SYLLABUS

PRACTICAL WEB DEFENSE
VERSION 1

The most practical and comprehensive training course on web application defense

eLearnSecurity has been chosen by students in over 140 countries in the world and by leading organizations such as:
COURSE DESCRIPTION

Practical Web Defense (PWD) is a fully guided and practical self-study course about how web applications are attacked in the real world and what you can do to mitigate every attack.

We illustrate exactly how each attack works, what the impact of each attack is, how to fix it, and how the exploit no longer works after the fix. **We also give you in-depth, practical advice about how to simplify your defense and how to implement attack mitigations that actually work.**

This course will:
- walk you through the process of identifying security issues,
- determine how they could be exploited,
- demonstrate proof-of-concept exploits,
- show you how to implement fixes for these security issues, and
- verify that the fixes work using your proof-of-concept exploits again.

Lessons are highly practical and will provide the knowledge and hands-on experience necessary to: identify, attack, fix and verify security issues in web applications.

Our labs will use relatively simple PHP applications and web services that implement vulnerabilities for all issues in the OWASP Testing Guide so that you know what the problem looks like in the real world. At the same time, these applications are simple enough so that even those without PHP experience will be able to apply the concepts to their relevant platform. Similarly, penetration testers without professional development experience are likely to be able to follow these examples without getting lost. This is particularly true given our guided Video-Lab course format, which will walk you through every step in the process.

We will also teach you what to do when you do not have access to source code or need to secure web applications without being able to change the code (i.e. until the next release or resources are available) using Virtual Patching.
WHO SHOULD TAKE THIS COURSE

We believe this course will be interesting and engaging for:

• Penetration testers

…who will be able to provide meaningful, actionable mitigation advice to customers. Some pentesters may also learn a thing or two attack-wise.

This course would also be beneficial to:

• Developers

…who will learn what they are defending from, how to implement effective defenses and how to verify if these defenses work.

We cover most areas in the OWASP Testing Guide and in many cases, go beyond that by providing real actionable mitigation advice that you can use in the real world. This course is practical, we won’t bore you with theories, and we hope you have as much fun taking the course as we had making it.

If you are willing to put the time to learn about practical web application attack and defense, then this course is for you.

WHO SHOULD NOT TAKE THIS COURSE

This course is probably not for you if you are looking for something that:

• Gives you a button to get all the vulnerabilities magically fixed
• Will give you a certification without any effort
• You can memorize to pass a multiple-choice test
• Will not make you think

HOW AM I GOING TO LEARN THIS?

The fun way of course!!! Don't worry, eLearnSecurity courses are very interactive, addictive, and presents content in such a way that appeals to all learning styles. During this training course, you will have to deal with several guided web defense challenges, that will provide you with relevant and hands-on practical application
experience. Don’t expect the outdated way of learning by reading pages and pages of theoretical methodologies.

NO BORING THEORIES ABOUT THE UNIVERSE HERE!

We take our learning to the next level by not only explaining how an attack works, but furthering your experience by showing you how it works in practice. With real examples, labs and mitigation advice that reflect real-world web application vulnerabilities and fixes.

DETAILED PRACTICAL INFORMATION ABOUT HOW WEB APPLICATION ATTACKS WORK

Many developers will not fix a security issue unless they understand exactly what the problem is, which is a valid point! We give you fully working examples of impact derived from security vulnerabilities so that:

- You have a reason to fix the problem
- You can show your development team exactly what the problem is
- You can prove to management that there is a problem and you can, therefore, get a budget
- You have a way to verify if the problem is really fixed
- Without knowing how the attack works, how will you know if the fix works?

DETAILED PRACTICAL INFORMATION ABOUT HOW WEB APPLICATION DEFENSES WORK

Many penetration testers might provide poor mitigation advice such as “you just need to validate your inputs”; however, in this course, whether you are a penetration tester or a developer you will learn how to implement effective web application defenses so that:

- You can provide effective and actionable mitigation advice to your customers (if you are a pentester)
- You can implement effective and actionable mitigations for any security problems you find/are told to fix (if you are a developer)
- Since you are also given a way to demonstrate security issues, you will be able to verify if your security fixes are really working.

DETAILED ILLUSTRATED PRACTICAL TECHNIQUES AND METHODOLOGY TO SIMPLIFY DEFENSE OF WEB APPLICATIONS.
ILLUSTRATED PRACTICAL USAGE OF SECURITY TOOLS TO FIND SECURITY ISSUES IMPLEMENT AND VERIFY FIXES

OWASP ZAP, OWASP OWTF, ModSecurity, cURL, Wireshark, Hydra, John, etc.

CAN I TRACK MY LEARNING PROGRESS?

...or will I only find out during the exam if I learned something?

The answer to these questions is very simple. Your achievements will tell. Each practical chapter of the course has a Video Lab associated with it. We will solve these together, while we explain to you all the necessary concepts. You are then free to practice the labs as long as you want. If you can solve a challenge, that demonstrates that you learned and properly understood the concepts.

IS THERE A FINAL EXAMINATION?

Yes. The final examination consists of two parts. The first part is a multiple-choice quiz test. Once you have passed this, you will proceed to the hands-on examination. During this second part of your exam you will have to solve a complex Web Defense Challenge.

WILL I GET A CERTIFICATE?

The PWD course leads to the eWDP certification.

Once you passed the complete final examination, you are an "eLearnSecurity Web Defense Professional" and will hold the eWDP certification.

You can print your shiny new certificate directly or have it shipped to you internationally.
ORGANIZATION OF CONTENTS

The student is provided with a suggested learning path to ensure the maximum success rate at the minimal effort.

- Module 1: Tool Introduction
- Module 2: Information Gathering
- Module 3: Configuration Management
- Module 4: Authentication
- Module 5: Authorization
- Module 6: Session Management
- Module 7: Business Logic Flaws
- Module 8: Data Validation
- Module 9: Cryptography
- Module 10: Denial of Service
- Module 11: Web Services
- Module 12: Client Side and Phishing
- Module 13: Error Handling and Logging
- Module 14: Applied Secure Coding Principles
- Module 15: Virtual Patching and Intrusion Detection
- Module 16: Securing Web Applications
Module 1: Tool Introduction

This module illustrates the basic usage of the main tools used during this course:

- Kali Linux
- OWASP ZAP
- OWASP OWTF
- cURL

Lab and videos included in this module

1. Tool Introduction
   1.1. Introduction
   1.2. OWASP ZAP Basics
      1.2.1. Starting ZAP
      1.2.2. Setting up interception of HTTP traffic
      1.2.3. Inspecting HTTP traffic
      1.2.4. Spidering a website
      1.2.5. Scanning a website
      1.2.6. HTTP request replay
      1.2.7. Tactical Fuzzing
      1.2.8. Break points: Interactive HTTP tampering
      1.2.9. ZAP Add-ons
      1.2.10. ZAP User Guide
      1.2.11. Further reading
   1.3. OWASP OWTF Basics
      1.3.1. Installing OWTF in Kali
      1.3.2. OWTF usage options
      1.3.3. Further reading
   1.4. cURL Basics
      1.4.1. Starting cURL
      1.4.2. Installing lynx
      1.4.3. Basic cURL usage
      1.4.4. Further reading
MODULE 2: INFORMATION GATHERING

This module focuses on examples of attack vs. defense, against information gathering attacks.

You'll see a quick demonstration of how attackers may attempt to target your website and how you can defend against each potential attack. This part of the course is very important to understand how a malicious attacker may attempt to exploit a security problem on your website and how each of these attacks can be mitigated.

Lab and video included in this module

2. Information Gathering
   2.1. Introduction
   2.2. Search Engines and Robots.txt
      2.2.1. Introduction
      2.2.2. What the problem is
      2.2.3. How can I see if I am vulnerable to this?
      2.2.4. How can I fix this?
      2.2.5. Further reading
   2.3. Comments and Metadata Information Leakage
      2.3.1. Introduction
      2.3.2. What the problem is
      2.3.3. How can I see if I am vulnerable to this?
      2.3.4. How can I fix this?
      2.3.5. Removing metadata before deploying to production
      2.3.6. Removing comments before deploying to production
      2.3.7. Further reading
   2.4. Web server software in use and patching
      2.4.1. Introduction
      2.4.2. What the problem is
      2.4.3. How can I see if I am vulnerable to this?
      2.4.4. How can I fix this?
      2.4.5. Patching and mitigation:
      2.4.6. Hiding the software version (Obscurity):
      2.4.7. Further reading
   2.5. Services, Port Scanning and Whois – App Discovery
      2.5.1. Introduction
      2.5.2. What the problem is
      2.5.3. How can I see if I am vulnerable to this?
2.5.4. Regular port-scans:
2.5.5. Regular verification of Whois records:
2.5.6. Regular verification of DNS records
2.5.7. How can I fix this?
2.5.8. The minimum number of ports are open on your web server
2.5.9. Whois records are generic-enough
2.5.10. DNS records do not reveal unintended servers/services
2.5.11. Further reading

2.6. Application Attack Surface – Entry Points
2.6.1. Introduction
2.6.2. What the problem is
2.6.3. How can I see if I am vulnerable to this?
2.6.4. How can I fix this?
2.6.5. Further reading

MODULE 3: CONFIGURATION MANAGEMENT

This module focuses on specific defense tactics against configuration management attacks.

Misconfiguration can sometimes be the weakest link in the security posture of a web application and is a common problem in modern web applications. This is illustrated by the fact that security misconfiguration has been listed among the OWASP Top 10 application security risks in the past two editions.

Lab and video included in this module

3. Configuration Management
3.1. Introduction
3.2. Admin Interfaces and Default Passwords
   3.2.1. Introduction
   3.2.2. What the problem is
   3.2.3. How can I see if I am vulnerable to this?
   3.2.4. How can I fix this?
   3.2.5. Further reading
3.3. Database Listeners and Connection Strings
   3.3.1. Introduction
   3.3.2. What the problem is
   3.3.3. How can I see if I am vulnerable to this?
   3.3.4. How can I fix this?
3.3.5. Further reading

3.4. Enabled HTTP Methods
3.4.1. Introduction
3.4.2. What the problem is
3.4.3. How can I see if I am vulnerable to this?
3.4.4. How can I fix this?
3.4.5. Further reading

3.5. Old, Backup and Unreferenced Files
3.5.1. Introduction
3.5.2. What the problem is
3.5.3. How can I see if I am vulnerable to this?
3.5.4. How can I fix this?
3.5.5. Further reading

3.6. File Uploads and File Extension Handling
3.6.1. Introduction
3.6.2. What the problem is
3.6.3. How can I see if I am vulnerable to this?
3.6.4. How can I fix this?
3.6.5. Further reading

3.7. Infrastructure and Application Configuration Management
3.7.1. Introduction
3.7.2. What the problem is
3.7.3. How can I see if I am vulnerable to this?
3.7.4. How can I fix this?
3.7.5. Further reading

**MODULE 4: AUTHENTICATION**

This module focuses on specific defense tactics against authentication attacks.

Authentication refers to protection mechanisms of content that require logging into a web application. Common protections are login, forgotten password or multiple factor authentication (MFA).

This module, will cover how these protections may be attacked and how defenders can implement tactical protections against these attacks.

Lab and video included in this module

4. Authentication
   4.1. Introduction
4.2. Credentials Transported Over an Encrypted Channel
   4.2.1. Introduction
   4.2.2. What the problem is
   4.2.3. How can I see if I am vulnerable to this?
   4.2.4. How can I fix this?
   4.2.5. Further reading

4.3. User Enumeration and Guessable Accounts
   4.3.1. Introduction
   4.3.2. What the problem is
   4.3.3. How can I see if I am vulnerable to this?
   4.3.4. How can I fix this?
   4.3.5. Further reading

4.4. Brute Force, Default Credentials and Lock Out
   4.4.1. Introduction
   4.4.2. What the problem is
   4.4.3. How can I see if I am vulnerable to this?
   4.4.4. How can I fix this?
   4.4.5. Further reading

4.5. Bypassing the Authentication Schema and ‘Remember Me’
   4.5.1. Introduction
   4.5.2. What the problem is
   4.5.3. How can I see if I am vulnerable to this?
   4.5.4. How can I fix this?
   4.5.5. Further reading

4.6. The Browser Cache and Autocomplete
   4.6.1. Introduction
   4.6.2. What the problem is
   4.6.3. How can I see if I am vulnerable to this?
   4.6.4. How can I fix this?
   4.6.5. Further reading

4.7. Password Policies
   4.7.1. Introduction
   4.7.2. What the problem is
   4.7.3. How can I see if I am vulnerable to this?
   4.7.4. How can I fix this?
   4.7.5. Further reading

4.8. Security Questions and Password Reset/Change
   4.8.1. Introduction
   4.8.2. What the problem is
   4.8.3. How can I see if I am vulnerable to this?
   4.8.4. How can I fix this?
4.8.5. Further reading

4.9. CAPTCHA
   4.9.1. Introduction
   4.9.2. What the problem is
   4.9.3. How can I see if I am vulnerable to this?
   4.9.4. How can I fix this?
   4.9.5. Further reading

4.10. Multiple Factor Authentication
   4.10.1. Introduction
   4.10.2. What the problem is
   4.10.3. How can I see if I am vulnerable to this?
   4.10.4. How can I fix this?
   4.10.5. Further reading

**MODULE 5: AUTHORIZATION**

This module focuses on specific defense tactics against authorization attacks.

Authorization attacks involve bypassing access permission to resources and accessing information that the user does not have permission to access.

In this module, we will study how authorization attacks work and how we can defend against them.

Lab and video included in this module

5. Authorization
   5.1. Introduction
   5.2. Path Traversal and LFI
      5.2.1. Introduction
      5.2.2. What the problem is
      5.2.3. How can I see if I am vulnerable to this?
      5.2.4. How can I fix this?
      5.2.5. Further reading
   5.3. Authorization Bypasses, Privilege Escalation and Direct Object References
      5.3.1. Introduction
      5.3.2. What the problem is
      5.3.3. How can I see if I am vulnerable to this?
      5.3.4. How can I fix this?
      5.3.5. Further reading
This module focuses on specific defense tactics against session management attacks.

Most websites that have a login page need to keep track of user sessions. It would be inconvenient for a user to have to login every time they click a link on your web application. Since HTTP is a stateless protocol, websites use mechanisms like platform libraries, cookies, etc. to keep track of user sessions.

Session management tries to balance security with convenience and is often chosen as an attack vector to impersonate application users.

Lab and video included in this module

6. Session Management
   6.1. Introduction
   6.2. Bypassing the Session Management Schema
       6.2.1. Introduction
       6.2.2. What the problem is
       6.2.3. How can I see if I am vulnerable to this?
       6.2.4. How can I fix this?
       6.2.5. Further reading
   6.3. Cookie Attributes
       6.3.1. Introduction
       6.3.2. What the problem is
       6.3.3. How can I see if I am vulnerable to this?
       6.3.4. How can I fix this?
       6.3.5. Further reading
   6.4. Session Fixation
       6.4.1. Introduction
       6.4.2. What the problem is
       6.4.3. How can I see if I am vulnerable to this?
       6.4.4. How can I fix this?
       6.4.5. Further reading
   6.5. Exposed Session Variables
       6.5.1. Introduction
       6.5.2. What the problem is
       6.5.3. How can I see if I am vulnerable to this?
       6.5.4. How can I fix this?
6.5.5. Further reading  

6.6. Cross-Site Request Forgery (CSRF)  
6.6.1. Introduction  
6.6.2. What the problem is  
6.6.3. How can I see if I am vulnerable to this?  
6.6.4. How can I fix this?  
6.6.5. Further reading  

MODULE 7: BUSINESS LOGIC FLAWS  

This module focuses on specific defense tactics against business logic attacks. 

Business logic attacks are perhaps some of the most difficult flaws to defend against because they are the kind of security problems that automated tools are particularly poor at finding. 

Given that most web applications are custom-made, business logic flaws vary in nature from one application to another. This makes business logic flaws challenging to find since there is no clear pattern to search for other than business complexity. 

Lab and video included in this module  

7. Business Logic Flaws  
7.1. Introduction  
7.2. Business Logic Flaws  
7.2.1. Introduction  
7.2.2. What the problem is  
7.2.3. How can I see if I am vulnerable to this?  
7.2.4. How can I fix this?  
7.2.5. Further reading  

MODULE 8: DATA VALIDATION  

This module focuses on specific defense tactics against data validation attacks. 

As you will see in this module, “data validation”, in many cases, depends on what the data is going to be used for later, and this may not be easy to anticipate. 

In the majority of situations, data validation can be summarized as a problem that happens when “code and input get mixed up” which typically happens when strings
containing “code” and the strings containing “input” are concatenated together and then executed.

You will see real-world examples of this issue throughout this module.

Lab and video included in this module

8. Data Validation
  8.1. Introduction
  8.2. Reflected and Stored Cross-Site Scripting (XSS)
    8.2.1. Introduction
    8.2.2. What the problem is
    8.2.3. How can I see if I am vulnerable to this?
    8.2.4. How can I fix this?
    8.2.5. Further reading
  8.3. HTTP Parameter Pollution (HPP)
    8.3.1. Introduction
    8.3.2. What the problem is
    8.3.3. How can I see if I am vulnerable to this?
    8.3.4. How can I fix this?
    8.3.5. Further reading
  8.4. Unvalidated Redirects and Forwards
    8.4.1. Introduction
    8.4.2. What the problem is
    8.4.3. How can I see if I am vulnerable to this?
    8.4.4. How can I fix this?
    8.4.5. Further reading
  8.5. ORM and SQL Injection (SQLI)
    8.5.1. Introduction
    8.5.2. What the problem is
    8.5.3. How can I see if I am vulnerable to this?
    8.5.4. How can I fix this?
    8.5.5. Further reading
  8.6. LDAP Injection
    8.6.1. Introduction
    8.6.2. What the problem is
    8.6.3. How can I see if I am vulnerable to this?
    8.6.4. How can I fix this?
    8.6.5. Further reading
  8.7. XML and XXE Injection
    8.7.1. Introduction
8.7.2. What the problem is
8.7.3. How can I see if I am vulnerable to this?
8.7.4. How can I fix this?
8.7.5. Further reading

8.8. SSI Injection
8.8.1. Introduction
8.8.2. What the problem is
8.8.3. How can I see if I am vulnerable to this?
8.8.4. How can I fix this?
8.8.5. Further reading

8.9. XPath Injection
8.9.1. Introduction
8.9.2. What the problem is
8.9.3. How can I see if I am vulnerable to this?
8.9.4. How can I fix this?
8.9.5. Further reading

8.10. MX Injection (IMAP/POP3/SMTP)
8.10.1. Introduction
8.10.2. What the problem is
8.10.3. How can I see if I am vulnerable to this?
8.10.4. How can I fix this?
8.10.5. Further reading

8.11. Code Injection and RFI
8.11.1. Introduction
8.11.2. What the problem is
8.11.3. How can I see if I am vulnerable to this?
8.11.4. How can I fix this?
8.11.5. Further reading

8.12. Command Injection
8.12.1. Introduction
8.12.2. What the problem is
8.12.3. How can I see if I am vulnerable to this?
8.12.4. How can I fix this?
8.12.5. Further reading

8.13. Buffer Overflow
8.13.1. Introduction
8.13.2. What the problem is
8.13.3. How can I see if I am vulnerable to this?
8.13.4. How can I fix this?
8.13.5. Further reading

8.14. HTTP Splitting/Smuggling
8.14.1. Introduction
8.14.2. What the problem is
8.14.3. How can I see if I am vulnerable to this?
8.14.4. How can I fix this?
8.14.5. Further reading

MODULE 9: CRYPTOGRAPHY

This module focuses on specific defense tactics against attacks to try to get around cryptographic security controls.

Since ancient times, humans have worried about interception and modification of messages sent from one location to another. Cryptography was invented to solve these problems.

The modern web applications of today try to solve the same ancient problem, i.e. prevent unwanted interception and modification of information. Thankfully, developers do not need to become elite cryptographers to develop web applications with solid cryptographic security controls, but it is important to understand how to adequately protect information.

Lab and video included in this module

9. Cryptography
   9.1. Introduction
   9.2. Cryptography and Data Storage
      9.2.1. Introduction
      9.2.2. What the problem is
      9.2.3. How can I see if I am vulnerable to this?
      9.2.4. How can I fix this?
      9.2.5. Further reading
   9.3. SSL/TLS Configuration
      9.3.1. Introduction
      9.3.2. What the problem is
      9.3.3. How can I see if I am vulnerable to this?
      9.3.4. How can I fix this?
      9.3.5. Further reading
   9.4. HTTP Strict Transport Security (HSTS)
      9.4.1. Introduction
      9.4.2. What the problem is
MODULE 10: DENIAL OF SERVICE

This module focuses on specific defense tactics against denial of service attacks.

Information security can be broken down into three essential features (CIA):
- Confidentiality
- Integrity
- Availability

A Denial of Service (DoS) is an attack against availability and therefore an information security concern.

Denial of Service attacks focus on crashing the system or at least making the system overuse enough resources (i.e. CPU, memory, disk, network, etc.) to prevent legitimate users from using it.

Lab and video included in this module

10. Denial of Service
   10.1. Introduction
   10.2. SQL Wildcard DOS
      10.2.1. Introduction
      10.2.2. What the problem is
      10.2.3. How can I see if I am vulnerable to this?
      10.2.4. How can I fix this?
      10.2.5. Further reading
   10.3. Regular Expression DOS (REDOS)
      10.3.1. Introduction
      10.3.2. What the problem is
      10.3.3. How can I see if I am vulnerable to this?
      10.3.4. How can I fix this?
      10.3.5. Further reading
   10.4. Allocation Logic Flaw DOS
      10.4.1. Introduction
      10.4.2. What the problem is
      10.4.3. How can I see if I am vulnerable to this?
10.4.4. How can I fix this?
10.4.5. Further reading

MODULE 11: WEB SERVICES

This module focuses on specific defense tactics against web service attacks.

Web Services are server-side programs that provide an API, or in other words, functionality that can be easily invoked programmatically.

Lab and videos included in this module

11. Web Services
   11.1. Introduction
   11.2. XML-RPC Basics
       11.2.1. Introduction
       11.2.2. Basic XML-RPC usage
       11.2.3. Fingerprinting XML-RPC web services
       11.2.4. Attacking XML-RPC web services
       11.2.5. XML-RPC defense guidelines
   11.3. JSON-RPC Basics
       11.3.1. Introduction
       11.3.2. Basic JSON-RPC usage
       11.3.3. Fingerprinting JSON-RPC web services
       11.3.4. Attacking JSON-RPC web services
       11.3.5. JSON-RPC defense guidelines
   11.4. SOAP Basics
       11.4.1. Introduction
       11.4.2. Basic SOAP usage
       11.4.3. Fingerprinting SOAP web services
       11.4.4. Attacking SOAP web services
       11.4.5. SOAP defense guidelines
   11.5. REST Basics
       11.5.1. Introduction
       11.5.2. Basic REST usage
       11.5.3. Fingerprinting REST web services
       11.5.4. Attacking REST web services
       11.5.5. REST defense guidelines
   11.6. Web Service Information Gathering
       11.6.1. Introduction
11.6.2. What the problem is
11.6.3. How can I see if I am vulnerable to this?
11.6.4. How can I fix this?
11.6.5. Further reading

11.7. SOAP/REST Action Spoofing
   11.7.1. Introduction
   11.7.2. What the problem is
   11.7.3. How can I see if I am vulnerable to this?
   11.7.4. How can I fix this?
   11.7.5. Further reading

11.8. XML Attacks
   11.8.1. Introduction
   11.8.2. How can I see if I am vulnerable to this?
   11.8.3. How can I fix this?
   11.8.4. Further reading

11.9. File Upload Attacks
   11.9.1. Introduction
   11.9.2. How can I see if I am vulnerable to this?
   11.9.3. How can I fix this?
   11.9.4. Further reading

11.10. Replay Attacks
   11.10.1. Introduction
   11.10.2. What the problem is
   11.10.3. How can I see if I am vulnerable to this?
   11.10.4. How can I fix this?
   11.10.5. Further reading

MODULE 12: CLIENT SIDE AND PHISHING

This module focuses on specific defense tactics against client-side attacks and phishing.

Lab and video included in this module

12. Client Side and Phishing
   12.1. Introduction
   12.2. DOM-Based Cross Site Scripting (DOM XSS)
      12.2.1. Introduction
      12.2.2. What the problem is
      12.2.3. How can I see if I am vulnerable to this?
      12.2.4. How can I fix this?
12.2.5. Further reading

12.3. Clickjacking (UI Redressing)
   12.3.1. Introduction
   12.3.2. What the problem is
   12.3.3. How can I see if I am vulnerable to this?
   12.3.4. How can I fix this?
   12.3.5. Further reading

12.4. Flash Files
   12.4.1. Introduction
   12.4.2. What the problem is
   12.4.3. How can I see if I am vulnerable to this?
   12.4.4. How can I fix this?
   12.4.5. Further reading

12.5. Silverlight Files
   12.5.1. Introduction
   12.5.2. What the problem is
   12.5.3. How can I see if I am vulnerable to this?
   12.5.4. How can I fix this?
   12.5.5. Further reading

12.6. HTML5
   12.6.1. Introduction
   12.6.2. What the problem is
   12.6.3. CORS
   12.6.4. Web Messaging
   12.6.5. Web Sockets and Server-Sent Events (SSE)
   12.6.6. Web Storage and Geolocation
   12.6.7. Additional events, tags, attributes and sandboxed iframes
   12.6.8. Further reading

MODULE 13: ERROR HANDLING AND LOGGING

This module focuses on specific defense tactics to handle error conditions and logging.

Sooner or later all web applications face unexpected conditions, typically due to input that was not anticipated but potentially also related to environmental factors such as server load. Regardless of the reason, web applications need to handle the following very carefully:

- Handle errors gracefully
- Log errors so that they can be reviewed later
Lab and video included in this module

13. Error Handling and Logging
   13.1. Introduction
   13.2. Error Handling and Logging
      13.2.1. Introduction
      13.2.2. What the problem is
      13.2.3. How can I see if I am vulnerable to this?
      13.2.4. How can I fix this?
      13.2.5. Further reading

**MODULE 14: APPLIED SECURE CODING PRINCIPLES**

This module focuses on general strategies to write secure code.

Defending against every possible attack is non-trivial. In addition to this, new attacks are being published every year, and it is unreasonable to expect developers to know and defend against every “cutting-edge attack that was published last week.” Because of this, this module will illustrate the general and widely accepted principles to produce secure code against known and future (not-yet-known) attacks.

The principles in this module aim to produce web applications that enforce the core pillars of information security:

- Confidentiality
- Integrity
- Availability

   14.1. Introduction
   14.2. Reduce the Attack Surface
      14.2.1. Principle description
      14.2.2. Practical examples
      14.2.3. Further Reading
   14.3. Strict Input Validation
      14.3.1. Principle description
      14.3.2. Practical examples
      14.3.3. Further Reading
   14.4. Aggressively Differentiate Input from Code
14.4.1. Principle description
14.4.2. Practical examples
14.4.3. Further Reading

14.5. Secure Access to Data and Function
14.5.1. Principle description
14.5.2. Practical examples
14.5.3. Further Reading

14.6. Secure Communications and Storage
14.6.1. Principle description
14.6.2. Secure communication examples
14.6.3. Secure storage examples
14.6.4. Further Reading

14.7. Least Privilege
14.7.1. Principle description
14.7.2. Web server examples
14.7.3. Database examples
14.7.4. Further Reading

14.8. Defense in Depth
14.8.1. Principle description
14.8.2. DOM XSS prevention in HTML5 postMessage example
14.8.3. Reflected XSS prevention example
14.8.4. Further Reading

14.9. Secure Defaults
14.9.1. Principle description
14.9.2. Practical examples
14.9.3. Further Reading

14.10. Make Security Simple and Obvious
14.10.1. Principle description
14.10.2. Data labeling example
14.10.3. Stopping execution as soon as a security check fails example
14.10.4. Centralized security control example
14.10.5. Further Reading
MODULE 15: VIRTUAL PATCHING AND INTRUSION DETECTION

This module defines strategies to:

- Mitigate security issues when fixing the root cause might not be immediately possible
- Detect and quickly respond to security breaches

This module will cover the following basic concepts:

- Web Application Firewall
- Virtual Patching
- Intrusion Detection
- Intrusion Prevention

Lab and video included in this module

15. Virtual Patching and Intrusion Detection
   15.1. Introduction
   15.2. ModSecurity Basics
       15.2.1. Introduction
       15.2.2. ModSecurity essentials in Kali Linux
       15.2.3. Rule Phases: 1-5
       15.2.4. Basic security rule structure and usage
       15.2.5. Rule Variables
       15.2.6. Rule Operators
       15.2.7. Rule transformations
       15.2.8. Rule Actions
   15.3. Virtual Patching
       15.3.1. Introduction
       15.3.2. Creating Virtual Patches
       15.3.3. Further reading
   15.4. Intrusion Detection
       15.4.1. Introduction
       15.4.2. Techniques
       15.4.3. Automated solutions
       15.4.4. Further reading
15.5. Intrusion Prevention
  15.5.1. Introduction
  15.5.2. Automated solutions
  15.5.3. Further reading

**MODULE 16: SECURING WEB APPLICATIONS**

This module focuses on the widely accepted best practices necessary to secure web applications.

It is important to note that securing web applications is a continuous process. Even a perfectly secure web application will eventually become vulnerable as new attack vectors are discovered, vulnerabilities against libraries and as software packages become known and published, etc.

Exam preparation lab and video included in this module

16. Securing Web Applications
  16.1. Introduction
  16.2. Preliminaries: Governance
    16.2.1. Introduction
    16.2.2. Strategy and Metrics
    16.2.3. Policy and Compliance
    16.2.4. Training
    16.2.5. Further reading
  16.3. Pre-Development: Threat Modeling and Design
    16.3.1. Introduction
    16.3.2. Types of attacker and motivations
    16.3.3. Attack models
    16.3.4. Example Design Requirements
    16.3.5. Further reading
  16.4. Development: Architecture
    16.4.1. Introduction
    16.4.2. Architectural decision examples
    16.4.3. Architecture review and validation
    16.4.4. Further reading
  16.5. Development: Code Reviews
    16.5.1. Introduction
    16.5.2. Static analysis tools
    16.5.3. Manual reviews
16.5.4. Further reading

   16.6.1. Introduction
   16.6.2. Dynamic analysis tools
   16.6.3. Manual dynamic analysis
   16.6.4. Further reading

16.7. Deployment: Hardening
   16.7.1. Introduction
   16.7.2. OS hardening
   16.7.3. Web server hardening
   16.7.4. Application hardening
   16.7.5. Further reading

16.8. Deployment: Penetration Testing
   16.8.1. Introduction
   16.8.2. Maximizing the value of a Penetration Test
   16.8.3. In-House Penetration Testing Team
   16.8.4. External Penetration Testing Companies
   16.8.5. Regular Penetration Testing
   16.8.6. Further reading

16.9. Post-Deployment: Regular Scanning
   16.9.1. Introduction
   16.9.2. Configuration Management program
   16.9.3. Vulnerability Management and regular scanning
   16.9.4. Further reading
We are eLearnSecurity.

Based in Santa Clara, California, with offices in Pisa, Italy, and Dubai, UAE, Caendra Inc. is a trusted source of IT security skills for IT professionals and corporations of all sizes. Caendra Inc. is the Silicon Valley-based company behind the eLearnSecurity brand.

eLearnSecurity has proven to be a leading innovator in the field of practical security training, with best of breed virtualization technology, in-house projects such as Coliseum Web Application Security Framework and Hera Network Security Lab, which has changed the way students learn and practice new skills.

Contact details:

www.elearnsecurity.com
contactus@elearnsecurity.com