MySQL® Conference & Expo

Presented by,
MySQL AB® & O'Reilly Media, Inc.

MySQL Security for Security Audits

Brian Miezejewski
MySQL Principal Consultat

Bio

- Leed Architect ZFour database 1986
- Senior Principal Architect American Airlines Enterprise Data Warehouse 1996-2001
- Director Database Architecture and Systems Travelweb.com (acquired by priceline.com)
- Managed and/or Architected large production systems in Oracle, Informix, MS SQL Server, ObjectStore, ZFour up to 14TB in size.
- MySQL PS Since 2006; Currently leading the Storage Engine and Server Enhancements practice

Experience

- General Accounting audits
  - Usually mainly focused on financial systems
  - Will overflow to the portion of your operational systems that feed the accounting system
- Sarbanes-Oxley (SOX)

  http://en.wikipedia.org/wiki/Sarbanes-Oxley_Act

  http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_public_laws&docid=f:publ204.107
- PCI - Payment Card Industry data security standard

  https://www.pcisecuritystandards.org/

Contents

Keys to Success

Payment Card Industry (PCI) requirements overview

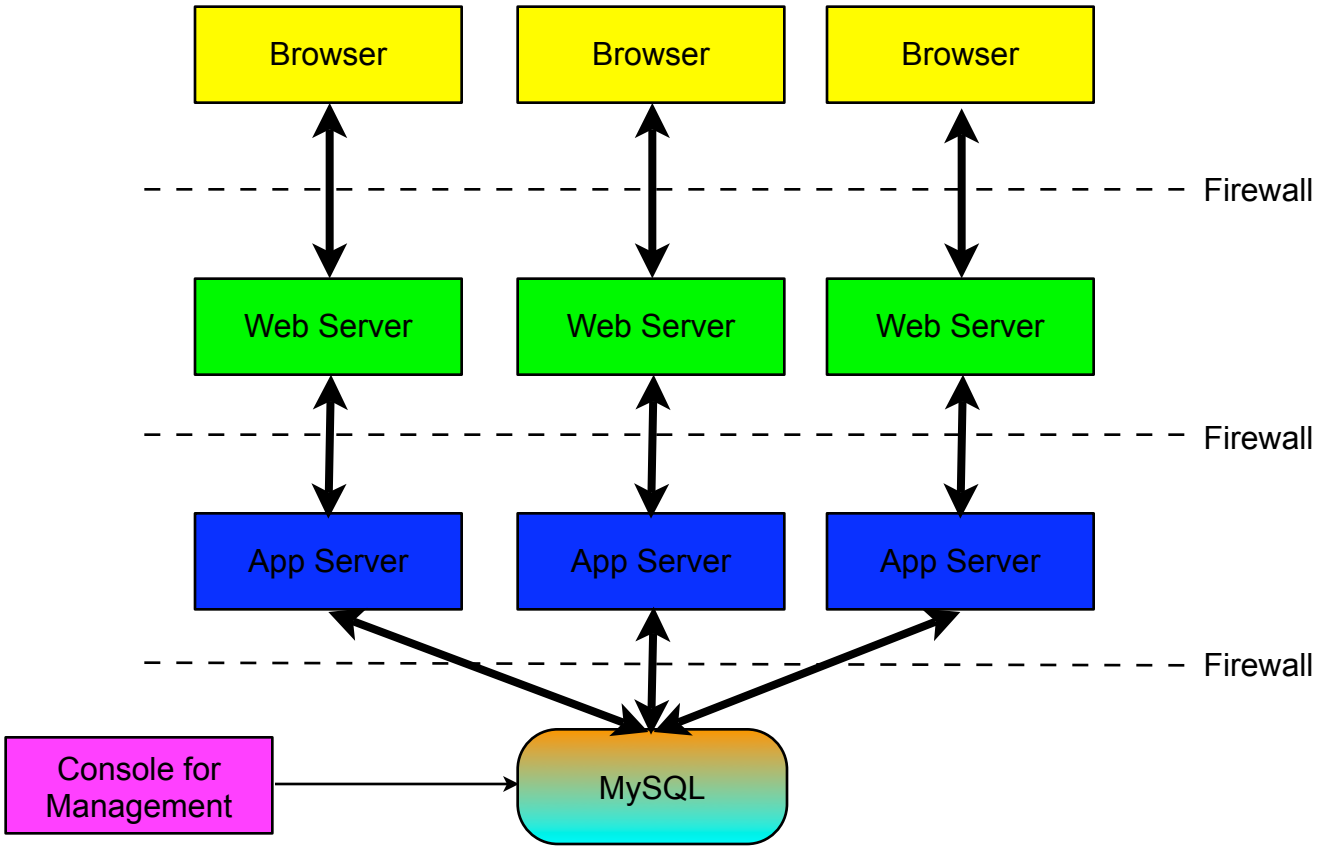
PCI requirements specifically related to MySQL

Other thoughts

Keys To Success

- Responsibility, Ownership, and Accountability
  - Roles (no, not that kind)
- Procedures and Policies
  - user add/create/modify
  - application and data add/create/modify
  - regular security reviews
- Documentation
  - Roles
  - Procedures and Policies
  - Change and Review Logs
  - Log of security related actions

Typical Application (Yea right!)



PCI Requirements I

Build and Maintain a Secure Network

1. Install and maintain a firewall configuration to protect cardholder data
2. Do not use vendor-supplied defaults for system passwords and other security parameters

Protect Cardholder Data

3. Protect stored cardholder data
4. Encrypt transmission of cardholder data across open, public networks

PCI Requirements II

Maintain a Vulnerability Management Program

5. Use and regularly update anti-virus software
6. Develop and maintain secure systems and applications

Implement Strong Access Control Measures

7. Restrict access to cardholder data by business need-to-know
8. Assign a unique ID to each person with computer access
9. Restrict physical access to cardholder data

PCI Requirements III

Regularly Monitor and Test Networks

10. Track and monitor all access to network resources and cardholder data
11. Regularly test security systems and processes

Maintain an Information Security Policy

12. Maintain a policy that addresses information security

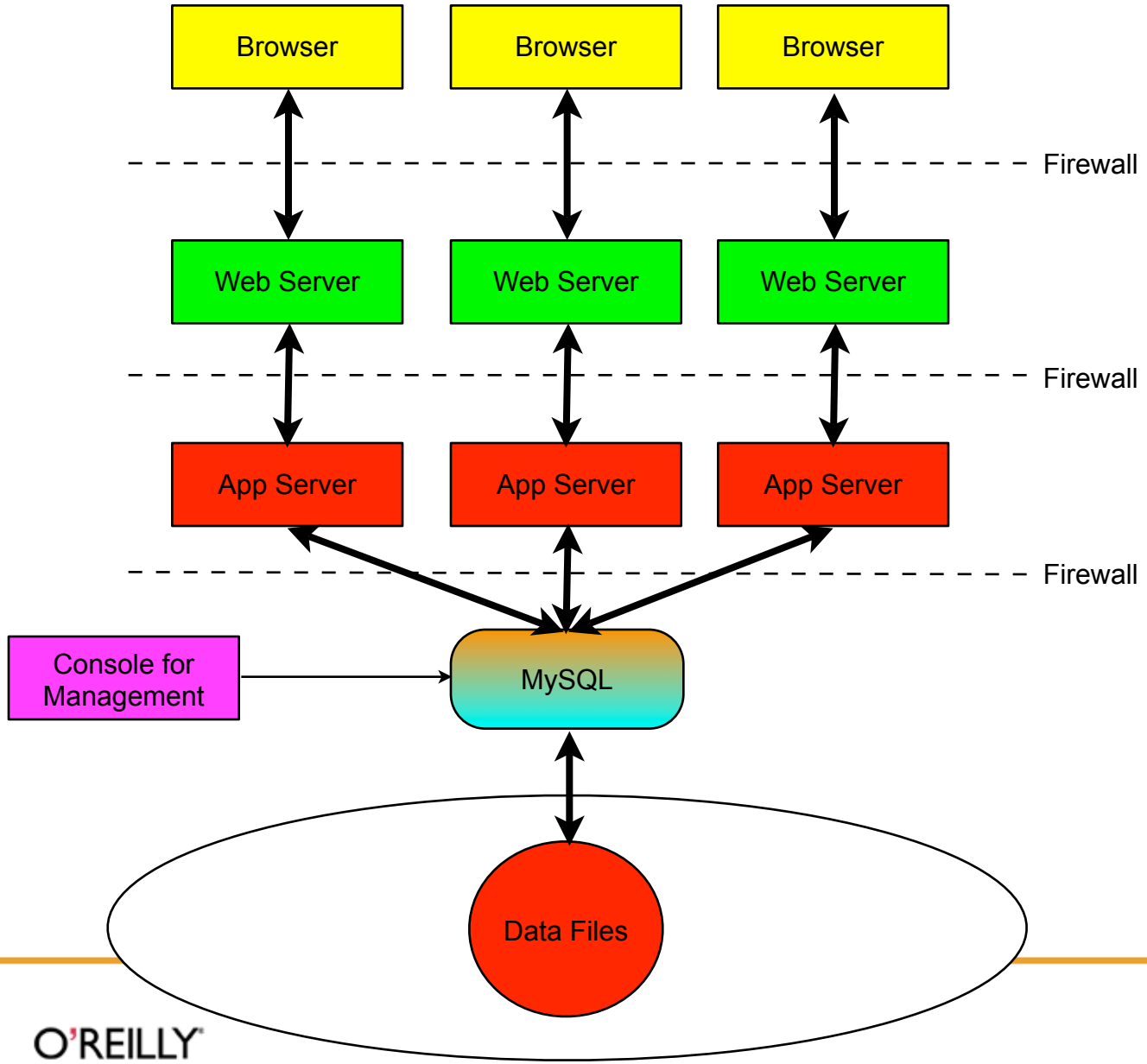
2. Do not use vendor-supplied defaults for system passwords and other security parameters

- MySQL installs with 3 or more default accounts:
  - SELECT User, Host, Password from mysql.user ;
  - http://dev.mysql.com/doc/refman/5.1/en/default-privileges.html

3. Protect stored cardholder data I

- Credit card numbers ***must*** be protected (encrypted) if they appear on storage (disk, tape, usb drive, etc.)
- Related customer data must also be protected if stored with the CC number
- Best place to encrypt data is ***in the application***
  - Encrypts communication of data
  - Encrypts accidental logging of the data
  - Encrypts data on disk
  - Separates the encryption from the data
- Consider using a public key on the application and giving the private key to accounting

Typical Application of today (Yea right!)



3. Protect stored cardholder data II

- If you ***have to use*** MySQL encryption functions be very careful with your logs:
  - Do not use binary logs prior to 5.1
    - Optionally encrypt disk with bin-log
    - Or increase the hardening of the database server
  - Use row-based replication in 5.1 and after
  - Do not turn on general query log
  - Be careful with slow query log
  - Don't log at application or between DB and app, i.e. proxy

3. Protect stored cardholder data II

```
mysql> insert into tab values( aes_encrypt('mypassword', 'mykey')) ;
Query OK, 1 row affected (0.00 sec)
```

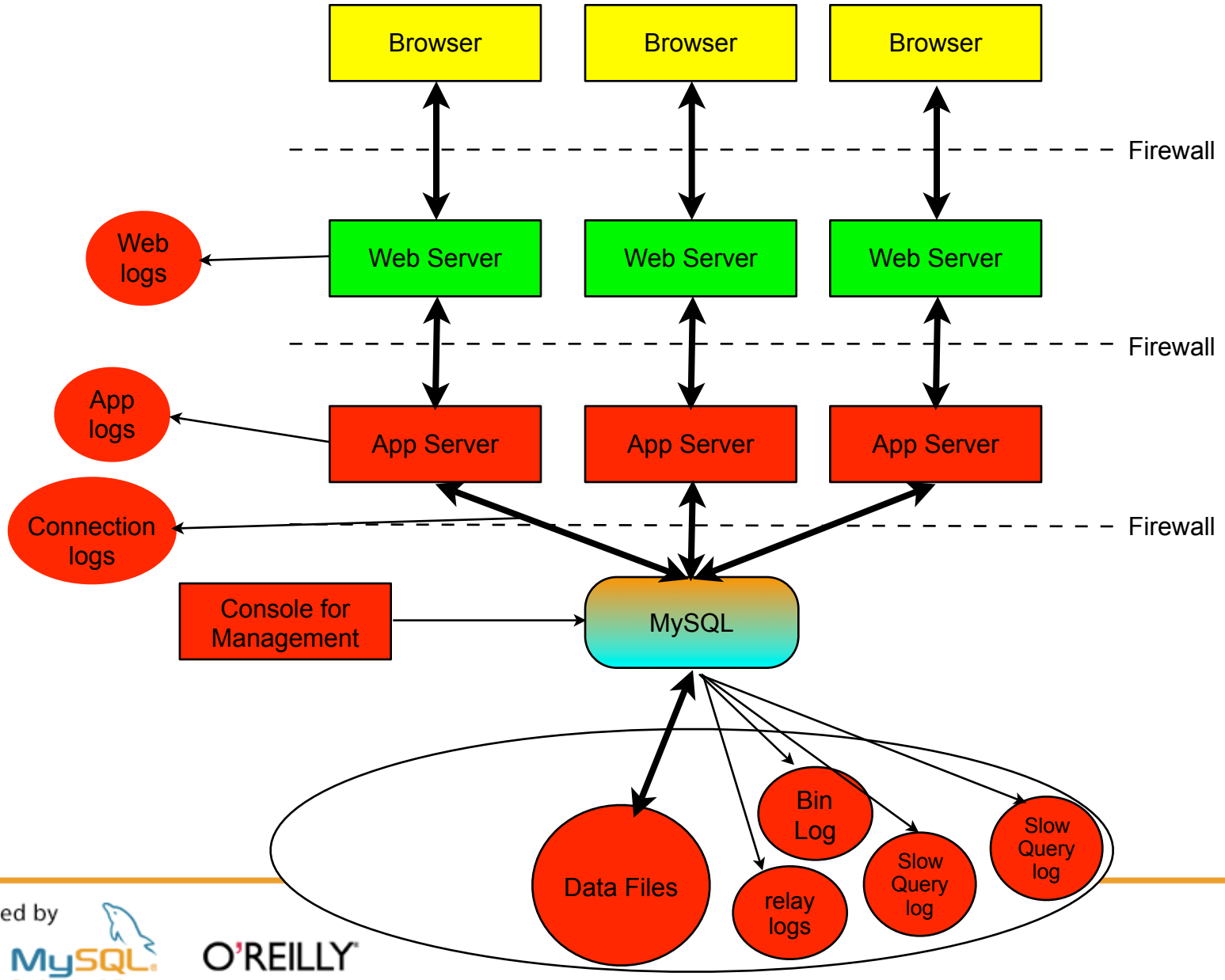
--------- binary log --------

```
SET TIMESTAMP=1208148932/*!*/;
insert into tab values( aes_encrypt('mypassword', 'mykey'))/*!*/
```

--------- general query log --------

```
080413 23:55:32              6 Query
insert into tab values( aes_encrypt('mypassword', 'mykey'))
```

Typical Application of today (Yea right!)



3. Protect stored cardholder data III

- Public key encryption makes it easier to hide private key, but not practical for all applications.
- Give only the minimal security access needed for a person to do their job, MySQL has 30 security privileges, ***learn them and use them!***
- PCI good source - ***review it!***
- Have a documented policy, follow it, log the security events, and manage security change.

3. Protect stored cardholder data IV

- Watch the Logs when when you do a GRANT using the mysql command line tool

```
mysql> grant all on *.* to 'me'@'localhost' identified by 'pwd' ;
Query OK, 0 rows affected (0.00 sec)
```

----- General Query Log -----

```
080414 13:47:12        1 Query
grant all on *.* to 'me'@'localhost' identified by 'pwd'
```

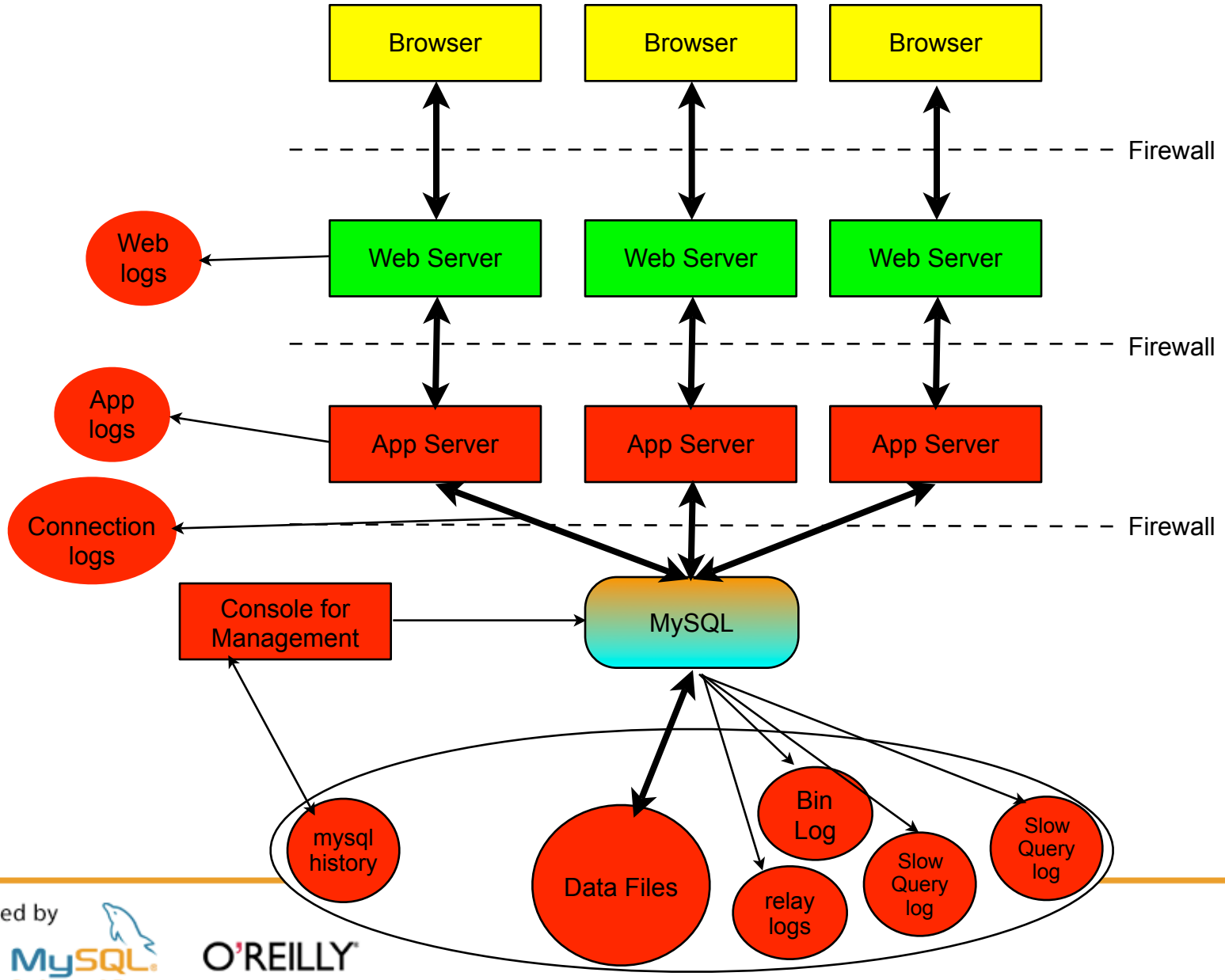
----- Bin Log -----

```
grant all on *.* to 'me'@'localhost' identified by 'pwd'/*!*/;
```

----- .mysql_history -----

```
grant all on *.* to 'me'@'localhost' identified by 'pwd' ;
```

Typical Application of today (Yea right!)



3. Protect stored cardholder data V

- Use mysqladmin in special account:

mysqladmin -u me --password=pwd password ppp

----- General Query Log -----

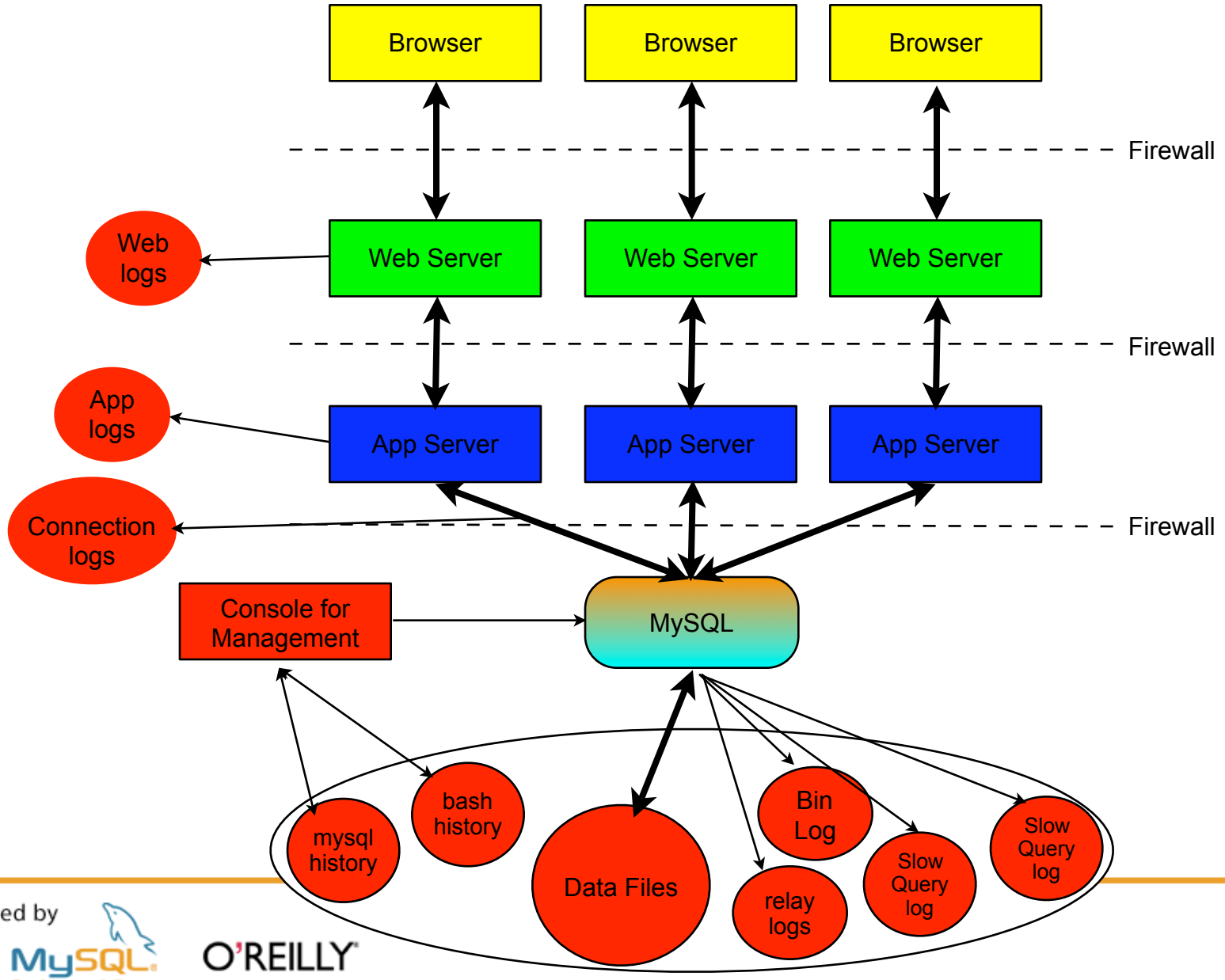
```
80414 14:05:07      2 Connect    me@localhost on
                    2 Query      SHOW VARIABLES LIKE 'old_passwords'
                    2 Query      set sql_log_off=1
                    2 Quit
```

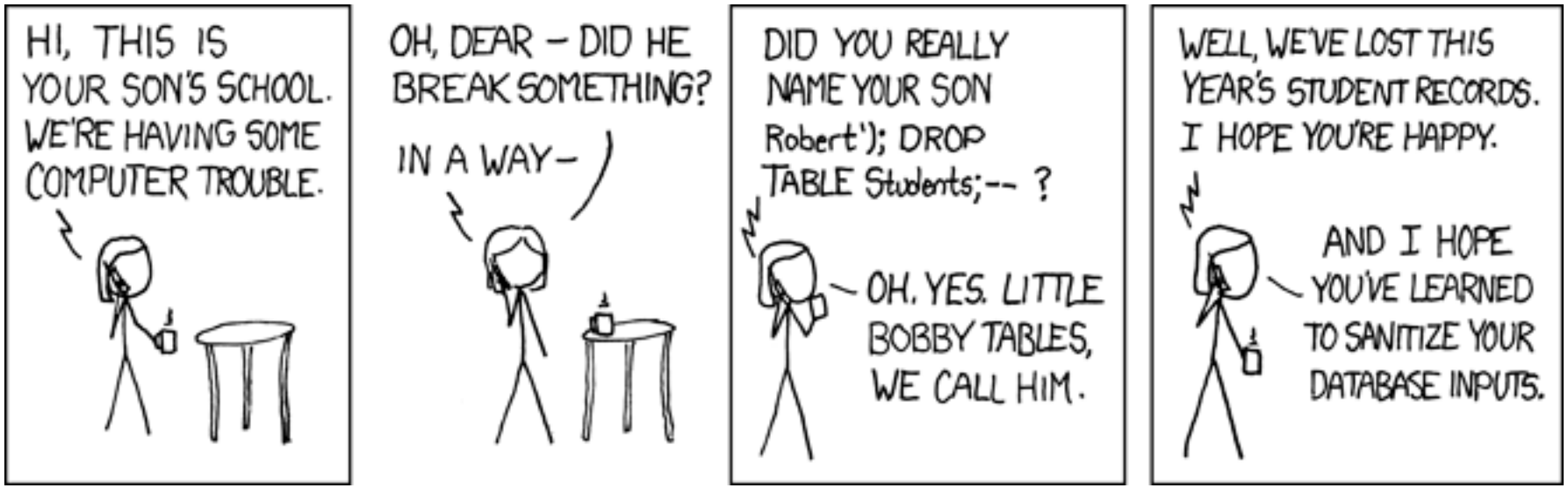
----- Bin Log ----

```
SET TIMESTAMP=1208199907/*!*/;
SET PASSWORD FOR
'me'@'localhost'='*9CF9BF8B3B3440167987159A2DCCE584D30D92E7'/*!*/;
```

- Need to disable history - set -o history
- Never use “mysql -u user -ppassword
- Write scripts to look at users .bash_history to check

Typical Application of today (Yea right!)




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.

xkcd.com

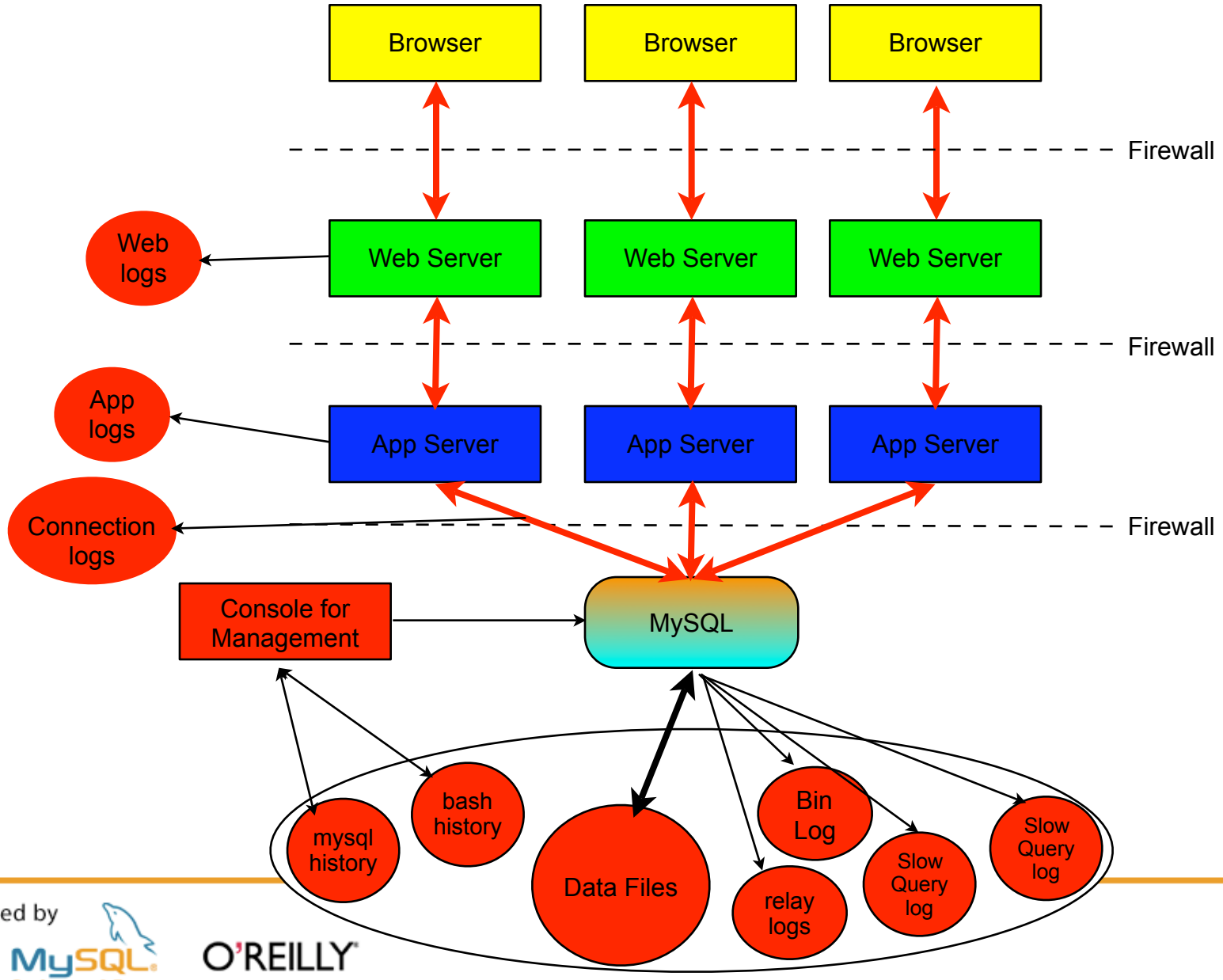
3. Protect stored cardholder data VI

- Protection with stored procedures
  - Create stored procedures for all operations and assigned each the minimum privilege it needs to to do its job.
  - Create a separate with only enough privilege to run the stored procedures
- Minimize security access for all monitoring tools access
- ***Protect your encrypted data, the more one has, the easier it is to crack!***

4. Encrypt transmission of cardholder data across open, public networks

- Note the clause “open, public networks”
- Can have separate closed dedicated network between application and database
  - Still needs firewall!
- Already taken care of if you do encryption in the application
- Use ssl connections

Typical Application of today (Yea right!)



6. Develop and maintain secure systems and applications I

- Have a regular process for identifying and applying security updates patches
  http://dev.mysql.com/tech-resources/articles/security_vulnerabilities.html
  http://forge.mysql.com/wiki/Security_Vulnerabilities_In_MySQL_Server
- Separate roles as much as possible
- Always perform security reviews for every application change
- Beware of extern applications that cache database data between the application, i.e memcache

8. Assign a unique ID to each person with computer access

- MySQL gives ***no help with:***
  - Aging passwords - PCI 90 days (SP)
  - No reuse of the last four passwords (SP)
  - Password quality checking
    - 7+ Chars
    - Alpha and Numeric
    - Don't use valid words: Use phrases - "I love to work on databases for MySQL and I think C++ is great" becomes "Il2woDBfM&ItC++ig8"

10. Track and monitor all access to network resources and cardholder data

- Help coming in 6.0, audit logging plugin.
  http://forge.mysql.com/worklog/task.php?id=3771
- Create script to monitor error log for failed logins and disable accounts based on failures
- Use triggers to monitor inserts, updates and deletes.
- Use stored procedures with built-in logging (to a table) to log accesses individual CC data.

Data Security Vulnerabilities

- Reasons for Vulnerability
  - ✓ Bad Policies or processes
  - ✓ Bad Design
  - ✓ Bad Software Configuration
  - ✓ Software Flaws
- Classes of Vulnerabilities
  - ✓ Invalid access - Hackers, corrupt or inept employees
  - ✓ Data in motion - Network connection
  - ✓ Static data - Disk storage, backups, logs, etc.

Brian Miezejewski
bmiezejewski@mysql.com