite la generación de números aleatorios seguros el cual es certificado por RSA**.**

Cuando se necesite transferir la llave de cifrado de la aplicación a la base de datos esta debe ir cifrada ya que de lo contrario un intruso podría tener acceso la llave transmitida, por ello se recomienda cifrar los datos de la red.

8.2.Cifrado de red

Oracle nos proporciona encriptación e integridad de los datos mientras viajan por la red, mediante Oracle net services, se encripta de forma simétrica el tráfico de red de la base de datos.

Proporcionando protección contra 2 tipos de ataques

- Ataque de modificación de datos, cuando un intruso intercepta paquetes en el tráfico de la red, los altera y los retransmite.
- Ataque de repetición, retransmitir repetidamente un conjunto completo de datos válidos por ejemplo, si un intruso logra interceptar un retiro bancario por $10.000 pesos y retransmitirlo 10 veces lo que se recibiría serian $100.000.

8.3.Almacenamiento de las llaves de cifrado

El almacenamiento de las llaves de cifrado es uno de los aspectos más importantes y difíciles a tener en cuenta, ya que la llave debe ser accesible para las aplicaciones o usuarios autorizados y al mismo tiempo ser inaccesible para alguien que está intentando acceder maliciosamente a los datos cifrados.

8.3.1. Almacenamiento de las llaves en la base de datos

Almacenar llaves de cifrado en la base de datos no siempre impide que el administrador de la base de datos acceda a los datos cifrados ya que un administrador de base de datos con todos los privilegios puede acceder a las tablas que contienen llaves de cifrado.

Como protección adicional el código que realiza la encriptación puede ser encapsulado mediante la Utilidad WRAP lo que permite ofuscar este código, Una alternativa para encapsular los datos es tener una tabla separada en la que se almacene la llave de encriptación y ejecutar la utilidad WRAP al procedimiento que hace la llamada de las llaves guardadas en la tabla, como beneficio de esta alternativa, los usuarios que tienen acceso directo a la tabla no pueden ver los datos confidenciales cifrados ni pueden recuperar las llaves de encriptación para descifrar los datos.

El acceso a los datos descifrados se puede controlar a través de un procedimiento que seleccione la llave de cifrado y la transforme antes de que se pueda utilizar para descifrar los datos, en caso de un usuario que tenga permisos de lectura sobre la tabla la llave almacenada no podría llegar a descifrar los datos ya que esta se transforma antes de su uso.

8.3.2. Almacenamiento de llaves en el sistema operativo

Cuando se almacenan las llaves de cifrado en un archivo plano del sistema operativo puede realizar llamadas desde PL/SQL para recuperarlas, sin embargo, si almacena claves de esta forma los datos encriptados serán tan seguros como el sistema operativo lo sea.

8.4. Importancia de cambiar las llaves

Como buena práctica de seguridad se dicta que se debe cambiar periódicamente las llaves de cifrado, para los datos almacenados esto requiere descifrar periódicamente los datos y luego volver a cifrarlos con la nueva llave, es muy probable que al cambiar la llave de cifrado no se pueda acceder a los datos y/o afecte el rendimiento de la base de datos.

9. Auditoria

López (2010) define auditoria como "Un análisis pormenorizado de un sistema de información que permite descubrir, identificar y corregir vulnerabilidades en los activos que lo componen y en los procesos que se realizan" (p. 22) de esta forma evaluar si las políticas de seguridad implementadas en la base de datos cumplen con su objetivo o monitorear acciones sospechosas de un usuario.

Existen dos roles destinados para esta funcionalidad, AUDIT_ADMIN el cual puede configurar y administrar las políticas de auditoria, consultar y analizar los datos de dichas políticas, normalmente es asignado a administradores de seguridad, el otro rol es AUDIT _VIEWER el cual está destinado a consultar y analizar los datos de la auditoria, este rol se suele asignar a auditores externos.

Al crear y habilitar una política de auditoria esta comienza a recopilar los registros de inmediato, no es necesario establecer parámetros de inicialización para habilitar la auditoria como sucedía en versiones anteriores a 11g, además se pueden tener varias políticas de auditoria ejecutándose al mismo tiempo.

Las políticas pueden ser tan simples como auditar las actividades de un solo usuario o complejas usando condicionales llegando a monitorear desde un nivel muy específico a nivel de columnas auditando las columnas más relevantes que contengan información sensible o a nivel de eventos, por ejemplo, generando un correo electrónico al administrador de base de datos cuando se realice un cambio en una columna sensible en horas no laborales.

Se recomienda auditar las acciones de los usuarios en la base de datos ya que permite la rendición de cuentas de las acciones de cualquier usuario tomadas en un esquema, tabla o columna en particular, nos permite determinar que usuarios están ejecutando acciones inapropiadas en relación a sus roles, investigando dicha activada sospechosa. Además, nos puede ayudar a redefinir los privilegios en caso de encontrar un usuario que tenga más privilegios de los esperados para su rol.

Conclusiones

- Como mínimo para asegurar una base de datos Oracle se debe activar una función de contraseñas fuertes, un control de acceso mediante el uso de roles y perfiles además de activar la función de auditoria.
- Usar una función de contraseñas fuertes que proporciona Oracle para prevenir que las contraseñas de los usuarios sean decifradas o adivinadas de forma fácil.
- Usar roles facilita la administración de privilegios de una forma centralizada y rápida.
- Se debe tener en cuenta al momento de cifrar la base de datos que esta tenga los recursos suficientes para que no afecte su rendimiento.
- En caso de cifrar la base de datos se debe cambiar periódicamente la llave de encriptación teniendo en cuenta la afectación que tiene con respecto al rendimiento y disponibilidad de los datos.
- Se debe proteger las columnas de las tablas que contengan información confidencial mediante Oracle Virutal Database o Oracle Data Redaction.
- La función de auditoria se complementa con el control de acceso basado en roles ya que permite monitorear si las acciones de los usuarios están sobrepasando lo que en su proceso esta definido.

Bibliografía

Huey, P., & Jeloka, S. (2016). Oracle Database Security Guide.

López, P. A. (2010). Seguridad informática. Editex.

Matischak, D., & Fuller, M (2013). Oracle Database 11g: Administration Workshop I

Montoya, J. A., & Restrepo, Z. (2012). Gestión de identidades y control de acceso desde una perspectiva organizacional. Ingenierías.

Solarte, F. N. S., Rosero, E. R. E., & del Carmen Benavides, M. (2015). Metodología de análisis y evaluación de riesgos aplicados a la seguridad informática y de información bajo la norma ISO/IEC 27001. Revista Tecnológica-ESPOL.

Urbano, R. (2014). Oracle Database Administrator's Guide.