MySQL®
Conference & Expo

Securely and Seamlessly Deploy MySQL with your Application

Lee Stigile, MySQL Systems Engineer

John David Duncan, MySQL Consulting Engineer

MySQL Conference & Expo

Agenda

- OEM Customers
- Reasons to Embed
- Commercial versus GPL Versions
- Deployment Options
- Files to Distribute
- OEM Security Challenges
  - Vulnerabilities
  - Tactics

MySQL Conference & Expo

Who embeds MySQL?

- Adobe Creative Suite
- Simply Accounting by Sage
- Quest Foglight (Network Monitoring)
- Symantec

- Many more customers:
  http://www.mysql.com/customers/embedded

MySQL Conference & Expo

Why embed?

- It's easier for your customer
- It shortens your customer's purchase process
- It's easier to support a pre-configured database

Why embed MySQL?

- Low total cost of ownership
- Full range of support offerings
- Full featured product versus express offerings
- Your customers IT organization will be familiar with MySQL

MySQL Conference & Expo

What version to embed?

MySQL Community Server

Community GPL

- Available under GPL
- Does not support "Commercial Distribution"

MySQL Embedded Server

Commercial Licence

- Sold to OEMs and ISVs
- Frees from applying GPL rules to source code and intellectual property
- Available for bundled or embedded use models
- OEM Support

MySQL Enterprise

Enterprise GPL

- Includes monitoring tools
- Monthly updates and quarterly service packs
- Around-the-clock support for the MySQL Enterprise server

MySQL Conference & Expo

You may need the commercial version if:

- Your application is not licensed under the GPL
- You distribute a MySQL server or MySQL connector with your non-GPL application

- References

http://www.gnu.org/licenses/gpl-faq.html

http://www.gnu.org/licenses/gpl-faq.html#IfLibraryIsGPL

http://www.gnu.org/licenses/gpl-faq.html#MereAggregation

MySQL Conference & Expo

Our recipe to build the commercial version:

- Take full GPL source
- Remove readline, pstack and netware directories
- Change license header
- “Commercial” version uses libedit instead of readline
- Version comment is “commercial”

MySQL Conference & Expo

Connectors

http://www.mysql.com/products/connector/

- **Java:** Connector/J or MXJ
- **.Net:** Connector/Net
- **C++:** Connector/C++
- **C:** libmysql, libmysqld, Connector/C
- **ODBC:** Connector/ODBC
- *And many others for PHP, Python, etc.*

MySQL Conference & Expo

Three deployment options

- Use the deeply embedded library
- Execute a silent windows install
- Launch MySQL from the command-line

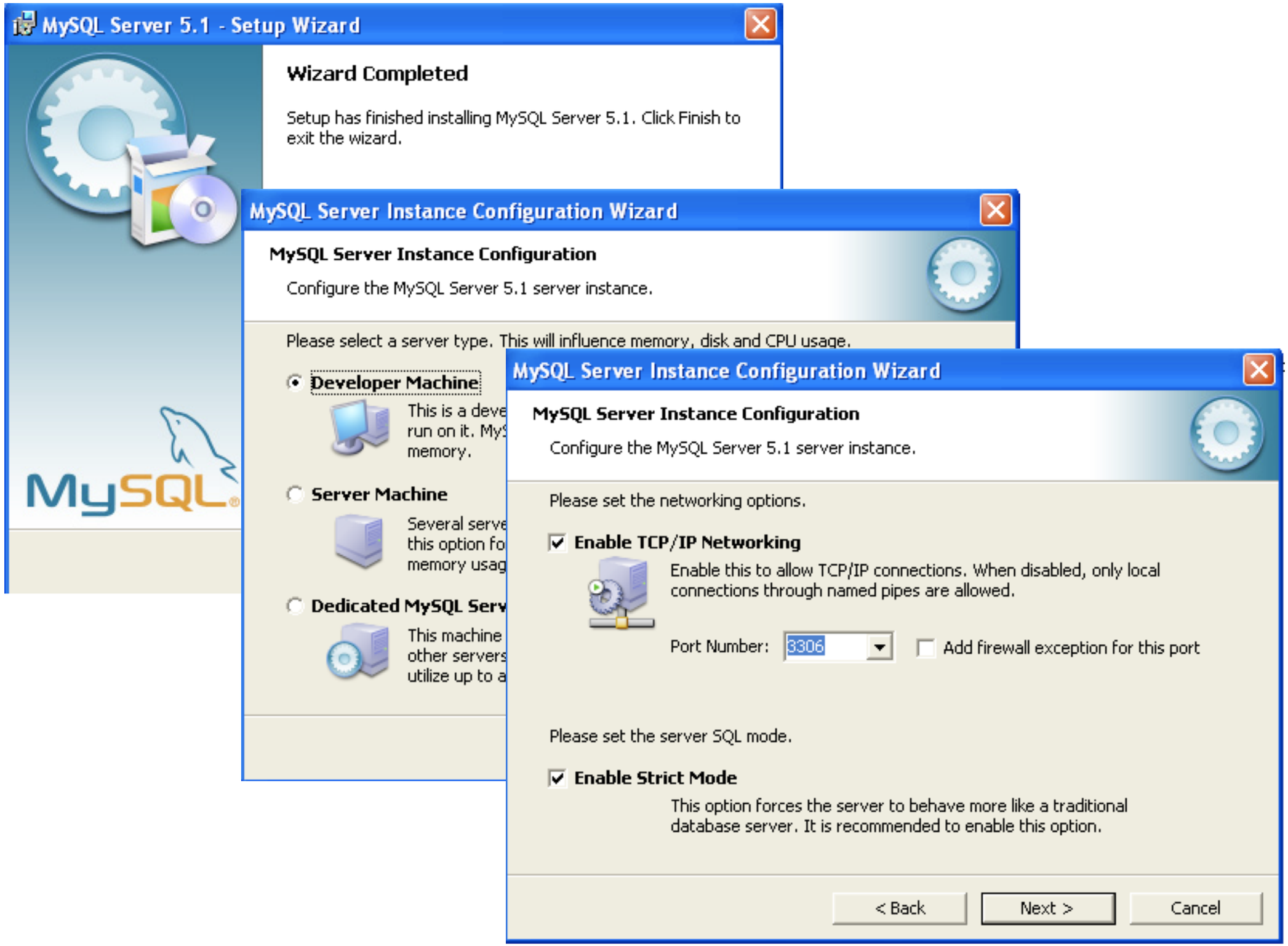
MySQL Conference & Expo

Option #1: Deeply Embedded Library

- Link the embedded library (libmysqld.dll) into your application
- Benefits: easy to deploy, small footprint
- Disadvantages: requires C or C++ calls to library, no external IP connections allowed

- For more information, see Anders Karlsson's notes from Tuesday's presentation: MySQL Embedded -- Getting Started with libmysqld

Option #2: Silent Windows Install



Option #2: Silent Windows Install

- Two step process

#1. Run msiexec, the windows installer, against the MySQL installation file

-Adds to start menu, adds registry values, add/remove from control panel

```
C:\>msiexec /q /log install.txt /i
    mysql-advanced-5.1.32-win32.msi
    datadir=”c:\installs\myapp” installdir=”c:\installs\myapp”
```

datadir=“c:\...” installdir=“c:\...”

Option #2: Silent Windows Install

- Two step process (continued)

  #2. Run MySQLInstanceconfig.exe, which is located in bin directory

  -Configures service, prepares my.ini, sets the root password, starts the service

```
C:\>MySQLInstanceConfig.exe -i -q
 “-lC:\mysql_install_log.txt”
 “-nMySQL Server 5.1.234” -pC:\installs\myapp”
 -v5.1.234  “tc:\installs\myapp\my-small.ini”
 “-cC:\mytest.ini ServerType=DEVELOPMENT
 DatabaseType=MIXED ConnectionUsage=DSS
 Port=3311 ServiceName=MySQLCust RootPassword=1234
```

Important link for MySQLInstanceconfig parameters:

http://svn.mysql.com/svnpublic/mysql-instance-config/trunk/readme.txt

Option #3: Launch MySQL from the command-line

- Popular method

```
C:\>"c:\mysqld.exe" --datadir="c:\program files\Myapp\data"
                    --language="c:\program files\Myapp\share\english”
```

- Possible options:
  - --skip-networking
  - --skip-innodb
  - --lower-case-table-names=1
  - --skip-grant-tables

Files to distribute

- Required:
  binary (mysqld.exe)
  errmsg.sys (--language=“c:\...”)

```
shell>"mysqld.exe" --datadir="c:\program files\Myapp\data"
                   --language="c:\program files\Myapp\share\english"
```

- Optional:
  Configuration files (--defaults-file=my.cnf)
  Character set and collation files

Data files to distribute -- logical

- Use mysqldump to create .sql script to import on target server

mysqldump -uroot --all-databases > data.sql

mysql < data.sql

http://dev.mysql.com/doc/refman/5.1/en/mysqldump.html

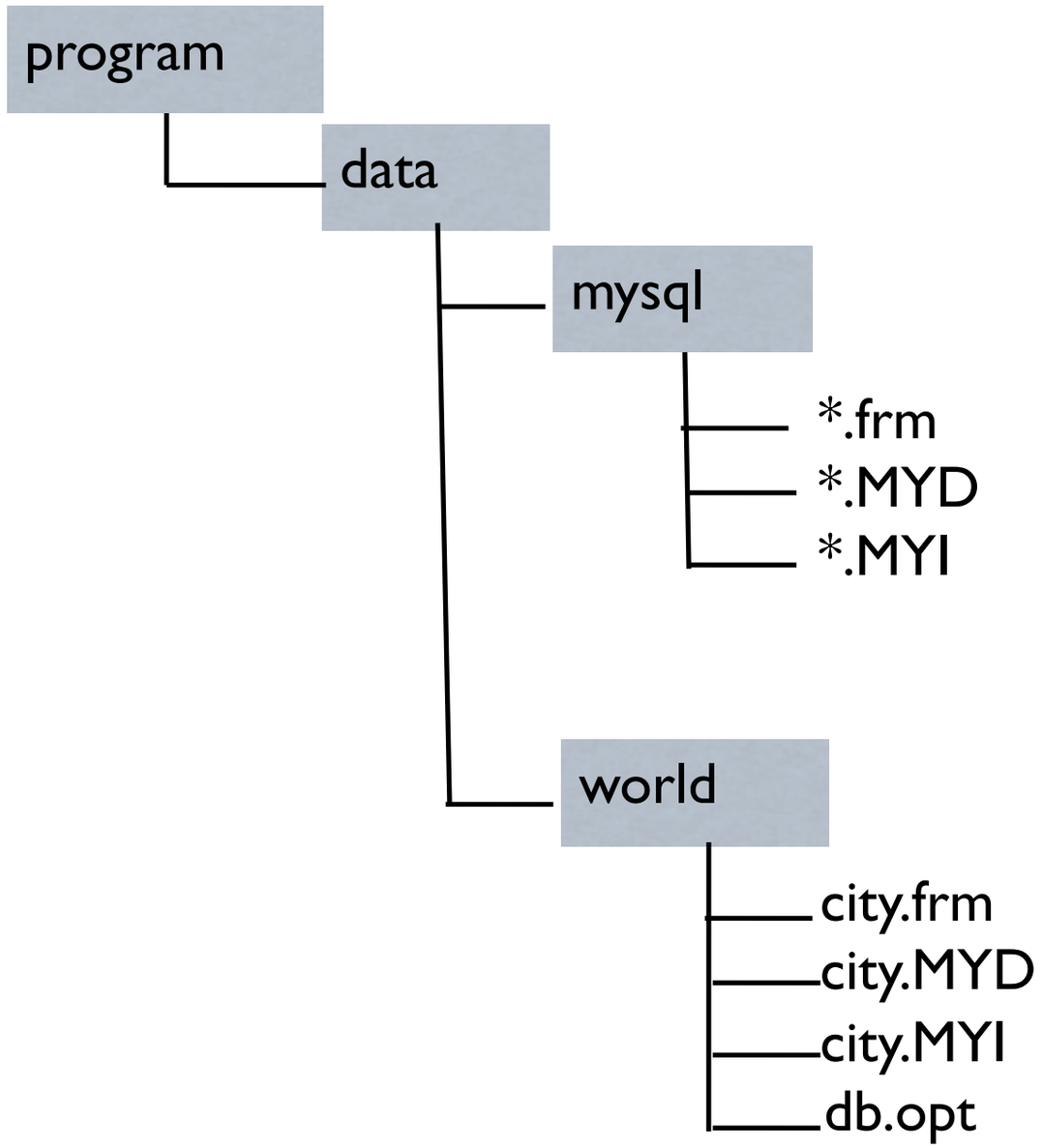
MySQL Conference & Expo

Storage Engines have different models for storage on disk

- **MyISAM**
  - 3 files in data directory
    - *table.frm (structure)*
    - *table.MYI (indexes)*
    - *table.MYD (data)*
- **InnoDB**
  - *table.frm* file in data directory
  - Data and indexes stored in a tablespace that is made up of datafiles:
    - *innodb_data_file_path = file1:size; file2:size ...*

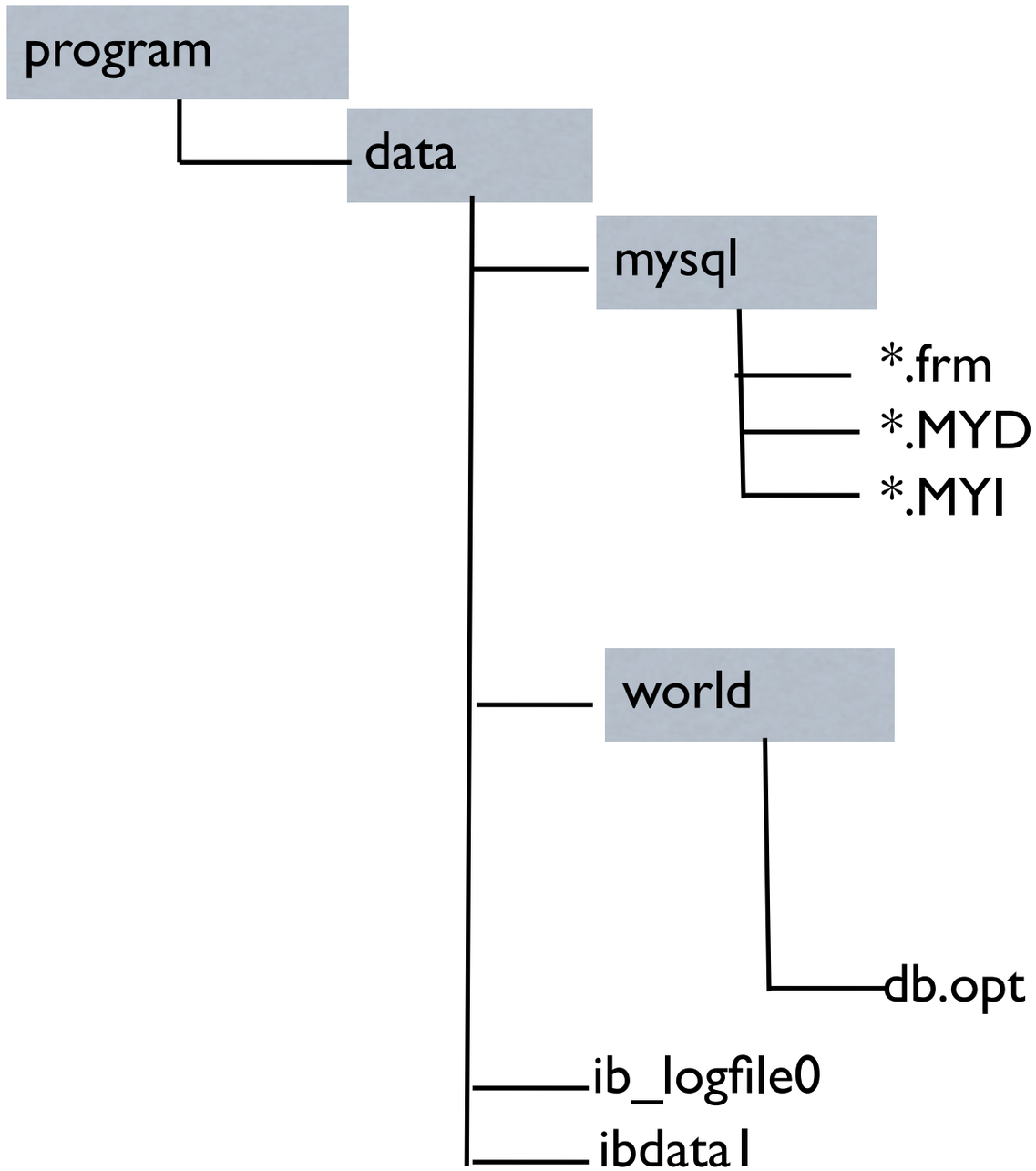
Raw MyISAM data files to distribute

- MyISAM:



Raw InnoDB data files to distribute

- InnoDB:



MXJ

- MXJ, a Java utility package for deploying MySQL
- Connection conn = DriverManager.getConnection("jdbc:mysql:mxj:///test", null);
  - Determines the appropriate binary for MySQL
  - Launches MySQL
  - Initializes the database
  - Shuts down when done

http://dev.mysql.com/doc/refman/5.0/en/connector-mxj

MySQL Conference & Expo

OEM Security Challenges

Bundled MySQL & Data Security

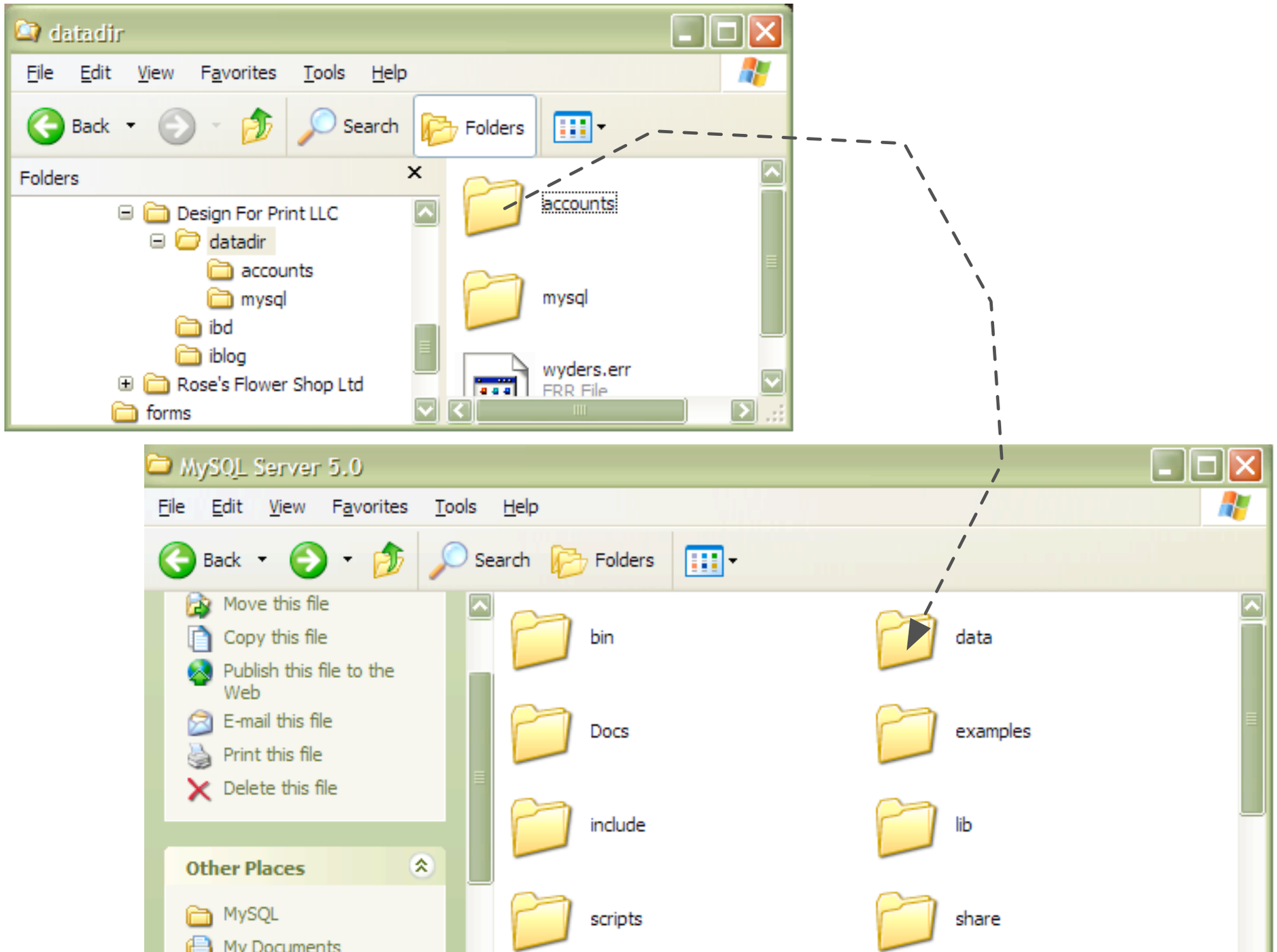
- MySQL is well-known
- ... easy to recognize
- ... and well-documented on the internet.

- MySQL data is no more secure than the physical database machine and its filesystem.

- **If you can access the files, you can access the data –** but at least an application can detect when this happens**.**

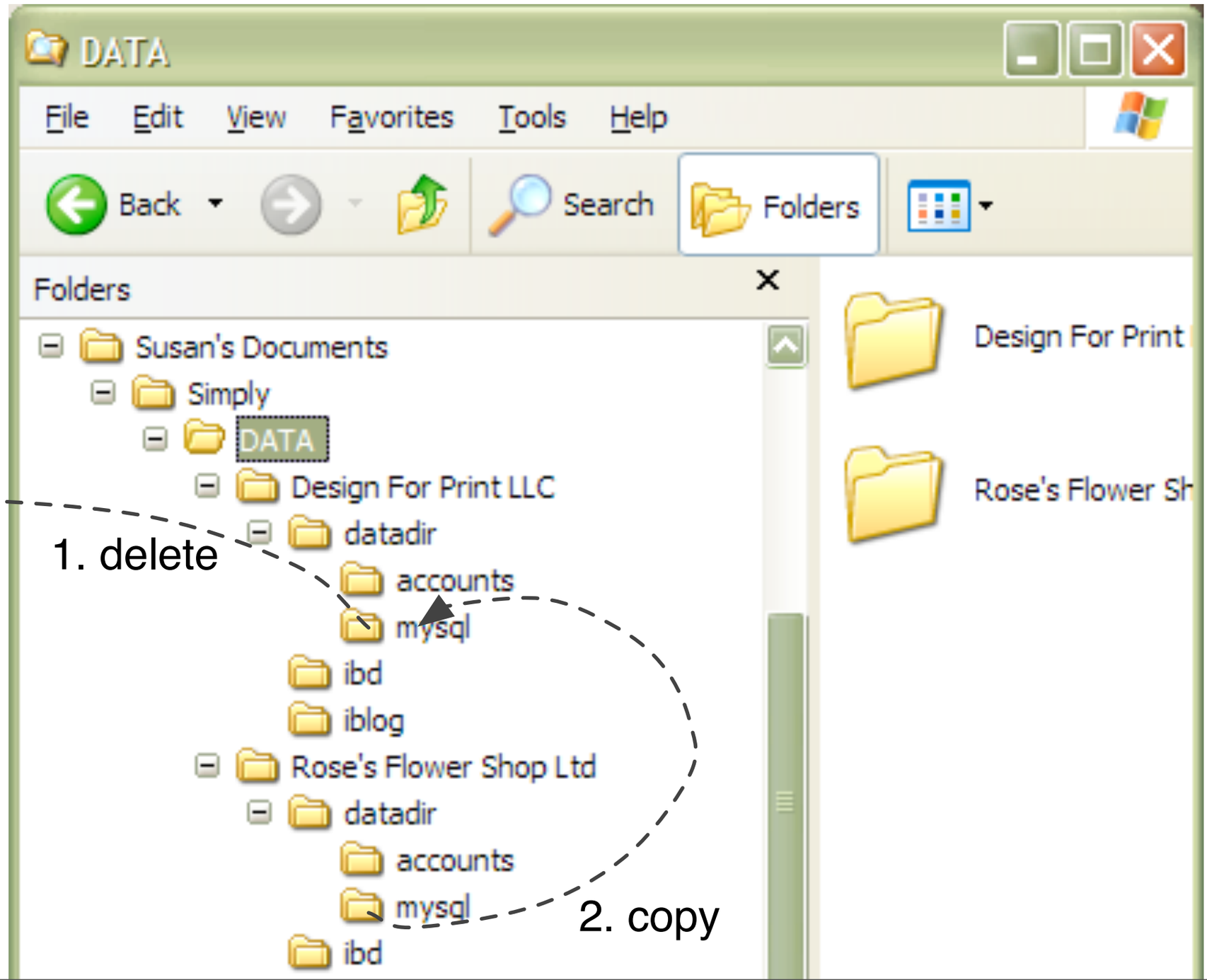
Exploit #1: Data file swap

- Bob wants access to private application data, so he just copies the data files to his own MySQL server.



Exploit #2: Grant table swap

- Bob wants access to private application data, so he swaps out the data files containing system privilege tables, then restarts the database server.



Does it matter?

- 1. Application data is not private anyway.
  - *e.g. common global metadata*
- 2. Application data is private, but physical and filesystem security are adequate.
  - *Any single-user application*
  - *Any dedicated database server maintained by I.T. staff*
- 3. Application data is private and filesystem security is not enough.
  - *desktop small-business accounting application*

MySQL Conference & Expo

Tactics

- Encryption
- Obscurity for InnoDB data files
- Application awareness of database tampering

Encryption

- **In the Database**
  - Column-level encryption in the database
    - AES_ENCRYPT()
    - DES_ENCRYPT()
  - Connection-level encryption using SSL
  - MySQL does *not* offer table- or database- level encryption

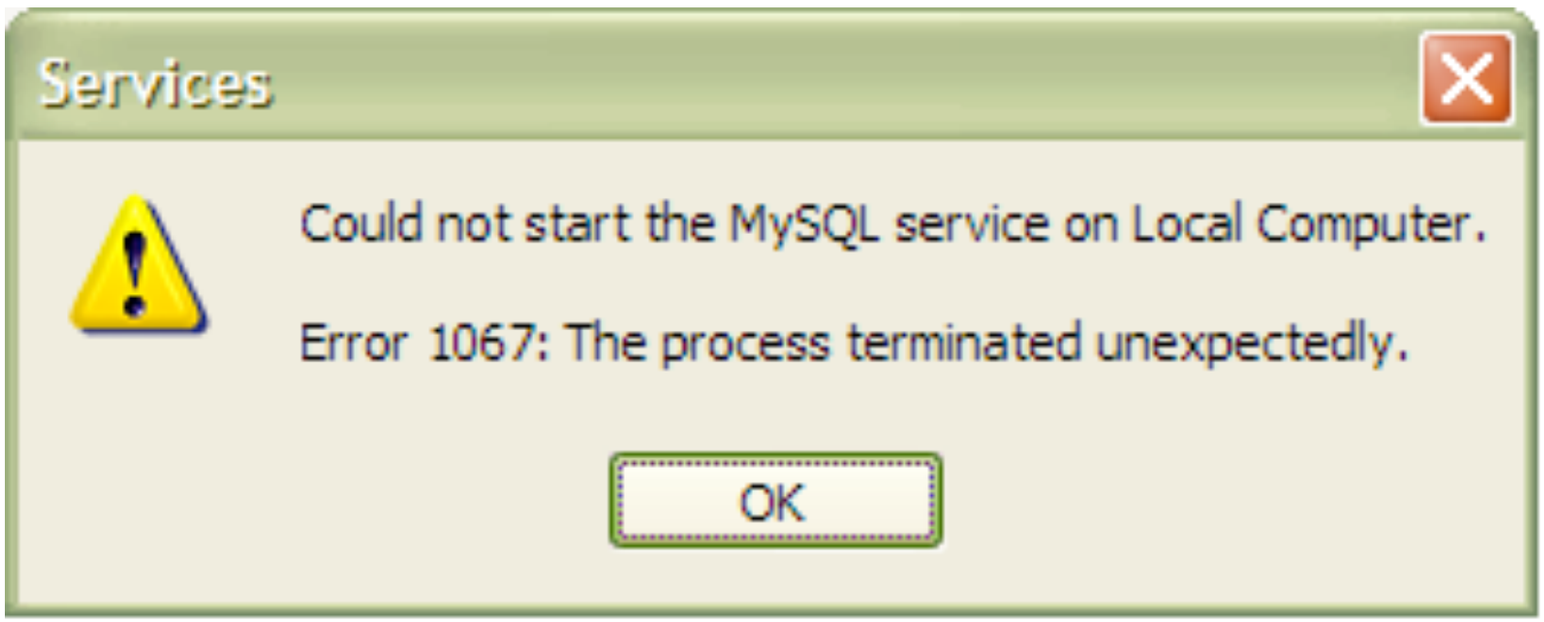
- **Outside the Database**
  - Filesystem encryption:
    - Windows: Dekart Private Disk
    - Linux: dm-crypt, cryptoloop
    - *etc.*

“Obscurity” for InnoDB Data Files

- Start MySQL Server, using an unusual set of InnoDB data files, supplied as a runtime argument

```
C:\>"c:\mysqld.exe" --datadir="c:\program files\Myapp\data"
    --language="c:\program files\Myapp\share\english”
    --innodb-data-file-path=”df_b:340000;df_l:2M;df_f:1M;
      df_n:11M:autoextend”
```

- Nobody will succesfully start the server without knowing the correct sequence of files



Application Awareness of Tampering

- **Tactic #1. CHECKSUM on grant tables**
  - All proper access to grant tables is via application
    - CHECKSUM TABLE mysql.user;
    - CHECKSUM TABLE mysql.host;
    - CHECKSUM TABLE mysql.db;
  - Checksums are stored:
    - in the filesystem
    - or in the Windows registry
    - or in an InnoDB table
  - Application can verify tables against stored checksums.

Application Awareness of Tampering

- **Tactic #2: The random username**
  - Create a strong link between a set of MySQL system grant tables and a set of InnoDB data files:
    - *Insert a random string into a grant table*
    - *Insert an MD5 hash of the string in an InnoDB table*
  - The application verfies the string against the hash at runtime.
  - This tactic can detect *both* the “data file swap” and the “grant table swap.”

The Random Username

```
CREATE TABLE t$privtest (m char(32)) engine=innodb;

CREATE VIEW  v$privtest AS
 SELECT 1 AS link_ok
 FROM mysql.user u, t$privtest t
 WHERE t.m = md5(u.user) ;


$random_user = random.string(16)  application code

GRANT USAGE on *.* to $random_user@"localhost";
INSERT INTO accounts.t$privtest VALUES
     (md5($random_user)) ;       // Create the link


SELECT * FROM v$privtest;        // Verify the link
```

More resources

- www.mysql.com/oem
  - case studies, data sheets, white papers
- www.mysql.com/training
- www.mysql.com/consulting
- dev.mysql.com/doc
- email us
  - lees@sun.com
  - jdd@sun.com

MySQL Conference & Expo

Thanks for attending.

Questions?