SEC542: Web App Penetration Testing and Ethical Hacking™

GWAPT
Web Application Penetration Tester
giac.org/gwapt

6 Day Program | 36 CPEs | Laptop Required

If an organization does not properly test and secure its web applications, adversaries can compromise critical systems, steal data, disrupt operations, and trigger regulatory fallout. Many still rely only on vulnerability scanners and assume these tools will reliably uncover real-world flaws.

SEC542 shows you how to move beyond push-button tools and perform focused, high-value web application penetration tests. You will learn a repeatable methodology to assess both Internet-facing and internal business applications that support sensitive workflows and data.

Through hands-on labs, you will practice finding and exploiting vulnerabilities such as SQL injection, XSS, deserialization bugs, SSRF, and file inclusion, then communicate business impact to stakeholders. This course lays a practical foundation in web application security; it will not make you an expert in a week, but it gives you the skills, process, and mindset to keep improving long after class ends.

You Will Be Able To

- Apply a structured OWASP-based web app testing methodology
- Map and probe web apps and APIs with modern tooling
- Exploit critical flaws, including injection, XSS, CSRF, SSRF, XXE, SSTI
- Chain smaller issues into remote code execution and data theft
- Automate testing with Python, Requests/httpx, and custom scripts
- Use Burp Suite, ZAP, ffuf, sqlmap, BeEF, and Metasploit effectively
- Assess authentication and access control, including bypass and privilege escalation

Business Takeaways

- Build a repeatable, defensible web application testing process
- Go beyond scanners to uncover real, exploitable attack paths
- Communicate technical findings clearly in business and risk terms
- Provide developers with focused, actionable remediation guidance
- Strengthen monitoring by recognizing logging and detection gaps
- Deliver professional reports, executive summaries, and debriefs for stakeholders
- Demonstrate how web app security supports overall organizational defense

Hands-On Web App Penetration Testing

SEC542 Web App Penetration Testing builds a complete methodology for testing modern web applications, walking students from reconnaissance and mapping through exploitation and reporting, using the OWASP Web Security Testing Guide and hands-on labs. Students review HTTP requests and responses, headers, cookies, HTTP methods, and TLS. Using intercepting proxies such as Burp Suite and OWASP ZAP, they learn to profile targets, enumerate attack surface, and spot configuration weaknesses.

The course moves into fuzzing, scanning, and APIs. Students practice input fuzzing, work with web APIs and OpenAPI definitions, and analyze authentication mechanisms and identity protocols, including JSON Web Tokens. Identity and access control weaknesses receive dedicated attention. Students perform username harvesting, password spraying, and account lockout testing, explore authentication bypass flaws, and move into authorization issues, including broken object-level and function-level authorization and privilege escalation. Students investigate prototype pollution, SQL injection, NoSQL injection, SSRF, and XML External Entity vulnerabilities. Students write Python scripts that use the Requests and httpx libraries to automate testing tasks. Insecure deserialization labs cover Java deserialization and Python pickling, and server-side template injection is explored.

SEC542 ties the technical content back to real pentesting work. Students discuss security logging and monitoring failures and examine issues that can lead to logic flaws in web applications. By the end of the course, students have a repeatable process for assessing web applications and a deep catalog of hands-on experience across the vulnerabilities that matter in modern environments.

"This course taught me to truly focus on the methodology while performing a pen test. During the capture-the-flag event, I realized how much time can be wasted if you fail to respect your methodology."

—Sean Rosado, **RavenEye**

GWAPT
Web Application Penetration Tester
giac.org/gwapt

GIAC Web Application Penetration Tester

The GIAC Web Application Penetration Tester (GWAPT) certification validates a practitioner's ability to better secure organizations through penetration testing and a thorough understanding of web application security issues. GWAPT certification holders have demonstrated knowledge of web application exploits and penetration testing methodology.

- Web application overview, authentication attacks, and configuration testing
- Web application session management, SQL injection attacks, and testing tools
- Cross-site request forgery and scripting, client injection attack, reconnaissance and mapping

Section Descriptions

SECTION 1: Introduction and Information Gathering

This first section of the web application penetration testing course covers essential techniques such as interception proxies, HTTP basics, information gathering, virtual host discovery, target profiling, HTTPS testing, and content spidering. Labs include configuring Burp Suite and conducting thorough assessments.

TOPICS: Web Application Penetration Testing Methodologies; Interception Proxies; HTTP Basics: Protocols, Requests and Responses; Virtual Host Discovery, Spidering, and Target Profiling; Security Testing Fundamentals

SECTION 2: Fuzzing, Scanning, APIs, and Authentication

Section 2 focuses on advanced web application security techniques, including fuzzing for vulnerability detection, information leakage analysis, and using Nuclei and Burp Suite Pro scanners. It also covers forced browsing for content discovery, API exploitation, various authentication methods, and federated identity protocols.

TOPICS: Master Techniques like Fuzzing and Information Leakage Detection; Utilize Advanced Scanning Tools like Burp Suite Pro; Use Forced Browsing to Find Unlinked Content; Identify and Exploit API Vulnerabilities Using Tools like Bruno; Federated Identity and Access Protocols

SECTION 3: Identity, AuthN/AuthZ Bypass, and Client-Side Attack

This section moves from username harvesting and blind password spraying through session management and authentication and authorization bypass, then into stored, reflected, and DOM-based XSS, payload construction, data exfiltration, and browser exploitation using tools such as Burp Suite, ffuf, DOM Invader, and BeEF.

TOPICS: Username Harvesting; Session Management and Token Randomness Analysis; Authentication and Authorization Bypass; Cross-Site Scripting Overview and Impacts; Client-side Testing, DOM, AJAX, Browser Developer Tools

SECTION 4: Prototype Pollution, Database and Command Injection, SSRF, and XXE

Students progress from prototype pollution and database injection (SQL and NoSQL) through command injection, SSRF, and XML external entities, learning to move from input-level flaws to full data access and system impact using tools like Burp Suite and sqlmap.

TOPICS: Prototype Pollution and Abuse of JavaScript's Inheritance Model; SQL and NoSQL Injection Techniques, Categories, and Impact; Database Injection Tooling and Automation with Burp Suite; Command Injection and Collaborator-Based Probing; SSRF and XXE Attacks

SECTION 5: CSRF, Serialization, SSTI, and Advanced Tools

This section advances from insecure deserialization, file inclusion, Python automation, SSTI, CSRF, and file upload exploitation to Metasploit-driven post-exploitation and the business side of penetration testing, tying technical attacks to logging, logic flaws, LLM risk, and reporting.

TOPICS: File Inclusion and Insecure Deserialization; Python Scripting and Pickling for Automating Web App Testing; Server-Side Template Injection; Security Logging and Monitoring Failures; Metasploit Framework Usage

SECTION 6: Capture the Flag

During Section 6, students compete in teams in the ranges.io platform, a powered web application penetration testing tournament. This capture-the-flag exercise lets them wield new or sharpened skills to answer questions, complete missions, exfiltrate data, and tackle progressive challenges with hints that support all skill levels and reinforce learning.

Who Should Attend

- General security practitioners
- Penetration testers
- Ethical hackers
- Web application developers
- Website designers, architects, and developers

NICE Framework Work Roles

- Security Control Assessor (OPM 612)
- Software Developer (OPM 621)
- Secure Software Assessor (OPM 622)
- System Testing and Evaluation Specialist (OPM 671)
- Information Systems Security Developer (OPM 631)
- Systems Developer (OPM 632)
- Vulnerability Assessment Analyst (OPM 541)
- Pen Tester (OPM 541)
- Exploitation Analyst (OPM 121)
- Target Developer (OPM 131)
- Cyber Ops Planner (OPM 332)

"Knowing everything from the Internet is not enough. This class has a sequential structure to understand the basics of pen testing."

— Vinita Mhapsekar, **Kaiser Permanente**