CSIT115 Data Management and Security

User Management

Dr Janusz R. Getta

School of Computing and Information Technology -
University of Wollongong

User Management

Outline

- Basic security guidelines
- Securing passwords
- Adding/removing user accounts
- Setting accounts resource limits
- Locking/unlocking accounts

Basic security guidelines

Do not ever give anyone (except `root` account) access to the user table in the `mysql` database !

Use `GRANT` and `REVOKE` statements to control access to MySQL and do not grant more privileges than necessary

Try

```
The first connection as 'root' user

mysql -u root
```

If it works all right then you must set password for `root` user !

Use the `SHOW GRANTS` statement to check which accounts have access to what

Then use the `REVOKE` statement to remove those privileges that are not necessary

Basic security guidelines

Do not store cleartext passwords in your database

Instead, use hashing functions like `sha2(), sha1(), md5()` or some other one-way hashing function and store the hash value

Do not choose passwords from dictionaries because special programs exist to break passwords

Invest in a firewall to protect you from at least 50% of all types of exploits in any software

Applications that access MySQL should not trust any data entered by users, and should be written using proper defensive programming techniques

Do not transmit plain (unencrypted) data over the Internet because such information is accessible to everyone who has the time and ability to intercept it and use it for their own purposes

Instead, use an encrypted protocol such as SSL or SSH

User Management

Outline

Securing passwords

Using `-pyour_password` or `--password=your_password` option on the command line like

Connecting as 'root' user with a visible password

```
mysql -u root -proot
```

is convenient but ... it is extremely insecure !

Using the `-p` or `--password` option on the command line with no password value like

Connecting as 'root' user with invisible password

```
mysql -u root -p
```

is less convenient but it is more secure

Use the `mysql_config_editor` utility to store authentication credentials in an encrypted login path file named `.mylogin.cnf`.

The file can be read later by MySQL client programs

Securing passwords

Store your password in a file with system variables in a section `[client]`

```
[client]
password=your_password
```
System veriables

To keep the password safe, the file should not be accessible to anyone but yourself

Store your password in the `MYSQL_PWD` environment variable

This method of specifying your MySQL password must be considered extremely insecure and should not be used.

User Management

Outline

- Basic security guidelines
- Securing passwords
- Adding/removing user accounts
- Setting accounts resource limits
- Locking/unlocking accounts

Adding/removing user accounts

Creating a new user

Creating a new user

```
CREATE USER jamesb IDENTIFIED BY 'mi6';
```

Altering a user

Altering a password of a user

```
ALTER USER jamesb IDENTIFIED BY 'cia';
```

Listing the users

Listing all users

```
SELECT USER FROM mysql.user;
```

Contents of mysql.user

```
+-----------+
| USER      |
+-----------+
| jamesb    |
| csit115   |
| mysql.sys |
| root      |
+-----------+
```

Adding/removing user accounts

Dropping a user

Dropping a user

```
DROP USER jamesb;
```

User name is up to 32 characters long

User account may have a password

Accounts instead of password may have authentication plugins that implement external authentication method

User name and passwords are stored in `mysql.user` table

Passwords stored in `mysql.user` table are encrypted using plugin-specific algorithm

When a user connects to the server there is an initial authentication step in which it provides a password that have a hash value equal to hashed password stored in `mysql.user table`

Adding/removing user accounts

After a user connects it can (depending on sufficient privileges) set or change password

When connecting a password is either provided in a command line or it is entered interactively during a login process

```
Connecting as 'csit115' user with a visible password ('csit115') and defult database used 'csit115'
mysql –u csit115 –pcsit115 csit115
```

```
Connecting as 'csit115' user with a visible password ('csit115')
mysql –u csit115 –p csit115
```

User Management

Outline

- Basic security guidelines
- Securing passwords
- Adding/removing user accounts
- Setting accounts resource limits
- Locking/unlocking accounts

Setting account resource limits

It is possible to set the limits for individual accounts on use of the following server resources:

- total number of queries an account can issue per hour (`MAX_QUERIES_PER_HOUR`)
- total number of updates an account can issue per hour (`MAX_UPDATES_PER_HOUR`)
- total number of times an account can connect to the server per hour (`MAX_CONNECTIONS_PER_HOUR`)
- total number of simultaneous connections to the server by an account (`MAX_USER_CONNECTIONS`)

Creating a user with a resource limit

```
CREATE USER jamesb IDENTIFIED BY 'mi6' WITH MAX_USER_CONNECTIONS 2;
```

Adding a resource limit

```
ALTER USER harryp WITH MAX_QUERIES_PER_HOUR 100;
```

Adding a resource limit

```
ALTER USER robinh WITH MAX_USER_CONNECTIONS 0;
```

Adding 3 resource limits

```
ALTER USER alcapone WITH MAX_QUERIES_PER_HOUR 20
                         MAX_UPDATES_PER_HOUR 10 MAX_CONNECTIONS_PER_HOUR 5;
```

Setting account resource limits

The server stores resource limits for an account in `mysql.user` table in a row corresponding to the account

Database server counts the number of times each account uses the resources

If an account reaches its limit on number of connections within the last hour, the server rejects further connections for the account until that hour is up

Similarly, if the account reaches its limit on the number of queries or updates, the server rejects further queries or updates until the hour is up

In all such cases, the server issues appropriate error messages

To reset the current counts to zero for all accounts, dba issues a `FLUSH USER_RESOURCES` statement

Setting account resource limits

The counts for an individual account can be reset to zero by setting any of its limits again

Per-hour counter resets do not affect `MAX_USER_CONNECTIONS` limit

All counts begin at zero when the server starts and the counts do not carry over through server restarts

User Management

Outline

- Basic security guidelines
- Securing passwords
- Adding/removing user accounts
- Setting accounts resource limits
- Locking/unlocking accounts

Locking and unlocking accounts

An account can be locked immediately after creation (`CREATE USER`) or at any time after it creation (`ALTER USER`)

Creating a user with a locked account

```
CREATE USER jamesb IDENTIFIED BY 'mi6' ACCOUNT LOCK;
```

Locking an account

```
ALTER USER harryp ACCOUNT LOCK;
```

Unlocking an account

```
ALTER user harryp ACCOUNT UNLOCK;
```

Account locking state is recorded in the `account_locked` column of the `mysql.user` table

References

MySQL 5.7 Reference Manual, 7.1 General Security Issues