UNICIENCIA 22
pp. 131-134
2008

# VULNERABILIDAD DE SISTEMAS GESTORES DE BASES DE DATOS

***Johnny Villalobos Murillo***
Facultad de Ciencias Exactas y Naturales, Escuela de Informática,
Universidad Nacional de Costa Rica
Heredia. Costa Rica
E-Mail: jvillalo@una.ac.cr

## RESUMEN

Existe una constante preocupación por la seguridad de las bases de datos; muchas veces la seguridad se ve afectada por la configuración de los procesos de conexión. En este ensayo se estudia como se configuran las conexiones hacia una base de datos, explicando los posibles errores y se proporcionan recomendaciones para disminuir el riesgo asociado a estos procesos.

**Palabras claves:** Bases de datos, seguridad.

## ABSTRACT

Exist a constant worry for Database Security, many times the security is affected for the security configuration connection, this paper will study how make security connection to a Database, explaining common mistakes and to give recommendations for keep low the associated risk to these process.

**Keywords:** Database, security.

## 1. CONEXIÓN A LA BASE DE DATOS

La instalación de un Sistema Gestor de Base de Datos (SGBD) es un procedimiento que requiere configurar un conjunto de parámetros, algunos de ellos sirven para especificar los diferentes procesos que el SGBD utilizará, otros están relacionados con áreas de memoria para realizar las transacciones y soluciones a consultas de las tablas de los modelos de datos y, finalmente, los parámetros requeridos por los procesos de conexiones desde los puntos remotos y locales a la base de datos.

El SGBD Oracle facilita la configuración de conexiones mediante un programa o unidad de procesos llamada SQL net, que a su vez provee una serie de funciones o programas, entre las cuales se encuentra el Listener. Cuando se instala el SGBD Oracle, se debe configurar el Listener del servidor y en caso de los clientes, cada uno de ellos debe contar en su computador con una serie de herramientas de cliente, entre las cuales el Listener no debe faltar y debe configurarse también para lograr conectarse al servidor de la base de datos.

En las instalaciones del cliente y del servidor se debe configurar el Listener y además se requiere administrarlo adecuadamente.

## 2. CONFIGURAR Y CONECTARSE

El propósito fundamental del Listener es crear un archivo de parámetros llamado TNSNAMES.ORA. Este archivo contiene los parámetros necesarios para establecer conexiones con la base de datos. Suponga que un cliente necesita conectarse a la base de datos de la compañía. Debido a que el cliente cuenta con equipo portátil, se le instala en su equipo las herramientas cliente, entre las cuales se encuentra el SQL net y el Listener. Posterior a las herramientas, se configura el Listener, y se proporcionan los siguientes parámetros: protocolo de comunicación que se emplea, nombre del servidor

Recibido 5/7/2006 / Aprobado 30/11/2006

donde se encuentra la base de datos, de no conocer el nombre del servidor, específicamente para acceso remoto, se proporciona la dirección IP donde se ubica, otro parámetro que debe indicarse es el número del puerto de comunicación. Una vez suministrada esta información, el Listener creará el archivo de parámetros con el siguiente contenido:

```
Cliente=(Description=
(Address=
        (Protocol=TCP)
        (Host=185.145.140.50)
        (Port=1521)
        (SID=Compañía)
))
```

En este caso la conexión será conocida como Cliente, y sirve para acceder a la base de datos llamada compañía, que reside en un servidor de base de datos localizado en la dirección IP suministrada.

Se pueden crear varias conexiones dentro del archivo TNSNAMES, una para cada base de datos que se desea acceder.

Debe aclararse en este momento que el archivo no forma parte de los objetos de la base de datos, es un archivo de texto que puede ser localizado en cualquier parte del directorio del disco del cliente. Debido a esto puede ser vulnerable si no se toman las precauciones necesarias.

## 3. OMISIONES DE SEGURIDAD

El archivo TNSNAMES, al ser un archivo de texto, el cual puede ser abierto con cualquier editor de texto, presenta un gran riesgo. Para disminuir el riesgo de que alguien conozca esta información, el Listener permite adicionar una clave al archivo y realizar cambios periódicos de claves. Lamentablemente muchas veces no se establece una clave o no se conoce sobre el Listener, por lo que el archivo queda vulnerable. El Listener puede ser invocado mediante el programa LSNRCTL.exe, y se puede crear la clave mediante el comando:

*Lsnrctl set password nueva_clave*

El Listener provee otras funciones que permiten administrarlo, como por ejemplo: activar, desactivar, cambiar clave, generar bitácora.

Estas funciones deben ser ejecutadas adecuadamente con una buena práctica de administración de seguridad, ya que como todo archivo encriptado o protegido con clave, se podría atacar mediante la fuerza bruta que tarde o temprano obtendría la clave. Se recomienda hacer cambios de clave en forma periódica como control para minimizar el riesgo.

### 3.1 Propagación del riesgo

En la instalación del SGBD, se crean dos usuarios administradores, SYS y SYSTEM, con claves ORACLE y MANAGER, respectivamente. Se recomienda que se cambie la clave una vez realizada la instalación, pero en algunos casos no se hace. El problema es que un atacante con poco conocimiento sobre el SGBD puede instalar herramientas de cliente en su máquina, y lograr descifrar el contenido del archivo de parámetros, así puede intentar conectarse a la base de datos como administrador; si puede acceder el daño sería irreparable, ya que como administrador puede eliminar y cambiar derechos, información y objetos. Se ha detectado que generalmente se cambia la clave de SYSTEM, pero SYS permanece igual. El problema se agudiza más si ataca por medio de SYS. En este caso, el SGBD proporciona un programa de administración llamado SERVER MANAGER; en este programa se puede conectar como SYS, mediante un alias llamado INTERNAL, el cual no requiere clave. Si no se toman las medidas respectivas de cambios de clave, las consecuencias del ataque serán devastadoras.

Veamos un ejemplo de ataque.

Paso 1. Obtener el nombre de la base de datos, la dirección IP y el puerto de comunicación.
Paso 2. Configurar el Listener desde la máquina del atacante.
Paso 3. Establecer la conexión.
Paso 4. Invocar el programa Server Manager.
Paso 5. Conectarse como Internal.
Paso 6. Cambiar las claves de los administradores.

➢ Alter user system identified by atacante
*No hay nada que hacer.*

## 4. VULNERABILIDAD DEL PROTOCOLO TNS

Oracle utiliza el protocolo TNS (sustrato de red transparente), en el cual se establecen los mecanismos y regulaciones para enviar paquetes de información entre el cliente y el servidor. Este protocolo no es del todo seguro, o tan seguro como deseamos, debido a esto paquetes de datos pueden ser atrapados en la red. Si se desea atacar el servidor, se puede alterar el tamaño del paquete o enviar un paquete espía, el cual puede tener un tamaño mínimo o un tamaño muy grande hacia el servidor.

El servidor no podrá procesar o entender lo que está recibiendo, lo que causaría un comportamiento no deseado, una respuesta inesperada, y en ese momento el servidor enviaría paquetes indicando errores, pero lamentablemente también enviaría información sobre él, el usuario o detalles que en manos del atacante, comprometen la seguridad.

## 5. OTROS PROBLEMAS CON LOS SGBD

Un sistema gestor de base de datos muy utilizado mundialmente, el SQL Server de Microsoft (SQLS), también tiene sus puntos débiles. Un aspecto que llama la atención es su sistema de encriptación, la forma de autenticar las conexiones y su almacenamiento en varias localizaciones, con el propósito de mejorar sus procesos, a su vez aumenta el riesgo y en este caso el impacto es fatal. Al igual que otros SGBD, la captura de una clave de administrador es, a nuestro juicio, el peor de los escenarios, ¿qué no podría hacerse?

El SQLS provee una serie de procedimientos para la administración de claves, claves de paquetes y claves de replicación, que por su naturaleza y utilización, pueden ser víctima a posibles vulnerabilidades, ya que existe facilidad para obtener las claves y descifrarlas.

### 5.1 Evaluando privilegios

Conocer los privilegios de los usuarios es para los atacantes otra posibilidad de producir daños en el SGBD. Una manera de hacerlo es colocando un programa espía (Caballo de Troya), otra consiste en crear un procedimiento almacenado con las sentencias necesarias para obtener la información necesaria, y crear un ataque para negarle a los usuarios la posibilidad de administración de sus objetos.

El problema con los privilegios puede ser muy grave, es posible heredar privilegios a los usuarios mediante el otorgamiento de permisos de ejecución a procedimientos almacenados. Un usuario que no tiene acceso directo a una tabla, puede tenerlo si se le otorga el privilegio de ejecución de un procedimiento que produzca transacciones (insert, update, delete) en la tabla. Esto podría ser una fuente de ataque.

### 5.2 Negación de Servicios

Independiente del SGBD, el ataque de negación de servicios es uno de los más frecuentes que se reciben en servidores a los que se logra penetrar. Consiste en consumir los recursos del SGBD, mediante procesos que son aplicados en ciclos infinitos, que se encargan de saturar las áreas de memoria. Estos procesos aparentemente no son dañinos en su forma, por ejemplo:

```
Create table t1 ( x int);
Exec (‘insert into t1 select ``x`` from t1
) while 1=1 exec (‘insert into t1 select *
from t1).
```

### 5.3 SQL injection

Suponiendo que se cuenta con medidas adecuadas de seguridad, para los problemas anteriores, existen otros peligros de ataque, por ejemplo, la inyección de estatutos SQL. Este ataque no es directo sobre la base de datos, se genera mediante las aplicaciones, programas, específicamente las aplicaciones WEB.

Los diseños de interfaces para los usuarios deben tomar en cuenta medidas respecto a esto. Un ejemplo clásico es el envío de una comilla ( ‘ ) en un input box de un formulario WEB. El conector de la base de datos abre la conexión, y ejecuta la instrucción, pensando que es válida, pasa y se ejecuta, pero el problema es que los procesos de análisis en el SGBD no saben que hacer, por lo que envían una respuesta de error, y en ese momento

proporcionan información clave. Este es el problema, la información del mensaje de error.

Otro ataque de este tipo es el que consiste en enviar sentencias SQL válidas, pero con lógica de ataque. Específicamente en la sección de selección de la sentencia.

Select .....
From .....
Where <condición válida>
or 'a' = 'a' < condición maliciosa >

La condición maliciosa es analizada con el operador "or", si no cumple la primera condición, la segunda sí (aquí está el problema). Existen muchos otros ejemplos más, pero todos se basan prácticamente en la misma lógica.

## 6. RECOMENDACIONES

Considere una buena política de seguridad, basada en prevención preferiblemente.

Asegure la configuración.
Cambio constante de claves.
Aplicación de software para monitoreo.
Buen diseño de interfaces para evitar la inyección.
Buenos procesos de encriptación.
Monitoreo constante.
Programas de auditoría.
Aplicación de parches.

Es necesario estar actualizado con los nuevos tipos de ataque, los administradores de las bases de datos o los administradores de seguridad deben enfocarse en la creación de estos controles, con el fin de lograr una seguridad razonable de la seguridad de las bases de datos.

Según un artículo publicado por Oracle sobre seguridad de bases de datos, la administración de bases de datos debe considerar los siguientes pasos para un protocolo de seguridad:

1. Instale solo lo requerido.
2. Cierre todas las cuentas expiradas.
3. Habilite la protección del Diccionario de Datos.
4. Otorgue sólo los privilegios necesarios.
5. Autentique correctamente sus clientes.
6. Restrinja el acceso al sistema operativo.
7. Restrinja el acceso a la red.
8. Proteja y encripte.
9. Use Firewall adecuadamente.
10. Instale parches.
11. Cambie los parámetros por defecto.

Y otros más. Pero lo principal es prevenir.

## REFERENCIAS

*Microsoft SQL Server Security*. Security in SQL 2005, www.microsoft.com/sql/technologies/security/default.mspx

*Oracle Database Security Check List*. Oracle White Paper.

*Oracle Database Security*. Oracle Press, november 2004.

*Protecting Databases*. Application Security, Inc. WWW.APPSECINC.COM.