CHAPTER V
CLOSING

5.1 Conclusion

Based on the research conducted on the Weskonek ERP System using the Grey-Box Penetration Testing approach with a hybrid methodology consisting of Penetration Testing Execution Standard (PTES), Common Weakness Enumeration (CWE), MITRE ATT&CK, and Common Vulnerability Scoring System (CVSS) v3.1, the following conclusions were obtained:

1. Based on the penetration testing results, the Weskonek ERP System still contains several potential security vulnerabilities that may affect the aspects of confidentiality, integrity, and availability. The vulnerabilities successfully identified and validated include SQL Injection, Cross-Site Scripting (XSS), Brute Force Login Page, and the risk of active session misuse. These findings indicate that the system requires security improvements, particularly in input validation, authentication management, cookie configuration, and user session protection. In general, the identified vulnerabilities are categorized within the Medium to High risk levels.
2. The vulnerability identification and analysis process was conducted based on the Penetration Testing Execution Standard (PTES) phases, namely pre-engagement interactions, intelligence gathering, threat modeling, vulnerability analysis, exploitation, post-exploitation, and reporting. The pre-engagement stage was used to determine the scope, target, and testing rules. The intelligence gathering stage collected information related to technologies, services, endpoints, parameters, directories, and the system's attack surface. The threat modeling stage was used to map potential threats against system assets. The vulnerability analysis stage was performed to identify and classify discovered weaknesses. The exploitation stage was conducted in a controlled manner to validate vulnerabilities. The post-exploitation stage was used to analyze the impact and severity of findings, while the reporting stage was used to compile testing results and provide mitigation recommendations.

3. The identified vulnerabilities were classified using the Common Weakness Enumeration (CWE) standard and mapped to attack tactics and techniques within the MITRE ATT&CK framework. SQL Injection was classified as CWE-89 because it is related to improper input handling in SQL queries and mapped to technique T1190: Exploit Public-Facing Application. Cross-Site Scripting was classified as CWE-79 because it is associated with insufficient input validation and sanitization on web pages and can be mapped to techniques T1190 and T1539: Steal Web Session Cookie when used to obtain user session information. Brute Force Login Page was classified as CWE-307 because the system does not implement adequate authentication attempt restrictions and was mapped to technique T1110: Brute Force. Meanwhile, the risk of active session misuse is related to weaknesses in session identifier protection and may lead to unauthorized session usage if the session identifier is obtained by unauthorized parties. This mapping demonstrates that the identified vulnerabilities are not only technical weaknesses but also have relationships with common attack patterns used by threat actors.
4. The severity level of each vulnerability was assessed using the Common Vulnerability Scoring System (CVSS) v3.1 as the basis for determining mitigation priorities. SQL Injection obtained a score of 8.3 with a High severity category because it can be exploited through the network using a valid user session and has a significant impact on data confidentiality and integrity. Brute Force Login Page obtained a score of 8.2 with a High severity category because it can be performed through the login page without requiring initial privileges and may lead to account takeover. Cross-Site Scripting obtained a score of 5.4 with a Medium severity category because it requires user interaction and has limited impacts on confidentiality and integrity. The risk of active session misuse obtained a score of 4.2 with a Medium severity category because active sessions may be misused if session identifiers are obtained, although previous sessions cannot be reused after logout.

The recommended mitigation actions include implementing parameterized queries or prepared statements, strengthening input validation and sanitization, applying output encoding, implementing Content Security Policy (CSP), configuring cookies with HttpOnly, Secure, and SameSite attributes, applying rate limiting, account lockout mechanisms, CAPTCHA on login pages, regenerating session identifiers after authentication, implementing appropriate session timeout policies, and fully invalidating sessions during logout. Therefore, the CVSS assessment results can be used as a basis for prioritizing security improvements in the Weskonek ERP System.

5.1.1 Suggestion

Based on the research results and conclusions obtained, several recommendations can be provided for the future development and improvement of the Weskonek ERP System security.

1. The Weskonek ERP System development team should improve input validation and sanitization mechanisms across all features that accept user input. Every input submitted to the system should be validated based on data type, character length, format, and usage context. This approach is essential to prevent vulnerabilities such as SQL Injection and Cross-Site Scripting.
2. Developers should implement parameterized queries or prepared statements for all database communication processes. The use of dynamic queries that directly combine user input must be avoided because it may create opportunities for SQL Injection exploitation.
3. The system should implement additional security mechanisms on the login page, such as rate limiting, temporary account lockout, CAPTCHA, and notifications for suspicious login attempts. These mechanisms can reduce the risk of brute force attacks performed automatically using various username and password combinations.
4. User session management should be strengthened by implementing session timeout, session regeneration after login, and complete session invalidation after logout. In addition, session cookies should be protected using

HttpOnly, Secure, and SameSite attributes to minimize the risk of session theft and misuse.

5. The system should implement Content Security Policy (CSP) and output encoding mechanisms to reduce the risk of malicious script execution in user browsers. These measures are important to strengthen protection against Cross-Site Scripting attacks.
6. System administrators should limit technical information exposed by the server, such as web server versions, frameworks, programming languages, and unnecessary configuration details. Such information may be exploited by attackers to identify vulnerabilities associated with the technologies being used.
7. Security testing should be conducted periodically, especially after feature updates, system upgrades, server migrations, or application configuration changes. Regular testing helps ensure that implemented security improvements do not introduce new vulnerabilities.
8. Future research may expand the testing scope by applying a broader security assessment approach, such as testing with multiple user privilege levels, conducting API security testing, performing deeper server configuration analysis, and including retesting activities after remediation has been implemented by developers.
9. Future research may also incorporate secure code review analysis if access to the source code is provided. This approach allows vulnerabilities to be analyzed more deeply at the code implementation level rather than relying solely on application-side scanning and exploitation results.
10. The company is recommended to establish a vulnerability management procedure, including periodic scanning schedules, finding documentation processes, remediation priorities, mitigation deadlines, and verification procedures. Such procedures can help the company maintain the security of the Weskonek ERP System continuously.