



RMOUG TRAINING DAYS
February 14-16, 2012 Denver, Colorado
Colorado Convention Center

www.rmouug.org

COLLEGATE PEAKS - PHOTO COURTESY OF JOHN KING, PAST DUES MEMBER

MySQL Security Essentials

Ronald Bradford
<http://ronaldbradford.com>

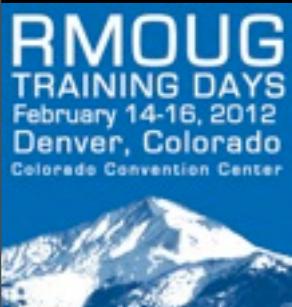
RMOUG
February 2012



EffectiveMySQL.com - Its all about **Performance** and **Scalability**

OBJECTIVE

- Identify MySQL security issues
- Improve MySQL installation process
- Identify data issues
- Improve data integrity and security
- Auditing options



ABOUT AUTHOR

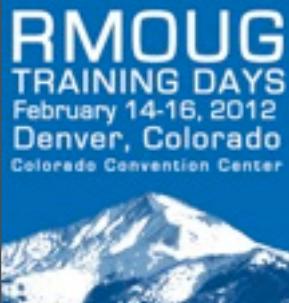
RONALD BRADFORD

- 12 years with MySQL / 22 years with RDBMS
 - Senior Consultant at MySQL Inc (06-08)
 - Consultant for Oracle Corporation (96-99)
- 7 years presenting MySQL content
- All time top MySQL blogger
- Published author
- Oracle ACE Director

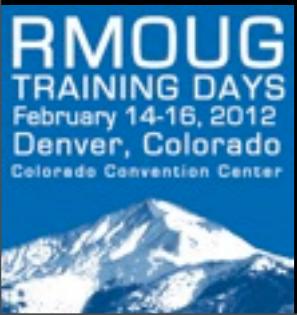
Available NOW
for consulting

<http://RonaldBradford.com>

EffectiveMySQL.com - Its all about **Performance** and **Scalability**



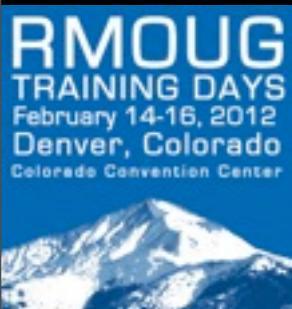
MySQL Security



[**EffectiveMySQL.com**](http://EffectiveMySQL.com) - Its all about **Performance** and **Scalability**

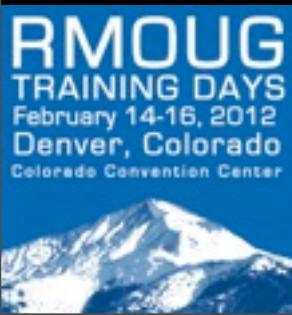
INSTALL SECURITY

- Default is terrible
- Minimum
 - \$ mysql_secure_installation
- Recommended
 - Operating System
 - Permissions & Privileges



Hacking MySQL

10 simple ways



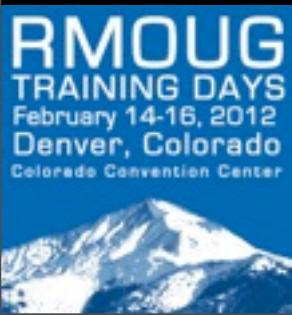
[**EffectiveMySQL.com**](http://EffectiveMySQL.com) - Its all about **Performance** and **Scalability**

HACKING MySQL

I

```
$ mysql -uroot
```

For a default MySQL
installation, there is no password for
super user account



EffectiveMySQL.com - Its all about **Performance** and **Scalability**

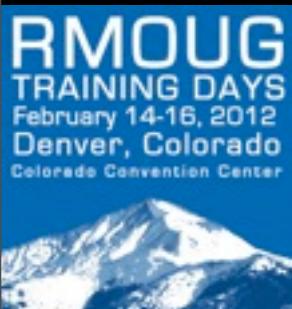
HACKING MySQL

2

```
$ mysql
```

```
mysql> USE test
```

For a default MySQL installation, there is anonymous access. More on this soon!



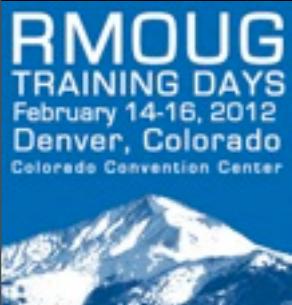
HACKING MySQL

3

```
$ sudo su -  
$ mysql  
$ cat ~/.my.cnf  
[client]  
user=root  
password=XXX
```

Many host providers have the superuser password in the default user config file

```
# Other users with logins  
$ find / -name ".my.cnf"
```

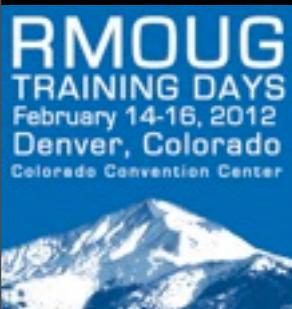


HACKING MySQL

4

```
$ grep mysql $HOME/.history  
$ grep mysql /home/*/.history  
  
mysql -uroot -pmysecret dbname
```

Shell commands leave an audit history. If password is specified on command line, it is in history

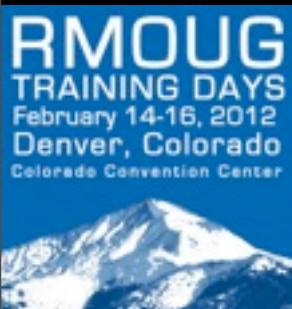


HACKING MySQL

4a

```
$ cat wordpress/wp-config.php  
$ cat drupal/sites/default/settings.php  
  
# Insert your application here
```

Applications with cleartext
passwords and user may have
excessive privileges

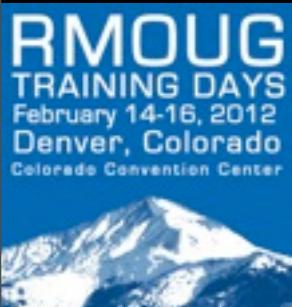


HACKING MySQL

5

```
$ mysql -u<some not privileged user>
mysql> SELECT host,user,password FROM mysql.user;
+-----+-----+-----+
| host      | user   | password |
+-----+-----+-----+
| localhost | root   |          |
| mactaz.local | root   |          |
| 127.0.0.1  | root   |          |
| ::1        | root   |          |
| localhost |          |          |
| mactaz.local |         |          |
+-----+-----+
```

Can you view MySQL privileges to find a user with more access?

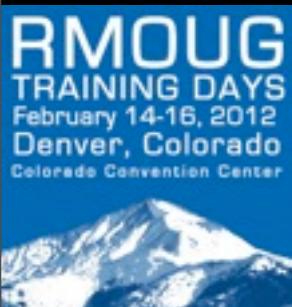


HACKING MySQL

6

```
$ cd [mysql-data-dir] # e.g. /var/lib/mysql  
$ cat master.info  
15  
mysql-bin.001536  
20160154  
10.0.0.1  
rep1  
XXX  
3306  
60  
  
$ mysql -urepl -pXXX -h10.0.0.1  
mysql> SHOW GRANTS;
```

Are datadir permissions secure?
Does replication connection have global
permissions?

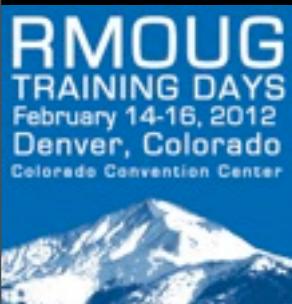


HACKING MySQL

7

```
$ cd [mysql-data-dir] # e.g. /var/lib/mysql  
$ strings mysql/user.MYD  
...  
dba*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19  
  
mysql> SELECT 'password',PASSWORD('password');  
+-----+-----+  
| password | PASSWORD('password') |  
+-----+-----+  
| password | *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 |  
+-----+-----+  
mysql> SELECT 'passwd',PASSWORD('passwd');  
+-----+-----+  
| passwd | PASSWORD('passwd') |  
+-----+-----+  
| passwd | *59C70DA2F3E3A5BDF46B68F5C8B8F25762BCCEF0 |  
+-----+-----+
```

Is datadir permissions secure?
Can list all users and password hash



HACKING MySQL

7a

```
mysql> CREATE SCHEMA IF NOT EXISTS hack;
mysql> use hack;
mysql> CREATE TABLE words(word VARCHAR(100));
mysql> LOAD DATA LOCAL INFILE '/tmp/passwords'
-> INTO TABLE words(word);

mysql -uroot -e "SELECT word,PASSWORD(word) FROM
hack.words" --skip-column-names > /tmp/mysql-passwords
$ cat /tmp/mysql-passwords
password *2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19
passwd *59C70DA2F3E3A5BDF46B68F5C8B8F25762BCCEF0
oracle *2447D497B9A6A15F2776055CB2D1E9F86758182F
changeme *7ACE763ED393514FE0C162B93996ECD195FFC4F5
xxx *3D56A309CD04FA2EEF181462E59011F075C89548
```

Brute force checking

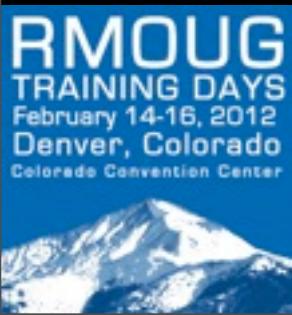
HACKING MySQL

8

```
# Replacing/Editing user.MYD
```

```
$ cd /tmp
$ echo "[mysqld]
datadir=/tmp" > my.cnf
$ scripts/mysql_install_db --defaults-file=/tmp/my.cnf
$ ls -l /tmp/mysql/user.*
-rw-rw---- 1 rbradfor wheel    320 Feb 13 11:27 /tmp/mysql/user.MYD
-rw-rw---- 1 rbradfor wheel   2048 Feb 13 11:27 /tmp/mysql/user.MYI
-rw-rw---- 1 rbradfor wheel  10630 Feb 13 11:27 /tmp/mysql/user.frm
```

If access data directory, you can
replace or hack users data



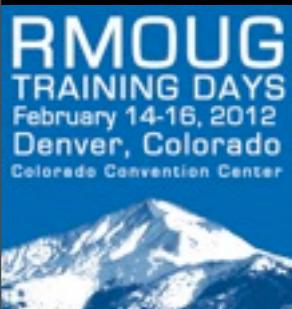
HACKING MySQL

9

```
[mysqld]
init-file=/path/to/init.sql
```

```
$ cat init.sql
UPDATE mysql.user SET password=password('fred');
FLUSH PRIVILEGES;
```

Run the SQL file when I start mysqld

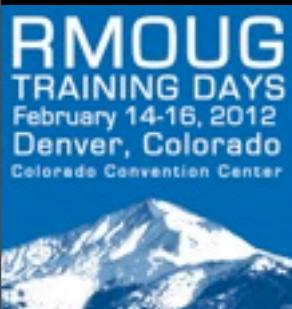


HACKING MySQL

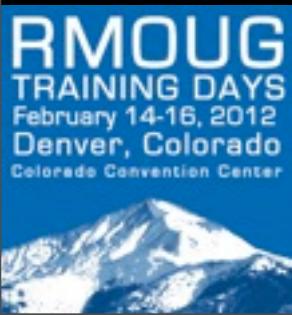
10

```
$ mysqld --skip-grant-tables  
  
$ mysql -uroot  
  
mysql> UPDATE mysql.user  
      -> SET password=PASSWORD('hacked');  
mysql> FLUSH PRIVILEGES;
```

If access to start and stop mysqld process, you can reset passwords.



Denial of Service (DOS)

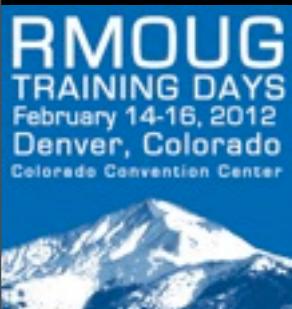


EffectiveMySQL.com - Its all about **Performance** and **Scalability**

BREAKING MySQL

```
$ mysql
mysql> USE test
mysql> CREATE TABLE filldisk(c VARCHAR(1000));
mysql> INSERT INTO filldisk VALUES(REPEAT('x',1000)),(REPEAT('y',1000)),
      (REPEAT('z',1000));
mysql> INSERT INTO filldisk SELECT a.c FROM filldisk a, filldisk b,
      filldisk c, filldisk d, filldisk e;
Query OK, 243 rows affected (0.05 sec)
mysql> INSERT INTO filldisk SELECT a.c FROM filldisk a, filldisk b,
      filldisk c, filldisk d, filldisk e;
Will it end?
mysql> SELECT POW(3,5);
+-----+
| pow(3,5) |
+-----+
|      243 |
+-----+
mysql> SELECT POW(243,5);
+-----+
| pow(243,5)      |
+-----+
| 847,288,609,443 |
```

Anonymous access.



BREAKING MySQL

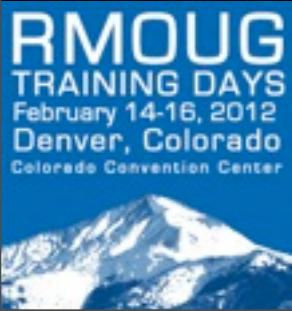
```
# Steal CPU cycles
mysql> SELECT MD5(a.c), MD5(b.c)
-> FROM filldisk a, filldisk b ORDER BY RAND();
```



EffectiveMySQL.com - Its all about **Performance** and **Scalability**

BREAKING MySQL

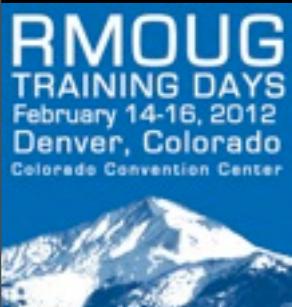
```
# Force disk I/O
mysql> SET SESSION tmp_table_size=1024*4; # 4K
mysql> SELECT ...
```



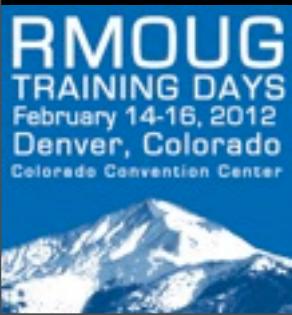
EffectiveMySQL.com - Its all about **Performance** and **Scalability**

INSTALL SECURITY

- \$ mysql_secure_installation
- Remove anonymous users
- Remove test database
- Remove non ‘localhost’ root users
- Set root password



OS Security

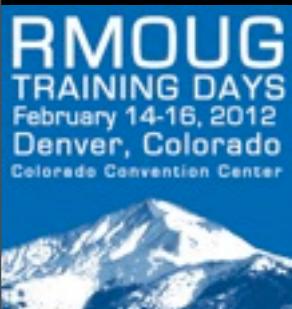


EffectiveMySQL.com - Its all about **Performance** and **Scalability**

OS SECURITY

- Defaults are not always secure
- Separate Data/Binary Logs/Logs/Configuration/Backups
 - Individual directory permissions
- Never run mysqld as 'root' user

- Minimize security risk
- Better auditability

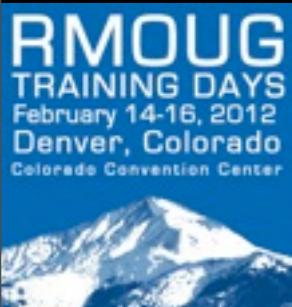


DATA SECURITY

```
$ ls -ld /var/lib/mysql
drwxr-xr-x 6 mysql mysql 4096 Oct 13 16:45 /var/lib/mysql

$ ls -l /var/lib/mysql
total 3749488
drwx----- 2 mysql mysql          4096 Oct  6 18:50 db1
drwx----- 2 mysql mysql          4096 Jun  6 14:11 db2
-rw-rw---- 1 mysql mysql 3298820096 Oct 16 20:42 ibdata1
-rw-rw---- 1 mysql mysql 268435456 Oct 16 20:42 ib_logfile0
-rw-rw---- 1 mysql mysql 268435456 Oct 16 20:42 ib_logfile1
drwx----- 2 mysql mysql          4096 Jun  6 09:37 mysql
srwxrwxrwx 1 mysql mysql          0 Oct 12 20:09 mysql.sock
```

Change default configuration for
socket=/var/run/mysql/mysql.sock



INSTALLATION

● Best Practice

/mysql
/etc
/data
/binlog
/log
/mysql-version

Single partition per
MySQL Instance

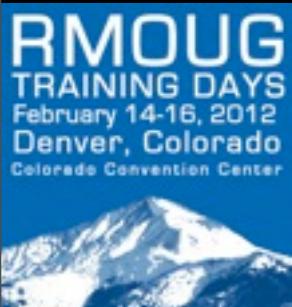
/etc/my.cnf
/etc/profile.d/mysql.sh
/etc/init.d/mysqld

Global files as symlinks
from partition

INSTALLATION

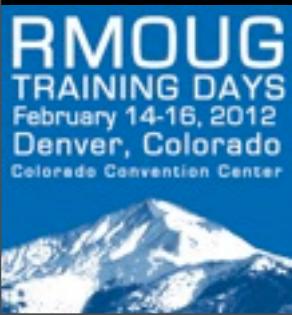
- Software installed by root
- Separate MySQL permissions for directories

```
$ chown -R root:root /mysql
$ chown -R mysql:mysql /mysql/{data,log,binlog,etc}
$ chmod 700 /mysql/{data,binlog}
$ chmod 750 /mysql/{etc,log}
```

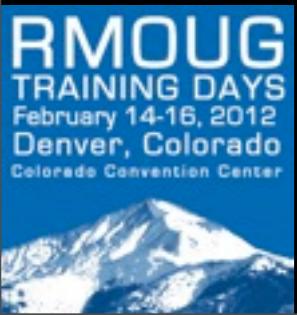


NETWORK

- MySQL listens on one TCP/IP port
 - Default port is 3306
 - --skip-networking disables TCP/IP
- Database does not require physical web access generally
 - Firewall management



User Privileges



[**EffectiveMySQL.com**](http://EffectiveMySQL.com) - Its all about **Performance** and **Scalability**

USER PRIVILEGES

Best Practice



```
CREATE USER goodguy@localhost IDENTIFIED BY 'sakila';  
GRANT CREATE, SELECT, INSERT, UPDATE, DELETE ON db.* TO  
goodguy@localhost;
```

Normal Practice



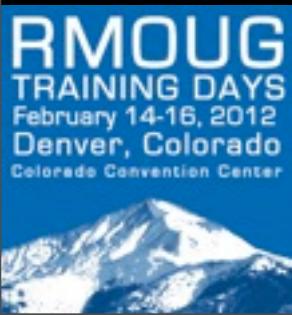
```
CREATE USER superman@'%' ;  
GRANT ALL ON *.* TO superman@'%';
```

<http://dev.mysql.com/doc/refman/5.5/en/create-user.html>

<http://dev.mysql.com/doc/refman/5.5/en/grant.html>

GRANT ALL ON *.* TO user@'%'

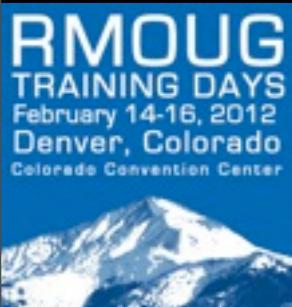
- GRANT ALL ON *.* TO user@'%'
 - *.* gives you access to all tables in all schemas
 - @'%' give you access from any external location
 - ALL gives you
 - ALTER,ALTER ROUTINE,CREATE,CREATE ROUTINE,CREATE TEMPORARY TABLES,CREATE USER,CREATE VIEW,DELETE,DROP,EVENT,EXECUTE,FILE,INDEX,INSERT,LOCK TABLES,PROCESS,PROXY,REFERENCES,RELOAD,REPLICATION CLIENT,REPLICATION SLAVE,SELECT,SHOW DATABASES,SHOW VIEW,SHUTDOWN,**SUPER**,TRIGGER,UPDATE,USAGE



SUPER

- SUPER

- Bypasses `read_only`
- Bypasses `init_connect`
- Can Disable binary logging
- Change configuration dynamically
- No reserved connection



SUPER/READ ONLY

```
$ mysql -ugoodguy -psakila db
mysql> insert into test1(id) values(1);
ERROR 1290 (HY000) : The MySQL server is running with the --read-only option so it cannot execute this statement
```



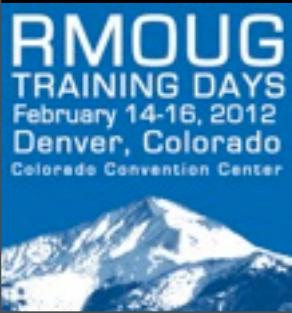
```
$ mysql -usuperman db
mysql> insert into test1(id) values(1);
Query OK, 1 row affected (0.01 sec)
```



SUPER / CONNECT

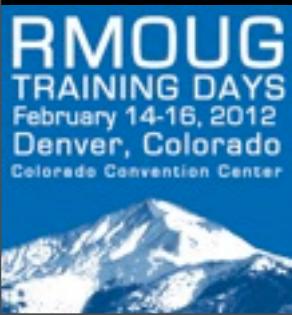
```
#my.cnf
[client]
init_connect=SET NAMES utf8
```

This specifies to use UTF8 for communication with client and data



EffectiveMySQL.com - Its all about **Performance** and **Scalability**

SUPER / CONNECT



[EffectiveMySQL.com](#) - Its all about **Performance** and **Scalability**

SUPER / CONNECT



```
mysql -ugoodguy -psakila db

mysql> SHOW SESSION VARIABLES LIKE 'ch%';
+-----+-----+
| Variable_name          | Value   |
+-----+-----+
| character_set_client    | utf8  |
| character_set_connection | utf8  |
| character_set_database   | latin1  |
| character_set_filesystem | binary   |
| character_set_results    | utf8  |
| character_set_server     | latin1  |
| character_set_system     | utf8    |
+-----+-----+
```

SUPER / CONNECT

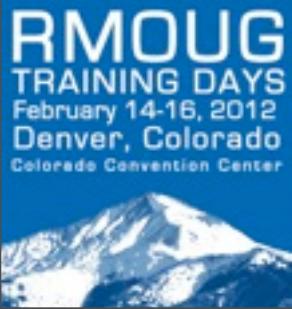


```
$ mysql -usuperman db
mysql -ugoodguy -psakila db                                J VARIABLES LIKE
mysql> SHOW SESSION VARIABLES LIKE 'ch%';
+-----+-----+-----+-----+
| Variable_name          | Value   |          | Value   |
+-----+-----+-----+-----+
| character_set_client    | utf8  |          | latin1  |
| character_set_connection | utf8  |          | latin1  |
| character_set_database   | latin1  |          | database |
| character_set_filesystem  | binary   |          | filesystem |
| character_set_results     | utf8  |          | latin1  |
| character_set_server      | latin1  |          | server   |
| character_set_system       | utf8    |          | stem     |
+-----+-----+-----+-----+
```

SUPER/BINARY LOG

```
mysql> SHOW MASTER STATUS;
+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+
| binary-log.000001 | 354 | | |
+-----+-----+-----+
mysql> INSERT INTO account VALUES (9, 'New', 100);
mysql> SET SQL_LOG_BIN=0;
mysql> UPDATE account SET balance=1,000 WHERE id=9;
mysql> SET SQL_LOG_BIN=1;
mysql> UPDATE account SET balance = balance - 50 WHERE id=9;
mysql> SHOW MASTER STATUS;
+-----+-----+-----+
| File | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+
| binary-log.000001 | 674 | | |
+-----+-----+-----+
```

The diagram consists of three yellow circles with black numbers inside them. Circle 1 is positioned above the first statement, circle 2 is to the right of the second statement, and circle 3 is to the right of the third statement.



SUPER / BINARY LOG

```
$ mysqlbinlog binary-log.000001 --start-position=354 --stop-position=674

# at 354
#100604 18:00:08 server id 1    end_log_pos 450 Query thread_id=1 exec_time=0
error_code=0
use mysql/*!*/;
SET TIMESTAMP=1275688808/*!*/;
INSERT INTO account VALUES (9, 'New', 100); 1
/*!*/;
# at 579
#100604 18:04:31 server id 1    end_log_pos 674 Query thread_id=2 exec_time=0
error_code=0
use mysql/*!*/;
SET TIMESTAMP=1275689071/*!*/;
mysql> UPDATE balance SET balance = balance - 50 WHERE id=9; 3
/*!*/;
DELIMITER ;
# End of log file
ROLLBACK /* added by mysqlbinlog */;
```

SUPER/CONNECTION



```
$ mysql -uroot
```

```
mysql> show global variables like 'max_connections';
```

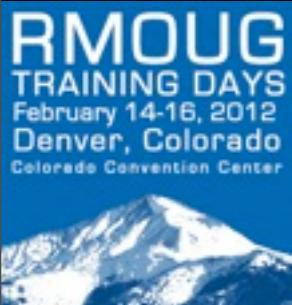
Variable_name	Value
max_connections	3

```
+-----+-----+
| Variable_name      | Value   |
+-----+-----+
| max_connections    | 3       |
+-----+-----+
1 row in set (0.07 sec)
```

```
mysql> show global status like 'threads_connected';
```

Variable_name	Value
Threads_connected	4

```
+-----+-----+
| Variable_name      | Value   |
+-----+-----+
| Threads_connected | 4       |
+-----+-----+
```



SUPER/CONNECTION

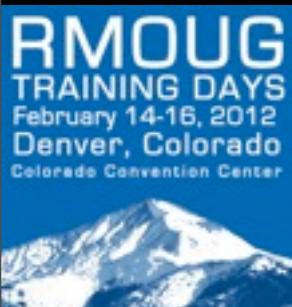


```
$ mysql -uroot
mysql> SHOW PROCESSLIST;
+----+----+----+----+----+----+----+----+
| Id | User | Host      | db    | Command | Time   | State    | Info
+----+----+----+----+----+----+----+----+
| 13 | root | localhost | db1   | Query   | 144    | User sleep | UPDATE test1 ...
| 14 | root | localhost | db1   | Query   | 116    | Locked    | select * from test1
| 15 | root | localhost | db1   | Query   | 89     | Locked    | select * from test1

mysql> KILL THREAD 13;
```



```
$ mysql -uroot
ERROR 1040 (HY000) : Too many connections
```



APPLICATION USERS

- Track Data Security
- Separation of responsibilities

- Application Viewer (Read Only Access)
 - SELECT
- Application User (Read/Write Access)
 - INSERT, UPDATE, DELETE, SELECT
- Application DBA (Schema Access Only)
 - CREATE, DROP, CREATE ROUTINE, SELECT, INSERT, UPDATE, DELETE, ...



SECURICH ACL

- User privilege via roles
- <http://securich.com>



Open source third
party package

NEW FEATURES

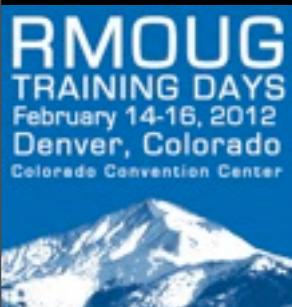
- User authentication interface (5.5)

<http://dev.mysql.com/doc/refman/5.5/en/authentication-plugins.html>

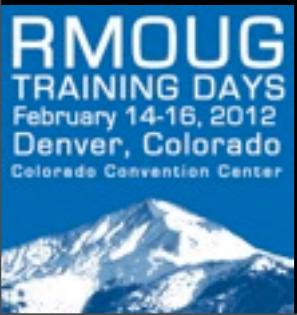
- PAM/LDAP Authentication (5.6 Enterprise)

http://blogs.oracle.com/MySQL/entry/new_commercial_extensions_for_mysql

http://blogs.oracle.com/mysql_joro/entry/mysql_55_brings_in_new_ways_to_authenticate_users



Data Integrity



EffectiveMySQL.com - Its all about **Performance** and **Scalability**

DATA INTEGRITY



```
CREATE TABLE sample_data (
    i TINYINT UNSIGNED NOT NULL,
    c CHAR(2) NULL,
    t TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP
) ENGINE=InnoDB DEFAULT CHARSET latin1;
INSERT INTO sample_data(i) VALUES (0) , (100) , (255);
Query OK, 3 rows affected (0.00 sec)
SELECT * FROM sample_data;
+----+----+-----+
| i | c | t           |
+----+----+-----+
| 0 | NULL | 2010-06-06 13:28:44 |
| 100 | NULL | 2010-06-06 13:28:44 |
| 255 | NULL | 2010-06-06 13:28:44 |
+----+----+-----+
3 rows in set (0.00 sec)
```

DATA INTEGRITY

```
mysql> INSERT INTO sample_data (i) VALUES (-1), (9000);
```



```
mysql> SELECT * FROM sample_data;
+---+---+---+
| i | c | t |
+---+---+---+
| 0 | NULL | 2010-06-06 13:28:44 |
| 100 | NULL | 2010-06-06 13:28:44 |
| 255 | NULL | 2010-06-06 13:28:44 |
| 0 | NULL | 2010-06-06 13:32:52 |
| 255 | NULL | 2010-06-06 13:32:52 |
+---+---+---+
5 rows in set (0.01 sec)
```

Data In != Data out

SQL_MODE

SQL_MODE =

STRICT_ALL_TABLES,

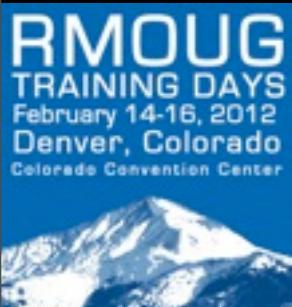
NO_ZERO_DATE,

NO_ZERO_IN_DATE,

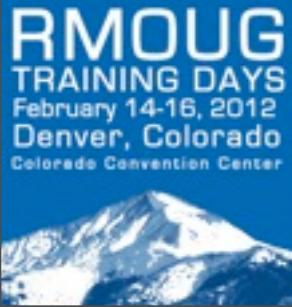
NO_ENGINE_SUBSTITUTION;

Minimum
Recommended
Configuration

<http://dev.mysql.com/doc/refman/5.5/en/server-sql-mode.html>



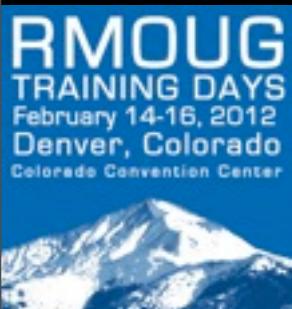
Installation



EffectiveMySQL.com - Its all about **Performance** and **Scalability**

INSTALLATION

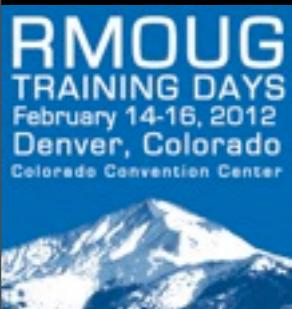
- Current version is 5.5
 - <http://dev.mysql.com/downloads/>
- RedHat distro current version is
 - 5.0.77 - 28 JAN 2009
- Ubuntu distro version is
 - 5.1.44 - 04 FEB 2010



UPDATES

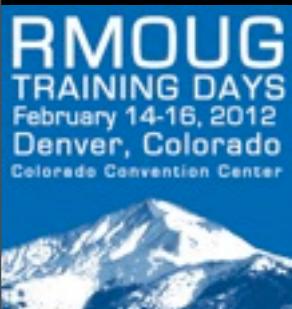
- MySQL is open source
 - More frequent updates
 - MySQL enterprise monthly updates
 - MySQL community ~ 6 months updates
- Release notes

<http://dev.mysql.com/doc/refman/5.5/en/news-5-5-x.html>



INSTALLATION

- MySQL 5.5 via distro
 - rpm available
 - Oracle has no yum repository
 - No immediate plans for community
 - Canonical (Ubuntu) has no ETA
 - .deb available **NEW**



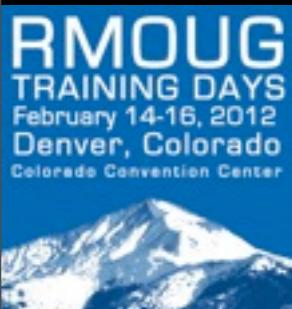
INSTALLATION

- Recommendations

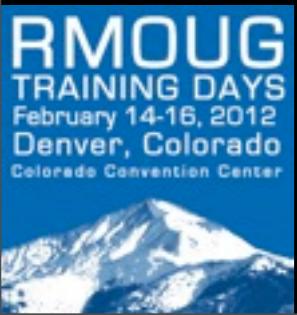
- Use MySQL created rpms
- Use MySQL Tar Binary

- Issues

- Dependencies
- PHP/Perl etc



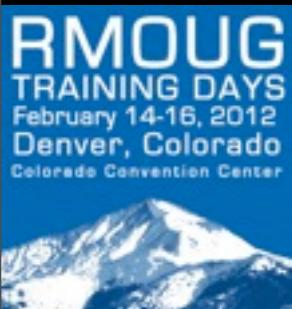
Auditing



EffectiveMySQL.com - Its all about **Performance** and **Scalability**

AUDITING OPTIONS

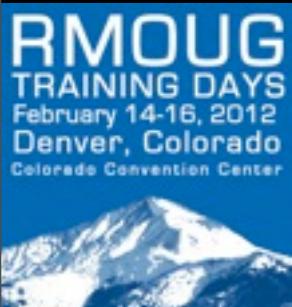
- Status variables
- Binary log
- Schema compare
- Audit plugin (5.5)
- Oracle Audit Vault



STATUS VARIABLES

```
mysql> SELECT variable_name, variable_value
-> FROM INFORMATION_SCHEMA.GLOBAL_STATUS
-> WHERE variable_name LIKE 'COM_ALTER%'
-> OR variable_name LIKE 'COM_DROP%'
-> OR variable_name LIKE 'COM_CREATE%';
```

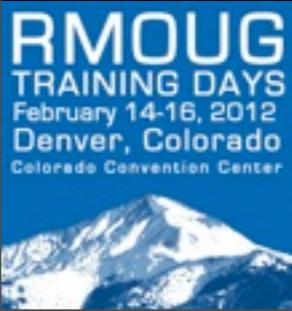
variable_name	variable_value
COM_ALTER_DB	0
COM_ALTER_DB_UPGRADE	0
COM_ALTER_EVENT	0
COM_ALTER_FUNCTION	0
COM_ALTER PROCEDURE	0
COM_ALTER_SERVER	0
COM_ALTER_TABLE	4
COM_ALTER_TABLESPACE	0



BINARY LOG

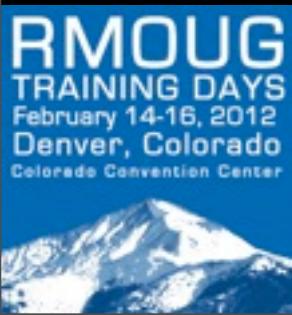
```
$ mysqlbinlog /path/to/file > binlog.txt  
$ grep -ie ALTER -e CREATE -e DROP binlog.txt
```

```
ALTER TABLE example ADD COLUMN i INT;  
ALTER TABLE example ADD INDEX (i);  
ALTER TABLE example DROP INDEX y;  
ALTER TABLE example DROP COLUMN y;
```



SCHEMA

```
$ mysqldump -uroot -p \  
--no-data --all-databases --skip-dump-date \  
| sed -e "s/AUTO_INCREMENT=[^\\ ] //"  
> schema.`date +%Y%m%d.%H%M`sql  
  
# Compare with a previous version  
$ diff schema.20111018.sql schema.20111017.sql
```



AUDIT PLUGIN

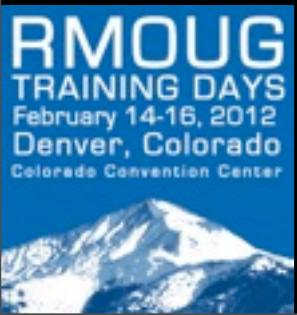
- Auditing interface (5.5)

<http://dev.mysql.com/doc/refman/5.5/en/audit-plugins.html>



Oracle Audit Vault

Better Security

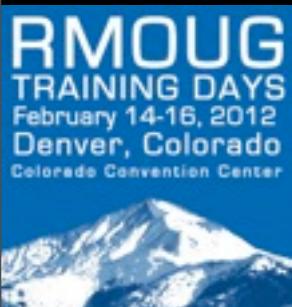


EffectiveMySQL.com - Its all about **Performance** and **Scalability**

RESTRICTING MySQLD

- Gazzang ezncrypt mask
- Also provides Transparent Encryption

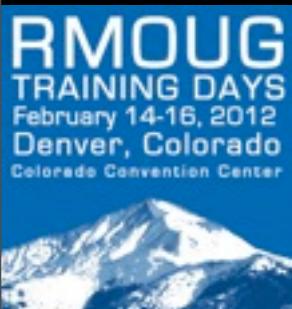
http://mikefrank.wordpress.com/2011/12/20/on-protecting-mysql-from-unwanted-use-of-skip_grant_tables/



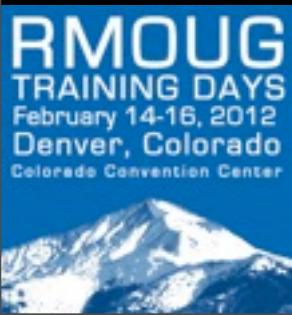
RESTRICTING MySQLD

- Compile from source
- --disable-grant-options
 - Removes --skip-grant-tables, --init-file and --bootstrap

<http://dev.mysql.com/doc/refman/5.1/en/source-configuration-options.html>



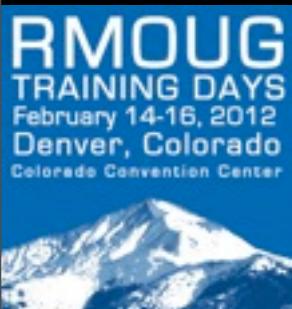
Conclusion



EffectiveMySQL.com - Its all about **Performance** and **Scalability**

CONCLUSION

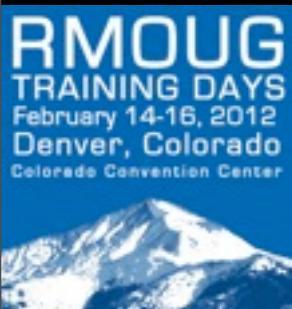
- MySQL is not secure by default
 - Responsibility on you (DBA/SA)
 - Starts with OS security
 - Use appropriate privileges

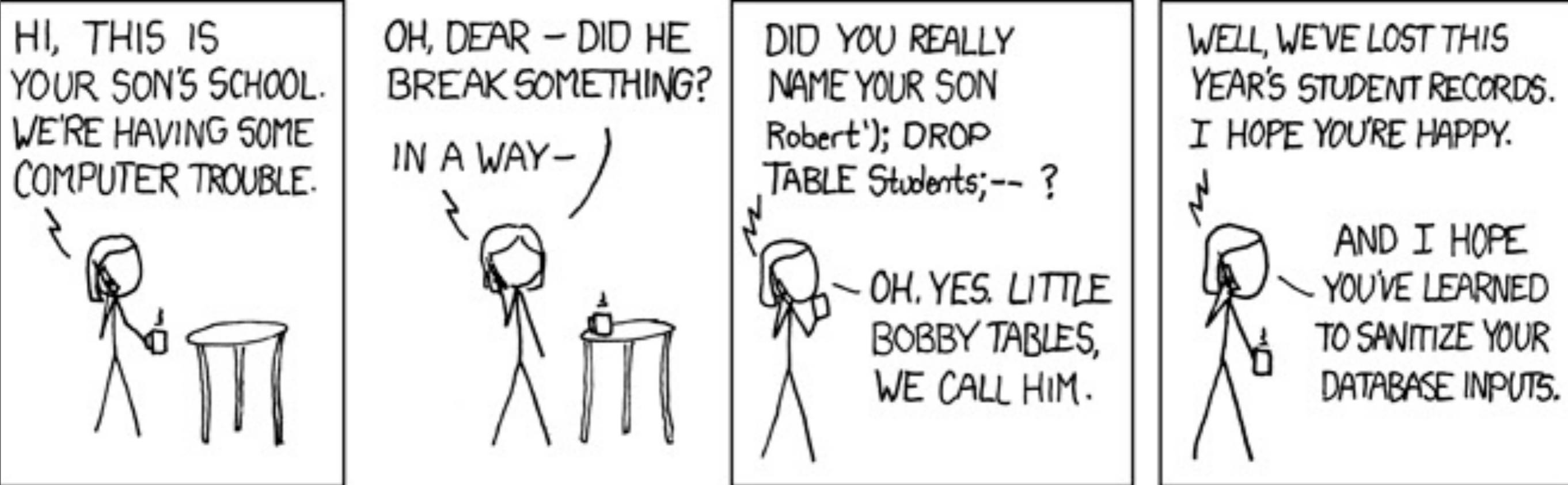


NOT COVERED

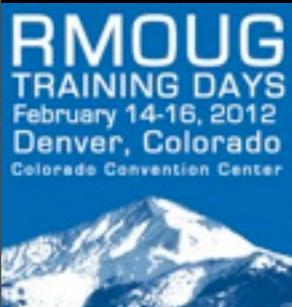
- Replication
- SSL
- Backups
- SQL Injection
- Sanitize your database inputs

<http://xkcd.com/327/>





<http://xkcd.com/327/>

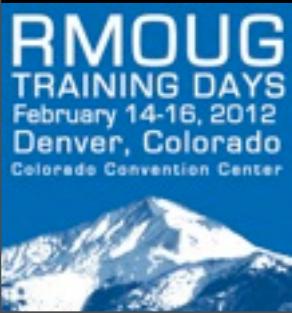


EffectiveMySQL.com - Its all about **Performance** and **Scalability**

REFERENCES

- Securing MySQL by Sheeri Cabral

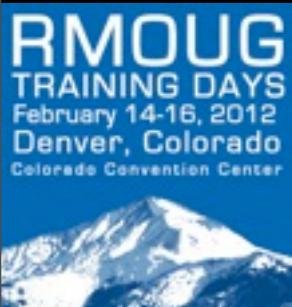
<http://sheeri.com/content/securing-mysql-and-how-be-rock-star-dba-pr>



REFERENCES

- OurSQL podcast
 - Episode 59: Security Blankets I
 - Episode 61: Security Blankets 2
 - Episode 55: MySQL Data encryption

<http://technocation.org/>



Em=psn

Ronald Bradford

<http://effectiveMySQL.com>