

SYLLABUS



WEB APPLICATION PENETRATION TESTING VERSION 3

The most practical and comprehensive training course on web application pentesting



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



COURSE GOALS

The Web Application Penetration Testing course (WAPT) is an online, self-paced training course that provides all the advanced skills necessary to carry out a thorough and professional penetration test against modern web applications.

Thanks to the extensive use of Hera Lab and the coverage of the latest research in the web application security field, the WAPT course is not only the most practical training course on the subject but also the most up to date.

This course, although based on the offensive approach, provides advice and best practices to solve security issues detected during a penetration test.

COURSE ORGANIZATION

The training course is completely self-paced with interactive slides and videos that students can access online without any limitation. Students have lifetime access to the training material.

Students can study from home, office or anywhere an internet connection is available.

This course, Web Application Penetration Testing v3, is integrated with Hera Labs, the most sophisticated virtual lab in IT Security. A minimum of 60 hours is advised. For more intensive use, 120 hours may be necessary. The Hera Lab provides a dedicated and isolated environment where a student can practice topics seen in the course.

TARGET AUDIENCE AND PRE-REQUISITES

The WAPT training course benefits the career of penetration testers and IT Security personnel in charge of defending their organization's web applications.

This course allows organizations of all sizes to assess and mitigate the risks their web applications are exposed to, by building strong, practical in-house skills.

Penetration testing companies can now train their teams with a comprehensive and practical training course without having to deploy internal labs that are often outdated and not backed by solid theoretical material.

A student who wants to enroll in the course must possess a solid understanding of web applications and web application security models.

No programming skills are required. However, snippets of JavaScript/HTML/PHP code will be used during the course.

WILL I GET A CERTIFICATE?



The WAPT course leads to the eWPTv1 certification.

The certification can be obtained by successfully completing the requirements, which is a practical penetration test exam that consists of complex, real-world web application that is hosted in our eLearnSecurity Hera Labs.

An eWPTv1 voucher is included in all the plans of the WAPT course.

ORGANIZATION OF CONTENTS

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

- Module 1: Penetration Testing Process
- Module 2: Introduction to Web Applications
- Module 3: Information Gathering
- Module 4: Cross-Site Scripting
- Module 5: SQL Injection
- Module 6: Authentication and Authorization
- Module 7: Session Security
- Module 8: Flash Security
- Module 9: HTML5
- Module 10: File and Resource Attacks
- Module 11: Other Attacks
- Module 12: Web Services
- Module 13: XPath
- Module 14: Penetration Testing Content Management Systems
- Module 15: Penetration Testing NoSQL Databases

MODULE 1: PENETRATION TESTING PROCESS

This module helps the penetration tester gain confidence with the processes and legal matters involved in a penetration testing engagement.

Students will learn methodologies and the best practice for reporting in order to become a confident and professional penetration tester.

This is a wealth of information that will be useful throughout the entire career of a penetration tester.

1. Introduction

1.1. Pre-engagement

- 1.1.1. Rules of Engagement
 - 1.1.1.1. Goal
 - 1.1.1.2. Scope of engagement
- 1.1.2. Timetable
- 1.1.3. Liabilities and Responsibilities
 - 1.1.3.1. Non-disclosure agreements
 - 1.1.3.2. Emergency Plan
- 1.1.4. Allowed Techniques
- 1.1.5. Deliverables

1.2. Methodologies

- 1.2.1. PTES
- 1.2.2. OWASP Testing Guide

1.3. Reporting

- 1.3.1. What do clients want?
- 1.3.2. Writing the report
 - 1.3.2.1. Reporting Phase
 - 1.3.2.2. Understanding your audience
 - 1.3.2.3. Report Structure
 - Executive Summary
 - Risk Exposure over time
 - Successful attacks by type
 - Vulnerabilities by cause
 - Vulnerability Report
 - Remediation Report
- 1.3.3. Report templates and guides

MODULE 2: INTRODUCTION TO WEB APPLICATIONS

During this introductory module, the student will learn and understand the basics of web applications.

In-depth coverage of the Same Origin Policy and cookies will help both experienced and non-experienced penetration testers gain critical foundational skills useful for the rest of the training course.

At the end of the module, the student will become familiar with tools such as Burp Suite and OWASP ZAP.

This module is an important introduction necessary for a heavily-practical, advanced course.

2. Introduction to Web Applications

2.1. HTTP/S Protocol Basics

- 2.1.1. HTTP Request
- 2.1.2. HTTP Response
- 2.1.3. HTTP Header Field Definitions
- 2.1.4. HTTPS

2.2. Encoding

- 2.2.1. Introduction
- 2.2.2. Charset
 - 2.2.2.1. ASCII
 - 2.2.2.2. Unicode
- 2.2.3. Charset vs. Charset Encoding
 - 2.2.3.1. Unicode Encoding
 - 2.2.3.2. HTML Encoding
 - HTML Entries
 - 2.2.3.3. URL Encoding (percent encoding)
 - 2.2.3.4. Base64

2.3. Same Origin

- 2.3.1. Origin definition
- 2.3.2. What does SOP protect from?
- 2.3.3. How SOP works
- 2.3.4. Exceptions
 - 2.3.4.1. Windows.location
 - 2.3.4.2. Document.domain
 - 2.3.4.3. Cross window messaging
 - 2.3.4.4. Cross Origin Resource Sharing

2.4. Cookies

2.4.1. Cookies Domain

2.4.1.1. Specified cookie domain

2.4.1.2. Unspecified cookie domain

2.4.1.3. Internet Explorer Exception

2.4.2. Inspecting the Cookie Protocol

2.4.2.1. Login

2.4.2.2. Set-Cookie

2.4.2.3. Cookie

2.4.3. Cookie Installation

2.4.3.1. Correct cookie installation

2.4.3.2. Incorrect cookie installation

2.5. Sessions

2.6. Web Application Proxies

2.6.1. Burp Suite

2.6.2. OWASP ZAP

MODULE 3: INFORMATION GATHERING

Every penetration test begins with the Information Gathering phase. This is where a pentester understands the application under a functional point of view and collects useful information for the following phases of the engagement.

A multitude of techniques will be used to collect behavioral, functional, applicative, and infrastructural information.

The students will use a variety of tools to retrieve readily-available information from the target.

3. Information Gathering

3.1. Gathering information on your target

3.1.1. Finding owner, IP, and emails

3.1.1.1. Whois

- Command line

- Web-based tool

3.1.1.2. DNS

3.1.1.3. Nslookup

- Find target ISP

- Netcraft

3.2. Infrastructure

3.2.1. Fingerprinting the web server

3.2.1.1. Netcat

3.2.1.2. WhatWeb

3.2.1.3. Wappalyzer

3.2.1.4. Web server modules

3.2.2. Enumerating subdomains

3.2.2.1. Netcraft

3.2.2.2. Google

3.2.2.3. Subbrute

3.2.2.4. Dnsrecon

3.2.2.5. TheHarvester

3.2.2.6. Zone transfer

3.2.3. Finding virtual hosts

3.3. Fingerprinting frameworks and applications

3.3.1. Third party add-ons

3.3.2. Mapping results

3.4. Fingerprinting custom applications

3.4.1. Burp target crawler

3.4.2. Creating a functional graph

3.4.3. Mapping the attack surface

3.4.3.1. Client side validation

3.4.3.2. Database interaction

3.4.3.3. File uploading and downloading

3.4.3.4. Display of user-supplied data

3.4.3.5. Redirections

3.4.3.6. Access control and login-protected pages

3.4.3.7. Error messages

3.4.3.8. Charting

3.5. Enumerating resources

3.5.1. Crawling the website

3.5.2. Finding hidden files

3.5.2.1. Back up and source code

3.5.2.2. Enumerating users accounts

3.5.2.3. Map

3.6. Relevant information through misconfigurations

3.6.1. Directory listing

3.6.2. Log and configuration files

3.6.3. HTTP verbs and file upload

3.7. Google hacking

3.7.1. Search operators

3.8. Shodan HQ

MODULE 4: CROSS-SITE SCRIPTING

In this module, the most widespread web application vulnerability will be dissected and studied in depth.

At first, you are provided with a theoretical explanation—this understanding will help you in the exploitation and remediation process.

Later, you will have the opportunity master all the techniques to find XSS vulnerabilities through black box testing.

4. Cross-Site Scripting

4.1. Cross-Site Scripting

4.1.1. Basics

4.2. Anatomy of an XSS Exploitation

4.3. The three types of XSS

4.3.1. Reflected XSS

4.3.2. Persistent XSS

4.3.3. DOM-based XSS

4.4. Finding XSS

4.4.1. Finding XSS in PHP code

4.5. XSS Exploitation

4.5.1. XSS and Browsers

4.5.2. XSS Attacks

4.5.2.1. Cookie Stealing through XSS

4.5.2.2. Defacement

4.5.2.3. XSS for advanced phishing attacks

4.5.2.4. BeEF

4.6. Mitigation

4.6.1. Input Validation

4.6.2. Context-Aware output encoding

4.6.3. Never trust user input

MODULE 5: SQL INJECTION

This module will contain the most advanced techniques in finding and exploiting SQL injections, from the explanation of the most basic SQL injection up to the most advanced.

Advanced methods will be taught with real-world examples using the best tools, and demonstrated on real targets.

You will not just be able to dump remote databases but also get root on the remote machine through advanced SQL Injection techniques.

5. SQL Injection

5.1. Introduction to SQL Injections

5.1.1. SQL Statements

5.1.1.1. SELECT

5.1.1.2. UNION

5.1.2. SQL Queries inside web applications

5.1.3. Vulnerable dynamic queries

5.1.4. How dangerous is a SQL Injection

5.1.5. SQLi attacks classification

5.1.5.1. In-band SQLi

5.1.5.2. Error-based SQLi

5.1.5.3. Blind SQLi

5.2. Finding SQL Injections

5.2.1. Simple SQL Injection scenario

5.2.2. SQL errors in web applications

5.2.3. Boolean-based detection

5.2.3.1. Example

5.3. Exploiting In-band SQL Injections

5.3.1. First scenario

5.3.2. In-band attack challenges

5.3.3. Enumerating the number of fields in a query

5.3.3.1. Different DBMS UNION mismatch errors

5.3.4. Blind enumeration

5.3.5. Identifying field types

5.3.6. Dumping the database content

5.4. Exploiting Error-based SQL Injections

- 5.4.1. MS SQL Server Error-based exploitation
- 5.4.2. The CAST Technique
- 5.4.3. Finding the DBMS version
- 5.4.4. Dumping the database data
 - 5.4.4.1. Finding the current username
 - 5.4.4.2. Finding readable databases
 - 5.4.4.3. Enumerating database tables
 - 5.4.4.4. Enumerating columns
 - 5.4.4.5. Dumping data
- 5.4.5. Video – Error-based SQLi
- 5.4.6. MySQL Error-based SQLi
- 5.4.7. PostgreSQL Error-based SQLi
- 5.4.8. Developing Error-based SQLi Payloads

5.5. Exploiting blind SQLi

- 5.5.1. String extraction
- 5.5.2. Detecting the current user
- 5.5.3. Scripting blind SQLi data dump
- 5.5.4. Exploiting blind SQLi
 - 5.5.4.1. String extraction
- 5.5.5. Optimize blind SQLi
- 5.5.6. Time-based blind SQLi

5.6. SQLMap

- 5.6.1. Basic syntax
- 5.6.2. Extracting the database banner
- 5.6.3. Information Gathering
- 5.6.4. Extracting the Database
- 5.6.5. Extracting the Schema
- 5.6.6. Video – SQL Injection
- 5.6.7. Video – SQLMap
- 5.6.8. SQLMap Advanced Usage
 - 5.6.8.1. Forcing the DBMS
 - 5.6.8.2. Fine tuning the payloads
 - 5.6.8.3. Aggressiveness and load
- 5.6.9. Conclusions

5.7. Mitigation Strategies

- 5.7.1. Prepare statement
 - 5.7.1.1. Implementation
- 5.7.2. Type casting
- 5.7.3. Input validation

5.8. From SQLi to Server Takeover

5.8.1. Advanced MySQL Exploitation

5.8.1.1. xp_cmdshell

5.8.1.2. Internet Network Host Enumeration

5.8.1.3. Port Scanning

5.8.1.4. Reading the File System

5.8.1.5. Uploading Files

5.8.1.6. Storing Command Results into a Temporary Table

5.8.2. Advanced MySQL Exploitation

5.8.2.1. Reading the File System

5.8.2.2. Uploading Files

5.8.2.3. Executing Shell Commands

5.8.3. Conclusions

MODULE 6: AUTHENTICATION AND AUTHORIZATION

Any application with a minimum of complexity requires authentication at some point.

The chances are that the authentication mechanisms in place are not sufficient or are simply broken, exposing the organization to serious security issues leading to a complete compromise of the web application and the data it stores.

In this module, the student will learn the most common authentication mechanisms, their weaknesses and the related attacks: from inadequate password policies to weaknesses in the implementation of common features.

6. Authentication and Authorization

6.1. Introduction

- 6.1.1. Authentication vs. Authorization
- 6.1.2. Authentication factors
 - 6.1.2.1. Single-factor authentication
 - 6.1.2.2. Two-factor authentication

6.2. Common Vulnerabilities

- 6.2.1. Credentials over unencrypted channel
- 6.2.2. Inadequate password policy
 - 6.2.2.1. Dictionary attacks
 - 6.2.2.2. Brute force attacks
 - 6.2.2.3. Defending from inadequate password policy
 - Strong password policy
 - Storing hashes
 - Lockout/Blocking requests
- 6.2.3. User enumeration
 - 6.2.3.1. Via error messages
 - 6.2.3.2. Via website behavior
 - 6.2.3.3. Via timing attacks
 - 6.2.3.4. Taking advantage of user enumeration
- 6.2.4. Default or easily-guessable user accounts
- 6.2.5. The remember me functionality
 - 6.2.5.1. Cache browser method
 - 6.2.5.2. Cookie method
 - 6.2.5.3. Web storage method
 - 6.2.5.4. Best defensive techniques

6.2.6. Password reset feature

- 6.2.6.1. Easily guessable answers

- 6.2.6.2. Unlimited attempts

- 6.2.6.3. Password reset link

6.2.7. Logout weaknesses

- 6.2.7.1. Incorrect session destruction

6.2.8. CAPTCHA

6.3. Bypassing Authorization

6.3.1. Insecure direct object references

- 6.3.1.1. Best defensive techniques

6.3.2. Missing function level access control

6.3.3. Parameter modification

- 6.3.3.1. Vulnerable web application

6.3.4. Incorrect redirection

- 6.3.4.1. Redirect to protect contents

- 6.3.4.2. Best defensive techniques

6.3.5. SessionID prediction

6.3.6. SQL Injections

6.3.7. Local file inclusion and path traversal

MODULE 7: SESSION SECURITY

Session-related vulnerabilities, along with extensive coverage of the most common attacking patterns are the subject of this module.

Code samples on how to prevent session attacks are provided in PHP, Java and .NET. At the end of the module, the student will master offensive as well as defensive procedures related to session management within web applications.

7. Session Security

7.1. Weaknesses of the session identifier

7.2. Session hijacking

7.2.1. Session Hijacking via XSS

7.2.1.1. Exploit session hijacking via XSS

7.2.1.2. Preventing session hijacking via XSS

PHP

Java

.NET

7.2.2. Session Hijacking via Packet Sniffing

7.2.3. Session Hijacking via access to the web server

7.3. Session Fixation

7.3.1. Attacks

7.3.1.1. Set the SessionID

7.3.1.2. Force the victim

7.3.1.3. Vulnerable web application

7.3.2. Preventing Session Fixation

7.4. Cross-Site Request Forgeries

7.4.1. Finding CSRF

7.4.2. Exploiting CSRF

7.4.3. Preventing CSRF

MODULE 8: FLASH SECURITY AND ATTACKS

Flash, although a dying technology, is still present on millions of websites. Flash files can expose a web application and its users to a number of security risks, which are covered in this module.

The student will first study the Flash security model and its pitfalls, and move on to using the most recent tools to find and exploit vulnerabilities in Flash files. After having studied this module, students will never look at SWF files the same way.

8. Flash Security and Attacks

8.1. Introduction

- 8.1.1. Actionscript
 - 8.1.1.1. Compiling and decompiling
- 8.1.2. Embedding Flash in HTML
 - 8.1.2.1. The allowScriptAccess attribute
- 8.1.3. Passing arguments to Flash files
 - 8.1.3.1. Direct reference
 - 8.1.3.2. Flash embedded in HTML
 - 8.1.3.3. FlashArgs attribute

8.2. Flash Security Model

- 8.2.1. Sandboxes
- 8.2.2. Stakeholders
 - 8.2.2.1. Administrative role
 - 8.2.2.2. User role
 - 8.2.2.3. Website role
 - 8.2.2.4. URL policy file
 - 8.2.2.5. Author role
- 8.2.3. Calling JavaScript from ActionScript
- 8.2.4. Calling ActionScript from JavaScript
- 8.2.5. Method NavigateToURL
- 8.2.6. Local shared object

8.3. Flash Vulnerabilities

- 8.3.1. Flash parameter injection
- 8.3.2. Fuzzing Flash with SWFInvestigator
- 8.3.3. Finding hardcoded sensitive information

8.4. Pentesting Flash Applications

- 8.4.1. Analyzing client-side components
- 8.4.2. Identifying communication protocol
- 8.4.3. Analyzing server-side components

MODULE 9: HTML5

This module provides an extremely in-depth coverage of all the attack vectors and weaknesses introduced by drafted as well as finalized W3C new standards and protocols.

We will go through the most important elements of HTML5 and especially the new CORS paradigm that completely changes the way the SOP is applied to most modern web applications. By mastering this module in theory and practice, the student will possess an arsenal of penetration testing techniques that are still unknown to the vast majority of penetration testers.

A number of Hera labs are available to practice topics covered within this module. This module will also bring a penetration tester's skills to the next level with next-generation attack vectors that are going to affect web applications for the next decade.

9. HTML5

9.1. Cross-Origin Resource Sharing

9.1.1. Same Origin Policy issues

9.1.2. Cross-Domain Policy in Flash

9.1.3. Cross-Origin Resource Sharing

9.1.3.1. Cross-Origin Ajax requests

9.1.3.2. Requests

Simple request

Preflighted request

Request with credentials

9.1.3.3. Access Control Headers

Access-Control-Allow-Origin

Access-Control-Allow-Credentials

Access-Control-Allow-Headers

Access-Control-Allow-Methods

Access-Control-Allow-Max-Age

Access-Control-Expose-Headers

Header origin

Access-Control-Request-Method

Access-Control-Request-Headers

9.2. Cross-Window Messaging

- 9.2.1. Relationship between windows
- 9.2.2. Sending messages
- 9.2.3. Receiving messages
- 9.2.4. Security issues
 - 9.2.4.1. Cross-Domain XSS

9.3. Web Storage

- 9.3.1. Different storages
 - 9.3.1.1. Local storage
 - 9.3.1.2. Session storage
- 9.3.2. Local storage APIs
 - 9.3.2.1. Adding an item
 - 9.3.2.2. Retrieving an item
 - 9.3.2.3. Removing an item
 - 9.3.2.4. Removing all items
- 9.3.3. sessionStorage APIs
- 9.3.4. Security Issues
 - 9.3.4.1. Stealing local storage via JS

9.4. WebSocket

- 9.4.1. Real-time applications using HTTP
- 9.4.2. WebSocket – a new W3C standard
 - 9.4.2.1. Benefits
- 9.4.3. WebSocket API
- 9.4.4. Security Issues

9.5. Sandboxed frames

- 9.5.1. Security issues before HTML5
 - 9.5.1.1. Redirection
 - 9.5.1.2. Accessing the parent document from iframe
- 9.5.2. HTML5 sandbox attribute

MODULE 10: FILE AND RESOURCE ATTACKS

During this module, the student will practice a number of vulnerabilities that affect web application files and resources.

The student will learn how to identify and exploit path traversal, file inclusion and unrestricted file upload vulnerabilities.

10. File and Resource Attacks

10.1. Path Traversal

- 10.1.1. Path conversion
- 10.1.2. Encoding
- 10.1.3. Best defensive techniques

10.2. File Inclusion Vulnerabilities

- 10.2.1. Local File Inclusion (LFI)
- 10.2.2. Remote File Inclusion (RFI)

10.3. Unrestricted File Upload

- 10.3.1. Vulnerable web application
 - 10.3.1.1. The attack
- 10.3.2. Best defensive techniques
 - 10.3.2.1. Filtering based on file content

MODULE 11: OTHER ATTACKS AND VULNERABILITIES

During this module, the student will practice a number of vulnerabilities that, despite being less known or publicized, are still affecting a number of web applications across many different programming languages and platforms.

Advanced clickjacking attacks are covered in depth with real-world examples and dissected real-world attacks.

The level of depth and the amount of practical sessions during this module will provide even seasoned penetration testers with new ways to break the security of their targets.

11. Other Attacks

11.1. Clickjacking

11.1.1. Understanding Clickjacking

11.1.2. Feasibility study

11.1.2.1. Case 1: Clickjacking is possible

11.1.2.2. Case 2: Clickjacking is not possible

11.1.3. Building of a malicious web page

11.1.4. Spreading the malicious link

11.1.5. Waiting for the victim click

11.1.6. Best defensive techniques

11.1.6.1. The old school

11.1.6.2. Using HTTP header X-Frame-Options

11.1.7. Likejacking in Facebook

11.1.8. Cursorjacking

11.2. HTTP Response Splitting

11.2.1. Typical vulnerable scenario

11.2.2. XSS through HTTP response splitting

11.2.3. Bypassing Same Origin Policy

11.2.3.1. Attack explained

11.2.3.2. Best defensive techniques

11.2.3.3. Defense in PHP

11.3. Business Logic Flow

11.3.1. Vulnerable web application

11.3.2. Best defensive techniques

11.4. Denial of Services

11.4.1. Different DoS attacks

11.4.1.1. DoS due to huge number of requests

11.4.1.2. DoS due to greedy pages

11.4.2. Best defensive techniques

MODULE 12: WEB SERVICES

Professional penetration testers should master all aspects related to web services testing.

Web services nowadays are the data and logic provider for a variety of thin and thick clients, from web application clients to mobile applications.

During this highly in-depth module, the student will first become familiar with web services paradigms and protocols and then learn all the most important related security issues.

WSDL and SOAP testing will be covered not only in theory but also in practice in our Hera Lab.

12. Web Services

12.1. Introduction

12.2. Web Services Implementations

12.2.1. XML-RPC

12.2.2. JSON-RPC

12.2.3. SOAP

12.2.4. RESTful

12.3. The WSDL Language

12.3.1. Interaction between client and server

12.3.2. Objects in the WSDL

12.3.2.1. Binding

12.3.2.2. PortType

12.3.2.3. Operation

12.3.2.4. Interface

12.3.2.5. Message

12.3.3. SOAP in action

12.3.4. Further reading

12.4. Attacks

12.4.1. WSDL Disclosure

12.4.1.1. Google hacking

12.4.1.2. Discovering WSDL files

12.4.1.3. Public Web Services

12.4.2. WSDL Scanning

12.4.2.1. Attack in action

12.4.3. SOAPAction Spoofing

12.4.3.1. Prerequisites for the attack

12.4.3.2. Attack in action

12.4.3.3. Best defensive techniques

12.4.4. SQLi through SOAP messages

12.4.4.1. Best defensive techniques

MODULE 13: XPATH INJECTION

XPath is the XML standard that allows web applications to query XML databases.

In this module, the student will learn advanced XPath injection techniques, in theory and practice in Hera lab.

13. XPath Injection

13.1. XML Documents and Databases

13.2. XPath

13.2.1. XPath expression and syntax

13.2.2. XPath vs. SQL

13.3. Detecting XPath Injection

13.3.1. Error-based injection

13.3.2. Blind injection

13.3.2.1. Detect true condition

13.3.2.2. Detect false condition

13.3.3. Exploitation

13.3.3.1. Bypass XPath query

13.3.3.2. Extracting the XML document structure

13.3.3.3. Finding out the root node

13.3.3.4. Finding the first child node name

13.3.3.5. Finding the content of a node

13.4. Best Defensive Techniques

MODULE 14: PENETRATION TESTING CONTENT MANAGEMENT SYSTEMS

This module covers the whole range of penetration testing activities against CMS, from information gathering, enumeration and brute force attacks, to host exploitation through vulnerable plugins and lateral movement through credential reuse. More specifically, the student will get accustomed to identifying vulnerabilities like XSS, SQLi, RCE, SOME and CSRF on WordPress and Joomla CMS, as well as chaining various vulnerabilities for maximum exploitation.

14. Penetration Testing Content Management Systems

14.1. Introduction

14.2. WordPress

14.2.1. Information Gathering

14.2.1.1. WPScan

14.2.1.2. Plecost

14.2.1.3. Nmap NSE Scripts

14.2.1.4. Directory Indexing/Listing

14.2.2. Exploitation

14.2.2.1. Bruteforce Attacks

15.3.2.4.1 Bruteforce with WPScan

25.3.2.4.1 Bruteforce with wpbf

14.2.2.2. Attacking Plugins

35.3.2.4.1 From XSS to RCE

45.3.2.4.1 Malicious Plugins for Post-Exploitation & Persistence

14.3. Joomla

14.3.1. Information Gathering

14.3.1.1. Joomscan

14.3.1.2. Joomla Scan

14.3.1.3. Extensions

14.3.1.4. Content Discovery

14.3.2. Exploitation

14.3.2.1. Bruteforce Attacks

14.3.2.2. Vulnerabilities in Joomla Core

MODULE 15: Penetration Testing NoSQL Databases

In this module, the student will learn how to manually identify and exploit vulnerabilities in NoSQL databases or NoSQL-powered web applications, as well as execute elaborate attacks against exposed NoSQL-related APIs. Transitioning from a compromised NoSQL database to full host exploitation, as well as effective data exfiltration methods are also covered in this module.

15. Penetration Testing NoSQL Databases

15.1. Introduction

- 15.1.1. Pentesting NoSQL Databases Methodology

15.2. NoSQL Fundamentals & Security

15.2.1. MongoDB

- 15.2.1.1. MongoDB Fundamentals

- 15.2.1.2. MongoDB Security

- 15.2.1.3. MongoDB Penetration Testing Tools

15.2.2. CouchDB

- 15.2.2.1. CouchDB Fundamentals

- 15.2.2.2. CouchDB Security

- 15.2.2.3. CouchDB Exploitation Examples

15.2.3. Elasticsearch

- 15.2.3.1. Elasticsearch Fundamentals

- 15.2.3.2. Elasticsearch Security

15.2.4. Memcached

- 15.2.4.1. Memcached Fundamentals

- 15.2.4.2. Memcached Security

- 15.2.4.3. Memcached Exploitation Example

15.2.5. Redis

- 15.2.5.1. Redis Fundamentals

- 15.2.5.2. Redis Security

15.3. NoSQL Exploitation

- 15.3.1. NoSQL Injections

- 15.3.2. NoSQL Injection Categories

- 15.3.2.1. PHP Tautology Injections

- 15.3.2.2. NoSQL Union Query Injection

- 15.3.2.3. NoSQL JavaScript Injection

- 15.3.2.4. Piggybacked Queries

- 15.3.2.4.1 Real-Life Piggybacked Query Attack

15.3.2.5. Cross-Origin Violations

15.3.2.5.1 Real-Life Cross-Origin Exploitation Against MongoDB

15.3.2.6. NoSQL Injection in MEAN Stack Applications

15.3.2.6.1 NoSQL Injection in MEAN Stack Applications – Example 1

15.3.2.6.2 NoSQL Injection in MEAN Stack Applications – Example 2

15.3.2.6.3 NoSQL Injection in MEAN Stack Applications – Example 3

15.3.2.6.4 NoSQL Injection in MEAN Stack Applications – Example 4

15.3.2.6.5 NoSQL Injection in MEAN Stack Applications – Example 5

ABOUT US

A background image of the Golden Gate Bridge in San Francisco, California, during a sunset. The sky is a deep red, and the bridge's towers and suspension cables are silhouetted against the water and sky.

We are eLearnSecurity.

Based in Santa Clara, California, and 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

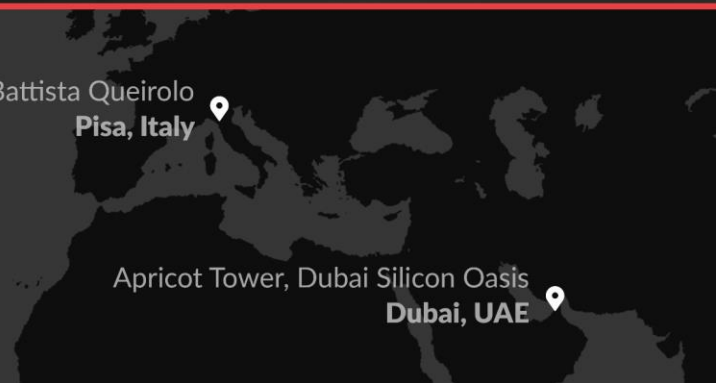
contactus@elearnsecurity.com

A dark map of the United States with a white location pin in California.

2040 Martin Ave.
Santa Clara, CA, USA

A dark map of Europe with a white location pin in Italy.

Via Gian Battista Queirolo
Pisa, Italy

A dark map of the Middle East with a white location pin in Dubai.

Apricot Tower, Dubai Silicon Oasis
Dubai, UAE