Web Application Hacking

Course 4027 – 40 Hours

Overview

This course is a practical guide to discovering and exploiting security flaws in web applications. By “web applications” we mean those that are accessed using a web browser to communicate with a web server. We examine a wide variety of different technologies, such as databases, file systems, and web services, but only in the context in which these are employed by web applications. Throughout the course, we spell out the specific steps you need to follow to detect each type of vulnerability, and how to exploit it to perform unauthorized actions.

What you will learn:

- Improve your understanding of web vulnerabilities and security
- Manual and automated application mapping
- Anatomy of server attacks; from injection to root
- Learn how to attack Authentication and Session
- Learn how to attack and exploit data-stores (SQL, noSQL)
- Learn Client attack techniques (XSS, CSRF)
- Learn how to attack Access Controls
- Learn how to attacking Back-End Components and Application Logic
- Gain better understanding in OS & service hardening
- Gain understanding in input sanitation and data validation
- Learn about web application filters and firewalls

Who Should Attend

- Penetration testers
- Web application developers
- Security vendors and service providers (IPS, WAF, Browser security)
- Web-focused QA with aspirations in security

Prerequisites

- Hands-on familiarity with web technologies (HTML, javascript, SQL, server-code)
- Good system skills (Linux or WIndows, preferably both)
- Good coding skills (any of these will do: java, python, c#, javascript, php …)
- Good understanding in networking technologies (IP, TCP, DNS, HTTP, HTTPs)

Course Contents

Part 1: Web application (in)security

- Server OS platforms
- Web servers (apache, nginx, weblogic, IIS etc.)
- Server-side application frameworks and programing languages
- Data-stores; Relational (SQL), non-relational (noSQL) and document-based
- Client-side technologies (Browsers, Browser plugins, HTML, HTML5, CSS, javascript etc.)
- Offensive toolset and practice targets

Part 2: Mapping web applications

- Web spidering (automated and user directed)
- Discovering Hidden Content
- Application Pages Versus Functional Paths
- Discovering Hidden Parameters
- Identifying Entry Points for User Input
- Identifying Server-Side Technologies and Functionality
- Mapping the Attack Surface
- Transmitting Data Via the Client
- Capturing User Data: HTML Forms
- Capturing User Data: Browser Extensions

Part 3: Authentication and session attacks

- Authentication Technologies
- Design Flaws in Authentication Mechanisms
- Brute-forcing logins
- Implementation Flaws in Authentication
- Securing Authentication
- The Need for State
- Weaknesses in Token Generation
- Weaknesses in Session Token Handling
- Securing Session Management

Part 4: Attacking data stores

- Exploiting a Basic Vulnerability
- Injecting into Different Statement Types
- Finding SQL Injection Bugs
- Fingerprinting the Database
- Extracting Data with UNION
- Performing Blind injections
- Automating SQLi testing with sqlmap
- Advanced Exploitation (FS interaction, and code execution)
- Bypassing Filters
- Second-Order SQL Injection
- Beyond SQL Injection: Escalating the Database Attack

- Using SQL Exploitation Tools
- SQL Syntax and Error Reference
- Preventing SQL Injection

Part 5: Attacking Users: Cross-Site Scripting

- Varieties of XSS
- Real-World XSS Attacks
- Payloads for XSS Attacks
- Delivery Mechanisms for XSS Attacks
- Finding and Exploiting XSS Vulnerabilities
- Preventing Reflected and Stored XSS
- Evading security filters and browser checks

Part 6: Attacking Back-End Components and Application Logic

- Injecting OS Commands
- Manipulating File Paths
- Injecting objects and (de)serializers
- The Nature of Logic Flaws
- Real-World Logic Flaws
- Avoiding Logic Flaws
- Hardening OS networking stacks
- Implement host-based firewalls and “good” logging
- Hardening web application servers (Apache, IIS)

Part 7: boot2root attack scenario against a server

- Scanning, enumeration and resource mapping
- Testing for possible injections
- Detect vulnerable parser at other side
- Inject code
- Gather system information
- Establish control connection
- Enumerate attack surface (resources, services, users, permissions, execution paths etc)
- Escalate privileges
- Establish persistence
- Pivot

Part 8: boot2root: now it’s your turn (optional)

- Final CTF
- Students work on their own and submit reports
- Workshop summary