Administración de Bases de Datos (2023-24, Grado en Ingeniería Informática. Plan 2010 Todos los grupos y Grado en Matemáticas + Ingeniería Informática. Plan 2019 Todos los grupos)

Píldora de conocimiento VITAMINADA: SEGURIDAD AVANZADA

Esquema del ejercicio

---

1. Conéctate a la base de datos como system.

2. Ejecuta **todos** los pasos necesarios para crear un wallet de tipo FILE, tal y como hemos visto en clase y en los videos, para permitir implementar TDE (Transparent Data Encryption) sobre columnas de las tablas que seleccionemos después.

Hay que tener en cuenta que en el proceso de creación del wallet se ha de elegir un directorio en el que **Oracle tenga permisos en tu máquina concreta**. Por ejm, en el directorio 'C:\app\alumnos\admin\orcl\xdb_wallet' o cualquier otro directorio de Windows en el que aparezca el **usuario de Oracle en el SO** (el nombre de este usuario suele empezar con 'ORA_"). Si no sabes cómo comprobarlo, pregunta al profesor antes de continuar.

Es necesario que entiendas bien TDE y todos los pasos que realizas. De lo contrario, te resultará muy difícil avanzar en la práctica. Para ello ve a la parte correspondiente de la documentación proporcionada en clase y estudiála antes de empezar. Encontrarás los pasos descritos en secuencia y explicados.

3. Todo el trabajo de tu proyecto LIFEFIT debería estar o estará en un espacio de tablas aparte. En el peor de los casos puede estar en el tablespace USERS. Asumiremos en adelante que usamos el esquema en el que estás desarrollando tu trabajo en grupo. Si no, no pasa nada, utiliza un esquema (usuario que tendrás que crear) de ejemplo, el que quieras. Más adelante, se volcará lo aquí aprendido al esquema final de LIFEFIT .

4. Usar una o varias tablas de tu trabajo en grupo susceptible de precisar que sus datos estén cifrados. Si no tuvieras nada creado en el momento de la realización de esta práctica, puedes crearte un par de tablas donde una de ellas fueran, por ejemplo, los clientes. Y, por supuesto, introducir algunos datos de ejemplo. Si tienes que crear estas tablas para la práctica, lee el paso siguiente ANTES de hacerlo.

5. Parece obvio que en esas tablas habrá una serie de columnas que almacenan información sensible. Identifícalas y haz que estén siempre cifradas en disco. PARA ESTA PRÁCTICA, ASEGURATE QUE HAYA AL MENOS UNA COLUMNA DE TEXTO NO CIFRADA Y AL MENOS OTRA CIFRADA con objeto de poder hacer comprobaciones en los siguientes pasos.

6. Una vez le has ordenado a Oracle que columnas deben de ir cifradas, comprueba que los cambios son efectivos mediante la consulta de la vista del diccionario de datos adecuada.

7. Prueba a insertar varias filas en una de esas tablas (y en todas aquellas tablas que sea necesario). A continuación, puedes forzar a Oracle a que haga un flush de todos los buffers a disco mediante la instrucción:

alter system flush buffer_cache;

Comprueba a continuación el contenido del fichero que contiene el tablespace con estos datos. Ese fichero lo podremos encontrar en el directorio en el que hayamos creado el tablespace en el que se encuentra la tabla que estamos utilizando.

No es necesario conocer el formato de dicho fichero. Simplemente tener en cuenta que los datos no cifrados aparecerán en claro. Y podemos hacer un buscar y los encontraríamos. Pero, ¿y si hacemos lo mismo con los que hemos decidido que se almacenen cifrados?

La manera más cómoda es utilizar una herramienta que extraiga los strings legibles. E.g.: https://docs.microsoft.com/en-us/sysinternals/downloads/strings

Si el fichero no es muy grande también se puede utilizar un editor (e.g. notepad) de texto para abrirlo y realizar búsquedas. Responde a las siguientes preguntas:

- ¿Se pueden apreciar en el fichero los datos escritos? ¿Por qué?

8. Vamos ahora a aplicar políticas de autorización más concretas mediante VPD. Quizás quieras consultar previamente la documentación de seguridad para refrescar los conceptos de VPD.

Supongamos que vamos a permitir a los clientes acceder a la BD y consultar sus datos. Si un cliente accede, sólo tendrá disponibles sus datos. Para ello, vamos a asumir que una de las columnas de la tabla cliente almacena su usuario de conexión a la BD (añade esta columna a la tabla si no la tiene ya). En el ejemplo a continuación asumimos que esta columna se denomina usuario, pero puede denominarse como desees.

Para ello, necesitaremos primero una función que forme los predicados de la cláusula WHERE. La crearemos en el esquema (con copiar y pegar, por ejemplo) en el que se encuentran las tablas:

```
create or replace function vpd_function(p_schema varchar2, p_obj varchar2)
  Return varchar2
is
  Vusuario VARCHAR2(100);
Begin
  Vusuario := SYS_CONTEXT('userenv', 'SESSION_USER');
  return 'UPPER(usuario) = ''' || Vusuario || '''';
End;
/



-- userenv = El contexto de aplicación

-- p_obj es el nombre de la tabla o vista al cual se le aplicará la política

-- p_schema es el schema en el que se encuentra dicha tabla o vista.

*/
```

9. Crearemos un usuario (cuyo nombre debe estar previamente presente en el campo user_name de alguna fila) de forma que podamos probar la política. Comprobaremos, que ese usuario, al conectarse, puede ver todos los datos de la tabla cliente(si no puede inicialmente, piensa por qué y soluciónalo).

A partir de ahora, además, ten en cuenta con que usuario vas a hacer cada cosa. Para crear, cancelar o borrar políticas se hará desde un usuario con permisos de DBA (lo haremos así por facilidad). Para probarlas se hará con el nuevo usuario que hemos creado precisamente para eso. En resumen, cada vez que se solicite llevar a cabo una acción, incluso si el enunciado no lo especifica, no debes dudar acerca de cual es el usuario que ha de hacerlo. Si dudas, pregunta al profesor.

Recordar también que siempre que creemos usuarios de prueba será asignándole los permisos MINIMOS necesarios para lo que queremos hacer (ni uno más).

Añadiremos la política (consulta las transparencias) a la tabla CLIENTES(desde un usuario con el role de DBA). Y después comprobaremos que ocurre después de añadir la política. Una aclaración, al añadir una política, ésta se encuentra activa por defecto.

Si en algún momento necesitas **desactivar** la política puedes usar:

```
begin

 DBMS_RLS.ENABLE_POLICY (        object_schema=>'el_nombre_del_esquema_en_el_que_está_la_tabla',    object_name=>'el_nombre_de_tu_tabla',

policy_name=>'nombre_politica', enable=>false);

end;
```

Si te has equivocado y quieres **borrar** y volver a crear la política:

```
begin

dbms_rls.drop_policy (

  object_schema=>'el_esquema',

  object_name=>'la_tabla',

  policy_name=>'el_nombre_de_la_politica' );

end;
```

10. ¿Qué ocurre cuando nos conectamos con ese usuario existente en la tabla CLIENTES y realizamos un select de todo? ¿Y cuando lo hace el propietario de la tabla?

11. Utilizando VPD, también podemos aplicar políticas sobre columnas, en lugar de sobre vistas o tablas enteras. Continuando con nuestro ejemplo, imaginemos que queremos permitir a estos clientes consultar todos los datos de la tabla excepto cuando también se solicita una columna determinada (ej. Telefono), en cuyo caso queremos que se muestren sólo los datos del usuario. Investiga en la documentación la función que ya hemos utilizado del paquete *dbms_rls* para añadir una política **nueva** (*dbms_rls.add_policy*). ¿Qué cambios deberíamos hacer para lograr nuestro objetivo? Tip: Desactiva previamente la política anterior para no tener conflictos en los resultados.

12. Qué desventajas pueden llegar a tener este tipo de control de acceso más específico? Si no encuentras la respuesta discútelo con el profesor.

13. A continuación se detallan los distintos tipos de usuarios que existirán en la BD. Para la correcta gestión de ellos se deberán crear los roles, usuarios, vistas, permisos, etc. que sean precisos. Se recomienda comenzar por el usuario Administrador porque resultará de utilidad para el resto de usuarios, roles y privilegios.

Los usuarios o grupos de usuarios serán los siguientes:

- **Administrador**: Control total y dueño de todo el esquema.
- **Gerente**: Gestión completa de centros, entrenadores y asignación de entrenadores a centros. Gestión alta/baja de clientes (únicamente datos administrativos).
- **Entrenador deporte y fitness**: Gestión completa de ejercicios, rutinas y asignación de ejercicios a rutinas. Asignación de rutinas a clientes, calendario de las mismas, estado del cliente y comprobar objetivos. Puede ver los vídeos de los cliente y posible retroalimentación por mensajería. Gestión de calendario de entrenadores y citas con clientes.
- **Entrenador nutrición**: Gestión de dietas y asignación a clientes. (un usuario)
- **Cliente**: Informa sobre su progreso (entrenamiento, valoración, indicadores de salud, objetivos). Sube vídeos de su progreso.
- Youtube?

- WhatsApp?

Los usuarios Administrador si llegaran a crearse más adelante Youtube y Whatsapp serán únicos en la base de datos. Sin embargo, para gerentes, entrenadores y clientes deberemos discriminar qué individuo es el que está trabajando, esto puede hacerse creando un usuario particular para cada uno de ellos o bien mantener un usuario común para cada grupo y añadiendo algún tipo de mecanismo que permita discriminar qué usuario es el que está conectado (por ejemplo controlando variables o el contexto de la sesión, usando tablas temporales basadas en sesiones -on commit preserve rows- que almacenen el identificador particular del entrenador/cliente que está conectado, etc).

Última modificación: martes, 12 de marzo de 2024, 13:40

◄ Entrega de la práctica de seguridad (Grupo A1)

Saltar a...

Entrega de la práctica (Grupo A1) ►