Vulnerable Security Training Lab Comprehensive Project Report

A Complete Web Application Security Testing Platform

Educational Use Only

Version 2.0 — December 2024

SECURITY WARNING

This document contains intentional vulnerabilities for educational purposes only. Do not use these techniques on systems you do not own or have permission to test.

Contents

Executive Summary

Project Overview: The Vulnerable Security Training Lab is an intentionally vulnerable web application designed for cybersecurity education, penetration testing practice, and security awareness training. This comprehensive platform provides hands-on experience with real-world web vulnerabilities in a safe, controlled environment.

Key Features:

- **16+ Different Vulnerabilities** covering OWASP Top 10 and beyond
- **Interactive Interface** with modern design and clear instructions
- **Realistic Scenarios** simulating actual production vulnerabilities
- **Safe Environment** with all vulnerabilities contained locally
- **Educational Content** including explanations, examples, and prevention tips
- **Progress Tracking** with challenge mode and completion tracking

1 Project Overview

1.1 Purpose & Objectives

The Vulnerable Security Training Lab serves multiple educational purposes:

- **Educational Tool:** Provides hands-on experience with real-world web vulnerabilities
- **Penetration Testing Practice:** Allows security professionals to practice exploitation techniques
- **CTF Training:** Serves as a Capture The Flag training platform
- **Developer Awareness:** Helps developers understand vulnerability manifestations
- **Security Testing:** Enables testing of security tools against known vulnerabilities

1.2 Target Audience

2 System Architecture

2.1 Technology Stack

```
vuln-lab/ |-- index.php |-- config.php | |-- sql/ | |-- login.php | |-- xss/
| |-- search.php | |-- upload/ | |-- upload.php | +-- files/ | +-- more/
```

light **Audience**	**Use Case**
Security Students	Learning web application security fundamentals
Penetration Testers	Practicing exploitation techniques
Developers	Understanding security vulnerabilities they might create
Security Analysts	Testing security tools and methodologies
CTF Players	Preparing for cybersecurity competitions
Security Educators	Teaching web application security concepts

Table 1: Target Audience and Use Cases

```
Frontend: HTML5, CSS3, JavaScript (vanilla)
Backend: PHP 7.4+
Database: MySQL (via config.php)
Server: Apache/NGINX with PHP support
Dependencies: None (pure PHP/MySQL)
```

2.2 Database Schema

```
CREATE DATABASE vulnsite;
USE vulnsite;

CREATE TABLE users (
    id INT PRIMARY KEY AUTO_INCREMENT,
    username VARCHAR(50),
    password VARCHAR(50),
    email VARCHAR(100),
    role VARCHAR(20) DEFAULT ’user’
);

INSERT INTO users (username, password, email, role) VALUES
(’admin’, ’admin123’, ’admin@test.com’, ’admin’),
(’john’, ’password’, ’john@test.com’, ’user’),
(’jane’, ’123456’, ’jane@test.com’, ’user’),
(’test’, ’test’, ’test@test.com’, ’user’);
```

Listing 1: Database Setup Script

3 Vulnerabilities Catalog

OWASP Top 10 2021 Coverage

light **OWASP ID**	**Category**	**Covered By**
A01:2021	Broken Access Control	IDOR, Path Traversal

light **OWASP ID**	**Category**	**Covered By**
A02:2021	Cryptographic Failures	JWT Weak Secrets
A03:2021	Injection	SQLi, XSS, Command, XXE, SSTI
A04:2021	Insecure Design	Race Conditions
A05:2021	Security Misconfiguration	File Upload
A07:2021	Identification Failures	CSRF, Session Issues
A08:2021	Software/Data Integrity	Deserialization
A10:2021	SSRF	Server-Side Request Forgery

Table 2: OWASP Top 10 2021 Coverage

4 Detailed Testing Guide

4.1 1. SQL Injection (SQLi)

Vulnerability Details

Location: `index.php` to `login.php`
Severity: Critical
CVSS Score: 9.8
Impact: Authentication bypass, data exfiltration

Vulnerable Code

```
// login.php
$username = $_POST[’username’];
$password = $_POST[’password’];
$query = "SELECT * FROM users WHERE username=’$username’
          AND password=’$password’";
```

Listing 2: Vulnerable SQL Query

Testing Payloads

- Authentication Bypass: `admin’ --`
- Union Injection: `’ UNION SELECT 1,2,3,4 --`
- Database Enumeration: `’ UNION SELECT 1,database(),user(),4 --`

Prevention

```
$stmt = $conn->prepare("SELECT * FROM users
                        WHERE username=? AND password=?");
$stmt->bind_param("ss", $username, $password);
$stmt->execute();
```

Listing 3: Secure Prepared Statement

4.2 2. Cross-Site Scripting (XSS)

Vulnerability Details

Location: `xss.php`
Severity: High
CVSS Score: 8.2
Impact: Session hijacking, credential theft

Testing Payloads

- `<script>alert('XSS')</script>`
- `<img src=x onerror=alert(1)>`
- `<svg onload=alert(1)>`
- `<body onload=alert(1)>`

5 Installation & Setup

5.1 Requirements

- PHP 7.4 or higher
- MySQL 5.7 or higher
- Apache or NGINX with PHP support
- Modern web browser

5.2 Installation Steps

1. Clone or download the repository
2. Configure database in `config.php`
3. Import the database schema
4. Set appropriate file permissions
5. Access via web browser

6 Security Considerations

LEGAL DISCLAIMER

This application contains intentional vulnerabilities for educational purposes only.
DO NOT:

- Deploy on public servers
- Use against systems you don't own
- Use for illegal activities

Conclusion

The **Vulnerable Security Training Lab** provides comprehensive hands-on learning for web application security. Use this knowledge responsibly to build more secure applications.

Remember: **With great power comes great responsibility.**