PEN 121:
Web Application Ethical Hacking and Penetration Testing

A 30-Hour Training Course

Course Description

This course dives you deep into the inner mechanisms of web applications and teaches you how these applications can be broken, analyzed, and penetrated. Web applications are complex and have many components, each of which has its attack surface. An increasing number of companies are developing their in-house web applications, and countless others are selling web apps to customers. It is of vital importance that these web apps are assessed thoroughly for any vulnerability and security weaknesses. The skills needed for such assessment are what you will gain out of this course. You will learn in this course all vulnerability classes about web applications, how each vulnerability class can be discovered, attacked, and mitigated. There will be in-depth coverage of the concepts, skills, and tools for Cross-Site Scripting (XSS) attacks, Cross-Site Request Forgery (CSRF) attacks, SQL Injection (SQLi) attacks, file inclusion attacks, command execution attacks, and many others. You will learn how to manually analyze a web application for vulnerabilities, in addition to using automated scanners and analyzers. Finally, the course will teach how web sessions can be broken and hijacked.

Who Should Attend?

This course is suitable for network security engineers, information security officers, network administrators, system analysts, web developers, full-stack developers, and all those who are interested in technically advancing their network security skills.

Duration

- Total Hours: 30 hours
- Total Days: 5 days or 10 days

Prerequisites

- Laptop: students must bring their laptops with the following specifications:
    - 8GB or more of RAM
    - USB
    - Wi-Fi (built-in or card)
- Kali Linux Live CD/DVD.
- Technical Knowledge: it is highly recommended that attendees have a background or familiarity with the following concepts:
    - HTML and CSS
    - PHP, JavaScript, and SQL

Outline

MODULE 01: Overview of Web Security

- Web Technology Architecture
- The HTTP Protocol
- The URL & Same Origin Policy (SOP)
- AJAX and Web Sockets
- Path Traversal and Directory Listing
- OWASP Testing Guide
- Web Pentest Tools: Nikto, Fiddler, Burp, ZAP

MODULE 02: Footprinting Web Applications

- Spidering and Crawling.
- Retrieving Cached/Archived Contents
- Analyzing Metafiles
- Webpage Comments and Metadata
- Identifying Inputs and Cookies
- Identifying Web Servers and Frameworks
- Enumerate HTTP Methods
- Error Code Analysis

MODULE 03: Web Authentication Attacks

- Basic Authentication
- Digest Authentication
- Certificate-based Authentication
- Integrated Windows Authentication
- Form-based Authentication

- Username Harvesting
- Weak Username and Password Policies
- Remember/Reset Password Functionalities

MODULE 04: BASIC INJECTION ATTACKS

- Dynamic File Inclusion Mechanism
- File Inclusion and Path Traversal
- Local File Inclusion (LFI)
- Remote File Inclusion (RFI)
- Server-Side Includes (SSI)

MODULE 05: OS COMMAND INJECTION ATTACKS

- Command Execution APIs
- Testing for Command Injection
- Blind Time-Based Command Injection
- Blind Network-Based Command Injection
- Establishing Shell Tunnels
- Web Shells

MODULE 06: SQL INJECTION ATTACKS

- The Basics of SQL Injection
- Bypassing Authentication
- Accessing the Database
- Blind SQL Injection
- Time-based SQL Injection
- UNION-based SQL Injection
- Automated SQL Injection using SQLMap
- Security Measures against SQL Injection

MODULE 07: CROSS-SITE SCRIPTING (XSS)

- First-Order XSS: Reflected XSS
- Second-Order XSS: Persistent (Stored) XSS
- DOM-based XSS
- Page Redirection with XSS
- Cookie Stealing with XSS
- Key Logging with XSS
- Bypassing XSS Filters

MODULE 08: SESSION MANAGEMENT ATTACKS

- Understanding Session IDs
- Cookie Weaknesses
- Session Hijacking
- Session Fixation
- Cookie Reuse

MODULE 09: CROSS-SITE REQUEST FORGERY (CSRF)

- Discovering CSRF Vulnerabilities
- Exploiting CSRF
  - GET-based CSRF Attacks
  - POST-based CRSF Attacks
- CSRF Counter-Measures
  - Protection Mechanisms with User Intervention
  - Protection Mechanisms with no User Intervention