YAYASAN AKRAB PEKANBARU
Akrab Juara : Jurnal Ilmu-ilmu Sosial
Volume 10 Nomor 4 Edisi November 2025 (1926-1941)

70

IMPLEMENTATION OF THE SEMONT APPLICATION AS A SIGNATURE-BASED INTRUSION DETECTION AND PREVENTION SYSTEM ON THE SMAN 1 RANCAEKEK COMPUTER NETWORK

Alwi Al Hadad, Hani Harafani
Program Studi Informatika Fakultas Teknologi Informasi Universitas Nusa Mandiri
(Naskah diterima: 1 October 2025, disetujui: 28 October 2025)

Abstract

SMAN 1 Rancaekek is highly dependent on web applications that are vulnerable to injection attacks (SQL Injection, XSS, RCE, LFI), web defacement, and brute force, exacerbated by the absence of real-time monitoring, which has led to serious security incidents. This study aims to design and implement the Semont (Sentinel Monitoring) application as a signature-based Intrusion Detection and Prevention System (IDPS) to detect and prevent such cyber attacks, monitor network traffic on Port 80 (HTTP) and Port 443 (HTTPS), and generate comprehensive reports. The research method involved network system analysis and Semont design. The main contribution of this research is the development of a lightweight, efficient, and intuitive IDPS solution capable of protecting sensitive data and website visual integrity with minimal overhead. The analysis results show that Semont successfully detected and blocked 100% of simulated attacks, significantly changing the security posture of the SMAN 1 Rancaekek website from vulnerable to secure, supported by detailed logging and real-time notifications to Telegram. In conclusion, Semont proved to be highly effective in detecting and preventing common cyber attacks, meeting the need for proactive defense in educational environments, although the signature-based method is limited to zero-day attacks, which can be improved through the integration of anomaly detection in the future.
Keywords: *Semont; IDPS; Signature; Siber; Injection*

Abstrak

SMAN 1 Rancaekek sangat bergantung pada aplikasi web yang rentan terhadap injection attacks (SQL Injection, XSS, RCE, LFI), web defacement, dan brute force, diperparah ketiadaan pemantauan real-time yang menyebabkan insiden keamanan serius. Penelitian ini bertujuan merancang dan mengimplementasikan aplikasi Semont (Sentinel Monitoring) sebagai Sistem Pendeteksi dan Pencegah Intrusi (IDPS) berbasis signature-based untuk mendeteksi dan mencegah serangan siber tersebut, memantau lalu lintas jaringan pada Port 80 (HTTP) dan Port 443 (HTTPS), serta menghasilkan laporan komprehensif. Metode penelitian melibatkan analisis sistem jaringan, perancangan Semont. Kontribusi utama penelitian ini adalah pengembangan solusi IDPS yang ringan, efisien, dan intuitif, mampu melindungi data sensitif serta integritas visual situs web dengan overhead minimal. Hasil analisis menunjukkan Semont berhasil mendeteksi dan memblokir 100% simulasi serangan, secara signifikan mengubah postur keamanan situs web SMAN 1 Rancaekek dari rentan menjadi aman, didukung logging detail dan notifikasi real-time ke Telegram. Sebagai kesimpulan, Semont terbukti sangat efektif mendeteksi dan mencegah serangan siber umum, memenuhi kebutuhan pertahanan proaktif di lingkungan pendidikan, meskipun metode

signature-based terbatas pada serangan zero-day yang dapat ditingkatkan melalui integrasi deteksi anomali di masa mendatang.
Kata Kunci: Semont; IDPS; Signature; Siber; Injection

I. INTRODUCTION

SMAN 1 Rancaekek, as a secondary educational institution, relies heavily on network infrastructure and web application services to support all teaching and learning processes and administrative activities. The research focused on the SMAN 1 Rancaekek web application service, with the domain https://smanrancaekek.sch.id, which serves as the primary access point for all school members to obtain information and manage data, such as information related to academic activities, personnel data, and administrative archives. The high frequency of access to this web service, coupled with the sensitive nature of the stored data, inherently makes this site a potential target for cyberattacks.

However, this rapid growth in reliance on information technology and connectivity is directly proportional to the increasing complexity and frequency of cybersecurity threats (Habibah, 2024). A computer network attack is defined as an attempt to gain unauthorized access to a network, with the aim of stealing data or carrying out other destructive actions. Intrusion, as a specific form of attack, is an unauthorized and unlawful attempt to access, modify, or control an information system/network, potentially rendering it unreliable or inoperable (Widodo & Aji, 2022). This fact is reinforced by the experience of SMAN 1 Rancaekek, which previously experienced a cybersecurity incident, in which an outsider successfully exploited a system weakness to access data and threatened to release it for personal gain if not paid a monetary reward (Widyarto & Hapsari, 2022). This modus operandi is known as the actions of a gray hat attacker, an individual who possesses the technical skills of a hacker but lacks the official permission or mandate to infiltrate a system.

This incident clearly demonstrates that system security cannot rely solely on trust or informal approaches but must be supported by a robust, multi-layered defense system to maintain data integrity and confidentiality. Among the various types of attacks, injection attacks are one of the most common and dangerous methods (Irawan et al., 2018). These attacks occur when hackers insert malicious commands or code into applications or programs that are then processed by the system. The impact of this attack can be very detrimental, including data theft, service disruption, data integrity damage, or even successfully bypassing system authentication mechanisms. From a technical perspective, the SMAN 1 Rancaekek

web application was identified as having vulnerabilities, particularly against SQL Injection. This type of web defacement infiltration generally occurs through security holes in the system, such as vulnerabilities in the Content Management System (CMS), plugins, or lack of software updates. The impact of this intrusion is very substantial, not only tarnishing the reputation and credibility of government agencies in the public eye, but also endangering the security of site visitors' data, which could be exposed to malicious content or other risks.

Another crucial issue is the lack of an active, real-time monitoring system to monitor activity on the web servers at SMAN 1 Rancaekek. This significantly hampers early detection of suspicious activity or indications of attacks. Therefore, security incidents are often only realized after the damage has already occurred and spread widely. Furthermore, according to the National Cyber and Crypto Agency (BKN) in its 2024 Cyber Security Landscape, educational institutions are often targeted by cyberattacks due to a lack of understanding of the importance of network and data security (Id-SIRTII/CC, 2023). This is further exacerbated by limited human resources (HR) within the school environment, who lack a full understanding of secure coding concepts for developing attack-resilient applications and are unfamiliar with the use of Intrusion Detection Systems (IDS) or Security Information and Event Management (SIEM) systems like Wazuh, which may be too complex for the school's IT team to implement and manage independently.

Given these conditions, there is an urgent need for a security system that not only continuously monitors traffic between the web and the server but also has the capability to automatically detect and mitigate potential attacks. An Intrusion Detection and Prevention System (IDPS) is a relevant solution designed to monitor and analyze networks or systems to identify vulnerabilities, report malicious activity, and implement preventative measures to address the development of computer-related threats (Elan Maulani & Faisal Umam, 2023). This study proposes the development and implementation of an application called Semont (Sentinel Monitoring) as a signature-based IDPS system. This signature-based method works with a pattern matching technique, which matches the pattern of data packets sent from users to servers with attack patterns registered in the database. This method is known to be efficient in detecting documented threats and has been proven to reduce the rate of detection errors (false positives and false negatives). However, its fundamental weakness lies in its inability to detect zero-day attacks or new attack variants that do not yet have signatures in the

database. Nevertheless, this approach was chosen based on the efficiency and accuracy of detection for common and well-known attacks, as well as considerations of ease of implementation and management in the SMAN 1 Rancaekek environment.

The Semont application is designed to detect and prevent various categories of vulnerability attacks such as injection (including XSS, SQL Injection, Remote Code Execution (RCE), Local File Inclusion (LFI), and the like), as well as brute force attacks. Furthermore, Semont will also have the capability to detect defacement of web directories and be able to immediately delete defaced pages. This system will focus on monitoring traffic between the web and the server that passes through Port 80 (HTTP) and Port 443 (HTTPS), which are the main access routes to school web services, and will generate comprehensive reports from the generated logs, making it easier for the school IT team to monitor and analyze incidents. The main goal of implementing this system is not only to detect attacks in real-time and prevent data compromise, but also to strengthen the school's overall cybersecurity infrastructure, even with the possibility of vulnerable web application code. The selection of Semont was also based on the consideration that its intuitive and easy-to-understand user interface (UI) will make it easier for the school IT team to operate and manage the system without requiring in-depth expertise in complex cybersecurity fields. With the Semont application, it is hoped that the security of school data and digital infrastructure can be significantly improved. This application can serve as a model for implementing an effective security system and raise awareness of the importance of cybersecurity in other educational settings.

The goal of this research is to implement the Semont application, which can be used to detect attacks in real time. It can also be used as a monitoring system with alarms or rapid notifications, as well as to strengthen the integrity and availability of the website services at SMAN 1 Rancaekek.

II. THEORETICAL STUDIES

Website injection attacks are one of the most common attack methods used by hackers. In these attacks, hackers insert malicious commands or code into applications or programs, which are then processed by the system. This process can have detrimental effects, such as data theft, service disruption, data integrity breaches, or even bypassing system authentication mechanisms. Hackers exploit weaknesses in user input validation processes,

which typically occur in input elements such as forms or URLs (Kurniawan et al., 2023). Injection vulnerabilities can often be exploited by attackers to execute malicious commands in web applications. While these attacks are easier to detect than more complex vulnerabilities, they are still dangerous. One effective preventive measure against these vulnerabilities is the use of a Web Application Firewall (WAF), which can help detect and block attacks at the application layer by analyzing suspicious patterns that could potentially harm the system (Hadad, 2025).

SQL Injection is an attack technique that exploits weaknesses in database systems connected to websites, allowing attackers to damage, modify, or even steal existing data. This attack can cause damage, loss, or theft of data stored on a website. This attack technique is typically carried out through the website URL, by adding specific parameters or using poorly protected CRUD (Create, Read, Update, Delete) operations. Open-source tools are often used by attackers to carry out this attack (Nugraha et al., 2024).

Cross-Site Scripting (XSS) is a type of client-side code injection attack, where the attacker inserts malicious scripts into legitimate web pages or web applications. The malicious scripts are then executed in the victim's browser when they visit the page. This attack can be carried out using various client-side programming languages, such as JavaScript, VBScript, ActiveX, Flash, and others. Although XSS is a dangerous cyberattack, many victims are unaware that they are being attacked (Mahdi Maulana Lubis et al., 2022). Cross-Site Scripting (XSS) attacks can be classified based on how they work and the location of the vulnerability. The generally accepted taxonomy divides XSS attacks into two main categories: reflected (non-persistent) and stored (persistent), and whether the attack occurs on the server or client side. Reflected XSS occurs when malicious script inserted by an attacker is directly inserted into the server response without first storing it. In this type of attack, data included in URL parameters is returned in the response displayed to the user. Reflected XSS typically only impacts one user at a time and is considered easier to detect and mitigate. Meanwhile, Stored XSS is more dangerous because the malicious data entered by the attacker is stored on the server, such as in a database. This malicious input is then re-inserted into the web page displayed to many users without first being filtered. Stored XSS can persist for a long time and impact many users, making it more difficult to detect and mitigate. Furthermore, there is client-side XSS, where the vulnerability lies in the code on the client

side, not the server. This type includes various exploits that occur in scripts or JavaScript code executed in the user's browser. Non-persistent client-side XSS, for example, can occur when an attacker-controllable resource, such as a parameter in a URL, is used in a malicious function, such as eval or document.write. Persistent client-side XSS, on the other hand, can occur when malicious data is stored in local storage such as cookies or localStorage, allowing the malicious script to remain active for a longer period of time (Somé, n.d.).

A payload is shell code executed on a target system after an exploit successfully accesses or compromises it. Two commonly used payload types are bind shell payloads and reverse shell payloads, each of which defines how a connection is established to the executed shell (Journal et al., 2021).

Log files are collections of data that record activities and events occurring on a system, application, network device, or other IT infrastructure. These log files contain critical information, such as application or system issues, warnings, and messages describing status and events that could impact the security and integrity of the system (Sándor R. Répás, 2024).

A Web Application Firewall (WAF) is a specialized firewall designed to filter and control HTTP traffic between web clients and application servers. WAFs are a critical component in maintaining robust application security. WAFs provide security at Layer 7 (the application layer), typically located between the primary firewall and the web server or application server. Unlike port-based network firewalls, WAFs take this a step further by providing additional protection for applications served over the internet (George & George, 2021).

An Intrusion Detection and Prevention System (IDPS) is a device or software application designed to monitor and analyze a network or system. This system functions to identify vulnerabilities, report malicious activity, and implement preventative measures to address the development of computer-related threats using a variety of effective response techniques (Azeez et al., 2020).

Signature-based is a technique for matching data packets sent from users to servers. If the packet matches a pattern registered in a database, it is identified as an attack (Saputra et al., 2022).

Pattern matching is a method used to detect or identify data packets that deviate from expected patterns in a sequence of actions (Al Rubaiei et al., 2020). This algorithm works based on input data evaluated by a centralized controller, referring to previous attack history and attack feature data. Simply put, this algorithm works by matching known attack patterns (through previously recorded data) to detect intrusions. The system examines incoming data against previous attacks and the characteristics of previously identified attacks (Remesh Babu et al., 2023).

III. RESEARCH METHODS

This research consisted of five main phases: Problem Identification, Semont Application System Design, Initial Testing, Semont Application Implementation, and System Testing and Evaluation.

The initial phase, problem identification, focused on understanding the context of the problem and determining the requirements of the system to be developed. During this phase, we conducted an in-depth analysis of cybersecurity issues at SMAN 1 Rancaekek, including recording past incidents, such as data extortion attempts by external parties (gray hat), and vulnerabilities related to injection attacks such as SQL Injection, XSS, RCE, and LFI. Next, we conducted penetration testing as bug hunters and discovered an SQL Injection vulnerability on the SMAN 1 Rancaekek website, which exposed sensitive data. We then determined the functional and non-functional specifications of the Semont application. We initially used Go Access running on the WSL (Windows Subsystem for Linux) environment to obtain log files and other critical information related to user activity on the site.

In the Semont application design phase, the initial step is to design the Semont application architecture which includes a network traffic monitoring module to monitor traffic passing through port 80 (HTTP) and port 443 (HTTPS), then designing a database used to store attack patterns (signatures) which are then used as a signature-based detection engine which can identify attack patterns such as XSS, SQL injection, RCE, LFI, brute force, and to scan for vulnerabilities (vulnerability scanning) using automated tools such as nuclei. Next, designing a prevention module so that it can prevent attacks such as deface attacks by automatically deleting defaced pages, and designing a log reporting module, then formulating a pattern matching algorithm to match packet patterns from users to servers with signatures in the database, and determining automatic responses When there is a matching signature,

integrating between the modules that have been developed so that they can function synergistically as a single IDPS (Intrusion Detection and Prevention System) system, and finally creating an interface design (UI / UX) to make it easier for users (school IT Team) to monitor and analyze incidents. In the implementation phase, it begins with the publication of the Semont application on the main domain of SMAN 1 Rancaekek, namely on the page https://www.smanrancaekek.sch.id/, with file placement in a special directory named /semont/. Semont is integrated at the web server level with automatic configuration to detect every HTTP (port 80) and HTTPS (port 443) request so that Semont can analyze every web traffic in real-time, enabling detection and prevention of intrusions at an early stage. In loader.php four core Semont files are loaded sequentially, namely config.php, logger.php, brute-force.php. and monitoring.php.

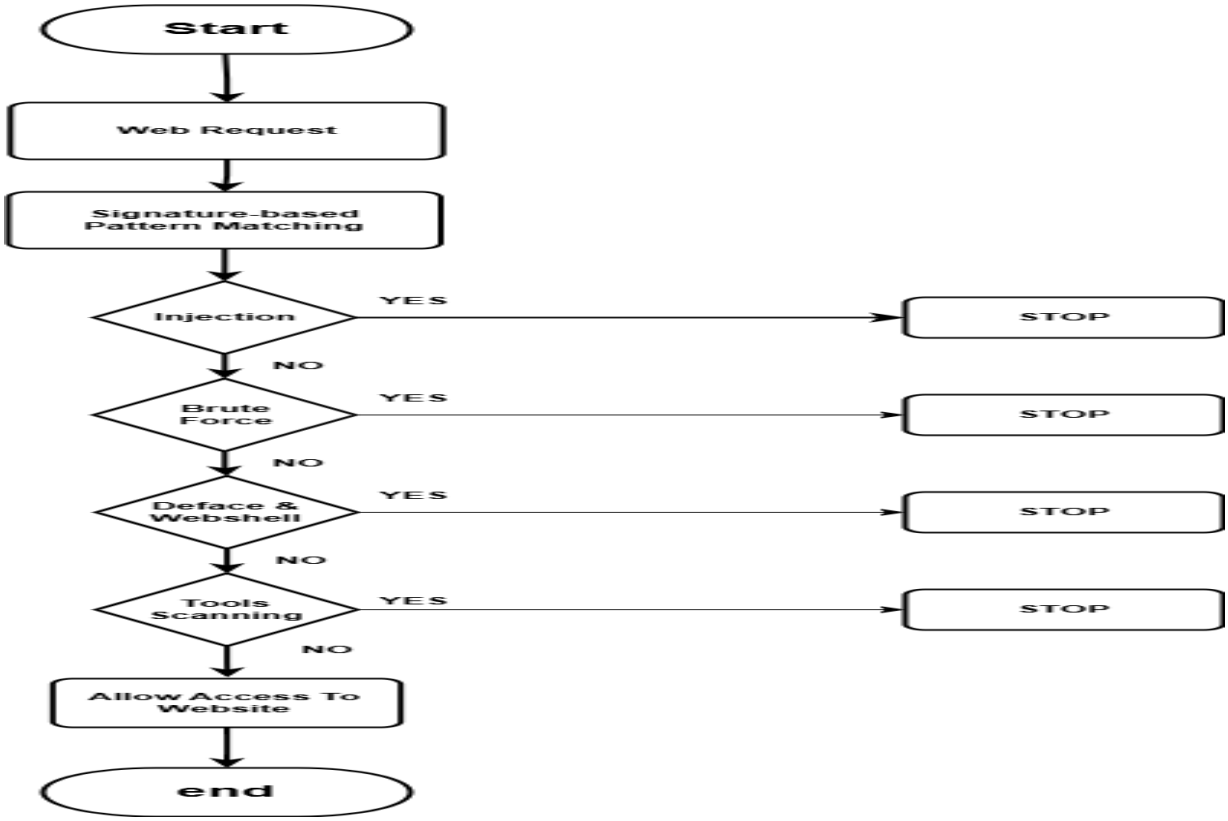

Figure 1. Signature-based method stages

The implementation phase begins with the publication of the Semont application on the main domain of SMAN 1 Rancaekek, at https://www.smanrancaekek.sch.id/, with files placed in a dedicated directory named /semont/. Semont is integrated at the web server level with automatic configuration to detect every HTTP (port 80) and HTTPS (port 443) request.

This allows Semont to analyze all web traffic in real time, enabling early intrusion detection and prevention. In loader.php, four core Semont files are loaded sequentially: config.php, logger.php, brute-force.php, and monitoring.php.

config.php serves as the central repository for all Semont system settings. Key configurations such as injection detection status (detect_injection), scanning tool detection (detect_scanning_tool), and IP blocking behavior (block_ip_on_detection, block_non_indonesia_ip) are loaded from the config.json file. This allows administrators to manage security functionality through the Semont user interface without having to modify the code directly. Furthermore, this file manages Telegram API tokens and a list of chat IDs from telegram_ids.json for sending real-time notifications, as well as IPInfo tokens for geolocating the attacker's IP address.

Furthermore, logger.php is the core of Semont's response mechanism. The semont_log() function is responsible for logging details of detected attacks to semont.log in a comprehensive format, including timestamp, IP address, device fingerprint, geographic location, detection reason, payload used, target URL, and User-Agent. Real-time notifications are a crucial feature implemented through the send_to_telegram() function. This module also implements specific detection for scanning tools using the semont_detect_scanning_tools() function. This system recognizes the User-Agent of common vulnerability scanning tools like SQLMap. If detected, this activity is logged, and the scanning IP is immediately intercepted or even blocked. Injection attack detection is also an integral part of logger.php through the semont_detect_injection() function. This function applies signature-based detection to payloads found in user input (via the GET, POST, COOKIE, and REQUEST methods).

The signatures used are designed to identify critical vulnerability patterns such as SQL Injection, Cross-Site Scripting (XSS), Remote Code Execution (RCE), and Local File Inclusion (LFI). Adaptive blocking is implemented through the semont_block_ip() function. If the number of attacks from a specific IP or device fingerprint exceeds a configured threshold, the system automatically blocks their access to the website. The user-agent-blocks.php file is a dedicated module responsible for detecting and blocking scanners and User-Agents containing injection payloads, with an HTTP 501 Not Implemented response. This module reads malicious User-Agent lists from various blacklist files such as block-user.txt, acunetix.txt, block.txt, dalfox.txt, dirb.txt, dirsearch.txt, nikto.txt, xspear.txt,

sqliv.txt, wapiti.txt, xss-strike.txt, jsqli.txt, Zed_Attack_Proxy_(ZAP).txt, and user-agents-nuclei.txt. In addition to matching against blacklists, this module also has specific User-Agent patterns for Acunetix Vulnerability Scanner and regex patterns to detect generic injection payloads (time-based SQL Injection, Remote Code Execution, Cross-Site Scripting, Path Traversal) inserted directly in the User-Agent. If a User-Agent is detected matching these criteria, the module logs its details using the semont_log() function (from logger.php) with the scanning-tool log type, and immediately responds with an HTTP 501 Not Implemented and terminates the connection, stopping further PHP script execution. This mechanism serves as an aggressive first line of defense against scanning tools and attacks that use User-Agents as a vector. brute-force.php is dedicated to detecting and preventing brute-force attacks, specifically on login pages. It monitors POST requests addressed to login URLs (/login.php, /auth.php) or containing credential parameters such as 'username' and 'password'. Configuring a threshold (SM_MAX_LOGIN_ATTEMPTS) and monitoring period (SM_MONITOR_LOGIN_PERIOD_SECONDS) allows for accurate brute-force detection. In response, a delay (SM_LOGIN_FAILURE_DELAY_SECONDS) is applied to each detected login attempt, significantly slowing down the attacker. If the threshold is exceeded, the attacker's IP address will be blocked and redirected to about:blank. In addition to logging, this module also applies rate limiting to general, non-login POST requests (SM_MAX_GENERAL_POST_REQUESTS), which serves to prevent flooding or simple Denial of Service (DoS) attacks. The monitoring.php file serves as an HTTP activity monitoring module, focusing on logging POST requests that may contain sensitive information such as login attempts or user IDs. The logSecurityActivity() function captures POST requests, extracts username or ID information from the POST data (either JSON or URL-encoded), and logs it to a dedicated log file (security_stream-YYYYMM.json). This module also features a cleanUpOldLogs() function that automatically rotates and cleans out old logs on a monthly and annual basis, maintaining log storage efficiency and availability. It's important to note that requests to the Semont dashboard itself are excluded from logging to avoid self-logging.

IV. RESEARCH RESULTS

Semont is implemented as an integrated protection layer within the web server. It can also be used as external storage for critical functions such as notifications and secure log storage. The Semont user interface is shown in Figure 2.

Because the Semont application is implemented in conjunction with the web server, the Semont login area can be accessed at https://www.smanrancaekek.sch.id/semont/sentinel-dashboard/login.php. Other Semont features are shown in Figure 3.

Figure 2. Semont Login Page

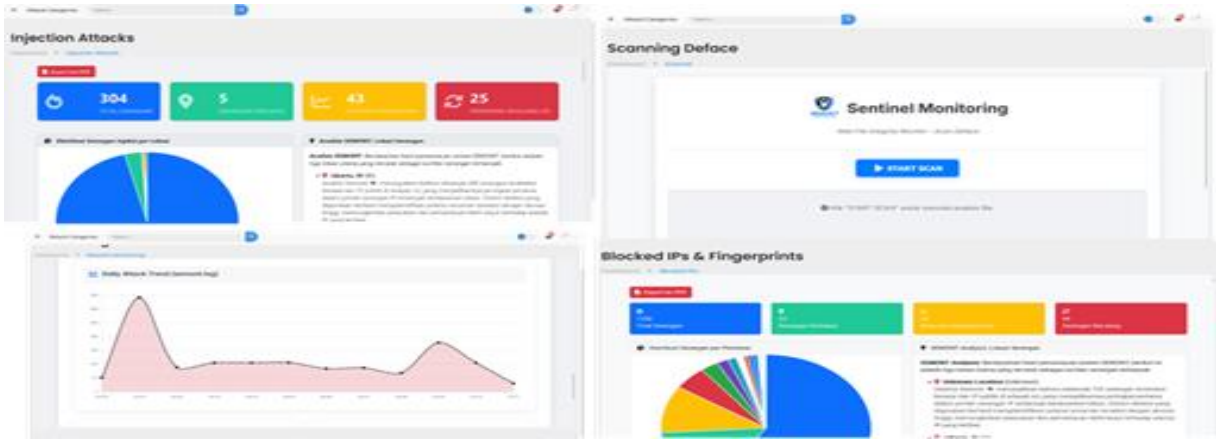

Figure 3. Semont Features

The test was conducted by simulating various types of cyberattacks on the SMAN 1 Rancaekek website, which has been implemented with Semont. The main target URL for the Injection and Scanning Tools tests was the endpoint /pengumumanlulus/percobaan.php and the parameter /percobaan-skripsi.php?id=10, while for the Brute Force test it was https://smanrancaekek.sch.id/skripsi/login.php.

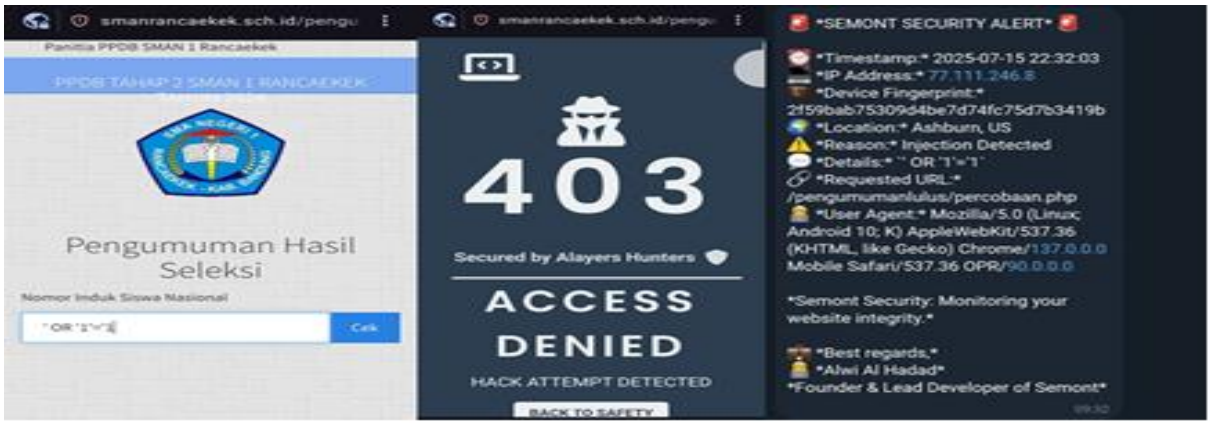

Figure 4. SQL Injection Attack Detection Results

Initially, an SQL injection attack was attempted to display a selection announcement for "Aleesha Fiorina Sukmana" using a manual injection exploit on a website without Semont. Semont successfully detected and prevented the injection attacks targeting uji.php and uji-skripsi.php. Each malicious payload was detected and logged in semont.log, a Telegram notification was sent, and the attacker's IP address was redirected to 404.php.

The second stage involved a brute-force attack on skripsi/login.php using the Burp Suite intruder, as shown in Figure 5. Semont successfully detected the brute-force attack.

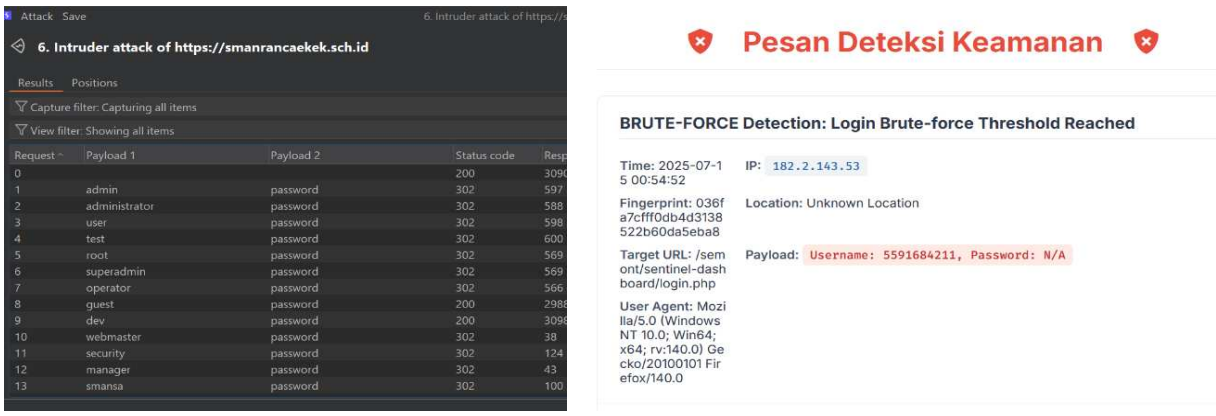

Figure 5. Brute Force Attack Detection Results

The third stage involves executing a web defacement attack to alter the website's appearance. This attack was successful before using Semont. However, once Semont was implemented, Semont successfully detected uploaded or modified files using the keyword "deface" or the webshell name or content. The detection results are displayed in the Deface Scanning UI. The file deletion feature also worked effectively, as can be seen in Figure 6.

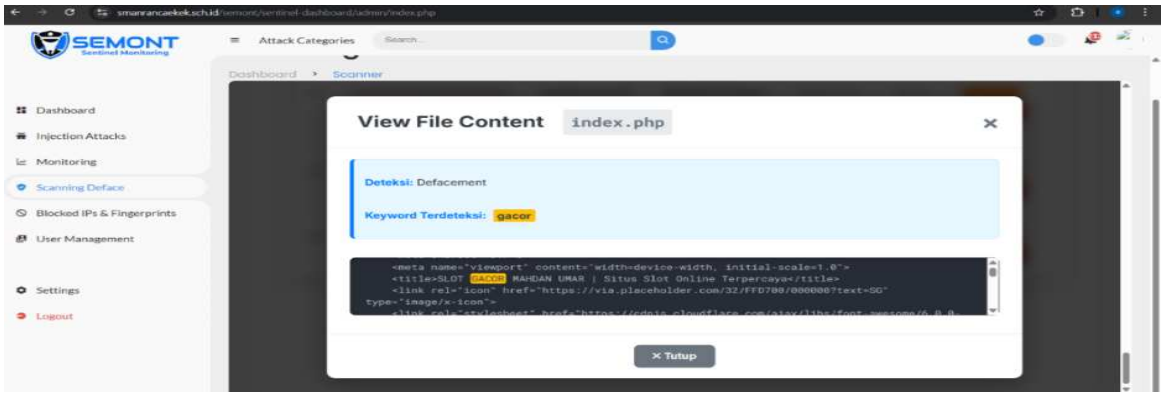

Figure 6. Defacement Scanning Results

This analysis shows that SEMONT effectively transformed the security posture of SMAN 1 Rancaekek's website from vulnerable to secure. Every type of attack that was previously exploited was now successfully detected and prevented by Semont, supported by detailed logging and real-time notifications to administrators.

V. CONCLUSION

The Semont application effectively intercepts all web traffic in real-time and consists of modules that work synergistically for detection and prevention. Semont's effectiveness in detecting and preventing cyberattacks has been proven to be very high. Through comparative testing, the SMAN 1 Rancaekek website, which was previously vulnerable to injection attacks, scanning tools, brute force attacks, as well as web defacement and webshell attacks, Semont successfully detected and blocked these attacks. This directly addresses the issues of website vulnerability and the lack of a proactive defense system. Semont's traffic monitoring and automated response functionality performed optimally. The system was able to comprehensively record attack details, send real-time notifications to Telegram administrators, and perform effective IP blocking and redirection. The direct remediation feature for web defacement also provided rapid control, addressing the issue of how the system can monitor and address incidents. The strength of this study lies in the implementation of a lightweight and efficient signature-based IDPS system, which generates minimal overhead on the web server, thus not disrupting website operations. The comparative approach (before vs. after implementation) empirically demonstrated a significant improvement in the website's security posture. Furthermore, the system provides an intuitive user interface and real-time notifications, which are very helpful for school IT teams. However, the limitations of signature-based methods are that they are ineffective in detecting

zero-day attacks or new attack variants not yet registered in the signature database. Furthermore, the scope of this research does not include in-depth development of statistical or rule-based anomaly detection methods, and it does not address the overall security of network infrastructure beyond the scope of web applications. Therefore, there is still a need to develop a mechanism to automatically update the Semont signature database from trusted cyber threat feed sources. This will improve the system's ability to detect the latest attacks and reduce reliance on manual updates. Adding an anomaly-based detection module (e.g., using machine learning algorithms to analyze abnormal traffic patterns) to complement the signature-based method will enable Semont to identify threats that do not have known signatures, including zero-day attacks. Developing more advanced data visualization features on the SEMONT dashboard, such as interactive graphs for attack trends, more flexible log data filters, and drill-down capabilities for more in-depth incident analysis.

REFERENCES

Al Rubaiei, M., Al Yarubi, T., Al Saadi, M., & Kumar, B. (2020). SQLIA detection and prevention techniques. *Proceedings of the 2020 9th International Conference on System Modeling and Advancement in Research Trends, SMART 2020*, *December 2023*, 115–121. https://doi.org/10.1109/SMART50582.2020.9336795

Azeez, N. A., Bada, T. M., Misra, S., Adewumi, A., Van der Vyver, C., & Ahuja, R. (2020). Intrusion Detection and Prevention Systems: An Updated Review. *Advances in Intelligent Systems and Computing*, *1042*(January), 685–696. https://doi.org/10.1007/978-981-32-9949-8_48

Elan Maulani, I., & Faisal umam, A. (2023). Evaluasi Efektivitas Sistem Deteksi Intrusi Dalam Menjamin Keamanan Jaringan. *Jurnal Sosial Teknologi*, *3*(8), 662–667. https://doi.org/10.59188/jurnalsostech.v3i8.907

George, A. S., & George, A. S. H. (2021). A Brief Study on The Evolution of Next Generation Firewall and Web Application Firewall This work is licensed under a Creative Commons Attribution 4.0 International License A Brief Study on The Evolution of Next Generation Firewall and Web Application Fir. *International Journal of Advanced Research in Computer and Communication Engineering*, *10*(5), 31–37. https://doi.org/10.17148/IJARCCE.2021.10504

Habibah, A. N. (2024). *Keamanan informasi dalam konteks teknologi komunikasi modern*. *2*(6), 965–971.

Hadad, A. Al. (2025). *Kerentanan IDOR: Kerentanan yang Unik dalam Aplikasi Web*. CSIRT NUSA MANDIRI.

Id-SIRTII /CC. (2023). Lanskap Keamanan Siber Indonesia. *Id-SIRTII /CC*, *70*, 1–107.

Irawan, A. S., Sakti Pramukantoro, E., & Kusyanti, A. (2018). Pengembangan Intrusion Detection System Terhadap SQL Injection Menggunakan Metode Learning Vector Quantization. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(6), 2295–2301.

Journal, I., Science, C., & Volume-, E. (2021). *Ethical Hacking and Penetrate Testing using Kali and Metasploit Framework Mujahid Tabassum Saju Mohanan Department of IT , University of Technology and Department of IT , University of Technology and Applied Sciences Applied Sciences Muscat , Oman Tripti . 1*, 9–22.

Kurniawan, A., Darus, M. Y., Ariffin, M. A. M., Muliono, Y., & Pardomuan, C. R. (2023). Automation of Quantifying Security Risk Level on Injection Attacks Based on Common Vulnerability Scoring System Metric. *Pertanika Journal of Science and Technology*, *31*(3), 1245–1265. https://doi.org/10.47836/pjst.31.3.07

Mahdi Maulana Lubis, M., Handoko, D., & Wulan, N. (2022). Analisis Implementasi Laravel 9 Pada Website E-Book Dalam Mengatasi N+1 Problem Serta Penyerangan Csrf dan Xss. *Januari*, *2023*(2), 173–187.

Nugraha, L. A., Kautsar, I. A., & Fitrani, A. S. (2024). SQL Injection: Analisis Efektivitas Uji Penetrasi dalam Aplikasi Web. *Smatika Jurnal*, *14*(01), 111–123. https://doi.org/10.32664/smatika.v14i01.1224

Remesh Babu, K. R., Saritha, S., Preetha, K. G., Sangeetha, U., & Izudheen, S. (2023). An Intelligent Pattern Matching approach with Deep Hypersphere Model for Secure Big Data Storage in Cloud Environment. *International Journal of Computer Information Systems and Industrial Management Applications*, *15*(2023), 166–175.

Sándor R. Répás, S. A. H. (2024). Anomaly Detection in Log Files Based on Machine Learning Techniques. *Journal of Electrical Systems*, *20*(3s), 1299–1311. https://doi.org/10.52783/jes.1505

Saputra, I. P., Utami, E., & Muhammad, A. H. (2022). Comparison of Anomaly Based and Signature Based Methods in Detection of Scanning Vulnerability. *International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, *2022-Octob*(October), 221–225. https://doi.org/10.23919/EECSI56542.2022.9946485

Somé, D. F. (n.d.). *MatriXSSed : a New Taxonomy for XSS in the Modern Web*. 4662–4672. https://doi.org/10.1145/3696410.3714774

Widodo, T., & Aji, A. S. (2022). Pemanfaatan Network Forensic Investigation Framework untuk Mengidentifikasi Serangan Jaringan Melalui Intrusion Detection System (IDS). *JISKA (Jurnal Informatika Sunan Kalijaga)*, *7*(1), 46–55. https://doi.org/10.14421/jiska.2022.7.1.46-55

Widyarto, E. Y., & Hapsari, D. K. (2022). Analisis Modus Operandi Tindak Kejahatan Menggunakan Teknik Komunikasi Love Scam Sebagai Ancaman pada Keamanan Sistem Informasi. *Syntax Idea*, *4*(9), 1352. https://doi.org/10.36418/syntax-idea.v4i9.1959