Capítulo 4
Seguridad y gestión de los usuarios

1. Introducción

La seguridad de los datos del sistema informático de la empresa no es solo responsabilidad del RSSI (Responsable de la Seguridad de los Sistemas de Información). Los técnicos y las buenas prácticas necesarias para protegerse deben ser conocidas y aplicadas por cualquier persona. Los elementos de la cadena de aplicaciones son interdependientes; la fortaleza de esta cadena va a depender de la solidez del eslabón más débil. Una prueba no efectuada por la aplicación, una cuenta de usuario que tenga demasiados derechos sobre la base de datos, una actualización de seguridad no efectuada o simplemente una torpeza en el servidor de producción pueden ser la causa del robo o deterioro de los números de tarjetas bancarias de sus clientes, por ejemplo. Las consecuencias pueden ser desastrosas tanto financieramente como en términos de imagen para la empresa. Si bien la seguridad absoluta no existe, vamos a tratar de mostrar todas las prácticas necesarias para minimizar los riesgos en este capítulo.

2. Securización del servidor MySQL

Las cuestiones relacionadas con la seguridad deben plantearse desde la instalación del servidor de base de datos. En la filosofía de MySQL, el usuario debe poder descargar, instalar y utilizar el SGBDR (Sistema de Gestión de Bases de Datos Relacionales) sin ninguna dificultad. Esta simplicidad es uno de los puntos fuertes de MySQL, ya que, por un lado, esto permitió democratizar el uso de una base de datos haciendo accesible este universo (muchos desarrolladores han experimentado las bases de datos mediante MySQL) y, por otro, esto da la posibilidad de probar fácilmente una aplicación. El inconveniente principal es que la instalación predeterminada es muy permisiva, sobre todo en materia de seguridad...

2.1 Securización de la instalación

El objetivo de esta sección no es volver a la instalación del servidor que se detalla en el capítulo Instalación del servidor, sino recordar algunos puntos cruciales en materia de seguridad, que deberá adaptar en función del tipo de instalación y de su sistema operativo.

2.1.1 Controlar los permisos

Debemos crear un grupo y un usuario dedicado para iniciar la instancia `mysqld`:

```
$ groupadd mysql
$ useradd -g mysql mysql
```

El directorio de datos contiene información sensible; por tanto, debe preservarse de actos dolosos o de la visualización por personas no autorizadas.

```
$ cd /usr/local/mysql/
$ chown -R root .
$ chown -R mysql data
$ chgrp -R mysql .
```

`mysqld` no debe, en ningún caso, ejecutarse con el administrador del sistema `root` en UNIX (o `administrador` con MS Windows). Un usuario con, por ejemplo, el derecho `FILE` puede crear archivos como `root`.

2.1.2 Poner contraseña a la cuenta root

Es el superusuario de MySQL (no confundir con el administrador del sistema `root` bajo UNIX). Cuenta, por lo tanto, con todos los derechos sobre el servidor. Debe protegerse mediante contraseña. Esto es válido para cualquier sistema operativo.

```
$ mysql -u root # conexión al servidor con el usuario root sin
contraseña
mysql> SET PASSWORD FOR root@localhost = password('m0T2p4ss3'); /*
el comando set password permite implementar una contraseña para una
cuenta de usuario */
```

Como veremos un poco más adelante, un usuario MySQL está compuesto por un nombre, «user», y el nombre del equipo desde el cual el usuario se conecta, «host». Podemos tener una cuenta `'root ''@'localhost'`, que permite conectarse al servidor con el usuario `root`, siempre que el cliente esté en el mismo equipo que el servidor MySQL (local), y, por ejemplo, una cuenta `'root'@'123.45.67.89'`, que permite conectarse como `root` desde el equipo que tiene como dirección IP 123.45.67.89. Son dos cuentas diferentes, que pueden tener derechos y contraseñas diferentes. Esta posibilidad que ofrece el servidor permite una gestión de derechos muy fina. Sin embargo, por experiencia, recomendamos solo tener una ocurrencia por usuario, en aras de simplificar la gestión de los derechos y, por lo tanto, disminuir el riesgo de errores. En otras palabras, guardemos solo el usuario `root'@'localhost'`. Si necesitamos administrar el servidor en remoto, en lugar de tener `'root'@'%'` (que permite conectarse con el usuario root desde cualquier equipo), usaremos el protocolo de comunicación segura SSH o equivalente.

```
mysql> SELECT user, host, password FROM mysql.user WHERE user = 'root' \G
*************************** 1. row ***************************
user: root
host: localhost
password: *228F5395471FEFD9B0BB291954D2189384329691
*************************** 2. row ***************************
user: root
host: 123.456.78.9
password: *63D85DCA15EAFFC58C908FD2FAE50CCBC60C4EA2

mysql> DROP USER 'root'@'123.456.78.9' ;

mysql> SELECT user, host, password FROM mysql.user WHERE user = 'root' \G
*************************** 1. row ***************************
```

```
user: root
host: localhost
password: *228F5395471FEFD9B0BB291954D2189384329691
```

Observación

Podemos renombrar la cuenta de administrador para que sea más difícil de encontrar por una persona malintencionada: `RENAME USER 'root'@'localhost' TO 'leader'@'localhost';`

2.1.3 Eliminar las cuentas anónimas

Para facilitar el manejo del software, podemos encontrar durante la instalación del servidor una cuenta anónima, es decir, una cuenta que tiene el campo `user` vacío. Esta permite conectarse al servidor con cualquier usuario, en local y sin contraseña. No hace falta argumentar más para comprender que, en un servidor de producción, ese usuario no debe existir. Hay que suprimirlo. Este principio debe extenderse a todas las cuentas no utilizadas; así eliminamos muchos problemas potenciales.

```
mysql> SELECT user, host, password FROM mysql.user WHERE user = '' \G
*************************** 1. row ***************************
user:
host: localhost
password:

mysql> DROP USER ''@localhost;

mysql> SELECT user, host, password FROM mysql.user WHERE user = '' \G
```

Observación

A partir de MySQL 5.7, no se crean cuentas anónimas durante la instalación del servidor, lo que sí sucedía en todas las versiones anteriores.

2.1.4 Eliminar el esquema test

El esquema `test` es creado por MySQL durante la instalación para todas las versiones salvo 5.7, y todos los usuarios tienen derecho a acceder en lectura y escritura sin que sea necesario especificar de forma explícita los derechos adecuados. Un usuario malintencionado podría llenar de datos el disco duro y así impedir el buen funcionamiento de nuestra aplicación.

```
mysql> SHOW SCHEMAS;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| test               |
+--------------------+

mysql> DROP SCHEMA test;

mysql> SHOW SCHEMAS;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
+--------------------+
```

Cabe señalar que esta recomendación se aplica también a todo esquema `test` que se cree a continuación, y también a cualquier esquema que comience por la palabra `test_`.

2.1.5 Securizar la instalación con la herramienta mysql_secure_installation

La aplicación de estas recomendaciones puede efectuarse con la herramienta `mysql_secure_installation`. Esta herramienta no está disponible en MS Windows; sin embargo, el instalador gráfico también ofrece la posibilidad de securizar la instalación.

```
$ mysql_secure_installation
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL
MySQL SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current
password for the root user. If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
```

2.2 Utilización de SSL

Los datos viajan en texto sin cifrar entre el cliente y el servidor MySQL. A partir de MySQL 4, podemos encriptarlos usando el protocolo SSL (*Secure Sockets Layer*), que permite garantizar la seguridad, integridad y confidencialidad de los datos intercambiados. Sin embargo, las operaciones adicionales tienen un impacto significativo sobre el rendimiento, reduciéndolo en cerca del 30 %.

2.2.1 Las opciones

SSL ofrece una lista de opciones. Existe más información disponible en el sitio de MySQL: https://dev.mysql.com/doc/refman/5.6/en/secure-connection-options.html. He aquí una breve descripción:

- `ssl` permite al servidor autorizar conexiones SSL y al cliente conectarse utilizando este protocolo.
- `ssl-ca` permite especificar el archivo que contiene el certificado de autoridad (CA).
- `ssl-capath` es el directorio de archivos que contienen los certificados de autoridad SSL.
- `ssl-cert` es el nombre del certificado SSL que debe utilizarse para establecer una conexión segura.
- `ssl-key` es el nombre del archivo de clave SSL que debe utilizarse para establecer una conexión segura.
- `ssl-cipher` es la lista de cifrados (*cipher*) autorizados que deben utilizarse con SSL.

2.2.2 Las principales etapas

Para poner en funcionamiento SSL, el servidor debe haber sido compilado con soporte para SSL, utilizando la opción `--with-openssl` o `--with-yassl`. Se requieren algunas etapas preparatorias antes de poder cifrar las comunicaciones. Las siguientes son las principales. Encontraremos más información en la documentación oficial de MySQL en la dirección:
https://dev.mysql.com/doc/refman/5.6/en/secure-connection-options.html