DEV300 Hardening PHP Web Apps

Course Overview

Web applications are routinely the source of many security vulnerabilities, especially as more and more move to the cloud. However, this is despite the fact it is often simple to fix most web applications vulnerabilities, before the code is released into the wild. The 'Hardening PHP Web Apps' course walks students through the list of the OWASP Top Ten vulnerabilities common in web application code and demonstrates various methods of secure coding to harden web applications. Specifically, the course focuses on examples A1 through A8 of the top ten list.

Each module contains instructor-led video lecture content to introduce the vulnerability and explores various mitigation measures, specific to each vulnerable code example. Each module also includes an interactive hands-on lab component, in which the student has the chance to experiment with real solutions to discover why some seemingly adequate code remediations are insufficient and others are more appropriate. The student is then challenged to complete a demanding "capstone lab" exercise that encourages the student to explore a novel web application and remedy sections where it is vulnerable.

Upon completion of this course, the student will understand how to identify many common web application vulnerabilities and gain valuable practical skills via engagement in meaningful web application security control measures.

Objectives

- Examine a web application design and implementation and identify potential vulnerabilities.
- Remediate the vulnerability by modifying the underlying code.

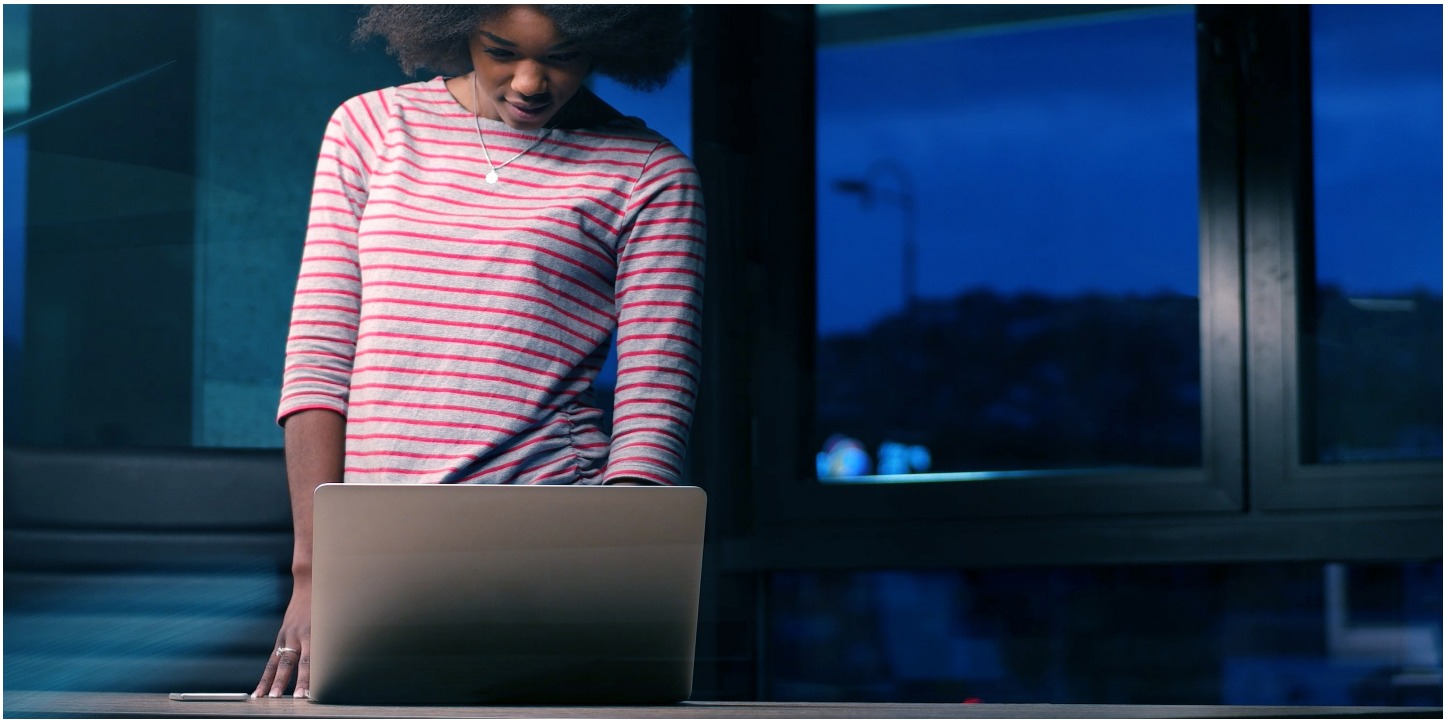
Prerequisite Knowledge

Before taking this course, students should be able to:

- Write web applications in PHP
- Identify basic and advanced examples of the OWASP Top Ten vulnerabilities
- Describe basic mitigation guidelines for fixing basic OWASP Top Ten vulnerabilities

Estimated Course Length: 12 hours

Module	Lecture	Labs	Estimated Completion Time (minutes)
0	Introduction		5
1	A1: Injection	Lab 1.1: Stopping SQL Injection with validation and prepared statements Lab 1.2: Stopping OS Command Injection with Data Validation	85
2	A2: Broken Authentication	Lab 2.1: Implementing Proper Authentication in PHP Lab 2.2: Enabling Google Authenticator in a PHP Web Application	100
3	A3: Sensitive Data Exposure	Lab 3.1: Password Hashing in PHP Lab 3.2: Proper error handling in PHP	70
4	A4: XML External Entities	Lab 4: Defending against XXE in PHP	35
5	A5: Broken Access Control	Lab 5.1: Basic Access Control in PHP Lab 5.2: Preventing Directory Traversal and LFI with Whitelisting in PHP	50
6	A6: Security Misconfiguration	Lab 6: Securing the PHP configuration	40
7	A7: Cross Site Scripting	Lab 7: Preventing XSS in PHP	55
8	A8: Insecure Deserialization	Lab 8: Secure Serialization in PHP	35
9	CSRF	Lab 9.1: Defending Against CSRF in PHP	35
10	File Upload Protection	Lab 9.2: Securely Handling File Uploads in PHP	50
11	Capstone	Lab 10: Capstone: Securing a Web Application From Top to Bottom in PHP	120
			Total: 11.5



Comtech
cyberstronger.com | comtechtel.com
275 West Street, Annapolis, MD 21401