A Case in Security on Relational Database Management Systems; MySQL

Prepared by: Jose H. Garcia

Introduction

There are two main security concerns in RDBMS:

- A RDBMS must protect the data stored in any database. Access to data requires users to have privileges per object. Objects can be tables, columns, etc. within the database as well as typical commands such as SQL statements (SELECT, UPDATE, DELETE, INSERT, etc.) or administrative commands such as GRANT, REVOKE, KILL, etc.
- Typically, RDBMS are client-server based therefore communications over the network are insecure.

11/29/01 2

Topics of Discussion

How MySQL takes care of the two main aspects in RDBMS security:

- MySQL implements security for user's access to data based on a policy defined in the control database (mysql).
- MySQL can use SSL communications between a client and the server. This is material well known to everyone, just the MySQL flavor.

11/29/01 3

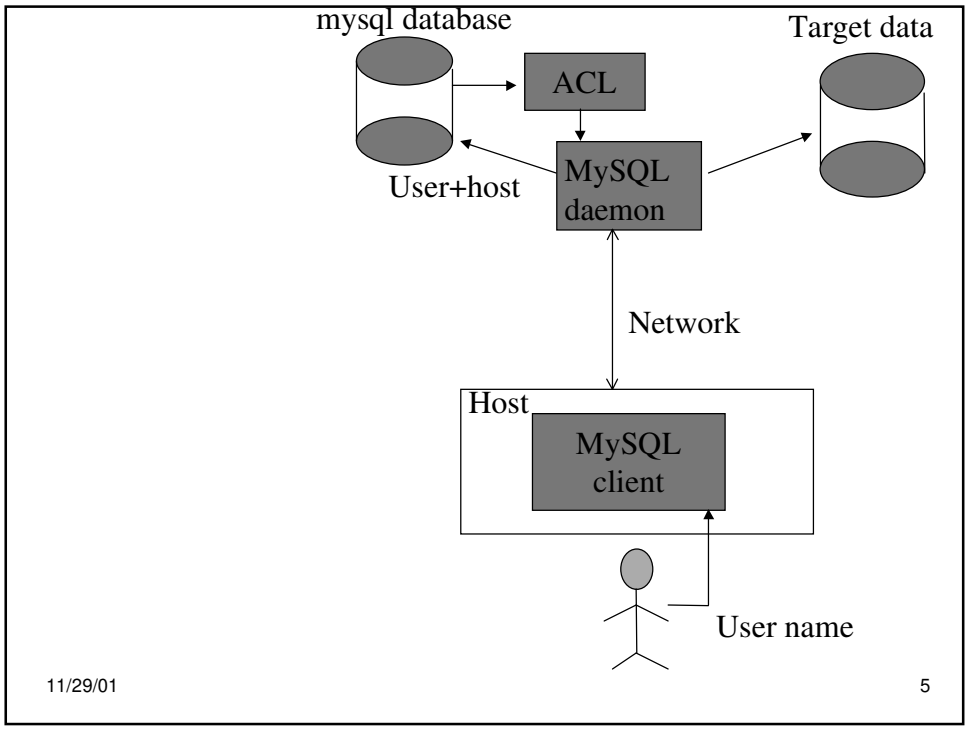
Topic One: Database security

This is where MySQL is rather unique. MySQL uses security based on Access Control Lists (ACLs).

The MySQL privilege system ensures that all users may do exactly the things that they are supposed to be allowed to do.

Identity is determined by the host from which you connect and the user name you specify.

11/29/01 4



MySQL access control involves two stages:

- Stage 1 (Connection verification): The server checks whether or not you are even allowed to connect. This is security based on the whole database as an object.
- Stage 2 (Request verification): Assuming you can connect, the server checks each request you issue to see whether or not you have sufficient privileges to perform it. Objects here are tables, columns, as well as commands.

11/29/01 6

How an ACL is created?

In the control database (mysql) there are 6 tables to control the privileges: **`columns_priv`**, **`db`**,**`func`**, **`host`**, **`tables_priv`** and **`user`**.

The server uses the **`user`**, **`db`**, and **`host`** tables at both stages of access control..

For the second stage of access control (request verification), the server may, if the request involves tables,additionally consult the **`tables_priv`** and **`columns_priv`** tables.

11/29/01 7

The **`user`** table scope fields determine whether to allow or reject incoming connections. For allowed connections, any privileges granted in the user table indicate the user's global (superuser) privileges.

The **`db`** and **`host`** tables are used together The **`db`** table scope fields determine which users can access which databases from which hosts.The privilege fields determine which operations are allowed. The **`host`** table is used as an extension of the db table when you want a given db table entry to apply to several hosts.

11/29/01 8

The **`tables_priv`** and **`columns_priv`** tables are similar to the db table, but are more fine-grained: they apply at the table and column levels rather than at the database level.

The whole picture is:
global privileges (from the user table)
OR (database privileges AND host privileges)
OR table privileges
OR column privileges

How to manage user's privileges?

- There are two main commands: **grant** and **revoke.**
- You may obtain the same effect by using insert and update statements directly on the mysql ACL tables but you must know very well what you are doing.
- Avoid giving admin privileges to users. Specially, avoid process (shutdown, kill, reload, flush, etc.) and file privileges

Topic Two: Secure login and communications

When login with the MySQL command line client you can provide you password as a command line option or to make it more secure you can enter the data via the standard input for the program.

Passwords provided by the client go "in the clear" over the network. You may use the compress method to make things harder to read or you could only use Unix sockets but still the only "secure" way is via SSL.

11/29/01 11

MySQL provides a set of functions for hashing;

- PASSWORD – MySQL hashing method.
- MD5 – no explanation required.
- ENCRYPT – hashing method used by Unix.

MySQL provides a method for encoding and decoding secret data with the functions ENCODE and DECODE.

All passwords provided to MySQL must be encrypted using MySQL encrypt method. This means data stored in the password columns of the ACL tables must be encrypted.

11/29/01 12

You have two options for secure channels between client and server with MySQL:

- Option 1: Use SSH to create a secure tunnel.

- Option 2: You must use MySQL 4.0 or higher. Obtain OpenSSL and install it, then compile the MySQL package to use SSL.

11/29/01 13

If you have an SSL enabled MySQL server you can control user's access so users can be forced to use an SSL Client.

If using an SSL connection, users can be required to provide a X509 certificate, inclusive we can control the issuer, subject and/or cipher.

11/29/01 14

For controlling the SSL features, use the grant command. For example:

```
GRANT ALL PRIVILEGES ON test.* TO root@localhost
 IDENTIFIED BY "goodsecret"
 REQUIRE SUBJECT "C=EE, ST=Some-State, L=Tallinn,
 O=MySQL demo client certificate,
 CN=Tonu Samuel/Email=tonu@mysql.com"
 AND ISSUER "C=FI, ST=Some-State, L=Helsinki,
 O=MySQL Finland AB,
 CN=Tonu Samuel/Email=tonu@mysql.com"
 AND CIPHER "EDH-RSA-DES-CBC3-SHA"
```

11/29/01 15

Finally, follow general security guidelines for Unix. Consider MySQL a low level daemon process.

Remember, root in MySQL is not ***root*** for the Unix system. Avoid giving any PROCESS or FILE privileges to MySQL users other that the root user.

NEVER run the MySQL daemon as ***root*** because then you are opening the host on which MySQL is running to the power of the SQL scripting language. This is equivalent of Visual Basic for Applications scripts (VBA) running wild on Windows machines.

11/29/01 16