Web & Mobile Security Laboratory Manual

Practical–1 (Cross-Platform): XAMPP & LAMP Setup + SQL Practice

Experiment Title: Installation and Configuration of XAMPP (Windows/Linux) and LAMP (Linux), Database Creation, and Execution of SQL Commands

Duration: 2 Hours (recommended: 30–40 min setup + 60–70 min SQL practice + 10 min viva)

Platforms: Windows (XAMPP), Linux (XAMPP), Linux (LAMP)

Software/Tools: Apache, MySQL/MariaDB, PHP, phpMyAdmin, Web Browser

1. Objectives

• Install and start a local web server stack using XAMPP (Windows/Linux)

• Verify Apache and MySQL services and access phpMyAdmin.

• Create a database and table, and execute essential SQL commands (DDL/DML/DQL).

• Understand the backend flow (Browser → Web Server → PHP → DB) that later enables SQL Injection labs.

• Record observations and prepare lab-ready outputs (screenshots, outputs, and queries).

2. Concepts / Theory

What is XAMPP?

XAMPP is a **cross-platform local web server package** used to **develop, test, and learn web applications** on a single machine.

Full Form

- **X** → Cross-platform (Windows, Linux, macOS)
- **A** → Apache (Web Server)
- **M** → MySQL / MariaDB (Database Server)
- **P** → PHP
- **P** → Perl

XAMPP allows to:

- Run a **web server locally**
- Create and manage **databases**
- Execute **PHP-based web applications**
- Practice **SQL commands**
- Perform **web security experiments** (SQLi, auth flaws, XSS, etc.)

Key point:
XAMPP simulates a **real web server environment** without using the Internet.

3. Components of XAMPP

Apache: Handles HTTP/HTTPS requests and serves web pages.
MySQL/MariaDB: Stores and manages application data.
PHP: Server-side scripting language for dynamic content.
Perl: Used for legacy scripting and automation.
phpMyAdmin: Web-based GUI tool to manage MySQL/MariaDB databases.

How XAMPP Works (Simple Flow)

Browser
↓
Apache (XAMPP)
↓
PHP Script
↓
MySQL Database
↓
Response to Browser

Web applications typically follow a client–server architecture. The browser (client) sends HTTP requests to a web server (Apache). Server-side code (PHP) processes inputs and interacts with a database server (MySQL/MariaDB) to store/retrieve data. The server then returns an HTTP response to the browser.

XAMPP vs LAMP

Feature	**XAMPP**	**LAMP**
OS	Windows, Linux, macOS	Linux only
Setup	One-click installer	Manual configuration
Target users	Students, beginners	Production servers
Security	Not hardened	Can be hardened
Use case	Learning & testing	Real hosting

Note: XAMPP is primarily used for development and learning, whereas LAMP is used for production servers.

P

PRACTICAL- This practical creates a controlled local environment to understand how databases are created and how SQL queries operate—forming the foundation for later security practicals (e.g., SQL Injection, Broken Authentication).

2.1 XAMPP (Concept)

XAMPP is a cross-platform local server package.
X = Cross-platform, A = Apache, M = MySQL/MariaDB, P = PHP, P = Perl.
XAMPP is primarily used for development and training labs (not recommended for Internet-facing production deployments).

2.2 LAMP (Concept)

LAMP is a Linux-based web stack:
L = Linux, A = Apache, M = MySQL/MariaDB, P = PHP.
LAMP is commonly used for production and enterprise deployments when hardened appropriately.
In Linux labs, LAMP is often preferred over XAMPP due to native package management and service control.

2.3 Key Security Note

XAMPP/LAMP stacks are used here for controlled learning. Insecure configurations, weak credentials, or mismanaged permissions can expose systems. Always run labs on isolated networks/VMs.

3. Pre-requisites / Requirements

• Administrator/root privileges on the machine

• Stable disk space (≈ 2–4 GB) and RAM (≥ 4 GB recommended)

• Web browser (Chrome/Firefox)

• Internet access only if downloading installers/packages (optional if already available)

4. Implementation A: Windows (XAMPP)

4.1 Steps

1. Download and install XAMPP (Apache + MySQL/MariaDB + PHP + phpMyAdmin).

2. Open XAMPP Control Panel.

3. Click Start for Apache and Start for MySQL.

4. Verify the services show ‘Running’.

5. Open browser and visit http://localhost/ (Apache test page).

6. Open phpMyAdmin at http://localhost/phpmyadmin.

4.2 Verification Checklist (Windows)

☐ Apache running (port 80) and test page loads at http://localhost/

☐ MySQL running (port 3306)

□ phpMyAdmin accessible at http://localhost/phpmyadmin

5. Implementation B: Linux (XAMPP)

5.1 Steps

1. Install Linux XAMPP (typically extracted/installed under /opt/lampp).

2. Start XAMPP stack: sudo /opt/lampp/lampp start

3. Verify status: sudo /opt/lampp/lampp status

4. Open browser and visit http://localhost/

5. Open phpMyAdmin (if enabled) at http://localhost/phpmyadmin or via XAMPP dashboard.

5.2 Useful Linux XAMPP Commands

```
Start:  sudo /opt/lampp/lampp start
Stop:   sudo /opt/lampp/lampp stop
Restart:sudo /opt/lampp/lampp restart
Status: sudo /opt/lampp/lampp status
```

6. Implementation C: Linux (LAMP)

6.1 Installation & Service Control (Ubuntu/Debian family)

Install Apache + MySQL + PHP (example):

```
sudo apt update
sudo apt install apache2 mysql-server php libapache2-mod-php php-mysql
```

6.2 Start/Stop/Status Commands

```
Apache:
  sudo systemctl start apache2
  sudo systemctl status apache2
MySQL/MariaDB:
  sudo systemctl start mysql
  sudo systemctl status mysql
Enable on boot:
  sudo systemctl enable apache2
  sudo systemctl enable mysql
```

6.3 phpMyAdmin (Optional but Recommended for Labs)

Install phpMyAdmin (Ubuntu/Debian example):

```
sudo apt install phpmyadmin
```

Then open: http://localhost/phpmyadmin

Note: On some Linux distros, phpMyAdmin requires enabling Apache configuration or PHP extensions.

6.4 Verification Checklist (Linux LAMP)

☐ Apache page loads at http://localhost/

☐ MySQL/MariaDB service is active

☐ phpMyAdmin loads at http://localhost/phpmyadmin (if installed)

7. SQL Commands & Practice (Common to XAMPP and LAMP)

These SQL commands are platform-independent and work on MySQL/MariaDB in Windows (XAMPP) and Linux (XAMPP/LAMP). Students must execute commands using phpMyAdmin (SQL tab) or CLI client.

7.1 Create Database & Table (DDL)

```
CREATE DATABASE websecurity;
SHOW DATABASES;
USE websecurity;

CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL,
  password VARCHAR(50) NOT NULL
);

SHOW TABLES;
DESCRIBE users;
```

7.2 Insert / Select / Update / Delete (DML/DQL)

```
INSERT INTO users (username, password) VALUES ('admin','admin123');
INSERT INTO users (username, password) VALUES ('test','test123');

SELECT * FROM users;
SELECT username FROM users WHERE id=1;

UPDATE users SET password='newpass' WHERE username='admin';

DELETE FROM users WHERE username='test';
SELECT COUNT(*) FROM users;
```

7.3 Boolean Logic Queries (Preparation for Blind SQLi)

```
SELECT * FROM users WHERE id=1 AND 1=1;
SELECT * FROM users WHERE id=1 AND 1=2;
```

Security Considerations

XAMPP installations are insecure by default. Default credentials, open services, and relaxed permissions are common. These weaknesses are intentionally used in security labs to help students understand real-world vulnerabilities.

8. Student Tasks / Deliverables

• Start Apache and MySQL/MariaDB and show running status (screenshot/output).

• Open phpMyAdmin and create database ‘websecurity’.

• Create ‘users’ table and insert at least 2 records.

• Execute SELECT queries (with and without WHERE).

• Update one password and delete one record; show final SELECT output.

• Write 5–8 lines describing the backend flow and why SQL understanding is essential for web security.

KEY POINTS to remember:

XAMPP is a local web server stack, not a hosting service.
• It supports Windows, Linux, and macOS.
• It simplifies learning web application security.
• XAMPP should never be exposed directly to the Internet.

9. Observations

Record the observed outputs during the experiment.

Action/Command	Expected Output	Observed Output (Student)
Apache start	Service running / localhost loads	
MySQL start	Service active	
CREATE DATABASE	Database created	
CREATE TABLE	Table created	
INSERT	Rows inserted	
SELECT	Rows displayed	
UPDATE/DELETE	Rows modified/removed	

10. Result

A local web server environment was successfully configured using XAMPP (Windows/Linux) and/or LAMP (Linux). A database and table were created, and SQL commands for insertion, retrieval, modification, and deletion were executed successfully.

11. Viva / Oral Questions

• Define XAMPP and LAMP. What are the components of each?

• Why is SQL considered platform-independent?

• What is the role of Apache and MySQL in a web application?

• Differentiate between DDL, DML, and DQL with examples.

• Why is the WHERE clause important in SQL Injection context?

• How do you start/stop Apache and MySQL services on Linux?

• Why is XAMPP not recommended for production deployment?

12. Post-Lab Questions (To be answered in record)

• Write the request flow: Browser → Apache → PHP → Database → Response (explain in 5–6 lines).

• List the default ports used by Apache and MySQL/MariaDB.

- Explain how improper handling of user input can lead to SQL Injection.