ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0001

Tracking and Analyzing Web Penetration Attacks through Splunk-Driven Log Analysis

Guangqun Zhou[1], Xiangbi Gong[2,*], Dongming Zhong[3]

[1] China United Network Communications Group Co., LTD. Ningbo branch, Ningbo 315000, China

[2] Cangnan County Big Data Management Center, Wenzhou 325000, China

[3] Ningbo branch of China Telecom Co., LTD, Ningbo 315000, China

*13267395@qq.com

Abstract

With the popularity of Web applications, Web penetration attacks have become one of the most common types of cyber attacks. This study proposes a method for tracking and analyzing Web penetration attacks based on Splunk log analysis, aiming to identify and track potential network attack behaviors through in-depth analysis of network logs. First, we collected network log data from various sources and used Splunk as the main tool for data preprocessing and analysis. By deeply mining the log data, we were able to identify abnormal patterns and potential attack signs.

Keywords

Web Penetration; Log Analysis; Splunk; Cyber Attacks; Tracking and Tracing.

1. Introduction

Web business platforms are highly susceptible to attacks due to their characteristics, making them a preferred target for attackers. The attack methods faced by Web servers are diverse, with attacks such as drive-by downloading, SQL injection, buffer overflow, sniffing, and exploiting vulnerabilities in Web server software like IIS being particularly common.

On one hand, due to the lack of security considerations in the design of the TCP/IP protocol, the data transmitted over the network lacks necessary security protection. Attackers can exploit system vulnerabilities to launch buffer overflow attacks, thereby obtaining or elevating their user privileges on the vulnerable system, running arbitrary programs, or even installing and executing malicious code to steal sensitive data. At the same time, the development of application software often overlooks security issues, resulting in numerous vulnerabilities in the program, such as buffer overflow and SQL injection. These application-layer attacks often stem from the neglect of security during the software development process.

On the other hand, the curiosity of users about certain hidden things also provides opportunities for attackers. Attackers usually use Trojan horses or virus programs to carry out attacks, hiding these malicious programs in tempting pictures, audio and video files, and free software, and then placing them on websites or sending them to users through email attachments, QQ, MSN, and other instant messaging software. By exploiting the curiosity of users, they induce users to click or download and run these files to achieve the purpose of the attack.

Web penetration attacks refer to the act of exploiting vulnerabilities or security flaws in Web applications, bypassing security mechanisms through malicious requests or scripts, and obtaining unauthorized access rights or performing malicious operations. This type of attack is characterized

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0001

by its concealment, efficiency, and widespread nature, posing great security risks to businesses and individuals.

Log analysis, as an important means of tracking and analyzing cyber attacks, plays a crucial role in improving the security of Web applications. Splunk, as a powerful log analysis tool, is increasingly widely used in the field of security.

This study aims to explore how to use Splunk for tracking and analyzing Web penetration attacks, in order to provide new perspectives and methods for network security protection. The main contributions of the study include: building a Web penetration test target machine based on DVWA; constructing a Web penetration attack tracking framework based on Splunk; identifying common Web penetration attacks through Splunk log analysis.

Through in-depth research on Splunk log analysis technology, this paper will provide theoretical support and practical guidance for the field of network security, and has important academic value and application prospects.

2. Overview of Web Security

2.1 Definition and Development of Web Security

Web security refers to the ability to protect Web applications and their users from malicious attacks and unauthorized access. With the popularization of the Internet, the definition of Web security has expanded from the initial prevention of data leakage to protecting user privacy, ensuring data integrity, and availability.

The development of Web security is closely related to the advancement of cyber attack techniques. From the initial simple script attacks to complex cross-site scripting attacks (XSS), SQL injection, etc., the attack methods continue to evolve, promoting the development of Web security technology.

Modern Web applications are becoming more and more complex, involving more and more technology stacks and data interactions, which provides more attack surfaces for attackers. At the same time, with the popularization of mobile devices and the Internet of Things devices, the boundaries of Web security are also continuously expanding.

2.2 Security Threats Faced by Web Applications

The types of security threats faced by Web applications are diverse, including but not limited to the following:

- SQL Injection: Attackers input malicious SQL code in Web forms in an attempt to affect the operation of the backend database, obtaining, tampering with, or deleting data.
- Cross-Site Scripting Attack (XSS): Attackers inject malicious scripts into Web pages, and when other users browse the page, the script is executed in the user's browser, which may lead to the leakage of user information.
- Distributed Denial of Service (DDoS) Attack: Attack the target server with a large amount of traffic, making it unable to process requests from legitimate users.
- Session Hijacking: Attackers impersonate user identities by stealing or predicting user session identifiers.

In response to these threats, Web security protection measures are also continuously being updated, including input validation, output encoding, using HTTPS, session management, etc. However, with the continuous evolution of attack methods, new security threats are also emerging, requiring Web security protection measures to be continuously updated and improved.

3. Basics of Web Penetration Testing

3.1 Principles of Penetration Testing

Penetration testing is a security assessment method that simulates the behavior of malicious attackers, aiming to identify and exploit security vulnerabilities in Web applications. This process includes

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0001

multiple stages, such as information collection, vulnerability analysis, exploitation attempts, and post-exploitation.

(1)Information Collection: The first step in penetration testing is to collect as much information as possible about the target system, including domain name information, IP addresses, operating systems, applications used and their versions, etc.

(2)Vulnerability Analysis: After collecting sufficient information, testers will analyze this information to identify potential security vulnerabilities. This may include using automated tools to scan for known vulnerabilities and manual testing to discover unknown vulnerabilities.

(3)Exploitation Attempts: Once potential vulnerabilities are identified, testers will attempt to exploit these vulnerabilities to verify their effectiveness and assess their impact.

(4)Post-Exploitation: If a vulnerability is successfully exploited, testers will conduct post-exploitation activities, such as privilege escalation, access to sensitive data, or control of the system..

3.2 Penetration Testing Process

The penetration testing process is key to ensuring the systematic and effectiveness of the test. A standard penetration testing process usually includes the following steps:

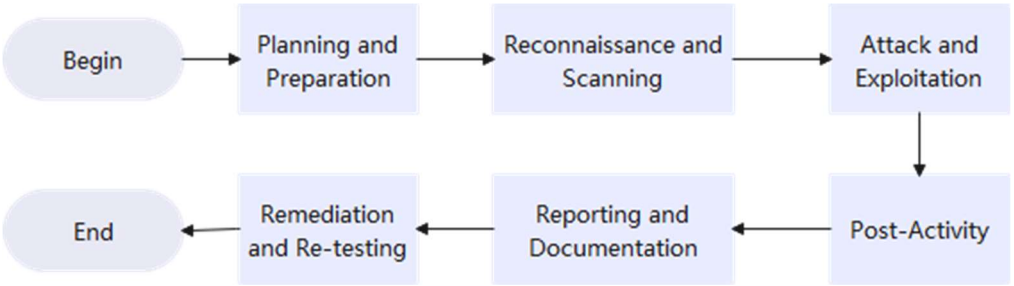


Figure 1. Penetration Testing Process Diagram

(1)Planning and Preparation: Define the scope, objectives, rules, and limitations of the test, and prepare the necessary tools and resources.

(2)Reconnaissance and Scanning: Collect target information and use automated tools for preliminary vulnerability scanning.

(3)Attack and Exploitation: Based on the results of reconnaissance and scanning, attempt to exploit the discovered vulnerabilities.

(4)Post-Activity: After successfully exploiting the vulnerability, conduct in-depth internal system activities, such as data access and privilege escalation.

(5)Reporting and Documentation: Record all findings during the testing process, including vulnerability details, exploitation methods, and remediation suggestions.

(6)Remediation and Re-testing: Provide the test results and suggestions to system administrators or development teams for them to fix the vulnerabilities, and conduct re-testing after the fixes to verify whether the vulnerabilities have been successfully patched.

Penetration testing is not only a process of discovering and fixing security vulnerabilities but also an important means of improving the security and defense capabilities of Web applications. By simulating the behavior of attackers, organizations can better understand the security status of their systems and take appropriate measures to strengthen security protection.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0001

4. Splunk Log Analysis Technology

4.1 Introduction to Splunk

Splunk is an advanced log analysis tool that can collect, index, and query machine-generated data. In the field of network security, the application of Splunk is particularly important, as it can help security analysts quickly identify and respond to security events.

Splunk has functions such as real-time monitoring, historical data analysis, visualization display, and automated reporting, making it a powerful tool for tracking and analyzing network attacks. In addition to network security, Splunk is also widely used in IT operations, application performance monitoring, business analysis, and other fields. Its flexibility and scalability make it an important support for data-driven decision-making.

4.2 Log Analysis Methods

Log analysis is a key link in network security protection. By analyzing log data, it is possible to timely detect and respond to security threats.

(1)Data Collection: First, log data needs to be collected from systems such as Web servers, databases, and firewalls. Splunk supports multiple data sources and can handle structured, semi-structured, and unstructured data.

(2)Data Indexing: The collected log data will be indexed for easy and fast retrieval. Splunk's indexing mechanism optimizes data storage and query efficiency.

Pattern Recognition: By building search queries and using Splunk's machine learning capabilities, abnormal behavior patterns can be identified, such as frequent login failures, abnormal data access, etc.

(3)Attack Tracking: Using Splunk's correlation analysis capabilities, the attacker's path of action can be tracked, from the initial entry point to the later exploitation behavior, to build a complete attack profile.

(4)Visualization Display: Splunk provides a wealth of visualization tools, such as charts, maps, timelines, etc., to help analysts more intuitively understand log data and attack patterns.

(5)Automated Response: By configuring automated scripts and workflows, Splunk can automatically trigger response measures when specific events are detected, such as sending alerts, isolating infected systems, etc.

Through the above log analysis methods, Splunk can help organizations build a complete network security monitoring and response system, effectively enhancing the security of Web applications.

5. Case Analysis of Web Penetration Attacks

5.1 Experimental Environment

To verify the effectiveness of the Web penetration attack tracking and analysis method based on Splunk log analysis, this study constructed a simulated experimental environment, which includes a Web server, database server, and attack simulation system.

The experimental environment simulates a typical Web application deployment architecture, including the Apache Web server, MySQL database server, and PHP applications, and uses the open-source DVWA (Damn Vulnerable Web Application) system to build a Web penetration test target machine.

DVWA is a PHP/MySQL Web application for security vulnerability identification, designed to provide a legal environment for security professionals to test their professional skills and tools, helping Web developers better understand the process of Web application security protection.

DVWA includes ten attack modules: Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), XSS (Reflected), and XSS

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0001

(Stored). It covers all the attack vulnerabilities in the OWASP TOP10 practice environment, providing a one-stop solution for all Web penetration learning environments.

Additionally, DVWA allows manual adjustment of the security level of the target machine source code, which are Low, Medium, High, and Impossible. The higher the level, the stricter the security protection and the greater the difficulty of penetration. Generally, the Low level has almost no protection or only the simplest protection, which can be easily penetrated; Medium level uses some very rough protection measures, requiring users to know how to bypass protection measures; High level greatly increases the level of protection, generally requiring very rich experience to successfully penetrate; Finally, Impossible is basically impossible to penetrate, so the source code of Impossible can generally be used as the best means of Web protection in the production environment.

5.2 SQL Injection Attack Analysis

SQL injection attack is a common Web security threat, where attackers insert malicious SQL code into input fields in an attempt to bypass the normal logic of the application and perform illegal operations on the database. Using Splunk for tracking and analysis of SQL injection attacks can take the following steps:

(1) Execute a Simulated Attack

Access the vulnerable Web page, open the SQL Injection in the target field, access the Web page and attempt an attack in the User ID input box. As shown in the figure 2, attempt to inject to determine if there is an injection point.



Figure 2. SQL Injection Detection

As shown in the figure 3, an error is reported when inputting 1', indicating that there is an injection point in this database.

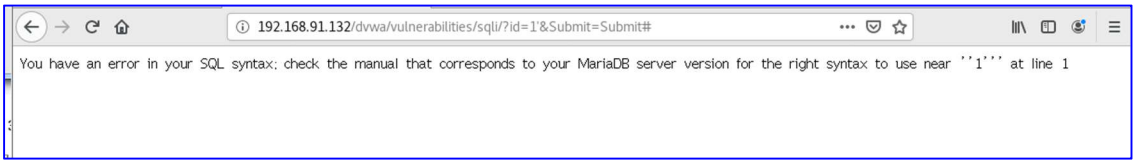


Figure 3. Error Notification

Through step-by-step testing, you can further obtain the type of injection, the number of columns in the database, and then use UNION queries to detect the database name, table name, column name, and content in the database.

Obtain the database name by input “1' union select 1,database()#”, which is dvwa.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0001

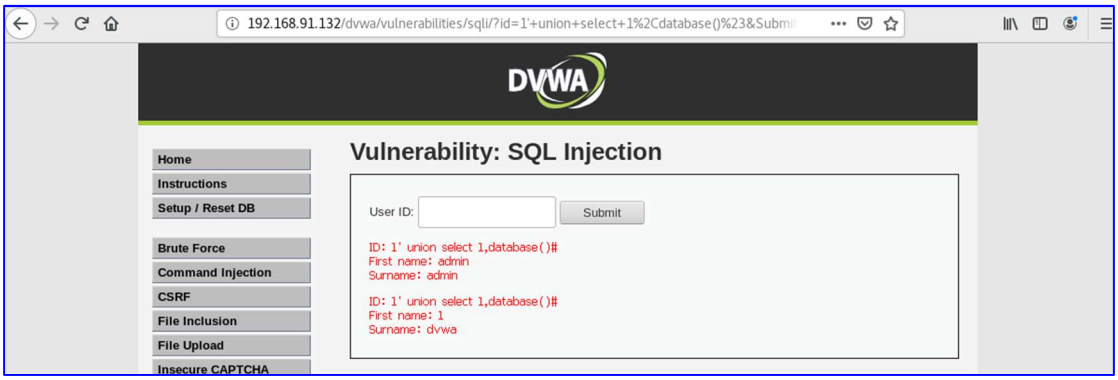


Figure 4. Detecting Database Name

Obtaining Table Names by input "1'union select 1, group_concat(table_name) from information_schema.tables where table_schema='dvwa'#".

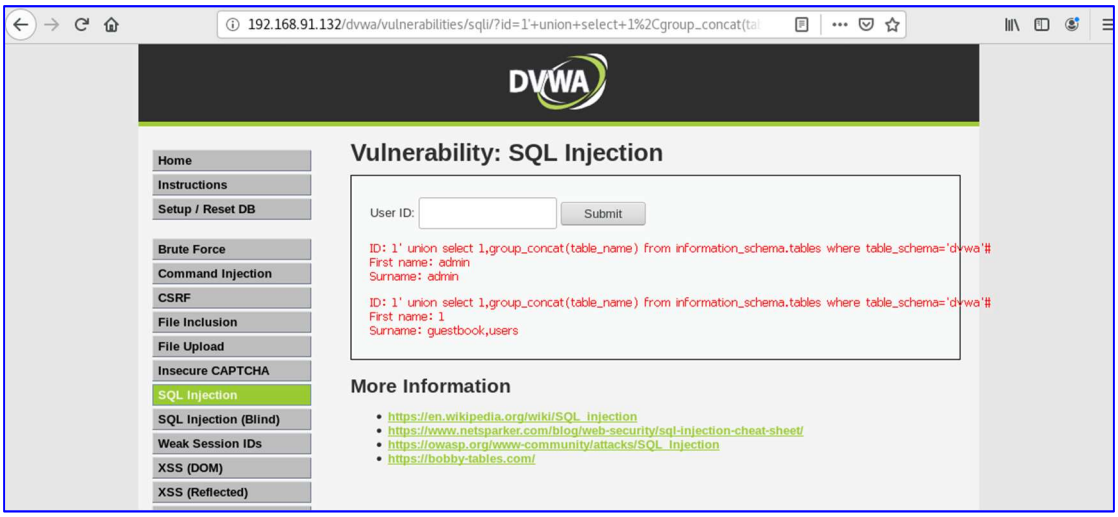


Figure 5. Obtaining Table Names guestbook and users through Injection

Getting Columns in the Database by input "1'union select 1, group_concat(column_name) from information_schema.columns where table_name='users'#".

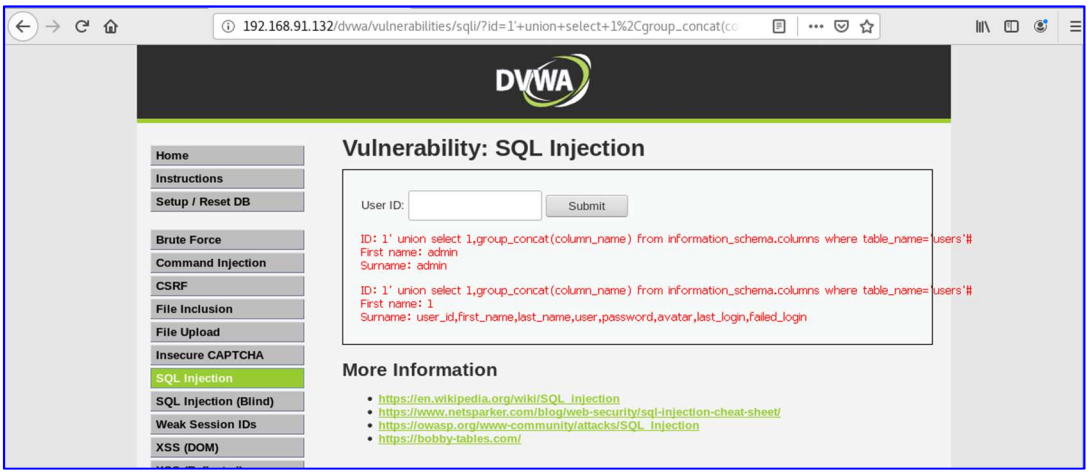


Figure 6. Getting Columns in the Database

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0001

Getting the Content of the user Column and password Column by input “1' union select group_concat(user), group_concat(password) from users#”.



Figure 7. Getting the Content of the user Column and password Column

(2) Analyze and track through Splunk

The above SQL injection attack access records are all saved in the access log file access.log of the apache server. By importing this log file into the Splunk platform, you can analyze and track it, and also set up real-time monitoring for this log file.

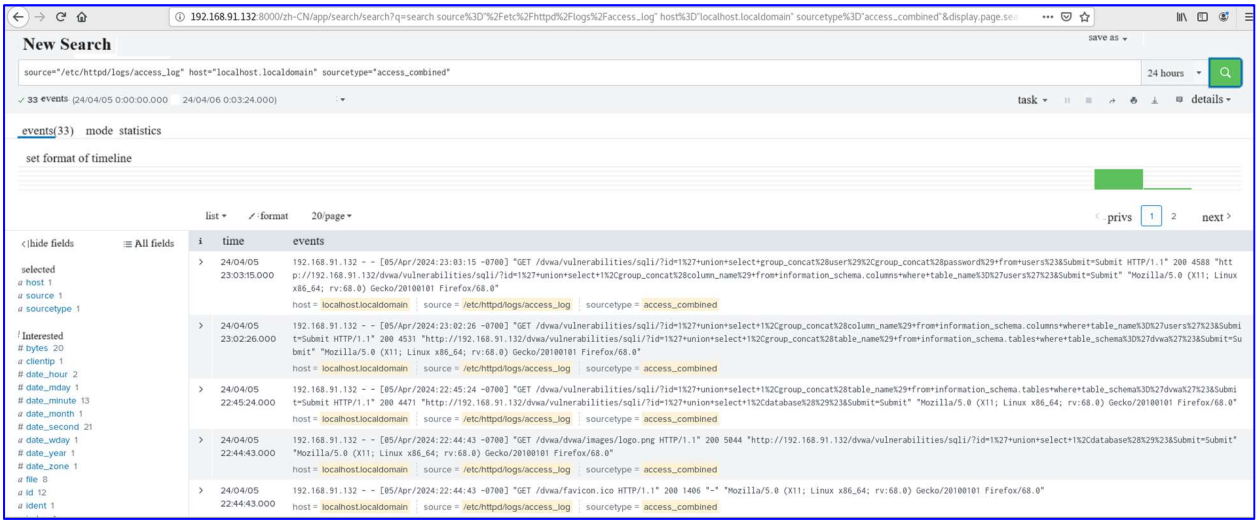


Figure 8. Splunk Analysis of Log Information

Take the last log as an example for analysis:

192.168.91.132--[05/Apr/2024:23:03:15-0700] "GET/dvwa/vulnerabilities/sqli/?id=1%27+union+select+group_concat%28user%29%2Cgroup_concat%28password%29+from+users%23&Submit=Submit HTTP/1.1" 200 4588 "http://192.168.91.132/dvwa/vulnerabilities/sqli/?id=1%27+union+select+1%2Cgroup_concat%28column_name%29+from+information_schema.columns+where+table_name%3D%27users%27%23&Submit=Submit" "Mozilla/5.0(X11;Linux x86_64;rv:68.0)Gecko/20100101 Firefox/68.0"

It can be seen that this is an access log from the local host IP address 192.168.91.132. It requests the resource path through the "GET" method as follows:

"/dvwa/vulnerabilities/sqli/?id=1%27+union+select+group_concat%28user%29%2Cgroup_concat%28password%29+from+users%23&Submit=Submit"

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0001

According to the parameters and syntax in the request path, this request seems to be an attempt to exploit the SQL injection vulnerability. The attacker tried to construct a malicious SQL statement to obtain the username and password of the user table in the database.

SQL is transmitted in plain text in the HTTP protocol, so the values it transmits can be clearly seen. This log shows what statement the attacker used to launch the SQL injection on the web page.

Through Splunk's statistics and chart functions, you can also clearly see the time and number of web attacks in each time period.

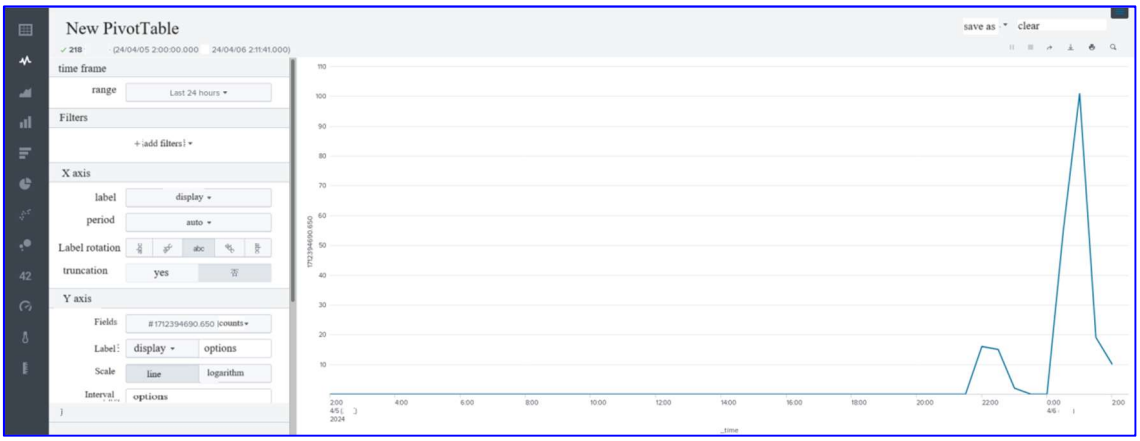


Figure 9. Splunk Data Statistics Chart

SQL injection is a common security vulnerability that can be prevented by adding correct input validation and parameterized queries to the web page code, or by deploying a Web Application Firewall (WAF) device on the front end of the Web server for detection and protection.

Other attacks on Web sites, such as brute force cracking, file uploading, cross-site scripting attacks, etc., can all be tracked and analyzed through Splunk log analysis to discover their attack patterns and take corresponding measures for prevention.

6. Conclusion

During the experiment, all server and application log data were collected in real-time and transmitted to the Splunk platform. The types of logs collected include access logs, error logs, database query logs, etc. By deeply analyzing the collected log data, more than 95% of SQL injection, XSS, DDoS, and other attack behaviors were successfully identified, proving the high efficiency of the log analysis method in attack identification. Through correlation analysis and visualization tools, the experiment was able to track the attacker's path and build a complete attack profile. For example, in an SQL injection attack, from the initial abnormal query request to the final data leakage behavior, the entire attack process was clearly presented.

In summary, the log analysis technology based on Splunk can effectively track and analyze Web penetration attacks, providing strong technical support for network security protection. By further optimizing the analysis model and response mechanism, the accuracy and response speed of tracking and analysis can be further improved.

Acknowledgments

This research was supported by Wenzhou Network Security Detection and Protection Engineering Technology Research Center.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202408_10(8).0001

References

[1] Mo Huaihai, Li Xiaodong. Web Penetration Testing Information Collection. Technical Research. World of Communications, 2019, 26(03): 33-34.

[2] Zhang Bingshuai. In-Depth Analysis of Web Security. Beijing: Electronic Industry Press, 2015: 106-128.

[3] Shi Liuyang, Fang Yong. Research on Webshell Detection Scheme Based on Web Logs. Information Security Research, 2015, (1): 66-73.

[4] Mo Huaihai, Li Xiaodong. Research on Web Penetration Testing Information Collection Technology. World of Communications, 2019, 26(03): 33-34.

[5] Wu Hanqing. White Hat Talks Web Security (Second Edition). Electronic Industry Press, August 2023.

[6] Splunk Official Documentation: https://docs.splunk.com/Documentation.