Introducing LAMP:
Linux, Apache, MySQL and PHP

Track 2 Workshop

PacNOG 7

July 1, 2010

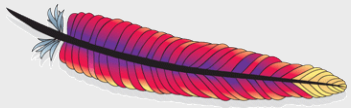
Pago Pago, American Samoa

What is LAMP?

One of the most popular dynamic web site environments in use today…

There are other flavors of this…

Linux	**Linux**	**FreeBSD**	**FreeBSD**
Apache	**Apache**	**Apache**	**Apache**
MySQL	**Postgresql**	**MySQL**	**Postgresql**
PHP	**PHP**	**PHP**	**PHP**



Why so popular?

From Wikipedia (I agree):

- **Easy to code:** Novices can build something and get it up and running very quickly with PHP and MySQL.
- **Easy to deploy:** Since PHP is a standard Apache module, it's easy to deploy a PHP application. Once you've got MySQL running, simply upload your .php files.
- **Develop locally:** It's easy to set up LAMP on your laptop, build your app locally, then deploy on the Web.
- **Cheap and ubiquitous hosting:** Even the cheapest Web hosts options allow you to run PHP and MySQL.

Why so popular cont.?

- MySQL is *fast* and can support large sites.
- PHP is relatively easy to learn and use.
- Many people already run and know Linux.
- Apache is ubiquitous.

So, is there anything “bad” about LAMP…?

LAMP Issues

PHP is susceptible to cross-site scripting (XSS) attacks.
http://en.wikipedia.org/wiki/Cross-site_scripting

So are other programming languages, but PHP, by default, does not verify user input as “reasonable”.

MySQL Injection Attacks. LAMP sites are vulnerable as you must filter user input for escaped characters:
http://en.wikipedia.org/wiki/SQL_injection

XSS and MySQL Injection

A few good references for dealing with these:

- http://en.wikipedia.org/wiki/Cross-site_scripting
- http://php.net/manual/en/function.mysql-real-escape-string.php
- http://www.tizag.com/mysqlTutorial/mysql-php-sql-injection.php
- http://www.netlobo.com/preventing_mysql_injection.html
- http://en.wikibooks.org/wiki/PHP_Programming/SQL_Injection
- http://old.justinshattuck.com/2007/01/18/mysql-injection-cheat-sheet/
- http://en.wikipedia.org/wiki/SQL_injection
- http://www.owasp.org/index.php/XSS_%28Cross_Site_Scripting%29_Prevention_Cheat_Sheet

XSS and MySQL Injection

The critical step is to safely read any data that is being input using built-in wrappers in PHP.

We will do this in our LAMP lab.

Steps to Using LAMP

1. Install a Linux server with Apache, MySQL and PHP.
2. Install the necessary modules so that Apache will execute (interpret) PHP code.
3. Install the necessary modules so that PHP can talk to MySQL.
4. Design and create an initial MySQL database for your project.
5. Populate the database with data if relevant.
6. Write PHP code to use this data and to dynamically generate web pages based on coding logic and available data.
7. Ensure you use proper coding and configuration method to secure your LAMP server.

LAMP Installation Lab

We will now install and configure LAMP for initial use in our classroom.