CSIT115 Data Management and Security

CSIT882 Data Management Systems

User Management

Subject Coordinators: Dr Chen Chen, Dr Thanh Le

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 a user table in a `mysql` database !

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

Try

```
mysql –u root
```

The first connection as 'root' user

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

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

Then, use the `REVOKE` statement to revoke the privileges, that are not necessary

Basic security guidelines

Do not store the cleartext passwords in your database

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

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

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

Applications, that access MySQL should not trust any data entered by the users, and it should be written using the 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 protocols such as SSL or SSH

User Management

Outline

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

Securing passwords

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

```
mysql -u root -proot
```
Connecting as 'root' user with a visible password

is convenient way to provide a password but ... it is extremely insecure !

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

```
mysql -u root -p
```
Connecting as 'root' user with invisible password

is less convenient and 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

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

Creating a new user

Altering a user

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

Altering a password of a user

Listing the users

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

Listing all users

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

Contents of mysql.user

In HTML view press 'p' to see the lecture notes

Adding/removing user accounts

Dropping a user

Dropping a user

```
DROP USER jamesb;
```

A user name is up to 32 characters long

A user account may have a password

The accounts instead of a password may have an authentication plugins, that implements the external authentication methods

The user names and passwords are stored in a relational table `mysql.user`

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

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

Adding/removing user accounts

After a successful connections a user can, depending on the sufficient privileges, set or change a 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 –pcsit115
```

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;
```

In HTML view press 'p' to see the lecture notes

Setting account resource limits

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

A database server counts the number of times each account uses each resource

If an account reaches its limit on number of connections within the last hour, the database 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 the 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, a database administrator 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 its creation (`CREATE USER`) or at any time after its 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;
```

If an account is locked then its state is recorded in an `account_locked` column of a relational table `mysql.user`

References

MySQL 8.0. Reference Manual, 6.2 Access Control and Account Management