PalominoDB

Securing MySQL

Presented by:

Sheeri K. Cabral - @sheeri

Senior DBA & Community Liasion, PalominoDB

www.palominodb.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

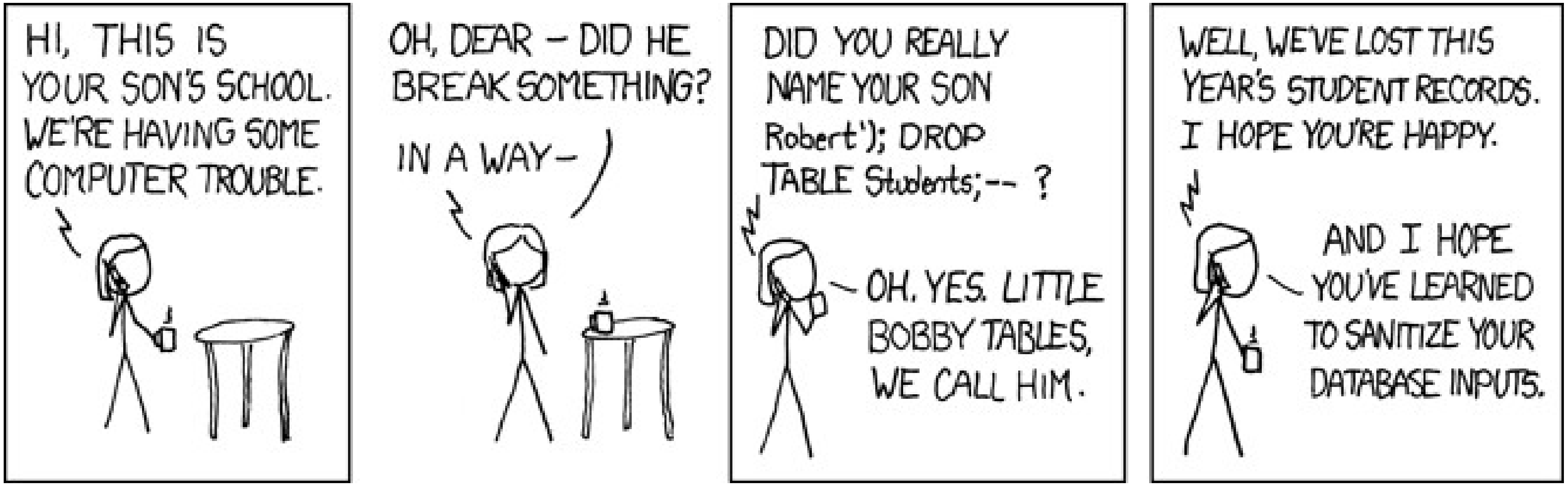
- http://bit.ly/kscope_sqlinject

```
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 - Code

Perl

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

Prepared Statements - Code

PHP

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

Prepared Statements - Code

Java

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

Prepared Statements - Code

.NET/C#

```
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

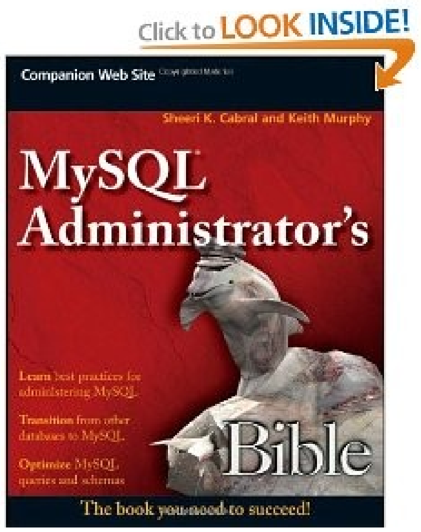
Creating Policies

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

Questions? Comments?

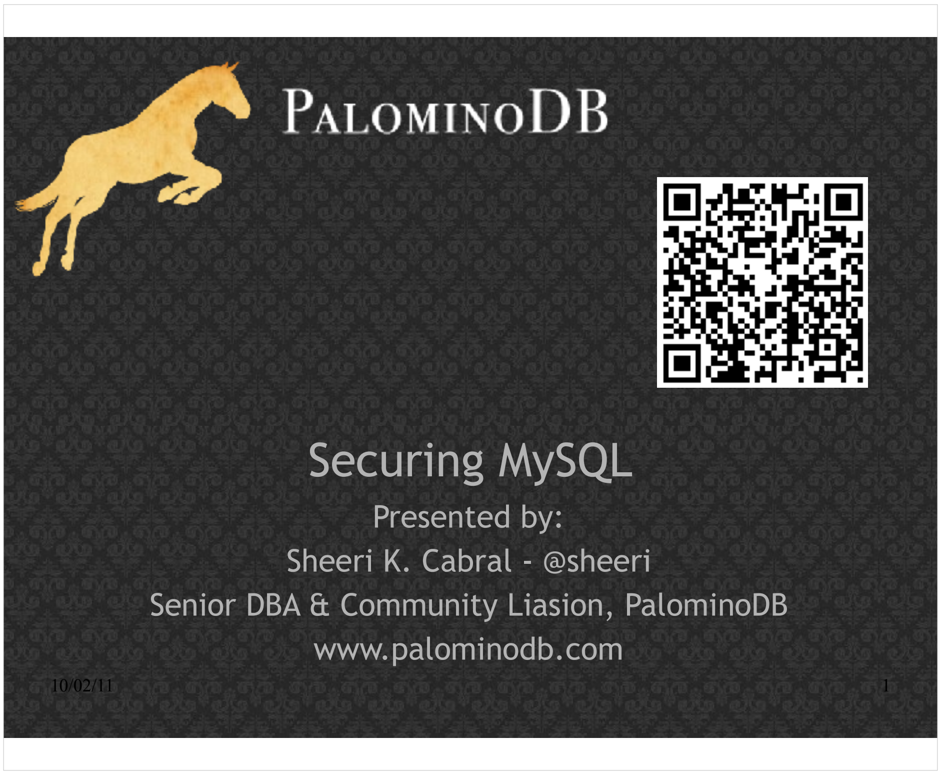
OurSQL podcast

MySQL Administrator's Bible


Click to LOOK INSIDE!
Companion Web Site
Sheeri K. Cabral and Keith Murphy
MySQL Administrator's Bible
Learn best practices for administering MySQL
Transition from other databases to MySQL
Optimize MySQL queries and schemas
The book you need to succeed!

kimtag.com/mysql

planetmysql.com

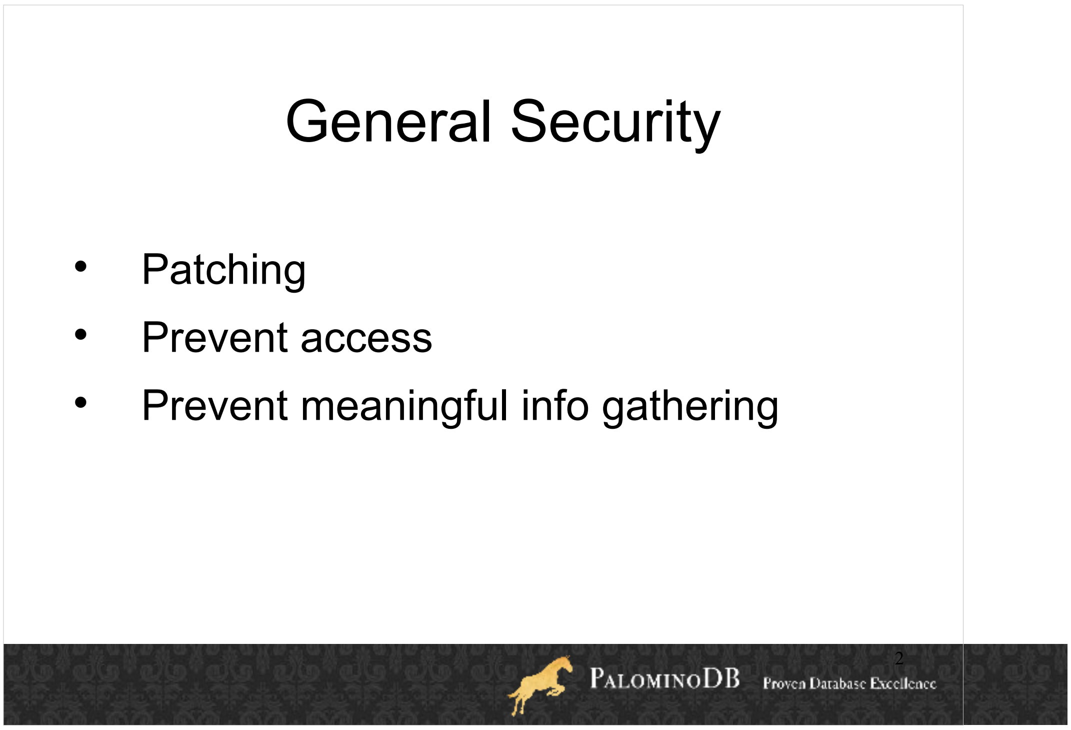


PALOMINODB

Securing MySQL

Presented by:

Sheeri K. Cabral - @sheeri
Senior DBA & Community Liasion, PalominoDB
www.palominodb.com



General Security

- Patching
- Prevent access
- Prevent meaningful info gathering

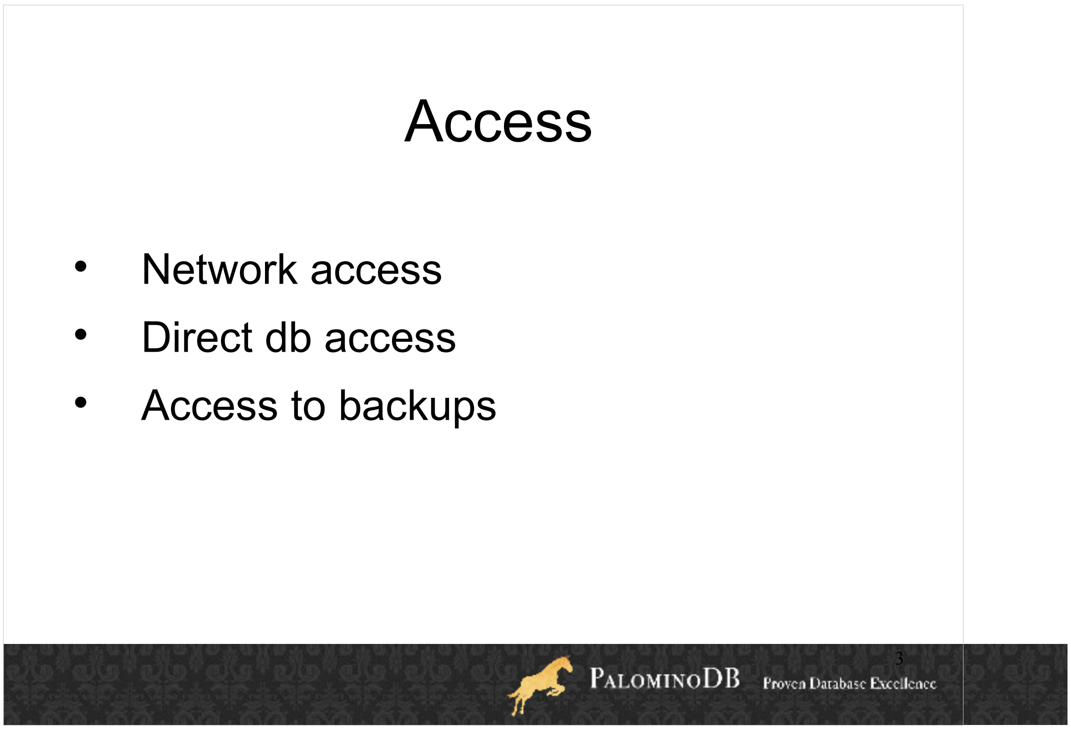
2

PALOMINODB Proven Database Excellence

MySQL has a new version each month! Can't patch every month, but you should upgrade every 6-12 months.
MySQL 5.1 GA Nov 2008
MySQL 5.5 GA Dec 2010

As for meaningful info gathering, e.g. encryption!

Preventing access includes permissions and ACL's but it's not limited to that

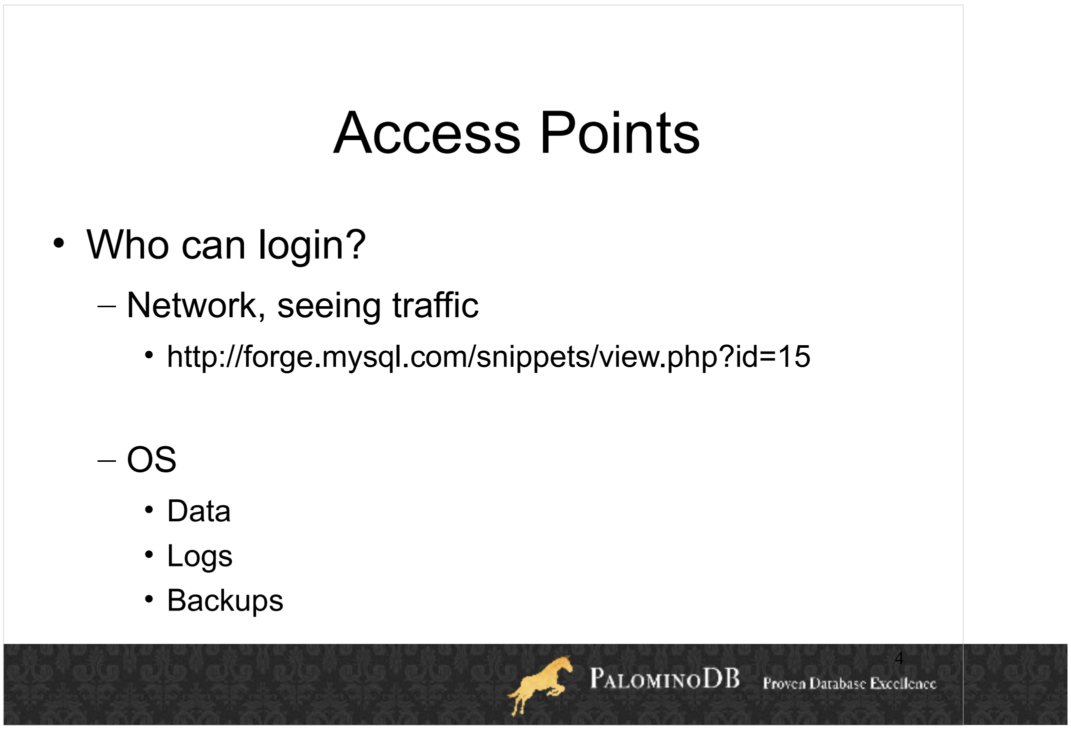


Access

- Network access
- Direct db access
- Access to backups

Can someone sniff traffic going across the network? What about replication or backups?

Can anyone try to login to port 3306 with telnet?



Access Points

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

4

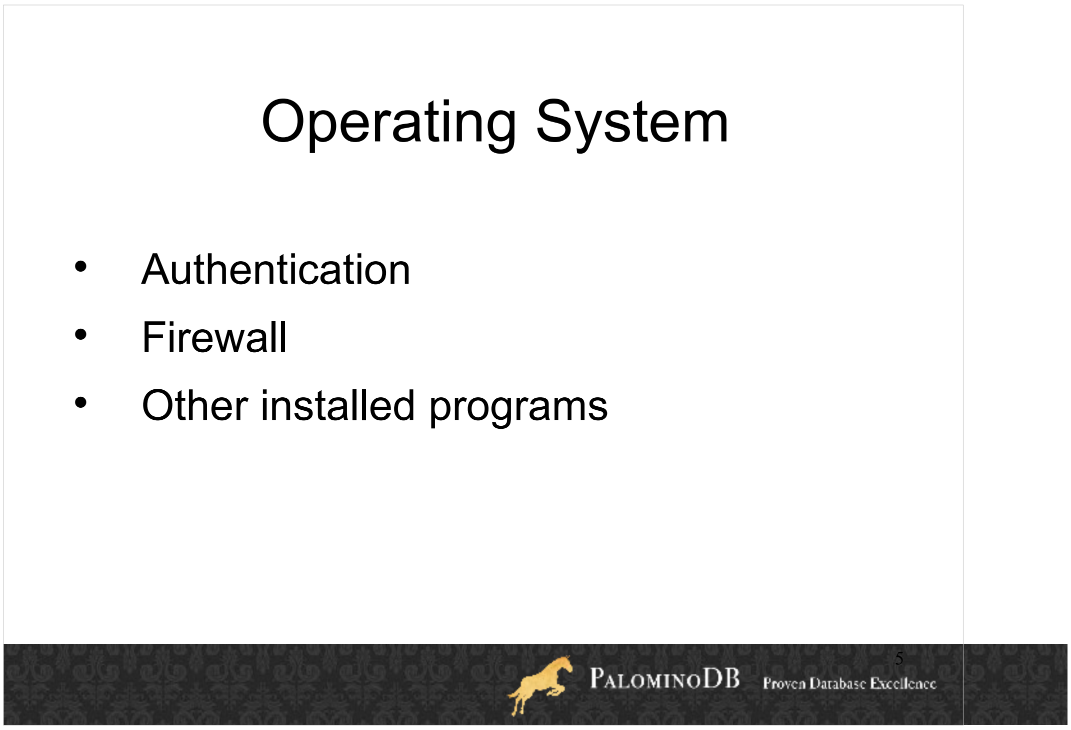
PALOMINODB Proven Database Excellence

Poor man's query profiler

Who can login to the OS and see the data? Strings on a MyISAM table can get data!

Who can login and read the logs? Slow, binary logs?

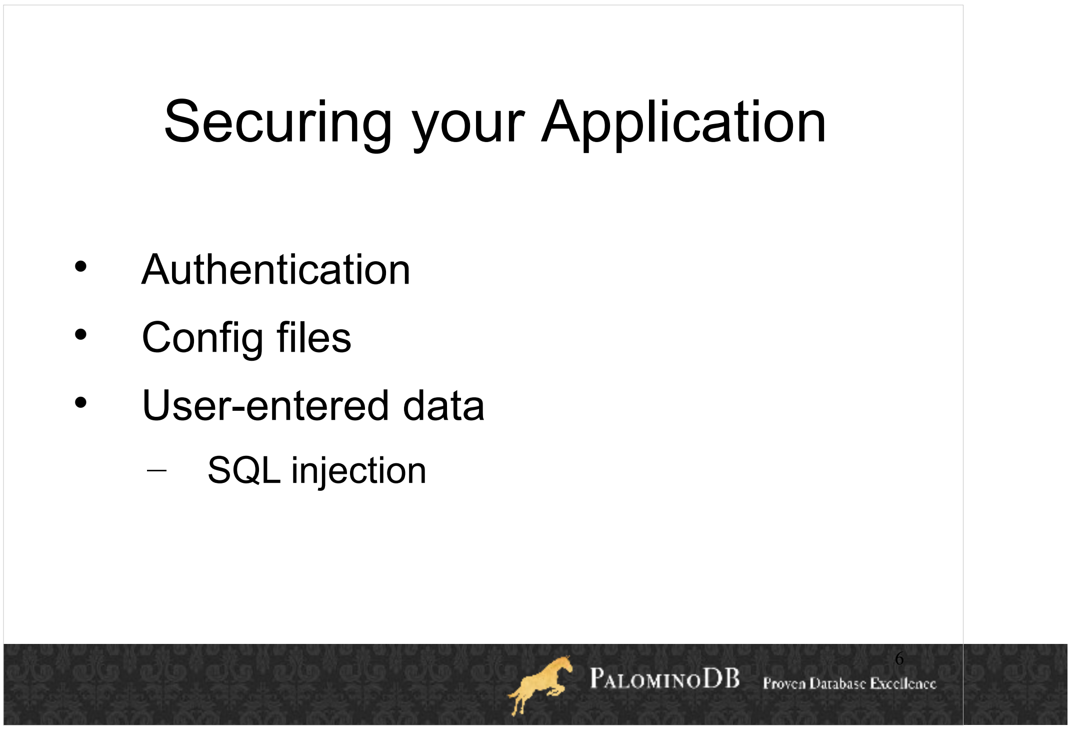
Read the backups?



Operating System

- Authentication
- Firewall
- Other installed programs

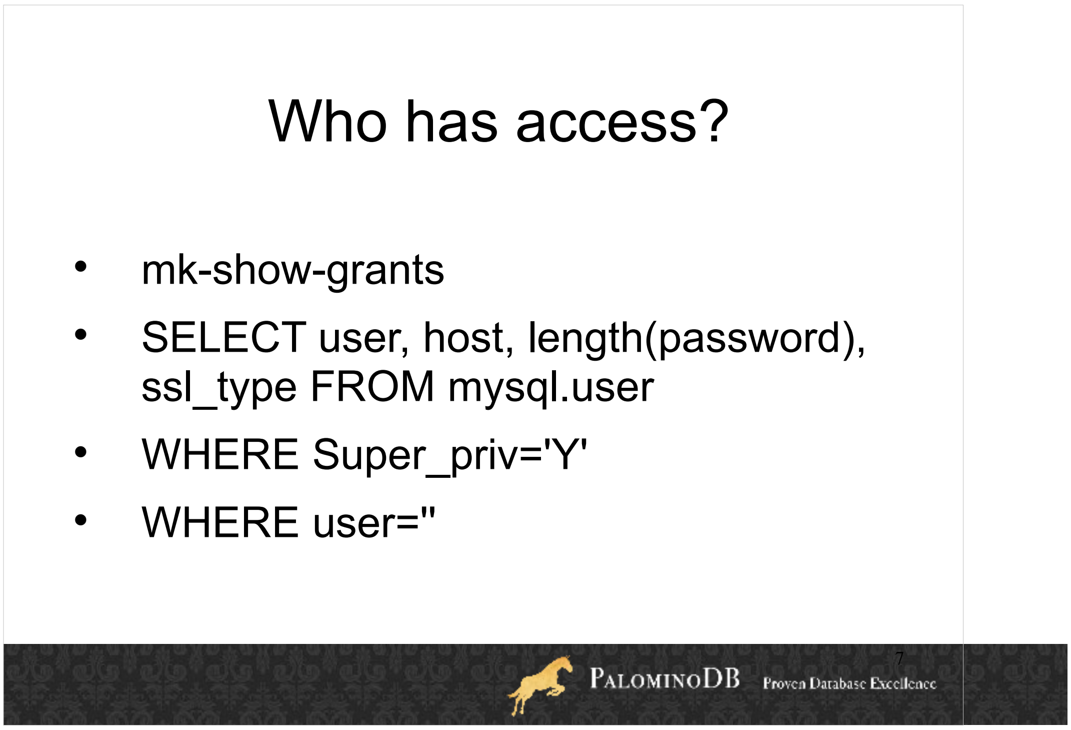
Are there other installed programs that are running on the same user, such as “nobody”? What other people have access due to the other installed programs?



Securing your Application

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

I talk about SQL injection and authentication more in-depth



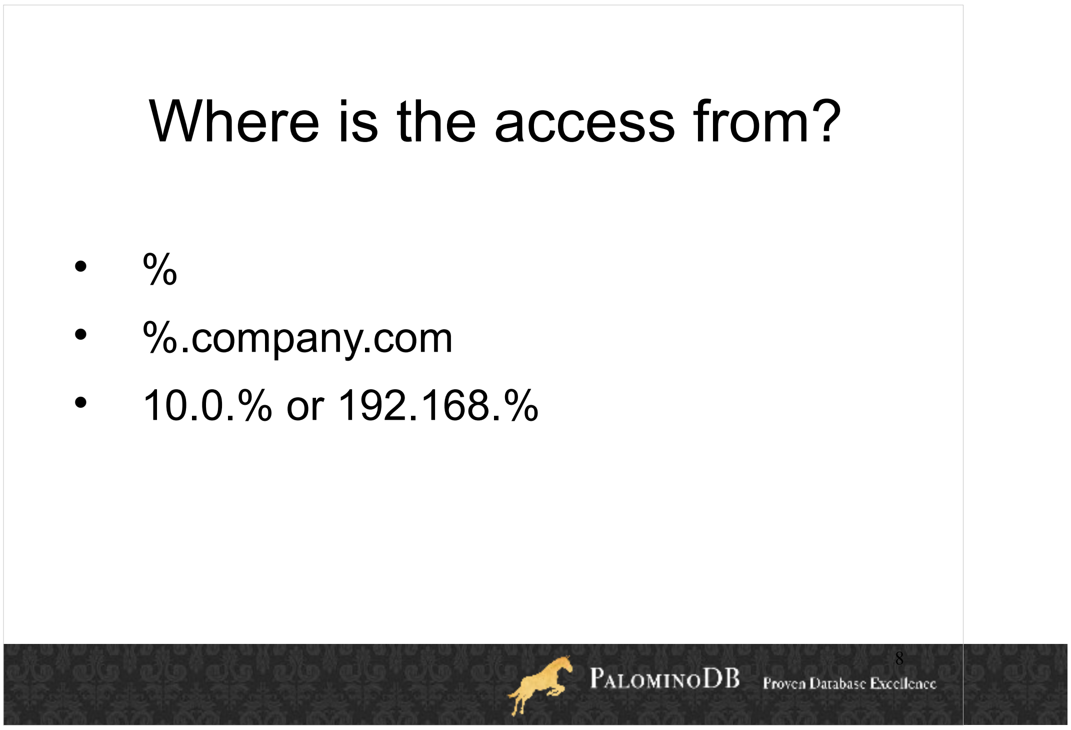
Who has access?

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

PALOMINODB Proven Database Excellence

Ways to see who has access

Super_priv can shutdown with mysqladmin shutdown. Also can write even if db is read_only. Also if max_connections is reached, 1 more user can login but only if they have the super priv.



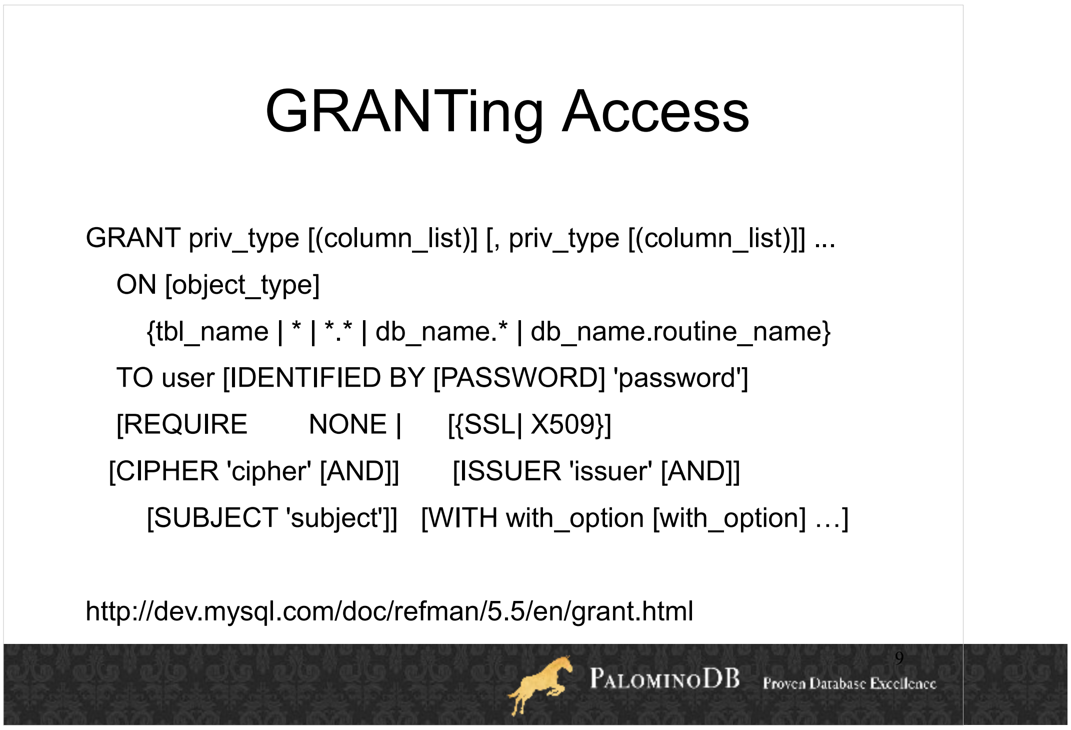
Where is the access from?

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

8

PALOMINODB Proven Database Excellence

This gets tricky because IP's can be spoofed, and if you're using Amazon EC2 or other cloud solutions (including traditional shared hosting) your IP might change without notice.

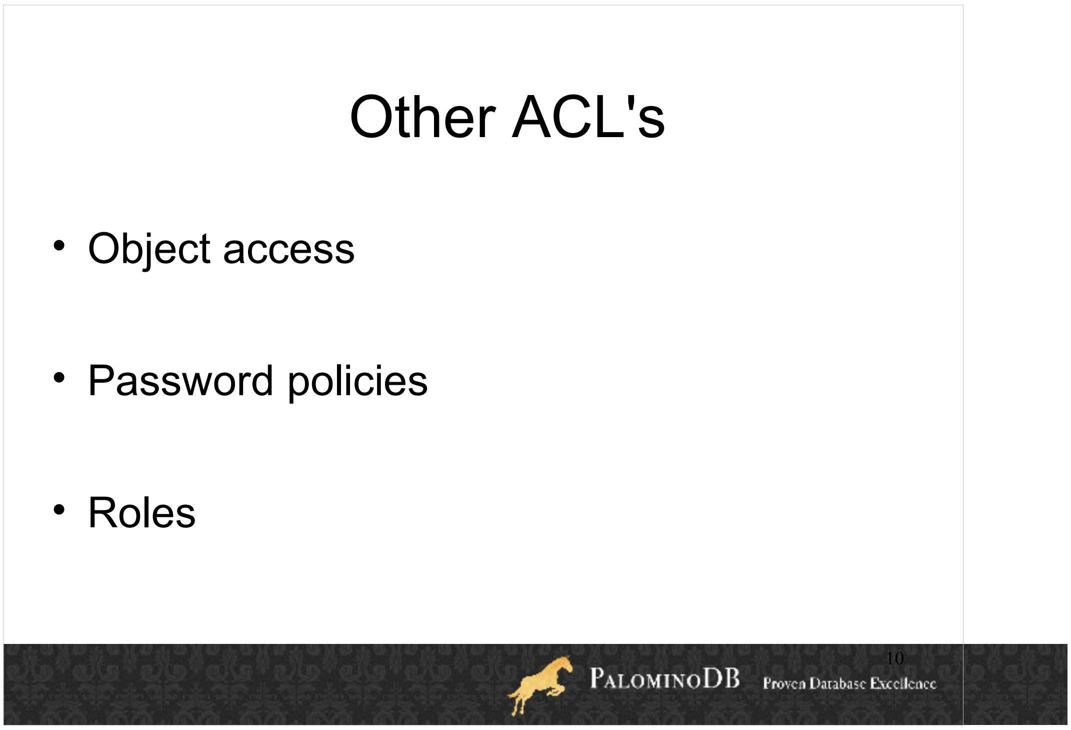


GRANTing Access

```
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

priv_type is the most important thing here, show the doc with the charts in it.



Other ACL's

- Object access
- Password policies
- Roles

Who can acces stored procedures/functions? Views?

You can also allow people to run commands they otherwise wouldn't by using stored procedures/functions, and you can allow them to see partial data by using views – a view definition is a SELECT query, so you can allow people to see certain columns and even certain rows only.

Access from...?

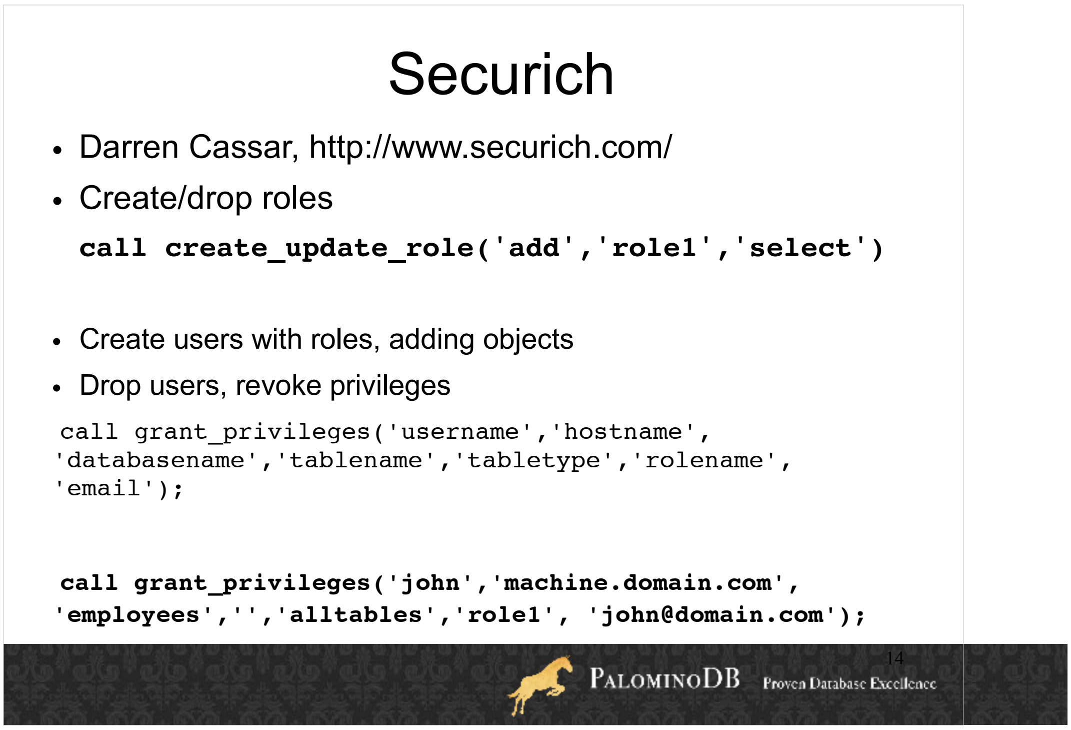
- localhost only, --skip-networking
- firewall
- Who can [attempt to] DOS you?

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');
```

PALOMINODB Proven Database Excellence

There's a good tutorial too!

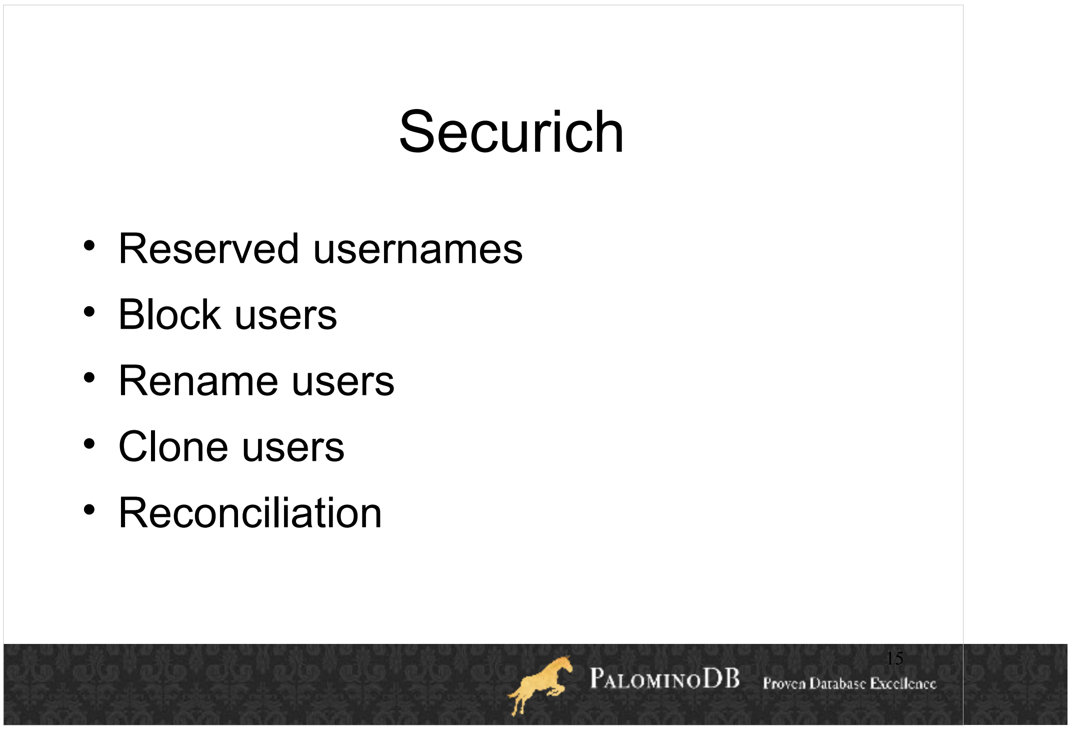
create_update_role either creates or updates the role as necessary

Can on ly drop roles if not in user

Grant privs limitation, if > then truncation happens:

FIELD	MAX LENGTH
username	16
hostname	60
databasename	64
tablename	64
tabletype	16
rolename	60
Emailaddress	50

`Tablename can be tblname, regular expression, '' for all, or a stored procedure name`



Securich

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

Block - Used to block a particular user, terminating his/her connections if necessary and leave the account around to be unblocked if necessary. This is a useful feature for when a user needs temporary rights.

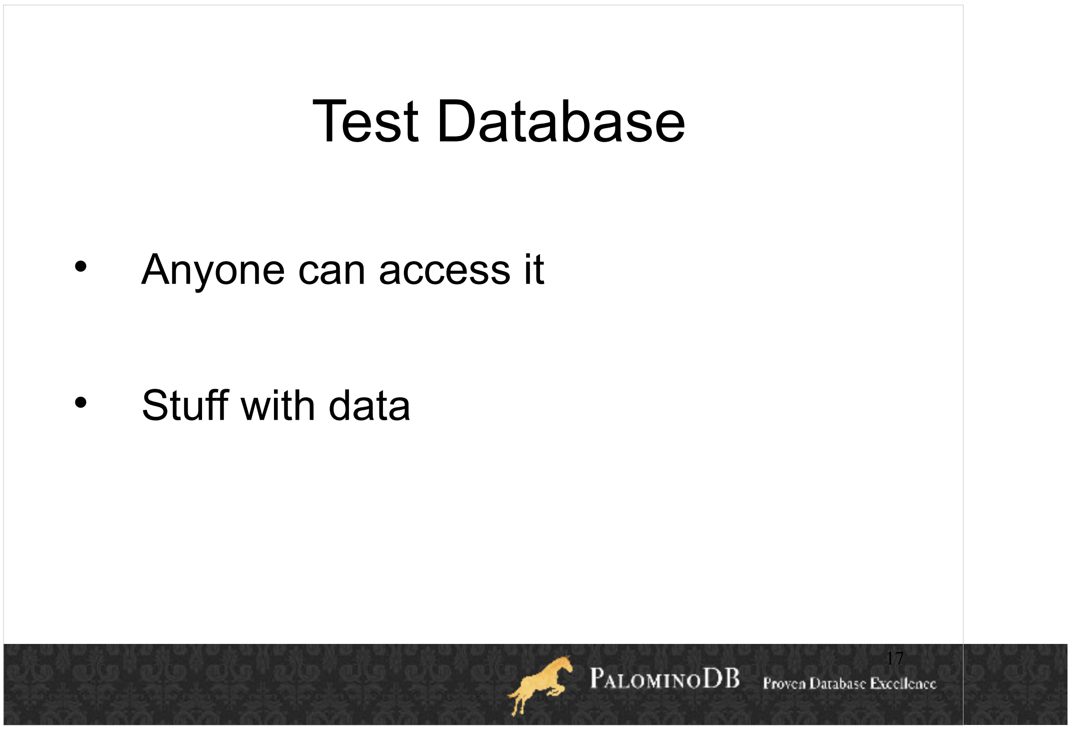
Can reconcile securich's internal db with what's in securich

- password_check();

This is password_check, a procedure used to check for password discrepancies between securich and mysql.

Server Options

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



Test Database

- Anyone can access it
- Stuff with data

PALOMINODB Proven Database Excellence

Cause DOS!

OS Files and Permissions

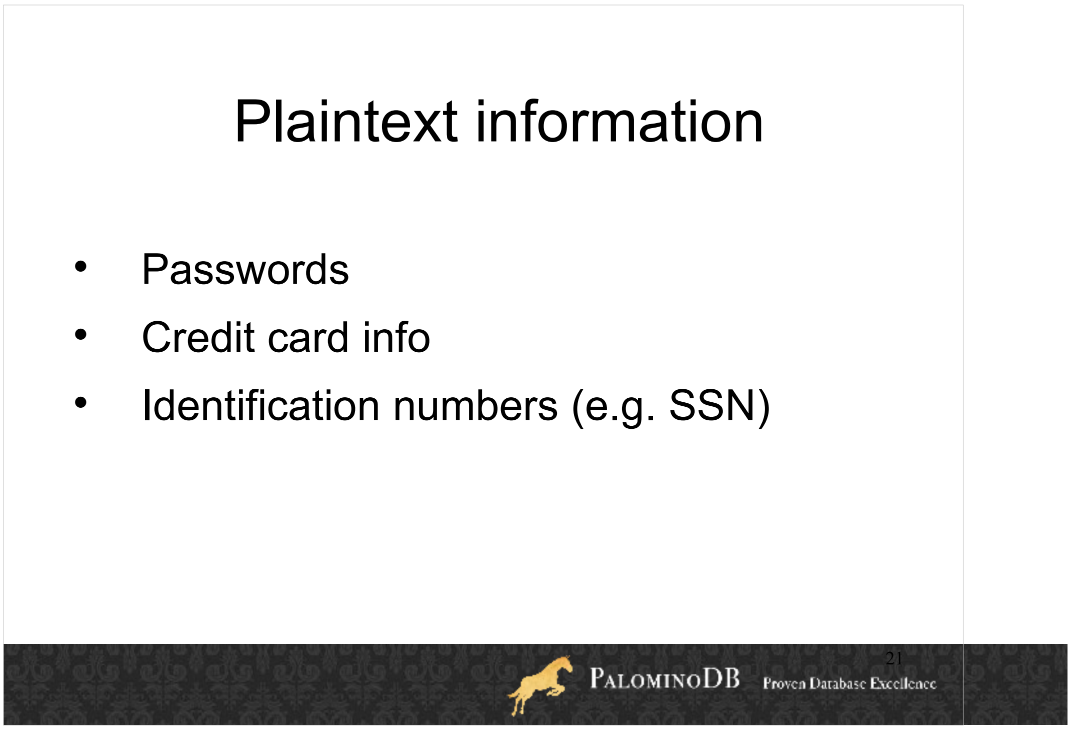
- mysql server user
- mysql server files & logs
- Passwords on commandline
- Office policies/runbook

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)

App users in mysql db, or app password?

mysql db is in memory, referred to every query. Don't make it too big if you don't have to!

User inputted data into mysql internal table == bad. Imagine html or injection in there...bad.

Can be stolen if db is compromised

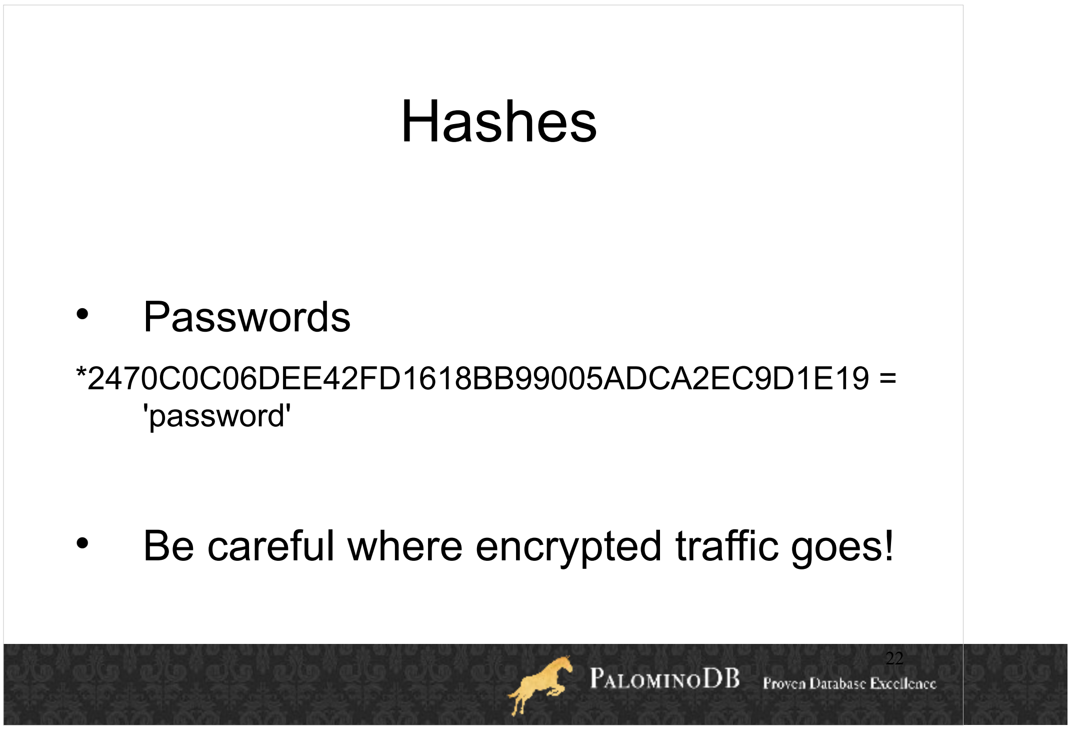
How are they transmitted?

Normally (most important)

On reset

What about hash transmittal – if you transmit the hash unencrypted, and others can get to db, they can get to customer.

Users may use them elsewhere



Hashes

- Passwords

*2470C0C06DEE42FD1618BB99005ADCA2EC9D1E19 = 'password'

- Be careful where encrypted traffic goes!

22

PALOMINODB Proven Database Excellence

Where are you encrypting?

The closer to the input source, the better

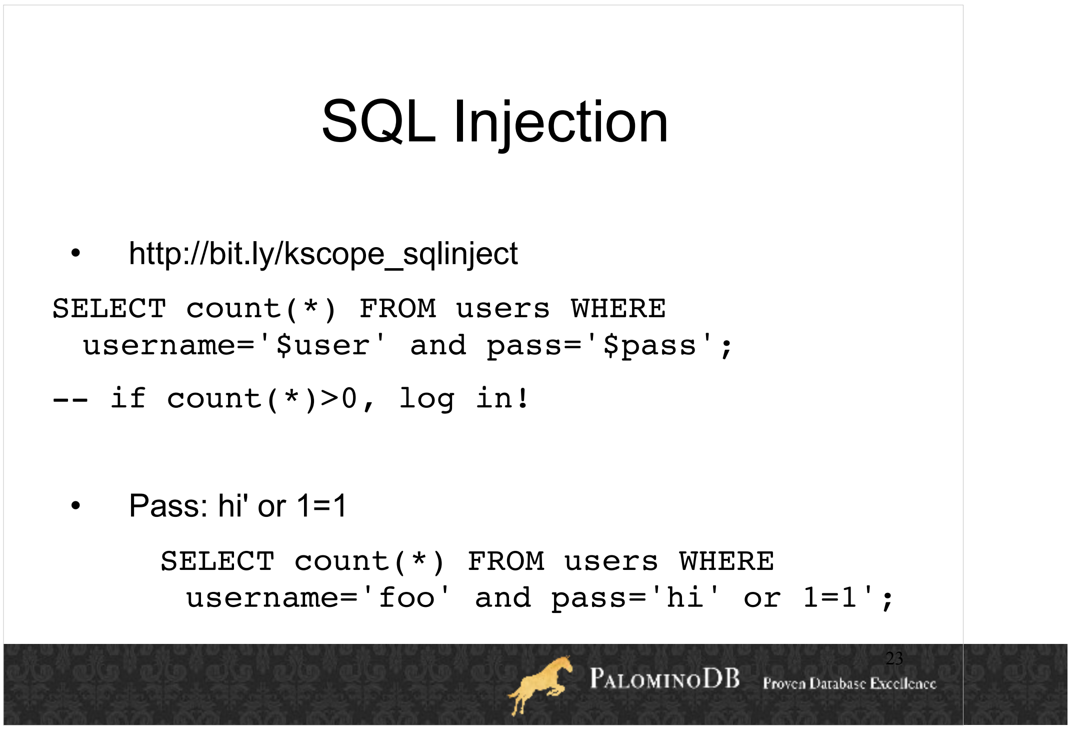
ie, Javascript for HTTP/HTTPS

How are you checking?

Password=hash('foo') ?? hash('foo') then send?

What if I make a web form on MY site that passes info to YOUR site? If checking is only on the page before, there's a problem! Only allow HTTP_REFER from inside...or specific pages.

You can google search for that password hash and find it in Google



SQL Injection

- http://bit.ly/kscope_sqlinject

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

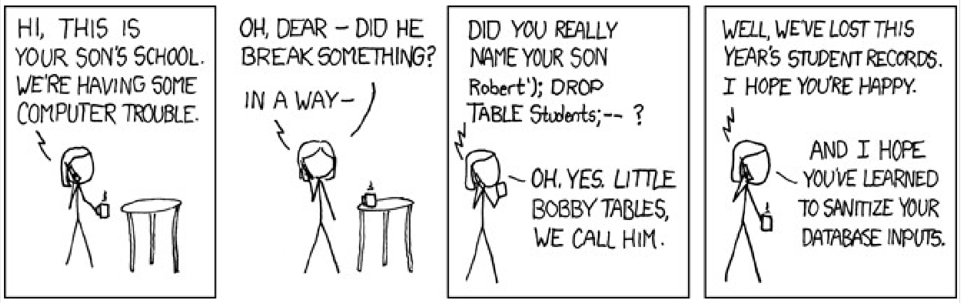
- Pass: hi' or 1=1

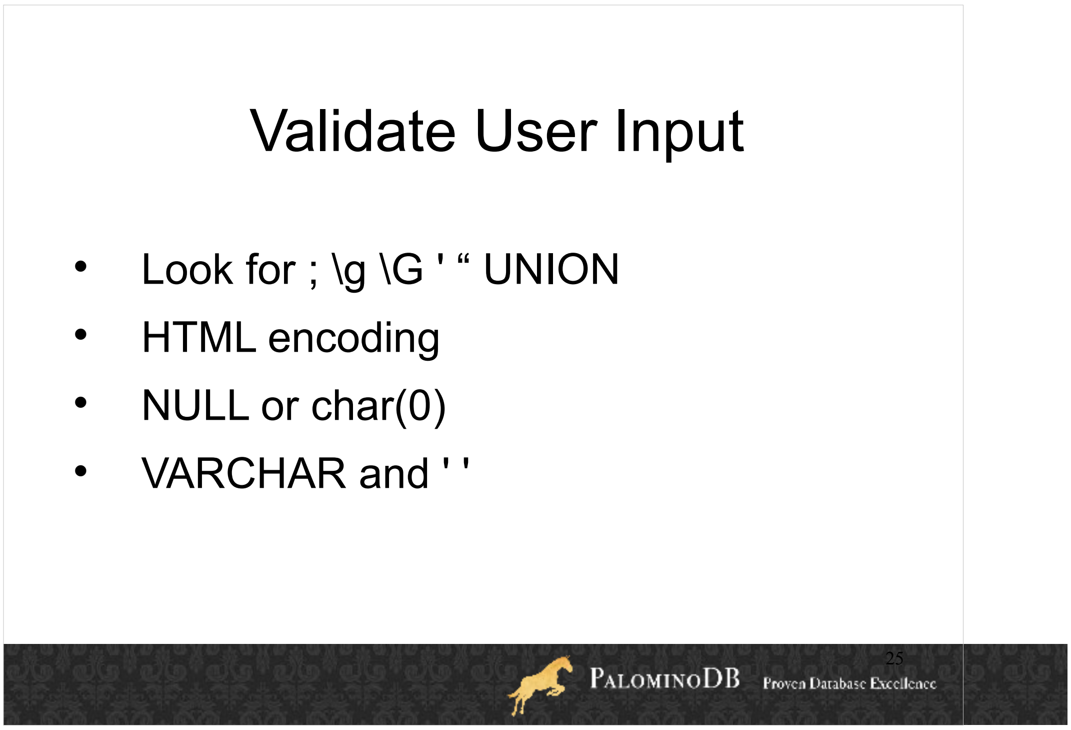
```
SELECT count(*) FROM users WHERE
 username='foo' and pass='hi' or 1=1';
```

I'm not going to talk much about SQL injection, but I'll give an overview:

Let's say you put in your password

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 ' '

PALOMINODB Proven Database Excellence

Disallow or escape ; \g \G " ' UNION (; won't always help, check if multi_query is allowed)

XSS - Do you allow HTML in stored forms? Including javascript? Personal ad and <G> in form renders weird. Not to mention <SCRIPT .... folks put links to their pay-per-click ads, whenever their page is clicked...

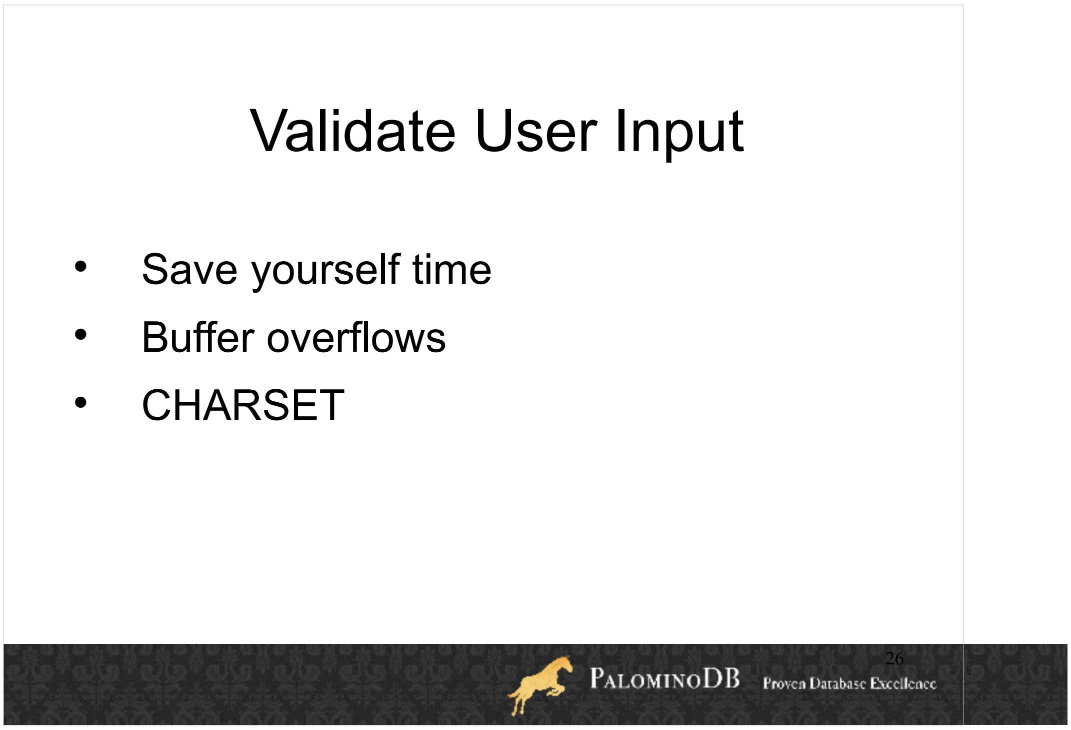
Type 0 XSS -- ?? page's client-side script, ie javascript, access URL request and uses info on that page for something in the current page, can be exploited – can put in another script.

Type 1 XSS – server gets data from client, client can put scripts in there. Reason to strip out HTML

Type 2 XSS – when this stuff is stored.

NULL / char(0) (mysql_query("/*".chr(0)."*/ SELECT * FROM table");)

' ' and varchar



Validate User Input

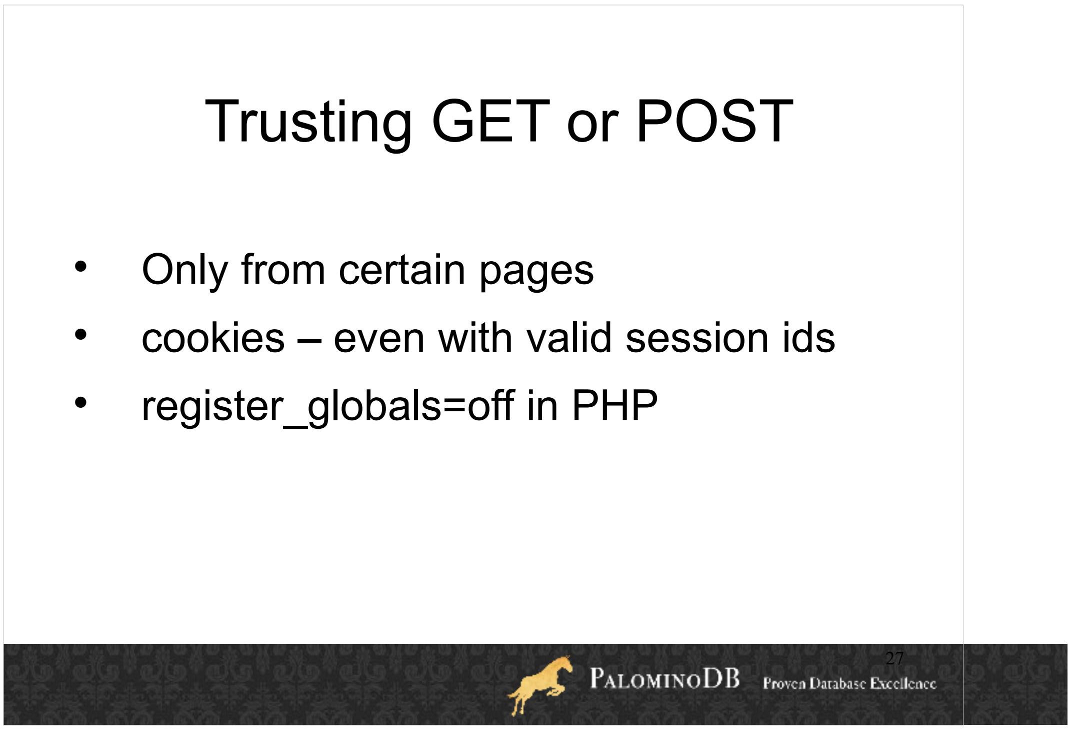
- Save yourself time
- Buffer overflows
- CHARSET

PALOMINODB Proven Database Excellence

Save yourself time, include e-mail checks if you can (php checkdnsrr)

Buffer overflows

What's your CHARSET? (length of INPUT TYPE=TEXT != # of bytes!)



Trusting GET or POST

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

PALOMINODB Proven Database Excellence

Easy to copy your web form and send it

HIDDEN fields too all you have to do is view source!

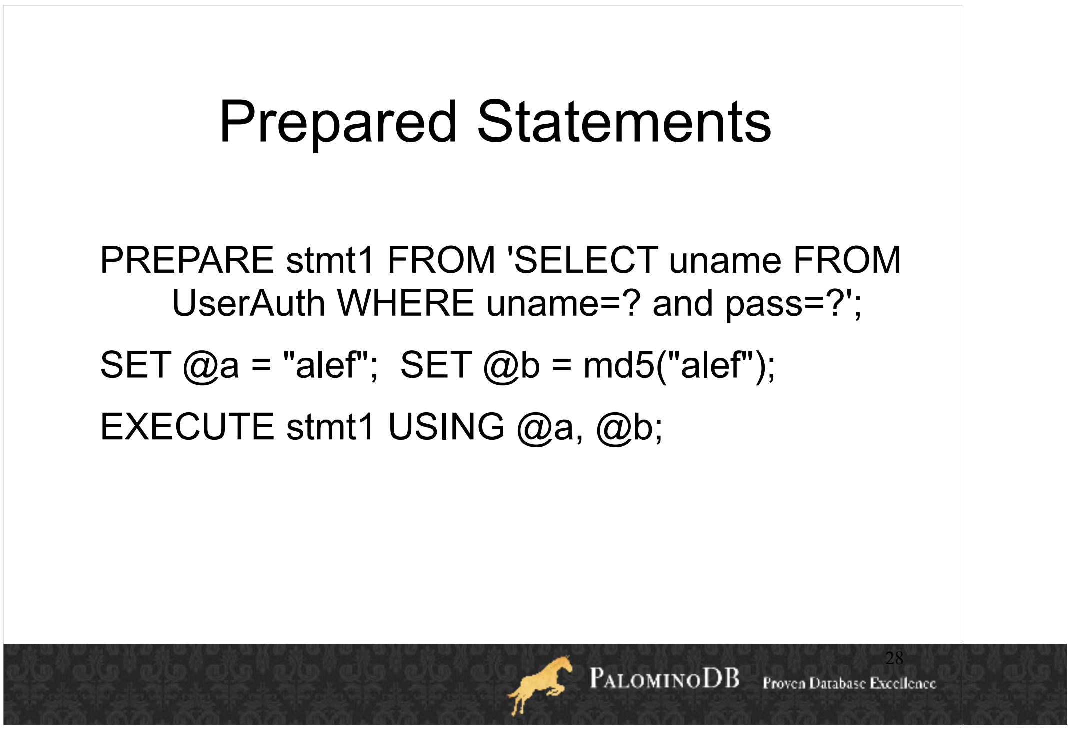
Valid user can do bad stuff, so even with a session ID don't trust unless it's your site

register_globals off in php to avoid POST params in GET context

index.php?$auth=true

Buffer overflows

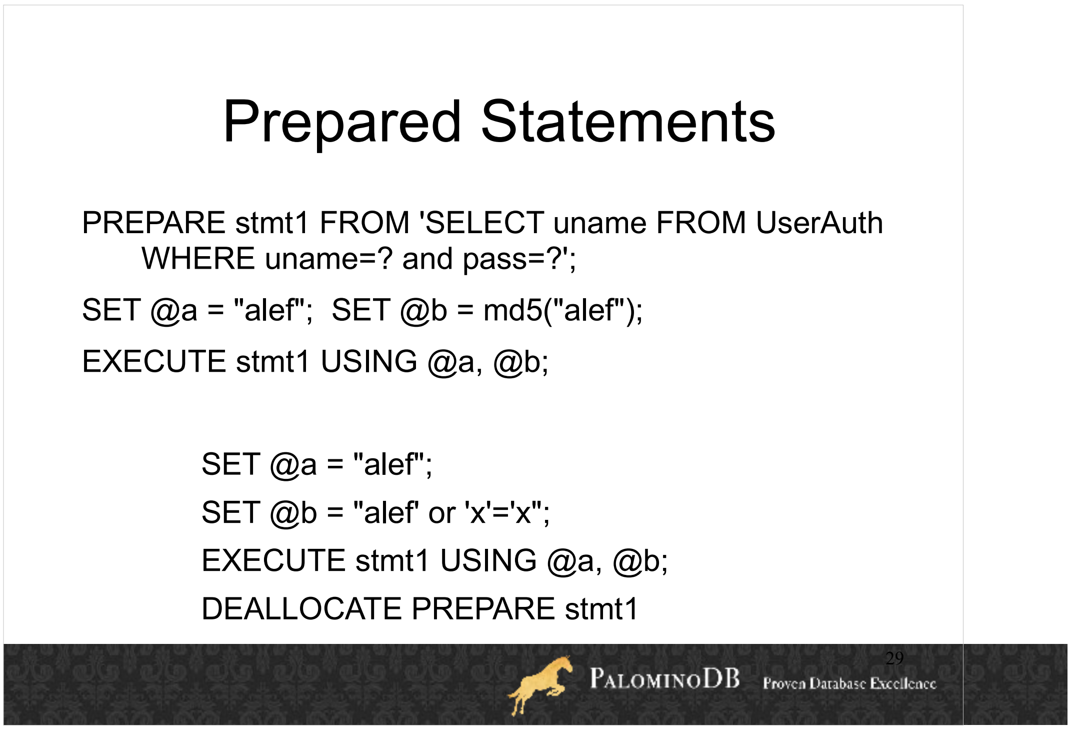
What's your CHARSET?



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;
```

Slow! Caches once per SESSION.



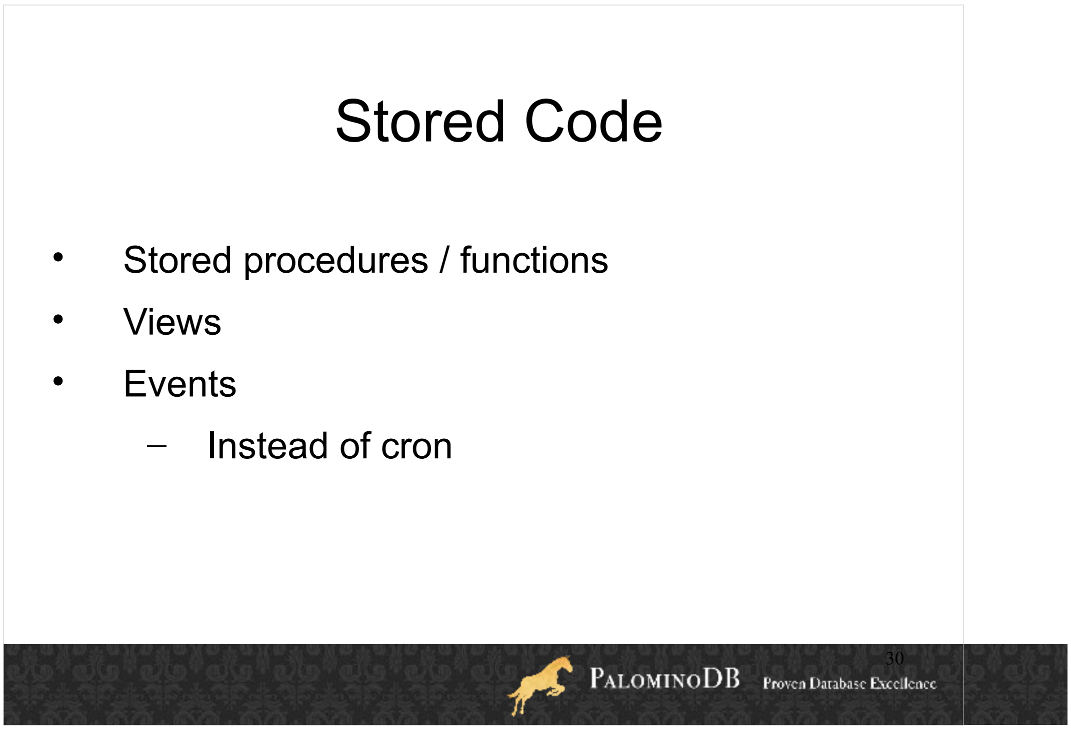
Prepared Statements

```
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 procedures? (MySQL 5)

Can use prepared statements in stored procedures, that's how I do dynamic tables in stored procedures



Stored Code

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

PALOMINODB Proven Database Excellence

Stored procedures? (MySQL 5)

Can use prepared statements in stored procedures, that's how I do dynamic tables in stored procedures

Prepared Statements - Code

Perl

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

Prepared Statements - Code

PHP

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

Prepared Statements - Code

Java

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

Prepared Statements - Code

.NET/C#

```
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() ){
      ...}
   }
```

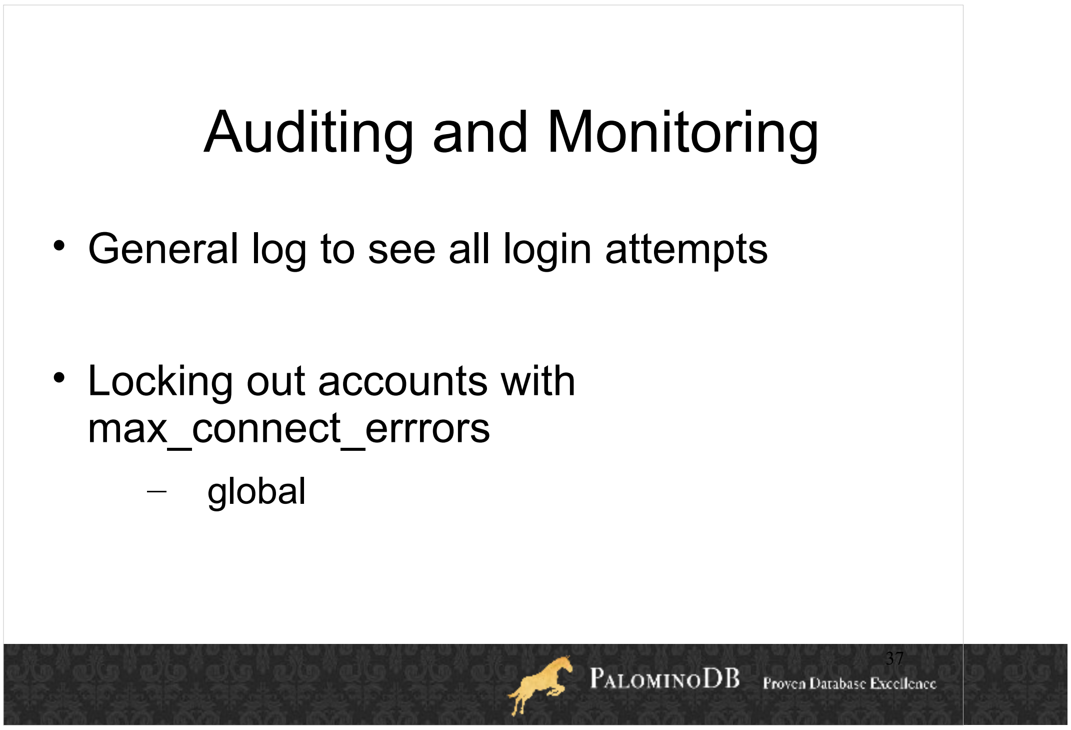
Encryption

- SSL is per-client
- Unencrypted MySQL data streams

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

Auditing and Monitoring

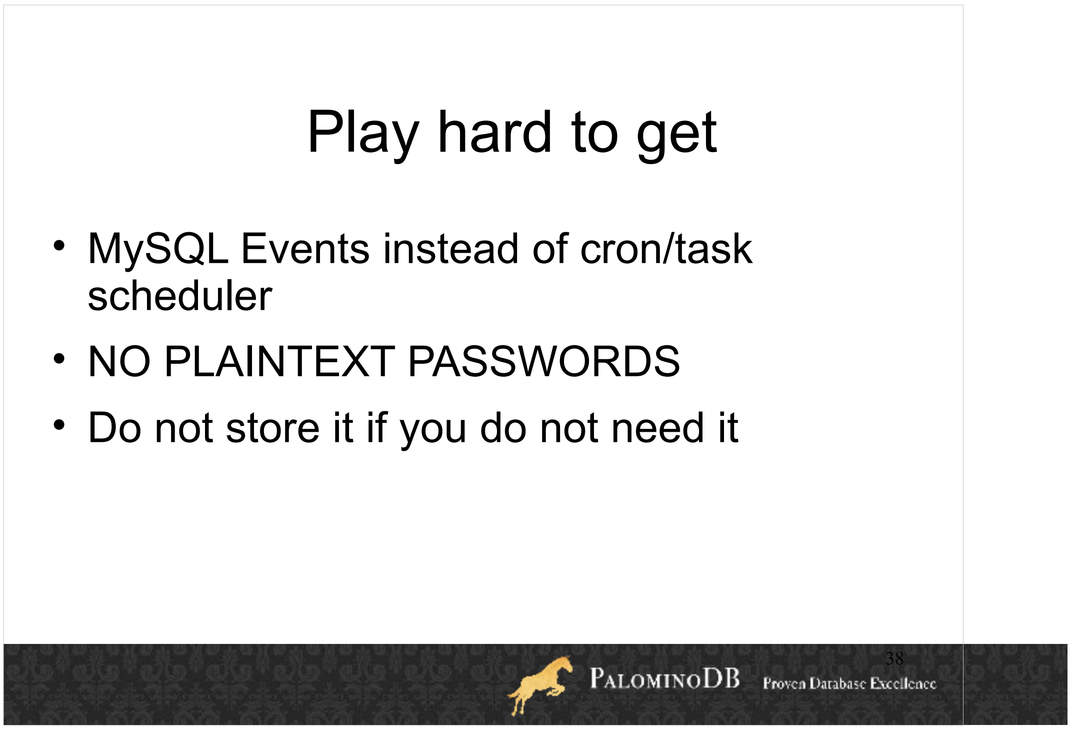
- Prevention is one part of security
- Auditing - review and assess security
- Monitoring – alerting of security issues



Auditing and Monitoring

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

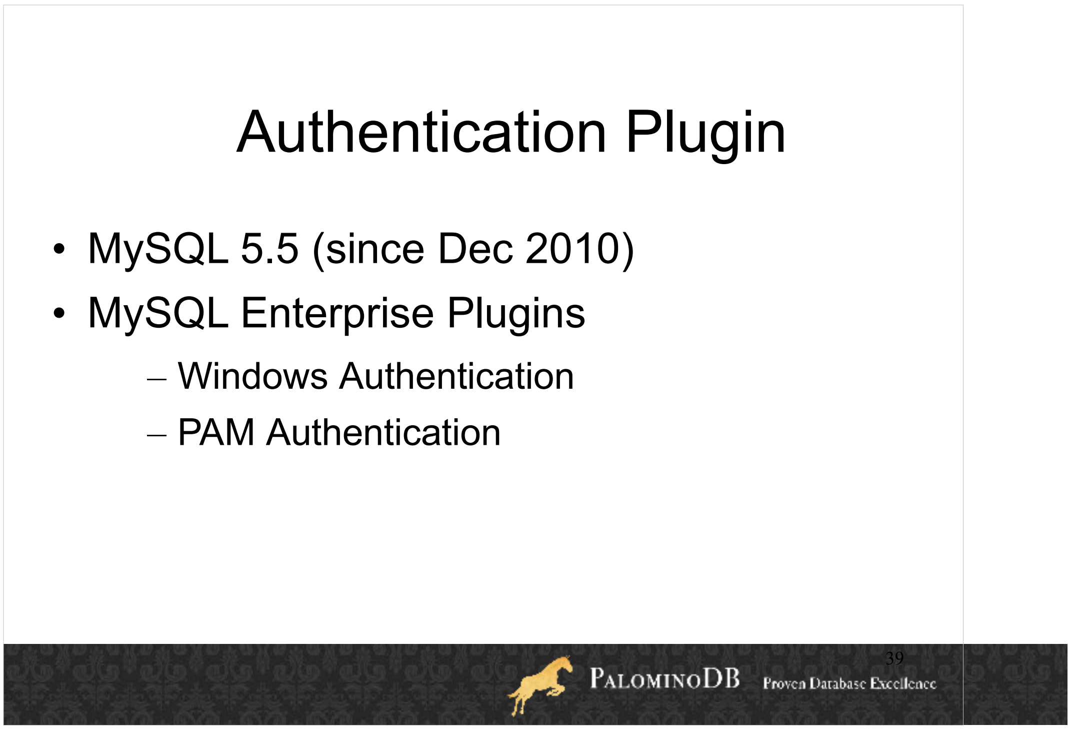
Flush hosts!



Play hard to get

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

No need to store passwords in cron if you use MySQL events. Bonus – it's backed up with the database!



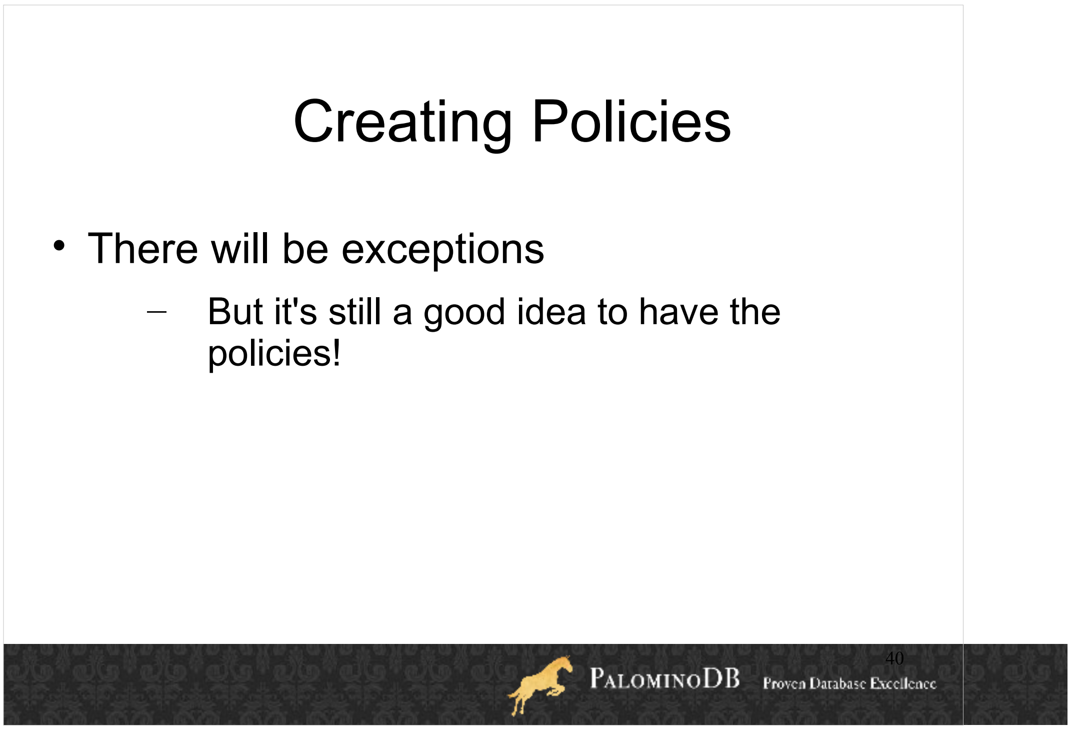
Authentication Plugin

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

39

PALOMINODB Proven Database Excellence

So far these are the only ones, none other yet, but we could use Kerberos auth.



Creating Policies

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

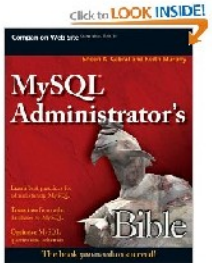
Personal accounts vs. role accounts, how often are each of those passwords changed? When ppl leave? Sometimes it's hard to change app passwords.

Encrypted connections/ replication?

Questions? Comments?

OurSQL podcast

MySQL Administrator's Bible



kimtag.com/mysql

planetmysql.com