LESSON: Web Application Security Fundamentals

Before you Begin

This is the second to 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 must catch up during these last two modules.

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 discover the fundamental components of web applications, distinguishing between client-side and server-side technologies. They will understand the roles of HTML, CSS, and JavaScript in creating dynamic online experiences, while also exploring server-side technologies such as PHP, Node.js, SQL, and ASP.NET for managing databases and application logic. The lesson introduces key web servers like Apache, Nginx, and IIS, highlighting their significance in hosting and delivering web content. Learners will gain insights into essential HTTP request methods, specifically GET and POST, and their roles in client-server communication with a focus on security considerations. Tools like Nmap and cURL will be explored for identifying supported HTTP methods, aiding in understanding server behavior and security. The lesson also emphasizes the crucial role of HTTP headers in requests and responses for web security and content interpretation. Learners will be introduced to OWASP, a non-profit

organization dedicated to enhancing software security, and its widely recognized OWASP Top 10, covering critical web application security risks. OWASP's Secure Coding Cheat Sheets will be discussed, offering guidelines and best practices for secure coding in various programming languages. Finally, the lesson introduces Burp Suite, an advanced web proxy tool, highlighting its key components, such as traffic interception, Repeater, Intruder, and Sequencer tools. Learners will understand how Burp Suite facilitates the identification and analysis of security vulnerabilities, making it a valuable asset in the field of web penetration testing.

Objectives

- Define web application dynamics.
- Explain client-side and server-side technologies.
- Describe web servers.
- Explain the Apache Server.
- Describe HTTP request methods and detection.
- Explain HTTP headers.
- Explain security enhancements with HTTP headers.
- Define OWASP and the OWASP Top 10.
- Explain OWASP Secure Coding Cheat Sheets.
- Describe traffic interception tools.
- Explain the Burp Suite and its components.

Lesson Activities and Teaching Strategies

Estimated Time	Lesson Portion	Directions
< 2 min	**Lesson Opener:** Web Application Security Fundamentals	● Introduce learners to the importance of web application security fundamentals in cybersecurity.
< 5 min	**Real World Scenario:** Web Application Security Fundamentals	● 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.

25 min	**Cyber Uncovered:** Web Technologies	● Explain the role of the client side in a web application and its components, highlighting the user interface and interactivity. ● Differentiate between the client side and server side of web applications, emphasizing the functions performed by each. ● Explore the significance of HTML in web pages, describing its foundational structure and how it collaborates with CSS and JavaScript. ● Define the role of CSS in web pages, detailing its responsibilities in visual presentation and styling. ● Describe the functionality of JavaScript as a scripting language used to introduce interactivity to web pages, providing examples of its applications. ● Explore server-side technologies (PHP, Node.js, SQL, ASP.NET) and their contributions to the backend functionality of web applications. ● Provide an overview of major web servers (Apache, Nginx, IIS, AWS), highlighting their unique features and importance in hosting web content. ● Introduce Apache 2 as a widely-used web server software, detailing its capabilities, such as customizable error messages and support for various programming languages. ● Guide learners on the process of installing and managing Apache 2 on Kali Linux using commands like sudo systemctl start apache2 and sudo systemctl enable apache2. ● Explain the significance of the apache2.conf file in Apache 2 configuration, emphasizing its impact on server-wide performance and 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.
30 min	**Lab:** Secure Apache2 Configuration	● 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 Break		
25 min	**Cyber Uncovered:** Request Methods and Headers	● Begin by introducing the concept of web requests, emphasizing their significance in internet interactions, such as fetching web pages and submitting form data. ● Explore the GET method, explaining that it is primarily used for requesting data from a specific resource and is suitable for retrieving web pages. ● Highlight the visibility of data in the URL and the implications for sensitive information. ● Move on to the POST method, describing its purpose in sending data to a server to create or update a resource.

		● Emphasize its use in form submissions and file uploads and its more secure nature for handling sensitive information. ● Introduce the importance of identifying supported HTTP methods for understanding web application behavior and security. ● Discuss the tools Nmap and cURL for HTTP method detection and provide practical examples for their usage. ● Transition to HTTP headers, explaining their integral role in requests and responses. ● Differentiate between request headers, response headers, and entity headers, emphasizing the information they convey. ● Guide the learners through enabling the headers module in Apache, stressing its role in enhancing security. ● Walk them through the steps of editing the configuration file, adding security headers, and applying the changes, ensuring a hands-on experience. ● Discuss key security headers and their functions, such as X-Frame-Options, Content-Security-Policy (CSP), X-Content-Type-Options, Strict-Transport-Security (HSTS), and X-XSS-Protection. ● Provide examples for each header to illustrate their purpose in securing web applications. ● 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		
25 min	**Cyber Uncovered:** OWASP	● Start the lesson by introducing OWASP as an international non-profit organization focused on enhancing software security. ● Highlight the diverse community, including corporations, educational institutions, and individuals, contributing to OWASP's mission. ● Discuss the significance of the OWASP Top 10 as a standard awareness document for developers in web application security. ● Emphasize its role in identifying and addressing the ten most critical security risks for web applications. ● Mention the periodic updates to keep the list relevant in the evolving landscape. ● Break down each of the OWASP Top 10 vulnerabilities, providing real-world examples and consequences. ● Explore broken access control, cryptographic failures, injection, insecure design, security misconfiguration, vulnerable and outdated components, identification and authentication failures, software and data integrity failure, security logging and monitoring failures, and server-side request forgery. ● Encourage discussions around the OWASP Top 10 vulnerabilities, prompting learners to share their understanding and experiences. ● Facilitate conversations about the potential impact of these vulnerabilities on web applications.

		● Introduce practical scenarios or case studies related to the OWASP Top 10 vulnerabilities. ● Encourage learners to analyze and propose solutions to address these vulnerabilities in hypothetical situations. ● Explain the purpose of OWASP Secure Coding Cheat Sheets in providing coding recommendations for PHP and other languages. ● Discuss the importance of following secure coding practices to mitigate potential vulnerabilities. ● Address the challenges developers face in implementing OWASP's recommendations and the consequences of neglecting secure coding practices. ● Discuss common pitfalls and provide insights on how to avoid them. ● Conclude by highlighting the impact of OWASP in shaping industry practices and improving software security. ● Discuss how developers can actively engage with OWASP resources to stay informed about evolving security threats. ● 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 Pulse Check (wait for ~75% respondent rate then close the poll)		
5 min Break		
25 min	**Cyber Uncovered:** Burp Suite	● Provide an overview of the current landscape of traffic interception tools and their significance for security experts. ● Highlight the three main tools leading the market: Burp Suite, Zap, and Fiddler. ● Introduce Burp Suite as a graphical tool used to test web application security. ● Mention that it's written in Java and developed by PortSwigger Web Security. ● Emphasize its popularity in web penetration testing and its features, such as extensibility, versatility, and support for independent add-ons. ● Explore the main tools within Burp Suite, showcasing their significance in web application security. ● Use visuals (from the provided link) to familiarize students with the Burp Suite Dashboard and its primary functions. ● Explain the importance of configuring Burp Suite as a proxy for intercepting and analyzing web traffic. ● Highlight the default listening address and port (127.0.0.1:8080) and the necessity of configuring the web browser to use the proxy. ● Suggest the use of the FoxyProxy browser add-on for simplified proxy settings. ● Detail the capability of Burp Suite to intercept and modify HTTP/S requests and responses.

		● Discuss the configuration of rules to mark messages for interception, emphasizing the proxy tab as a key location for intercept options. ● Explore the main feature of traffic manipulation in Burp Suite, focusing on the Intercept and Repeater tools. ● Explain the process of delaying packets, inspecting traffic in Raw mode or Hex, and the options for editing, forwarding, or dropping traffic. ● Highlight the Repeater tool's role in handling repetitive transmissions. ● Explain how a previously captured packet can be manipulated and sent again, making it effective for testing specific client-server communication. ● Introduce the Intruder tool and its purpose in brute-forcing parameters and headers using various payloads. ● Walk through the process of marking packets for substitution, automatic identification of parameters, and clearing or manually selecting specific content. ● Provide an overview of the different attack types within Burp Suite's Intruder, including sniper, battering ram, pitchfork, and cluster bomb. ● Conclude the lesson by introducing the Sequencer tool and its main functionality in testing for randomness. ● 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
30 min	**Lab:** Intercept and Access with Burp Suite	● 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 HTTP communications and web application security best practices. 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
	Additional Time Filler (if needed)	● Kahoot ● Discuss interview prep and questioning ● Use breakout rooms for additional lab practice ● Continue Real World Scenario Conversation