Offensive Security: Hacking Web Applications

Workshop & Hands-On Technical Lab Overview

Target Audience	Security Architects, Security Engineers, Software Engineers. No hacking experience required! A basic understanding of web applications is assumed.
Duration	4–6 Hours (Hands-On Technical Lab and Workshop)
Focus Area	Offensive Cyber Operations, Adversary Mindset, Attack Surface Exploitation
Format	On-site workshop for up to 15 attendees (English, Swedish, or Spanish)
Requirements	Internet access. Attendees bring their own laptops. No software installation required.

Objective

To defend our applications, we need to understand our adversary. We must understand the intruders' techniques and how they can attack our in-house applications using common web vulnerabilities.

This 6-hour practical workshop/lab avoids relying on automated hacking tools. To learn the true mechanisms behind these attacks, we will instead use a standard web browser (Firefox) equipped with built-in developer tools.

Practical Exercises

Attendees will operate within an isolated lab environment, systematically executing each phase. Each phase will reveal more details to the attacker, turning the exercises into a gamified treasure hunt!

- **Reconnaissance:** Browse the target application as an unauthenticated user using a web browser to understand its business logic. What functionality and endpoints can be accessed without authentication?
- **SQL Injection:** Locate an entry point vulnerable to SQL injection. Can we leverage this flaw to extract password hashes directly from the database?
- **CSRF (Cross-Site Request Forgery):** After cracking a hash retrieved from the previous step to gain authenticated access, analyze the new application surface to uncover a CSRF vulnerability. Craft a malicious phishing email to trigger this exploit against another user.
- **XSS (Cross-Site Scripting):** Exploit XSS vulnerabilities to manipulate client-side context and further accomplish the operational objective.
- **File Upload Vulnerabilities:** Analyze file upload functions within the application. Can we abuse unrestricted file uploads to execute code or escalate privileges?
- **Command Injection:** Identify a blind command injection flaw. Even though the application returns no output back to the attacker, can we still leverage it to achieve a reverse shell?
- **Container Security:** Round off the lab by demonstrating what happens when the vulnerable application is deployed inside a Kubernetes cluster. Learn how attackers steal cluster secrets and move laterally between compromised containers.