MySQL Security: More Than Just ACL's

Sheeri Cabral

Senior DB Admin/Architect, Mozilla

@sheeri www.sheeri.com

General Security

- Patching
- Prevent access
- Prevent meaningful info gathering

Access

- Network access
- Direct db access
- Access to backups

Access Points

- Who can login?
  - Network, seeing traffic
    - http://forge.mysql.com/snippets/view.php?id=15
  - OS
    - Data
    - Logs
    - Backups

Operating System

- Authentication
- Firewall
- Other installed programs

Securing your Application

- Authentication
- Config files
- User-entered data
  - SQL injection

Who has access?

- mk-show-grants
- SELECT user, host, length(password), ssl_type FROM mysql.user
- WHERE Super_priv='Y'
- WHERE user=''

Where is the access from?

- %
- %.company.com
- 10.0.% or 192.168.%

GRANTing Access

```
GRANT priv_type [(column_list)] [, priv_type [(column_list)]] ...
    ON [object_type]
        {tbl_name | * | *.* | db_name.* | db_name.routine_name}
    TO user [IDENTIFIED BY [PASSWORD] 'password']
    [REQUIRE        NONE |      [{SSL| X509}]
   [CIPHER 'cipher' [AND]]       [ISSUER 'issuer' [AND]]
        [SUBJECT 'subject']]   [WITH with_option [with_option] …]
```

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

Other ACL's

- Object access
- Password policies
- Roles

Access from...?

- localhost only, --skip-networking
- firewall
- Who can [attempt to] DOS you?

ACLs – to do what?

- --local-infile=0
- --skip-symbolic-links
- GRANT
  - MAX_QUERIES_PER_HOUR
  - MAX_UPDATES_PER_HOUR
  - MAX_CONNECTIONS_PER_HOUR

Changing ACLs

- Who changes ACLs?
- How are ACL changes audited?
- When do ACL changes happen?

Securich

- Darren Cassar, http://www.securich.com/
- Create/drop roles

```
call create_update_role('add','role1','select');
```

Create users with roles, adding objects

Drop users, revoke privileges

```
call grant_privileges('username','hostname', 'databasename',
'tablename','tabletype','rolename', 'email');
```

```
call grant_privileges('john','machine.domain.com',
 'employees', '','alltables','role1', 'john@domain.com');
```

Securich

- Reserved usernames
- Block users
- Rename users
- Clone users
- Reconciliation

Server Options

- --bind-address
- --skip-name-resolve
- --skip-show-database

Test Database

- Anyone can access it
- Stuff with data

OS Files and Permissions

- mysql server user
- mysql server files & logs
- Passwords on commandline
- Office policies/runbook

How Does Your Data Flow?

- Where is data encrypted?
- Where do errors go?
  - Are those logs checked?
- Where does the traffic flow?

Separating Admin Apps

- Same data, different interface
- Performance, e.g. reporting
- Only allowed from VPN?
  - Public vs. easily accessible

Plaintext information

- Passwords
- Credit card info
- Identification numbers (e.g. SSN)

Hashes

- Passwords

*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 = 'password'

- Be careful where encrypted traffic goes!

SQL Injection

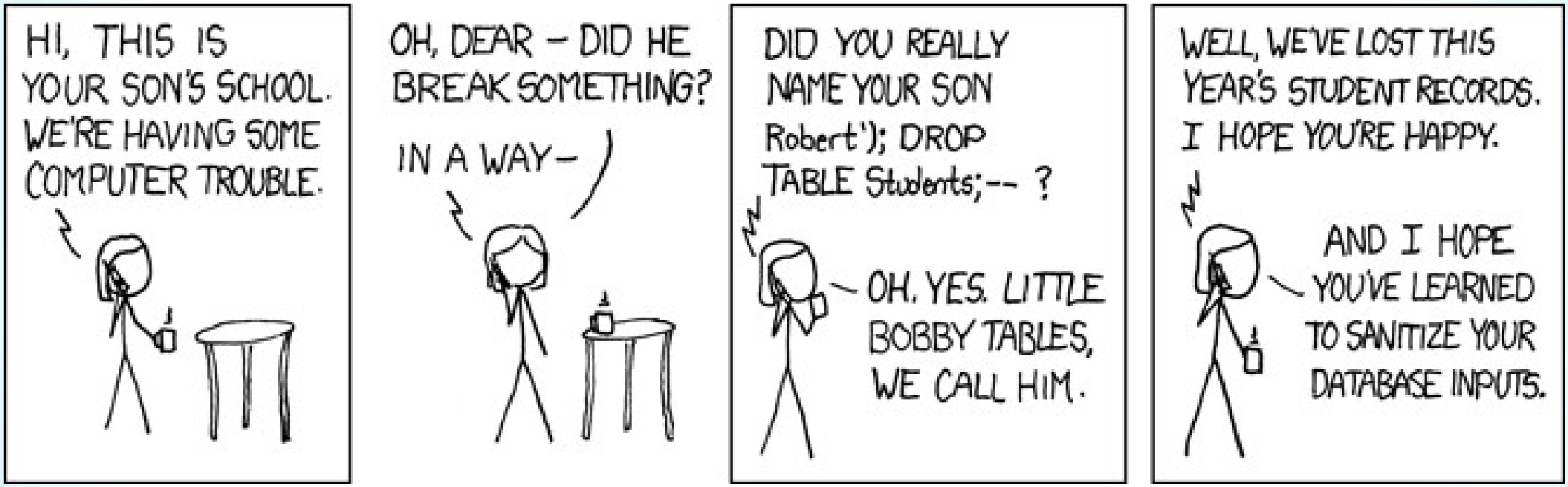
- http://bit.ly/kscope_sqlinject

```
SELECT count(*) FROM users WHERE
  username='$user' and pass='$pass';
-- if count(*)>0, log in!
```

- Pass: hi' or 1=1

```
SELECT count(*) FROM users WHERE
  username='foo' and pass='hi' or 1=1';
```

What + How


HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR - DID HE BREAK SOMETHING?
IN A WAY-
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Validate User Input

- Look for ; \g \G ' “ UNION
- HTML encoding
- NULL or char(0)
- VARCHAR and ' '

Validate User Input

- Save yourself time
- Buffer overflows
- CHARSET

Trusting GET or POST

- Only from certain pages
- cookies – even with valid session ids
- register_globals=off in PHP

Prepared Statements

```
PREPARE stmt1 FROM 'SELECT uname FROM
    UserAuth WHERE uname=? and pass=?';

SET @a = "alef";  SET @b = md5("alef");

EXECUTE stmt1 USING @a, @b;
```

Prepared Statements

```
PREPARE stmt1 FROM 'SELECT uname FROM
    UserAuth WHERE uname=? and pass=?';
SET @a = "alef";  SET @b = md5("alef");
EXECUTE stmt1 USING @a, @b;
```

```
SET @a = "alef";
SET @b = "alef' or 'x'='x";
EXECUTE stmt1 USING @a, @b;
DEALLOCATE PREPARE stmt1
```

Stored Code

- Stored procedures / functions
- Views
- Events
  - Instead of cron

Prepared Statements Perl Code

```
$query = $sql->prepare("SELECT uname FROM
UserAuth WHERE uname = ? AND pass = ?");
$query->execute($uname, $pass);
```

Prepared Statements PHP Code

```
$stmt = $mysqli->prepare("SELECT uname
  FROM
   UserAuth WHERE uname = ? AND pass = ?");
$stmt->bind_param($uname, $pass);
$stmt->execute();
```

Prepared Statements Java Code

```
PreparedStatement pstmt =
  con.prepareStatement("SELECT uname
  FROM
  UserAuth WHERE uname = ? AND pass
= ?");
  pstmt.setString(uname, pass);
  ResultSet rset = pstmt.executeQuery();
```

Prepared Statements .NET/C# Code

```
using(SqlCommand cmd = new SqlCommand("SELECT
   uname FROM UserAuth WHERE uname = @uname
   AND pass = @upass",con)) {

        cmd.Parameters.AddWithValue("@userName",
          userName);

        cmd.Parameters.AddWithValue("@pass", pass);

            using( SqlDataReader rdr =
          cmd.ExecuteReader() ){

            ...}

          }
```

Encryption

- SSL is per-client

- Unencrypted MySQL data streams

```
shell> tcpdump -l -i eth0 -w -src
  or dst port 3306 | strings
```

Auditing and Monitoring

- Prevention is one part of security
- Auditing - review and assess security
- Monitoring – alerting of security issues

Auditing and Monitoring

- General log to see all login attempts
- Locking out accounts with max_connect_errrors
  - global

Play hard to get

- MySQL Events instead of cron/task scheduler
- NO PLAINTEXT PASSWORDS
- Do not store it if you do not need it

Authentication Plugin

- MySQL 5.5 (since Dec 2010)
- MySQL Enterprise Plugins
  - Windows Authentication
  - PAM Authentication

Creating Policies

- There will be exceptions
  - But it's still a good idea to have the policies!

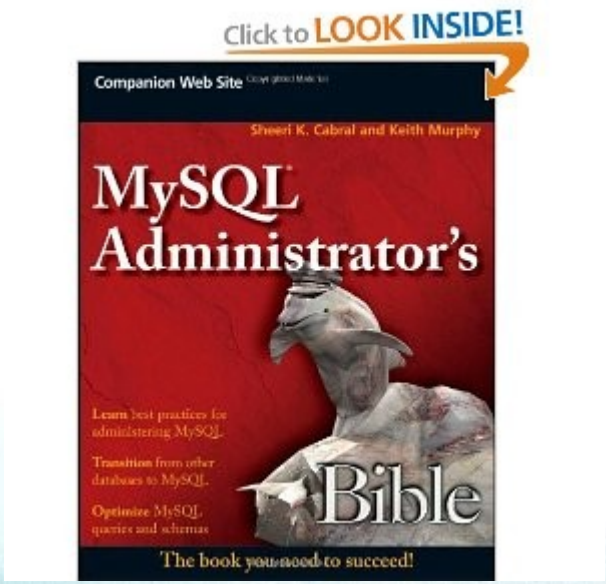
Questions? Comments?

OurSQL podcast

- www.oursql.com

MySQL Administrator's Bible

- tinyurl.com/mysqlbible



kimtag.com/mysql

planetmysql.com