Jurnal E-Komtek
Vol. 9, No. 2 (2025) pp. 434-439
https://jurnal.politeknik-kebumen.ac.id/index.php/E-KOMTEK
p-ISSN : 2580-3719 e-ISSN : 2622-3066

Web Application Security Testing Against SQL Injection Attacks Using SQL Map

Teguh Rizki Saputra[1], Hafiq Ibnu Wardana[2], Alfian Nur Fariq[3], Riko Cahyono[4], Susanto[5]

[1-5] Department of Informatics Engineering, Universitas Semarang, Indonesia, 50196

hafiqibnuw@gmail.com

https://doi.org/10.37339/e-komtek.v9i2.2260

Published by Politeknik Piksi Ganesha Indonesia

Artikel Info
Submitted:
02-01-2025
Revised:
03-01-2026
Accepted:
06-01-2026
Online first :
06-01-2026

Abstract

Web application security is a major concern due to the growing prevalence of cyberattacks, particularly SQL injection attacks, which compromise the integrity, confidentiality, and availability of data. This study aims to assess the vulnerability of web applications to SQL injection attacks through penetration testing. This test is conducted using SQLMap, a tool that detects and exploits vulnerabilities via boolean-based and error-based blind SQL injection techniques. In addition, this study implements and tests the protection capability of the addslashes ()-based input filtering method in PHP. The test results indicate that SQLMAP is widely used because protection against this vulnerability can provide a robust approach to securing web applications. Thus, web applications are expected to be protected against attacks that damage existing data and systems.

Keywords: *SQL Injection, Web Application Security, Penetration Testing, SQLMAP, Input Filtering*

Abstrak

Keamanan aplikasi web menjadi perhatian utama akibat meningkatnya ancaman serangan cyber, terutama serangan injeksi SQL, yang mengancam integritas, kerahasiaan, dan ketersediaan data. Penelitian ini bertujuan untuk mengukur kerentanan aplikasi web terhadap serangan SQL injection dengan menggunakan metode pengujian penetrasi. Pengujian ini dilakukan dengan menggunakan tool berupa SQLMAP yang dapat mendeteksi dan mengeksploitasi kerentanan melalui teknik blind SQL injection berbasis boolean dan teknik error-based injection. Selain itu, penelitian ini juga mengimplementasikan dan menguji kemampuan proteksi menggunakan metode input filtering berbasis addlashes() di PHP. Hasil pengujian menunjukkan bahwa SQLMAP banyak digunakan karena perlindungan terhadap kerentanan ini dapat memberikan solusi kuat tentang cara melindungi aplikasi web. Dengan begitu, aplikasi web diharapkan aman dari serangan yang merusak data dan sistem yang ada

***Kata-kata kunci**: SQL Injection, Keamanan Aplikasi Web, Pengujian Penetrasi, SQLMAP, Filterisasi Input*

1. Introduction

Web application security is a major concern with the rise of cyberattacks, including SQL Injection. This attack allows attackers to insert malicious SQL commands into database queries, allowing them to access sensitive data, modify or delete data, and take full control of the application **[1]**. As they evolve, modern web applications face security risks from three main aspects: integrity, confidentiality, and availability. SQL injection attacks often occur due to the modification of SQL commands via unsafe input. This technique enables attackers to access the database, manipulate data, and cause significant damage to systems connected to it **[2]**. With database integration becoming increasingly deep in almost all modern web applications, SQL Injection remains one of the most common attack techniques. Attackers can exploit security holes in login forms or through URLs to execute invalid SQL commands. This often results in significant data corruption and user information leakage **[3]**. Web application security relies heavily on protecting data entry points, such as login forms. One effective method is to use the addslashes() function in PHP for input filtering. This technique works by adding escape characters to user input, thereby preventing the execution of malicious SQL code **[4]**.

Technological developments also offer significant opportunities to enhance web application security through penetration testing with tools such as Kali Linux. This Linux distribution provides popular tools, such as SQLMap, for detecting and exploiting SQL Injection vulnerabilities. Using a penetration testing-based approach, security vulnerabilities in web applications can be identified before attackers exploit them **[5]**. However, a report from OWASP shows that SQL Injection remains one of the 10 most critical web security risks to date. This technique enables attackers to access the database, steal sensitive data, or manipulate existing data. One solution to prevent this attack is to implement a Web Application Firewall (WAF), which detects and blocks suspicious requests before they are executed **[6]**.

In web applications, SQL injection is a major concern. Web-based systems often lack adequate input validation, creating opportunities for attacks that can compromise data integrity and availability **[7]**. Basic SQL injection techniques, such as the use of the string 'OR 1=1--', are still frequently employed by attackers to exploit vulnerabilities in systems lacking robust input validation. In addition, these exploits can be developed by manipulating server responses or forcing error pages to reveal database structures **[8]**. To detect and prevent SQL Injection vulnerabilities, the use of testing tools such as SQLMAP and OWASP ZAP has proven effective.

SQLMAP, for example, allows automated testing of various SQL injection techniques, such as boolean-based blind and error-based injection. This tool helps detect vulnerabilities before attackers exploit them, thereby improving the security of web applications **[9]**. This study focuses on testing web application security against SQL injection attacks using SQLMap. Using this tool, testing is conducted to evaluate SQL Injection vulnerabilities, as observed in attacks targeting web servers. This research is expected to contribute to the development of mitigation strategies to protect important data stored in web applications **[10]**.

2. Method

This study employs web application security testing using SQL Injection techniques, conducted through a penetration testing approach. The following is an explanation of the methods used:

1. Penetration testing

Penetration testing is conducted to evaluate the vulnerability of web applications to various attacks, such as SQL injection. This process simulates attacks from the attacker's perspective, enabling the identification of weaknesses in the web system before malicious actors exploit them. Penetration testing aims to identify vulnerabilities in web systems that attackers could exploit, such as SQL injection, which can damage databases or allow attackers to gain full control of the system.

2. Translated with DeepL.com (free version) Use of SQLMAP security tools

SQLMAP as an effective tool for testing vulnerabilities to SQL Injection attacks. With its ability to automatically and deeply exploit. SQLMap can also handle various SQL Injection techniques, including boolean-based and error-based blind and error-based injection. This tool can not only detect vulnerabilities but also exploit them, enabling a more thorough assessment of potential threats that may be faced by web database systems.

3. Input filtering method

The filtering method is used as an application-level protection against SQL Injection attacks by filtering user-provided data. One of the common techniques used in this process is 'addslashes()' in PHP. This technique works by appending escape characters to the input, thereby preventing the execution of unauthorized SQL commands. Implementing this method is essential for strengthening the security of web applications against attacks originating from

unverified input.

3. Results and Discussion

This research focuses on testing the security of web applications against SQL Injection threats through a penetration testing approach. The testing process using the SQLMAP tool aims to assess the vulnerability of web applications through an in-depth, automated approach to various injection techniques, including boolean-based blind and error-based injection.

1. Perform SQL Injection using the command in the command line "python sqlmap.py -u http://testphp.vulnweb.com/product.php?Pic=1". If the test is successful, it will display the contents of the database; here, the database contains 2 databases, namely Acuart and Information_schema, Information_schema in **Figure 1.**

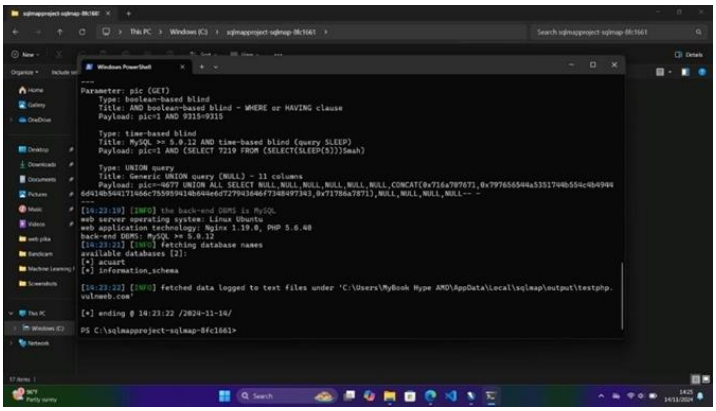

Figure 1. System Testing 1

2. Try to access the contents of the Information_schema database using the command "python sqlmap.py -u http://testphp.vulnweb.com/product.php?pic=1 -D database_name". The results show that the database contains a table with important data about the website. A complete explanation is provided in **Figure 2.**

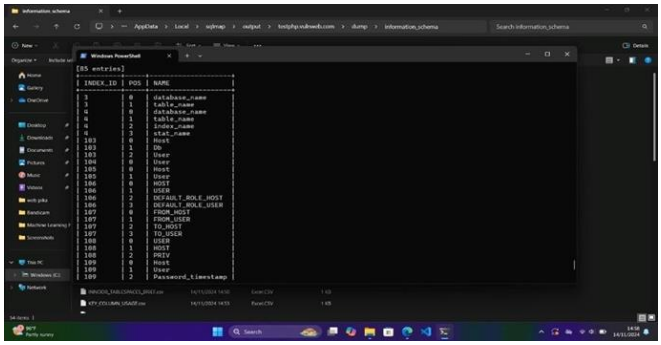

Figure 2. System Testing 2

.

3. Bagian 3. To access the contents of the product table from the Acuart database using the command "python sqlmap.py -u http://testphp.vulnweb.com/product.php?pic=1 -D database_name -T products –dump". By using the command above, the contents of the products table will appear below. Complete explanation See Figure 3.

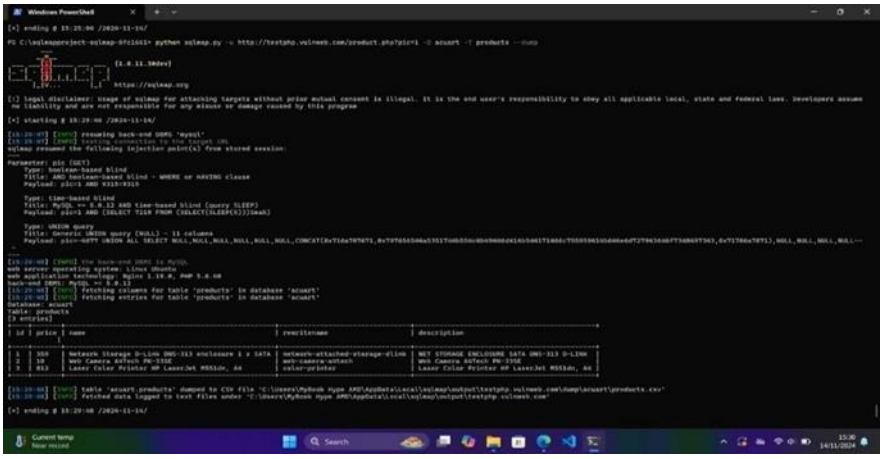

Figure 3. System Testing 3

4. Conclusion

Web application security is critical, particularly against SQL injection attacks, which can compromise data integrity, confidentiality, and availability. This study shows that SQL Injection threats remain among the most critical web security risks. Testing with SQLMap-style tools has proven effective for detecting and exploiting various vulnerabilities, including boolean-based blind and error-based injection. In addition, the addslashes()-based input filtering method in PHP has proven effective in preventing the execution of malicious SQL commands, thereby increasing the system from possible attacks. The results of this study provide important insights in efforts to improve web application security, especially in industries that store sensitive data. By combining penetration testing with mitigation techniques such as input filtering, web applications can be significantly protected against SQL injection attacks. Hopefully this strategy can be an effective preventive step to maintain the security of the entire data system and web applications.

References

[1]. H. F. R. U. Naomi Augusta, "Website Security System with Multi Methods to Prevent SQL Injection," National Seminar on Corisindo, 2024.
[2]. M. D. F. S. I. G. Tino Imam Maulana Pratama, "Analysis of Attacks and Security on SQL Injection: A Study," JIIFKOM (Journal of Informatics & Computer Science), STTR Cepu, 2022.
[3]. I. A. K. A. S. F. S. Luthfi Arian Nugraha, "SQL Injection: Effectiveness Analysis of Penetration Testing in Web Applications," SMATIKA: STIKI Informatics Journal, 2024.

[4]. R. N. I. K. Yovie Ferdianto, "Application of Admin Login Security and Input Filtering to Prevent SQL Injection," Jurnal Informatika Rekayasa Perangkat Lunak (JATIKA), 2023.
[5]. R. F. E. M. S. S. Yehezkiel Natanael, "Information Security Analysis for Website Users Using Kali Linux Through SQL Injection Techniques," Journal of Informatics Engineering (TEKINFO), 2024.
[6]. W. A. P. R. A. Bangkit Wiguna, "Implementation of Web Application Firewall to Prevent SQL Injection Attacks on Websites," Journal of Information and Communication Technology, 2020.
[7]. H. S. L. A. Ade Bastian, "Analysis of Data Education Application (DAPODIK) Security Using Penetration Testing and SQL Injection," Infotect Journal, 2020.
[8]. H. H. K. A. I. M. Andika Saputra, "Addressing Security in SQL Injection and How to Prevent It," Proceedings of the Annual Research Seminar 2017 in Computer Science and ICT, 2017.
[9]. D. F. R. Rakhmadi Rahman, "Network Penetration Testing Using OWASP ZAP and SQLMAP to Identify Website Security Vulnerabilities," Journal of Information Systems Research, 2024.
[10].B. M. R. D. R. H. R. D. A. Y. A. S. Ade Riyanti, "SQL Injection Penetration Testing on Website Database Security Vulnerabilities Using SQLmap," Journal of Internet and Software Engineering, 2024.
[11].K. P, Industrial Ecology, Yogyakarta: Andi Offset, 2013.
[12].O'Brien and Marakas, Management Information System, 16th ed., New York: McGraw Hill, 2013.
[13].M. Muslihudin and Oktavianto, Analysis and Design of Information Systems Using Structured Model and UML, Yogyakarta: Andi Offset, 2016.
[14].R. A. Sukamto and M. Shalahuddin, Software Engineering: Structured and Object-Oriented, Bandung: Informatika, 2015.
[15].B. Hartono, Computer-Based Management Information Systems, Jakarta: Rineka Cipta, 2013.
[16].I. Mulyani, E. Satria, and A. D. Supriatna, "Development of Short Message Service (SMS) Gateway for Academic Information Services," Journal of Algorithms, vol. 9, no. 2, pp. 389-397, May 2013.

.