LESSON: Advanced Web Application Security

Before you Begin

This is the very last lesson of this course. Instructors may want to spin up the first lab of the module to get it ready and stable before doing it live when the lab slide comes up. Note that if, for some reason, you are behind in the slides or labs in terms of pacing or timing, you absolutely must finish up in this module.

For this lesson, instructors are required to ensure the following activities are completed:

- Review the “Lesson Opener” and “Real World Scenario” with the learners prior to starting the module.
- Throughout the module, you will find “Consider the Real World Scenario” slides. Review the questions found on these slides, tie the concepts back to the scenario discussed at the start of the lesson as well as content you are presenting, and encourage the learners to share their thoughts.
- Ensure learners are given opportunities for breaks throughout the lesson. The pacing guide below provides recommended breaks. However, there are additional breaks added in the slide deck, please use them only if needed.
- For each lesson, you will find a “Pulse Check” slide which is the opportunity for instructors to open a poll to gather feedback from the learners. Leave the poll open for about 1 minute and after you close the poll, immediately share the results with the learners. Encourage the learners to share their thoughts. This information will help the instructors as well as the learners better understand where they are with regards to the lesson.
- Labs are to be demonstrated live for each module. This is a must do. The demonstration of labs is the top priority for the lead instructor. While demonstrating each lab, encourage students to participate and explore.
- At the end of each lesson, it is important to take a few minutes to review the key concepts for the lesson, provide guidance on what the learners can do to prepare for the next lesson, and wrap up with open Q&A.

Summary

In this lesson, learners will explore two pivotal web application vulnerabilities: Cross-site scripting (XSS) and SQL injection. They'll grasp the intricacies of XSS, understanding its categories, such as stored XSS, reflected XSS, and DOM-based XSS, along with mitigation strategies involving regular expressions, encoding functions, and protective HTTP headers. Transitioning to SQL injection, learners will comprehend the manipulation of databases, unauthorized data access, and the structural differences between relational and non-relational databases. Key SQL database elements, including tables, columns, and primary keys, will be covered, alongside the significance of the information schema in cybersecurity. The lesson will delve into SQL queries in MySQL, SQL injection types, practical attack examples, and mitigation strategies, offering a comprehensive understanding of these critical security vulnerabilities and how to safeguard against them.

Objectives

- Define cross-site scripting (XXS) and its key points.
- Define JavaScript and its key attributes.
- Describe the three different ways that JavaScript is integrated into HTML pages.
- Explain the three types of XSS attacks.
- Describe XSS attack goals and XSS mitigation strategies.
- Define SQLi, its key points, and types.
- Explain the database's two types.
- Describe the SQL database structure and features.
- List and explain SQL queries in MySQL.
- Explain bypassing login authentication.
- Describe the SQLi attack flow.
- Define SQLi enumeration vectors and their main goals.
- Describe SQLMap as the prominent tool in SQLi automation.
- List SQLi goals and main mitigations.

Lesson Activities and Teaching Strategies

Estimated Time	Lesson Portion	Directions
< 2 min	**Lesson Opener:** Advanced Web Application Security	● Introduce learners to the importance of advanced web application security in cybersecurity.
< 5 min	**Real World Scenario:** Advanced Web Application Security	● Review the real world scenario challenge and inform learners that you will be constantly coming back to this scenario throughout the lesson to discover how to solve and apply concepts to this real situation.
35 min	**Cyber Uncovered:** XSS	● Provide an overview of cross-site scripting (XSS) as a prevalent security vulnerability in web applications. ● Emphasize the injection of malicious scripts by attackers and the resulting impact on user sessions and data security. ● Discuss JavaScript's role as a versatile scripting language that enhances web user experience. ● Highlight its client-side execution, interactivity, and universality across modern web browsers. ● Outline the three primary types of XSS attacks: Stored XSS, reflected XSS, and DOM-based XSS. ● Explain how each type exploits vulnerabilities in web applications. ● Explore the characteristics of stored cross-site scripting, emphasizing its permanence, wide impact, and common targets. ● Mention its severity as a security breach where an attacker's script is permanently stored on the server.

		● Examine the characteristics of reflected cross-site scripting, focusing on user-triggered execution and its non-persistent nature. ● Discuss its delivery methods, often distributed through emails or messages requiring user interaction. ● Explain how DOM-based XSS attacks exploit client-side scripting by manipulating the Document Object Model (DOM) in the browser. ● Highlight characteristics such as client-side execution, DOM manipulation, and triggers through user actions. ● Present the goals of XSS attacks, including session hijacking, data theft, malware spreading, defacement, phishing, and clickjacking. ● Discuss how attackers aim to achieve specific outcomes through these malicious activities. ● Discuss the essential components of effective XSS mitigation: Secure coding practices and protective HTTP headers. ● Provide examples of secure coding practices, such as input validation with regular expressions (RegEx) and encoding functions like htmlspecialchars and htmlentities. ● Emphasize the role of protective HTTP headers like Content-Security-Policy (CSP), X-Frame-Options, and HttpOnly Cookies in mitigating XSS risks. ● Use the provided code example, showcasing the use of htmlentities() in PHP scripts and the implementation of HTTP headers for added security. ● Be prepared to discuss the implication of the real world scenario presented at the beginning of class on network types and devices. There are specific prompts that you should ask learners to reflect on to apply this concept to the real world scenario.
5 min Break		
40 min	**Lab:** XSS with ChatGPT	● Remind learners to use this lab to practice and apply the concepts they have learned throughout the day. ● Learners will receive direct feedback on their lab to properly assess their knowledge and determine where they might need additional assistance.
5 min Pulse Check (wait for ~75% respondent rate then close the poll)		
5 min Break		
35 min	**Cyber Uncovered:** SQL Injection	● Highlight the critical nature of SQL injection as a vulnerability targeting the data layer of applications. ● Emphasize how attackers exploit this vulnerability to manipulate databases, bypass security measures, and execute administrative operations. ● Discuss the potential consequences of SQL injection, focusing on data compromise, security bypass, and the potential for extensive damage. ● Connect these key points to real world scenarios to illustrate the practical implications of SQL injection attacks.

		● Differentiate between relational and non-relational databases, outlining their structures, data integrity, and use cases. ● Use examples of popular databases to provide context and help students understand the practical applications of each type. ● Break down the fundamental elements of SQL databases, including tables, columns, rows, primary keys, and indexes. ● Illustrate the importance of these elements in maintaining data integrity and facilitating complex data manipulation. ● Explain the critical role of the information schema in providing insights into a database's metadata. ● Discuss its standardized structure, data accessibility, and cross-DBMS compatibility with a specific focus on cybersecurity perspectives. ● Provide an in-depth understanding of essential SQL queries in MySQL, such as SELECT, ORDER BY, UNION SELECT, and WHERE. ● Include practical examples and walk-throughs to enhance comprehension of query syntax and functionality. ● Categorize SQL injection attacks into in-band SQLi, inferential SQLi, and out-of-band SQLi. ● Highlight characteristics and examples of each type, making sure the distinctions between them are clear. ● Explore how SQL injection can be exploited to bypass login pages, focusing on the injection payload and the manipulated query. ● Use a practical example to demonstrate the step-by-step process of bypassing authentication. ● Discuss the significance of enumeration vectors in SQL injection attacks. ● Cover version, user, and structure enumeration, explaining how attackers use these vectors to gather specific information about the database. ● Introduce SQLMap as a premier SQL injection automation tool, emphasizing its functionality, versatility, and database support. ● Showcase key features of SQLMap, including database fingerprinting, data retrieval, and remote code execution capabilities. ● Outline the specific goals of SQL injection attacks, including access and exfiltration, database control and manipulation, authentication bypass, launching additional attacks, and service disruption. ● Discuss the real world implications of these goals and their potential impact on organizations. ● Teach effective strategies to mitigate SQL injection risks, focusing on input validation, sanitization, and the use of prepared statements with parameterized queries. ● Discuss the role of web application firewalls (WAFs) as a protective measure against SQL injection, highlighting their effectiveness in blocking known attack patterns.

		● Be prepared to discuss the implication of the real world scenario presented at the beginning of class on network types and devices. There are specific prompts that you should ask learners to reflect on to apply this concept to the real world scenario.
5 min Break		
40 min	**Lab:** SQL Injection Attacks	● Remind learners to use this lab to practice and apply the concepts they have learned throughout the day. ● Learners will receive direct feedback on their lab to properly assess their knowledge and determine where they might need additional assistance.
10 min	**Lesson Closure**	● For this lesson, spend just a few minutes reminding the learners what the key ”take-aways'' were from the lesson and what they should do to prepare for the next module. The take-aways discussion should include key concepts such as understanding Cross-Site Scripting (XSS) and SQL Injection (SQLi) vulnerabilities. Students should review this information prior to moving to the next module. ● Recommend that the students read-ahead and come prepared for the next lesson. ● Q&A
5 min	**End-of-Course Survey**	● Allocate 5 minutes to facilitate the completion of the End-of-Course Survey. ● Encourage learners to provide honest and constructive feedback about their learning experience.
	Additional Time Filler (if needed)	● Kahoot ● Discuss interview prep and questioning ● Use breakout rooms for additional lab practice ● Continue Real World Scenario Conversation