

# Staying Safe

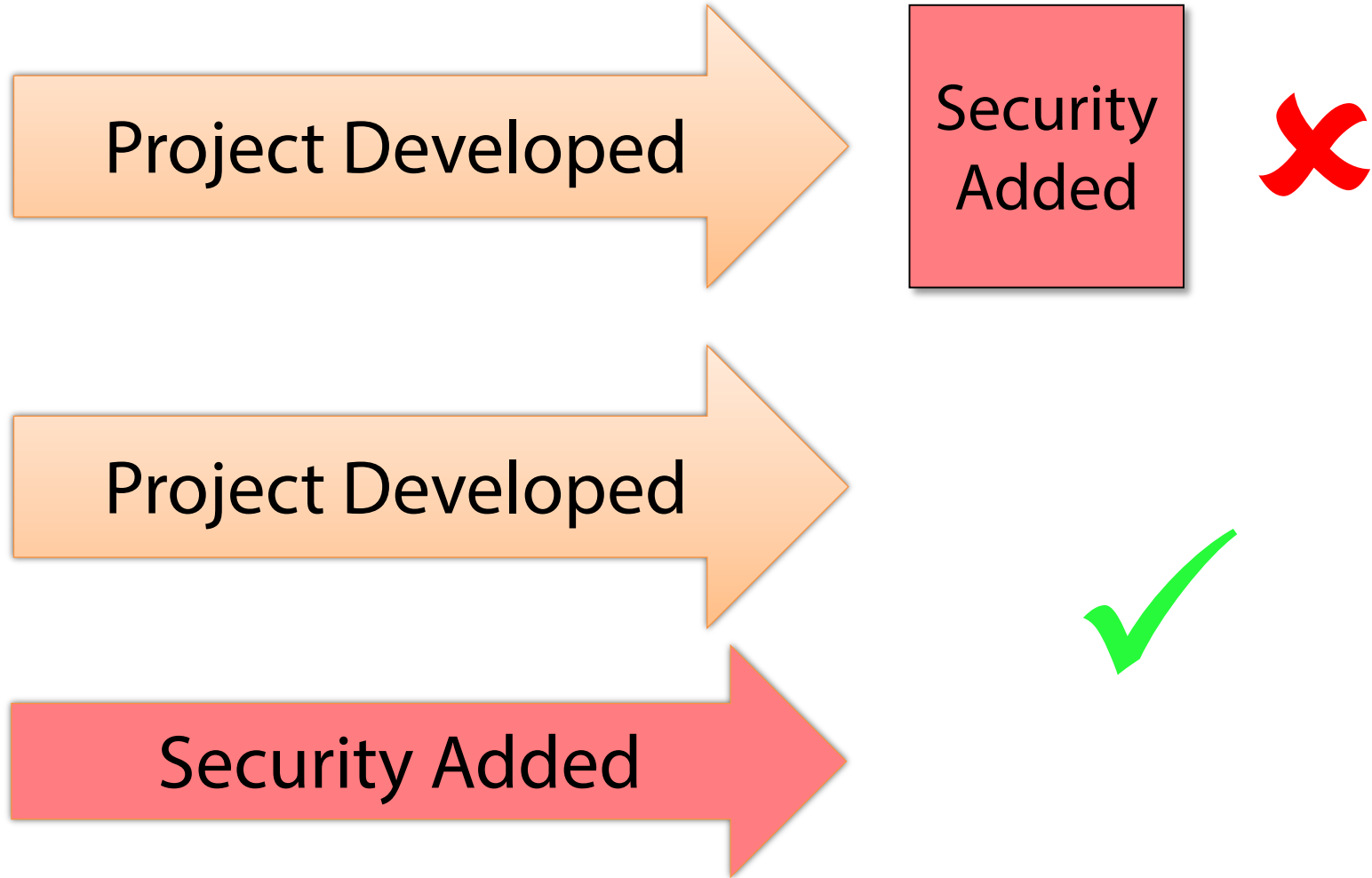


# Lesson Objectives

- In this lesson we will:
- Discuss best practices for keeping a web application secure
- Learn how to create a packet-filtering firewall
- Discuss SQL injection attacks
  - What they are
  - How to defend
- Back up the web site
- Back up the database



# Security is Not an Afterthought



# The "CIA" Security Model

- **Confidentiality**

- Allow the "right people" access to information
- Deny access to the "wrong people"

- **Integrity**

- Data has not been changed (accidentally or maliciously)
- Data comes from the place you think it came from (not an imposter)

- **Availability**

- Data and services should be available when they are supposed to be
- Many organisations have near total-dependency on their IT services



# Defense in Depth

- Attackers prefer to pick the low-hanging fruit
- More barriers → less intruders



Defence	Layer
Physical placement	Physical
Firewall	Network
Infrastructure maintenance	Operating system
Restricting file permissions	Operating system
Apache access controls	Application
"Least privilege" MySQL accounts	Application
Good coding practices	Application
Make regular backups	Application



# Stating the Obvious

- **Don't run (or install) services you don't need**
  - Minimise the "attack surface"
- **Choose strong passwords**
  - Easy to remember, hard to guess
- **Keep up to date with your Linux distribution's security updates**
- **Disable direct root login (especially via `ssh`)**
  - Restrict use of `su`
- **Make regular backups**

Test Your Password	
Password:	.....
Hide:	<input checked="" type="checkbox"/>
Score:	75%
Complexity:	Strong

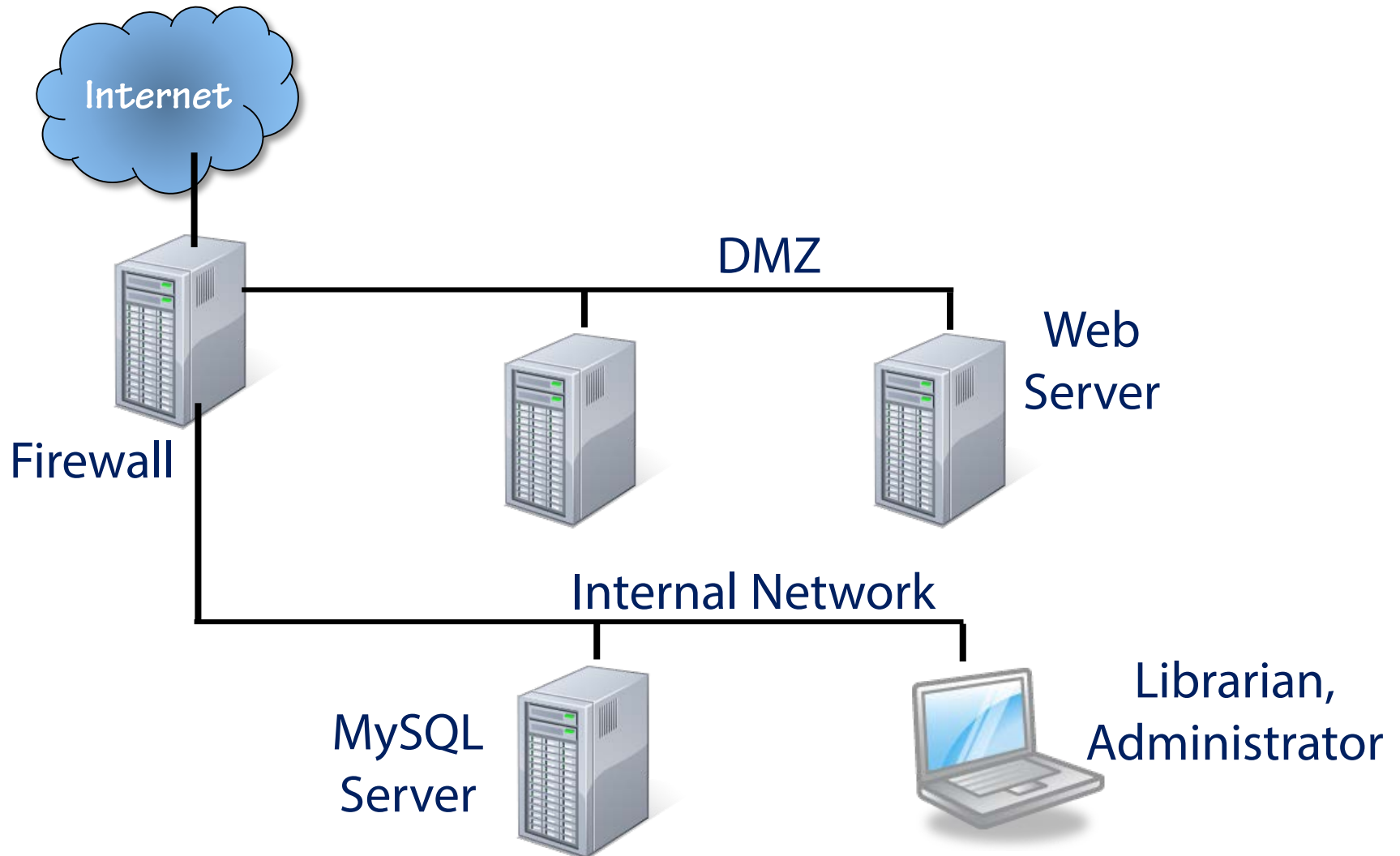


# The Principle of Least Privilege (Again)

- The principle of least privilege says that programs (or users) should only have the minimum access to resources that they need to do their legitimate work
- At the operating system level
  - Do *not* run services as `root`
  - E.g. on CentOS, MySQL server runs as "`mysql`", `httpd` runs as "`apache`"
  - Set file ownership, groups and permissions carefully
- At the application level
  - Create non-root accounts in MySQL
  - Assign minimum privileges

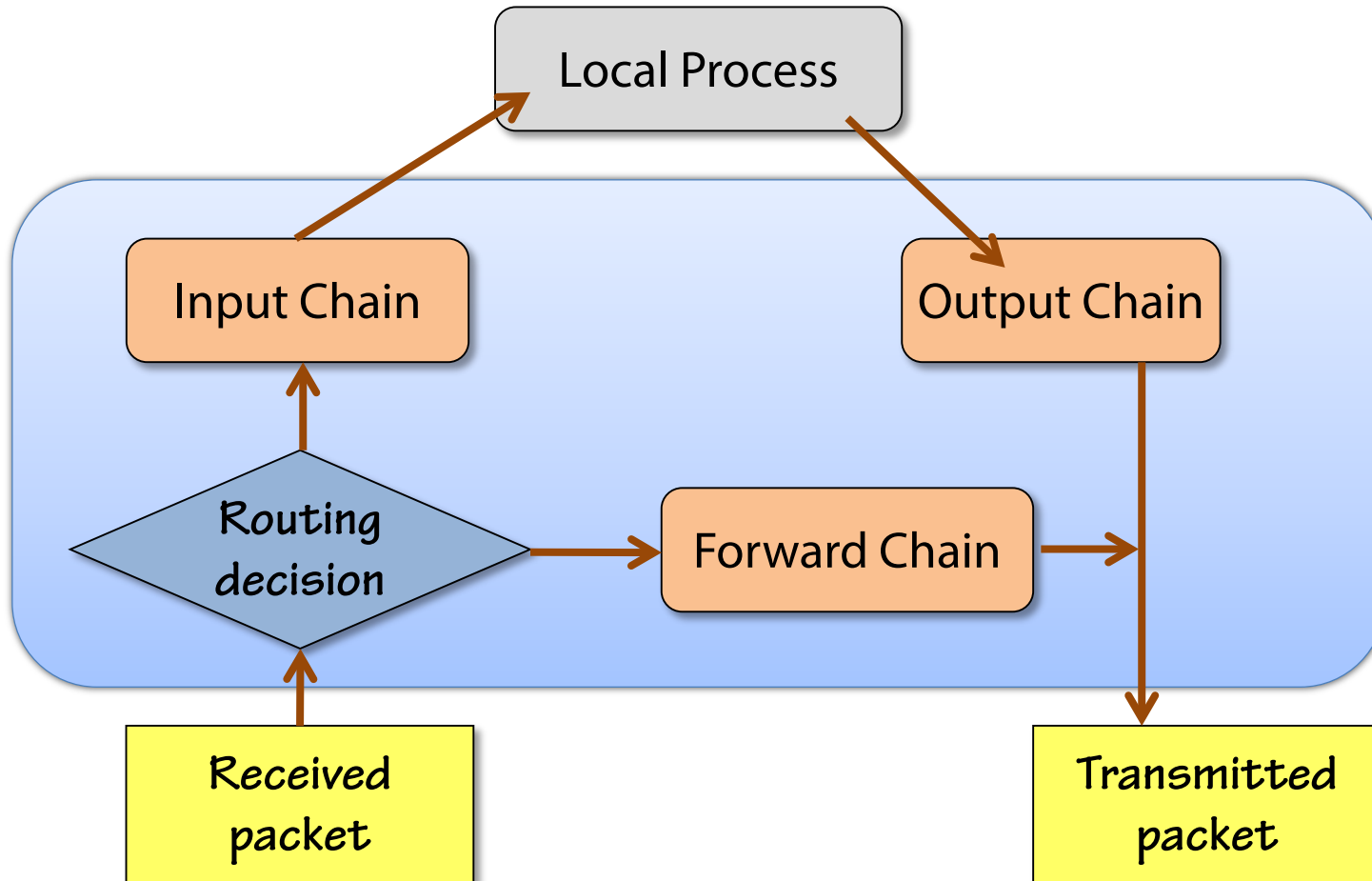


# Placement





# Firewalls (Netfilter)



# Injection Attacks

- Injection attacks occur when untrusted input is sent to an interpreter
  - Data from PHP's `$_GET`, `$_POST`, `$_COOKIE` is *untrusted*
  - Used to create a SQL command or query or Linux shell command

## 1 User enters untrusted input

Enter book title:

foo' OR 'x'='x

"title" input field

```
$title = $_GET['title'];  
$query = "select * from books" .  
"where title = '" . $title . "'";
```


## 2 Application constructs query

## 3 Query is executed


```
Select * from books where title = 'foo' or 'x'='x'
```

## 4 The entire table is retrieved !


# A More Malicious Attack



Hello, this is your  
healthcare  
receptionist



Are you really called  
Mary');DROP TABLE PATIENTS;-- ?




Only we've just lost all  
our medical records ...

# Defending Against SQL Injection

- **Positively vet all user supplied input**
  - Match against a regular expression

```
$d = trim($_POST["date"]);  
if (!ereg("^[0-9]{4}-[0-9]{2}-[0-9]{2}$", $d)  
    // input is not a valid date
```

- **Connect to the database using an account with the minimum necessary privileges**
- **Use prepared statements** 
  - Parsed *once*, without contamination from untrusted input



# Backing up the Web Site

- **Standard Linux backup solutions**
  - Backup the contents of `/var/www`
- **Small-scale tools:**
  - `tar`, `rsync`, `scp -r`
- **Enterprise tools**
  - `backuppc`, `bacula`
- **Cloud backup**
  - Ubuntu One, DropBox, many more ...
- **Don't forget the config files:**
  - `/etc/my.conf`, `httpd.conf`



*Always backup to a  
different machine,  
preferably at a  
different site*

# Backing up the Database

- Your data is probably your most important asset
- Prolonged data loss can cause irrecoverable financial damage or even closure of business



## Backup options

`mysqlhotcopy`

`mysqldump`

**Percona XtraBackup**

**Free, open source**

**MySQL Enterprise Backup**

**Commercial**

# mysqlhotcopy

- Simple perl script
- Instructs `mysqld` to lock and flush the database tables, then directly copies the underlying files
- Fast



Backup this database ...



```
mysqlhotcopy -u root -p rootpw library /tmp
```



... to this directory



Does not work with InnoDB

# mysqldump



- **mysqldump backs up the database to a text file**
  - Writes the SQL CREATE and INSERT statements needed to re-create and re-populate the tables
  - Can also back up triggers, stored procedures and events

Includes CREATE DATABASE  
and USE commands in the file

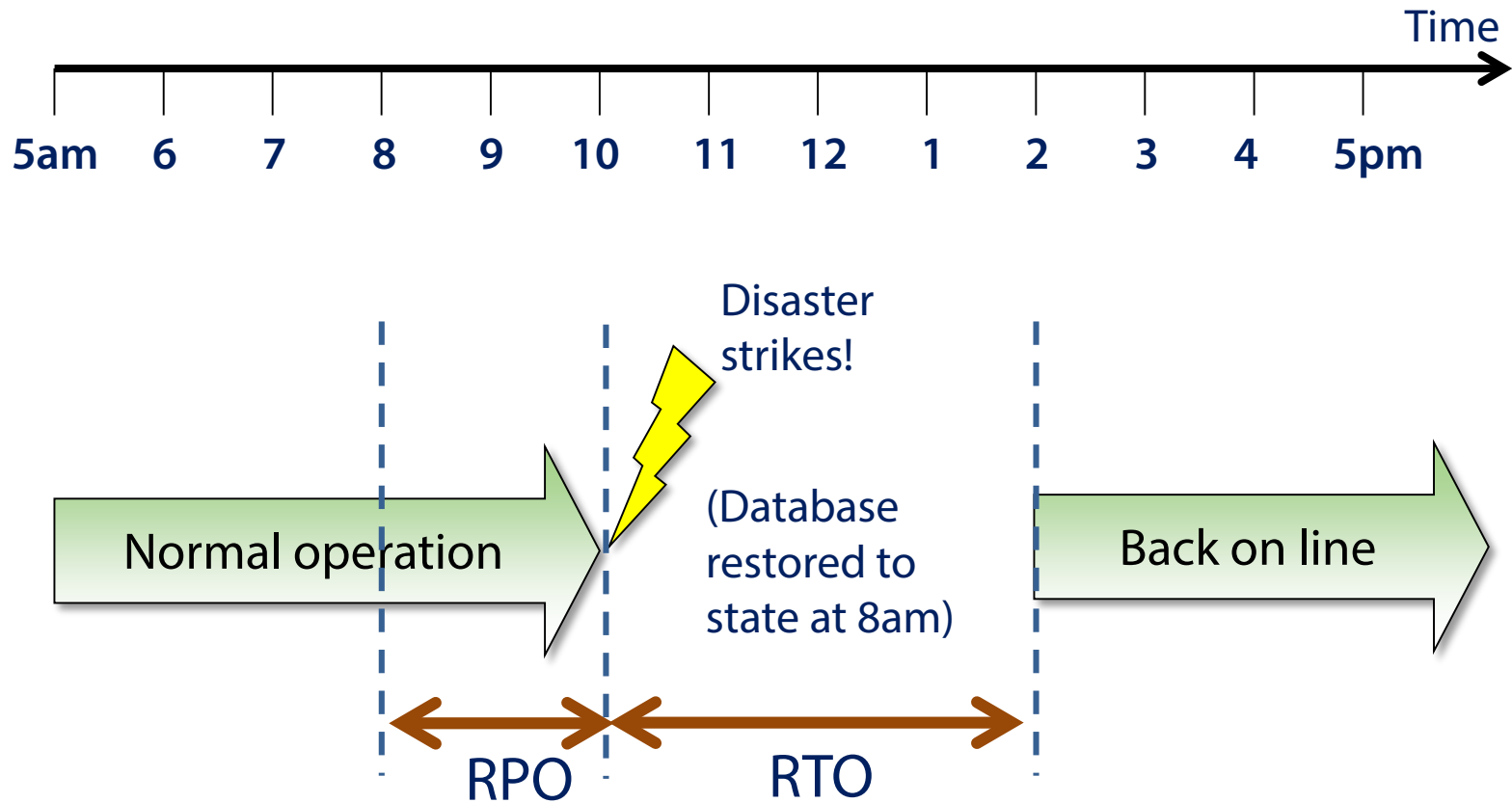
```
mysqldump --single-transaction --database library > bak.sql
```

Useful with InnoDB to dump  
tables within a transaction

The database to  
be backed up



# RTO and RPO



RTO = "Recovery Time Objective"

RPO = "Recovery Point Objective"

# MySQL Log Files

- **MySQL can write various log files**
  - In CentOS, only the error log is written by default

Log file	Contains	Used for
Error log	Server startup and shutdown actions, and operational errors.	Debugging
General query log	All client connections and queries	Debugging (how is my database being used?)
Slow-query log	All "slow" queries (>10 seconds by default).	Performance analysis and optimisation
Binary log	Every action that modifies the database (e.g. INSERT, UPDATE, DELETE, but not SELECT)	Recovering after data loss, replicating to slave server

# Enabling Logging

- Either add parameters to the command line used to start `mysqld`:

```
--log-error=/var/log/mysqld.log
```

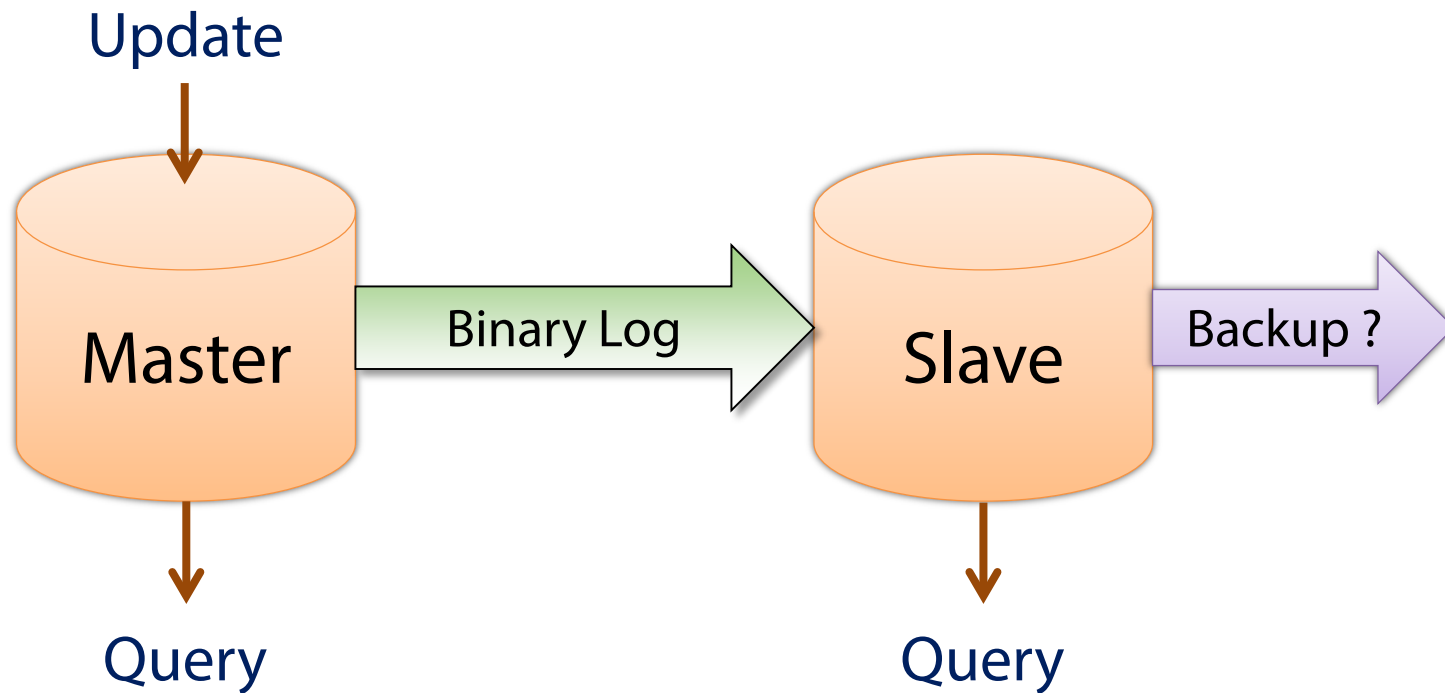
- ... or ... add entries to `/etc/my.cnf`:

```
[mysqld]  
...  
log-bin=/var/log/mysql/bin  
log=/var/log/mysql/querylog
```

- `mysqld` will create the log files but *not* the directory

# Replication

- Replication can improve performance and availability

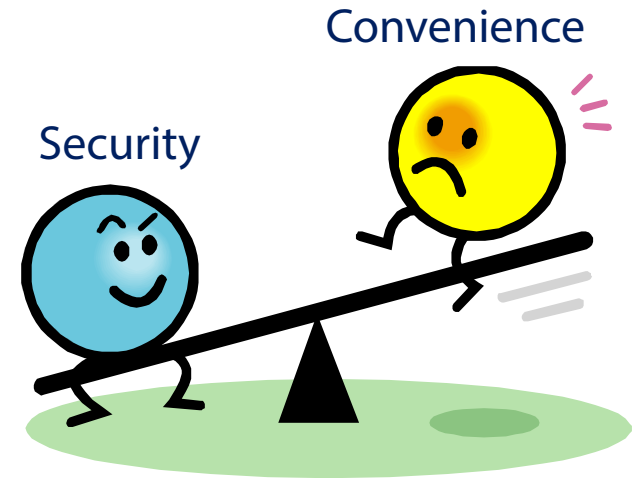


# MySQL Enterprise Backup

- Part of MySQL Enterprise Edition from Oracle
- "Hot" online backups (transactions are not interrupted)
- Incremental backups
- Partial backups (selected tables)
- Built-in compression
- Point-in-time recovery
- Fast (up to 50 times the speed of mysqldump)
- Multi-platform
  - Linux, Windows, MAC, Solaris

# Lesson Summary

- **We have examined some ways of making our web site more secure**
  - Physical placement
  - Packet filtering firewall
  - Principle of least privilege
  - Defending against SQL injection
  - Backing up the database
- **Defense in Depth!**
  - There is no such thing as absolute security
- **Security is an ongoing process**



## **Coming up in Lesson 11:**

### **Going Further**

A few suggestions for further study