Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 6, No.2, April 2024, pp. 0240209-01 ~ 0240209-09
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v6i2.18307

Enhancing Web Server Security against Layered Cyber Threats in Healthcare

Muhammad Fajar Rifai, Syaiful Hendra*, Hajra Rasmita Ngemba, Ryfial Azhar, Rahmah Laila

Information Technology Department, Faculty of Engineering, Tadulako University, Jl.Soekarno Hatta No KM 9 Palu 94148, Central Sulawesi, Indonesia

*syaiful.hendra.garuda@gmail.com

Abstract. *Information technology plays an important role in improving operational efficiency at Torabelo Hospital. The server system in use today faces security and optimization challenges. This research analyzes the impact and recommends solutions to improve server security and optimization. The findings show that the server system is vulnerable to various types of attacks and performance degradation. This can negatively impact hospital operations and put patients at risk. The recommended solution is to implement Squid as reverse proxy, WAF (Web Application Firewall), and Snort as IDS (Intrusion Detection System). System testing showed that this solution successfully detected and prevented various common attacks. This research provides insights to health IT professionals to improve the security and performance of their server systems and improve healthcare services to patients at Torabelo Hospital.*

Keywords: Hospital, Security, Squid, Web Application Firewall, Healthcare

(Received 2024-01-31, Accepted 2024-02-13, Available Online by 2024-03-08)

1. Introduction

In today's digital age, information technology is at the heart of operational efficiency in many sectors, including the healthcare industry. As an integral part of its healthcare system, Torabelo Hospital has adopted a server system to manage patient data, organize medical procedures, and provide quick access to critical information. However, these advancements pose serious server security and optimization challenges that impact the hospital's operations. Security issues that could result in sensitive patient data leakage or delays in clinical information delivery due to lack of server optimization require in-depth understanding and appropriate solutions[1]. Torabelo Hospital's server security issues have drawn attention as potentially serious. Security threats such as hacking, malware, and even unauthorized access will result in losses and compromise the privacy and security of patients' personal information [2,3]. Server security in healthcare is a key issue when protecting sensitive patient data. Various cyber-attacks such as malware, phishing, and DDoS can compromise servers and cause data leakage. Unauthorized access through weak passwords, excessive privileges, and stolen credentials can compromise patient privacy [16, 17]. Patient data security is an important aspect of modern healthcare systems. Protecting

sensitive medical records requires ensuring various aspects such as network security, access, data encryption, and user identity management. Such protection is important to maintain patient privacy, improve healthcare quality, and comply with applicable regulations[18]. Effective information security risk management is essential to prevent or minimize the impact of unwanted incidents. This can be achieved by identifying, analyzing and evaluating risks and implementing appropriate risk mitigation measures [19].

Information security focuses on three main pillars: confidentiality, integrity, and availability. These pillars are reinforced by authentication, authorization, auditing, and non-denial. Proper implementation will build trust and improve efficiency in the digital age [20]. Information security such as protecting information and devices in the workplace. Minimizing damage from various threats. Its aspects include privacy, identification, authentication, authorization and accountability. Its presence ensures a smooth and secure workflow [21]. Information security is an important consideration when creating an integrated clinical environment. Development should be based on a fundamental system that ensures confidentiality of patient data, integrity of information, controlled access, and accountability of all actions. The implementation of these systems creates a strong fortress that protects sensitive information, ensures smooth workflow, and builds trust between patients and medical staff. Information security is not the responsibility of one party alone; all elements of the healthcare ecosystem must work together to create safe and reliable services [22]. Information security is like a solid fortress that protects a company's valuable assets from various threats. Its presence minimizes losses and helps the company achieve its goals with strong internal and external control systems. Its implementation demonstrates an organization's commitment to data security and stakeholder trust [21].

In addition, lack of server optimization slows down data access and processing, which are critical factors in a medical environment that can impact timely diagnosis and treatment of patients. Therefore, it is important to understand the impact of these security issues and lack of optimization on overall hospital operations and develop appropriate strategies to address them. The purpose of this study is to document and analyze in detail the security challenges faced by Torabelo Hospital's servers and to determine the impact of the lack of optimization of medical servers. By understanding the causes of these issues, this research provides valuable insights for healthcare IT professionals and other stakeholders to take effective steps in improving the security and optimization of their server systems, thus ultimately expected to support the provision of sound medical services. It is also expected to provide better and more efficient services for patients at Torabelo Hospital.

2. Methods

2.1. Stages of research

The hospital's IT infrastructure consists of a TCP/IP network with Internet access, servers with Ubuntu operating systems, EMR and HIS applications, and hardware such as servers, firewalls, routers, and workstations. The use of Ubuntu operating system on the hospital's server improves the security and stability of the IT system. The EMR application helps hospitals store and manage patient records electronically, thus improving the efficiency and accuracy of medical documentation. The HIS integrates various administrative and clinical functions of the hospital and improves the efficiency and effectiveness of hospital operations. High-speed internet connectivity ensures smooth operation of the IT system and access to patient data. At this stage of the research, the researcher will outline the steps taken in this study. The following steps are carried out as follows :

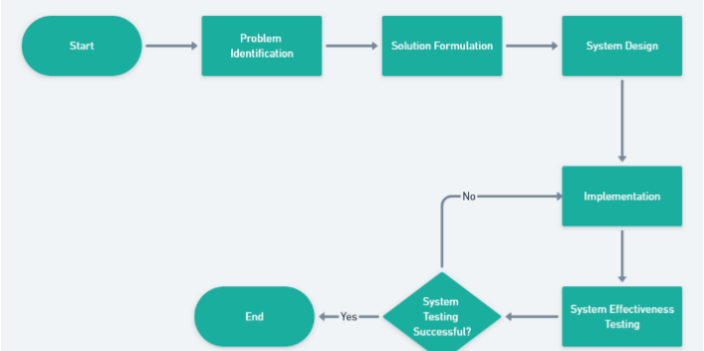

Figure 1. Research Flow Chart

During the problem identification phase, the entire web system requires additional protection against potentially harmful cyber attacks such as Deface, SQL injection, XSS, and DDoS. In addition, it requires a system that can monitor and detect suspicious activity[4,5]. The formulated solution involves using Squid reverse proxy as the first layer of defense[3,6]. Squid acts as an intermediary that directs incoming traffic to the Apache web server. Additionally, the solution includes the implementation of a web application firewall (WAF) to filter and prevent attacks such as tampering, SQL injection, and XSS on incoming traffic. Properly configured IDS (Intrusion Detection System) and IPS (Intrusion Prevention System) settings are also required for early detection and further prevention[7-9]. The system design considers the Apache configuration as the main web server and is configured to handle requests after passing through the Squid reverse proxy[10,11]. The Squid configuration filters incoming traffic and forwards it to Apache for processing. Squid's built-in WAF implementation to monitor and prevent the attacks mentioned above. IDS and IPS are installed at the network layer to monitor traffic, detect suspicious activity, and automatically prevent attacks if detected. The implementation includes configuring Apache as a web server, setting up Squid reverse proxy, installing and configuring WAF to prevent certain attacks, and adding IDS to your network environment to monitor and protect traffic flowing through your system, and installing IPS[12,13].

2.2. System Design

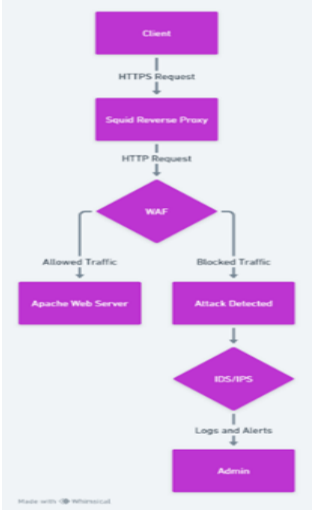

Figure 2. System Design

A system that combines Apache as a web server and Squid as a reverse proxy and designs a configuration that sets up Squid as the first layer before forwarding access to the Apache web server to secure the web infrastructure[2]. In this configuration, Squid acts as the initial gateway that receives requests from users. Squid completes some additional validation and processing phases before

forwarding the request to Apache. This allows the use of access control policies and caching, which improves security by optimizing requests to internal servers before they reach the main web server[14].

The application of Web Application Firewall (WAF) when designing this system is an important step to protect web applications from various malicious attacks such as Deface, SQL Injection, XSS, and DDoS attacks. WAF works by analyzing HTTP requests and filtering incoming content to the web application to ensure that the received data does not contain malicious payloads or programmatic attacks. With proper configuration[15], WAF provides a strong and adaptable layer of defense against evolving security threats by setting access policies and blocking attacks before they reach the web application[9].

The implementation of an Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) into the system strengthens the defense layer against attacks. IDS monitors and analyzes network traffic to detect suspicious behavior patterns and attacks in progress. IPS, on the other hand, acts as an active defense against attacks by shutting down or blocking traffic identified as potential threats. With IDS and IPS, the system becomes more responsive in detecting, responding to, and protecting your infrastructure from various malicious attacks[5,10].

3. Results and Discussion.

3.1. Squid Reverse Proxy Configuration

The first step is to install Squid on the Ubuntu server, then configure the reverse proxy as the first layer before switching to the Apache web server in "Squid. Conf.

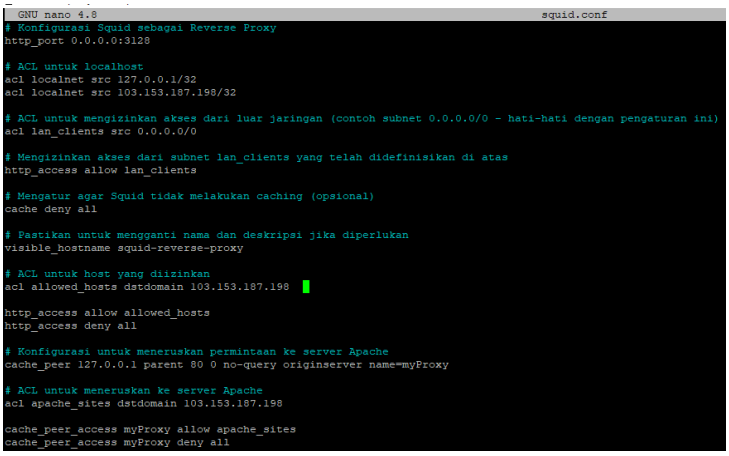

Figure 3. Squid Reverse Proxy Configuration

3.2. Web Application Firewall (WAF) Configuration

The steps taken in the WAF configuration are to install the ModSecurity "Library" and then activate "ModSecurity" in the Apache configuration at the end of the file.

LoadModule security2_module /usr/lib/apache2/modules/mod_security2.so

Then, define the security rules in ModSecurity.conf.

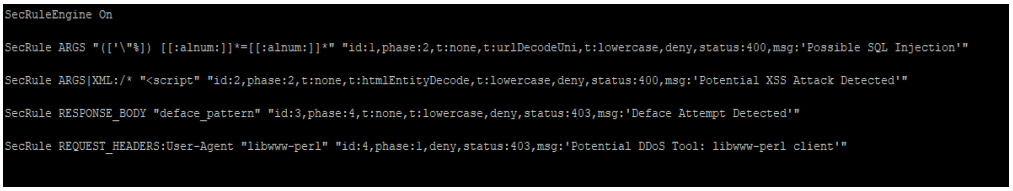

Figure 4. Rules on WAF

Add "/etc/modsecurity/webproxy_rules.conf" at the end of the "ModSecurity.conf" block and configure the VirtualHost block to enable ModSecurity.

```
<IfModule security2_module>
 SecRuleEngine On
 SecRule ARGS "([\'\"%]) [[:alnum:]]*=[[:alnum:]]*" "id:1,phase:2,t:none,t:urlDecodeUni,t:lowercase,deny,status:400,msg:'Possible SQL Injection'"
 SecRule ARGS|XML:/* "<script" "id:2,phase:2,t:none,t:htmlEntityDecode,t:lowercase,deny,status:400,msg:'Potential XSS Attack Detected'"
 SecRule RESPONSE_BODY "deface_pattern" "id:3,phase:4,t:none,t:lowercase,deny,status:403,msg:'Deface Attempt Detected'"
 SecRule REQUEST_HEADERS:User-Agent "libwww-perl" "id:4,phase:1,deny,status:403,msg:'Potential DDoS Tool: libwww-perl client'"
</IfModule>
```

Figure 5. Rules on VirtualHost Blocks

3.3. Intrusion Detection System (IDS) and Intrusion Prevention System (IPS) Configuration

Install snort and configure integration with waf located in snort.conf

```
 GNU nano 4.8                                                    snort.conf
#IDS
alert tcp any any -> any 80 (msg:"Possible Deface Attempt Detected"; content:"deface_pattern"; sid:100001; rev:1;)

alert tcp any any -> any 80 (msg:"Potential SQL Injection Detected"; content:"SELECT * FROM"; sid:100002; rev:1;)

alert tcp any any -> any 80 (msg:"Potential XSS Attack Detected"; content:"<script>"; sid:100003; rev:1;)

alert tcp any any -> any 80 (msg:"Potential DDoS Attack Detected"; flags:S; threshold: type both, track by_dst, count 100, seconds 10; sid:100004; rev:1;)

#IPS
alert tcp any any -> any 80 (msg:"SQL Injection Detected"; flow:to_server,established; content:"SELECT"; nocase; sid:100001; rev:1;)

alert tcp any any -> any 80 (msg:"Cross-Site Scripting (XSS) Attempt"; flow:to_server,established; content:"<script>"; nocase; sid:100002; rev:1;)

alert tcp any any -> any 22 (msg:"SSH Brute Force Attempt"; flow:to_server; flags:A+; threshold: type both, track by_src, count 5, seconds 60; sid:100003; rev:1;)

alert tcp any any -> $HOME_NET 21 (msg:"FTP Port Scan Detected"; flags:S; threshold: type both, track by_src, count 5, seconds 60; sid:100004; rev:1;)

alert icmp any any -> $HOME_NET any (msg:"ICMP Flood Detected"; dsize:0; threshold: type both, track by_src, count 50, seconds 10; sid:100005; rev:1;)
```

Figure 6. Snort integration with WAF

3.4. Rules Setting

Setting rules by pointing to Snort (IDS) first before passing to squid (as a reverse proxy) and WAF (Web Application Firewall). Using the management tool "iptables" to manage NAT (Network Address Translation) and routing.

```
root@webproxy:~# ^C
root@webproxy:~# ^C
root@webproxy:~# sudo iptables -A INPUT -p tcp --dport 80 -j REDIRECT --to-port 8888
root@webproxy:~# ^C
root@webproxy:~# sudo iptables -A INPUT -p tcp --dport 8888 -j REDIRECT --to-port 3128
root@webproxy:~# ^C
root@webproxy:~# sudo iptables -A INPUT -p tcp --dport 3128 -j REDIRECT --to-port 8080
root@webproxy:~# ^C
root@webproxy:~#
```

Figure 7. Rules Integration

This rule specifies that all packets arriving on port 80 (the default port for HTTP) are redirected to port 8888, where Snort is running. Snort is an intrusion detection system (IDS/IPS) that examines incoming traffic to detect potential security threats. It then redirects the traffic forwarded by Snort (running on port 8888) to Squid, running on port 3128. And forwards traffic routed through Squid (running on port 3128) to a WAF (such as ModSecurity) running on port 8080

3.5. Black Box Testing

To verify the security and robustness of the entire system, black box testing is performed with a predefined one with the aim of evaluating the external side of the system without detailed knowledge of its internal structure. Tests can be seen in the table.

Table 1. Black Box Testing

Features	Result Expectations	Result
Apache (Web Server)	accessed through a browser directs traffic to Apache as a web server	Valid
Squid (Reverse Proxy)	squid reverse proxy as the first layer before pointing to the Apache web server	Valid

Web Application Firewall (WAF)	WAF addresses attacks through the firewall in the form of defacers, SQL Injection, XSS, and DDOS.	Valid
Intrusion Detection System (IDS)	detects suspicious activity on the network. Sends an alert to the network or system administrator for further investigation.	Valid
Intrusion Prevention System (IPS)	detects suspicious activity, and takes action to prevent attacks, such as blocking traffic or changing system settings.	Valid

3.6. Integration Testing

This stage is to test the interaction between components that have been configured previously. Integration testing is done using Apache JMeter tools. Table 3 is the result of integration testing using Apache Jmeter.

Table 2. Integration Testing

Sample	**Before**			**After**		
Performance Test	Sample Time (ms)	Latency	Connect Time (ms)	Sample Time (ms)	Latency (ms)	Connect Time (ms)
10 User 1 Seconds 3 Loop	Min 174 Max 969	Min 114 Max 420	Min 173 Max 346	Min 71 Max 176	Min 71 Max 175	Min 73 Max 88
Stress Test	Sample Time (ms)	Latency	Connect Time (ms)	Sample Time (ms)	Latency	Connect Time (ms)
222 User 1 Seconds 2 Loop	Min 176 Max (969)	Min 110 Max 942	Min 177 Max 1128	Min 71 Max 401	Min 71 Max 210	Min 72 Max 114

Based on the test comparison table. Tests have shown that integrating squir reverse proxy as the first layer of the web server can significantly improve performance. This is evidenced by a 59% reduction in minimum sample time, a 58% reduction in maximum latency, and a 90% reduction in maximum connection time. These improvements show that integrating the Squid reverse proxy improves the efficiency and responsiveness of the web server, thereby improving the quality of service to users. The minimum sampling time before configuration was 174 ms and after configuration was 71 ms. The maximum latency before configuration was 420 ms and after configuration was 175 ms. The maximum connect time was 1128 ms before configuration and 114 ms after configuration. This performance improvement shows that integrating Squid reverse proxy allows the web server to process requests more quickly and efficiently, providing a better user experience.

3.7. Penetration Testing

This stage is to ascertain potential security or vulnerabilities in the system. Penetration testing is carried out with the aim of identifying and exploring security gaps.

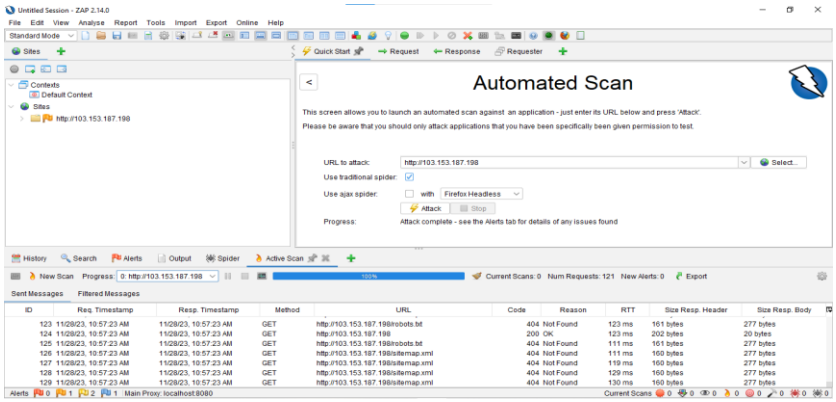

Figure 8. Penetration Testing

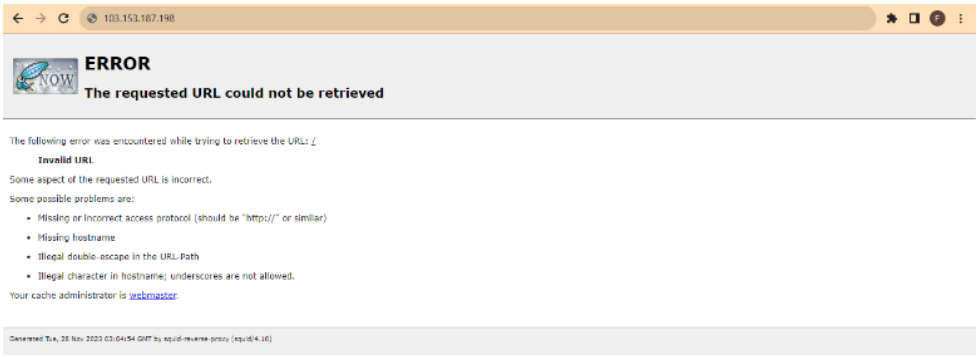

Figure 9. Squid Reverse Proxy Results After Penetration Testing

Penetration testing confirmed the success with the penetration results in Figure 7 showing that the configuration was successful where there were 0 critical vulnerabilities (red flags), 1 moderately severe vulnerability (orange flag), 2 vulnerabilities with the lowest level (yellow flag). from the penetration test, squid also prevented the attack by not forwarding the attack request to the main server and providing an error response to the attacker. Squid has proven effective in defending against attacks by blocking malicious requests and sending error messages to attackers. However, serious vulnerabilities must be addressed and continuous monitoring is essential. These upgrades have demonstrated a secure and efficient system, however continuous optimization is required to ensure maximum security.

4. Conclusion

After running a series of configurations, including implementing Squid as a reverse proxy, WAF (Web Application Firewall), and Snort as an IDS (Intrusion Detection System) and testing the effectiveness of the system, overall, the system successfully detected and prevented many common attacks such as SQL injection, XSS, and DDoS attacks. However, from the test still has a gap where there are still very critical vulnerabilities (high risk) which may cause great losses. if future researchers want to use this journal as a reference, it is hoped that they can develop even better security.

Acknowledgement

The researchers would like to express their deepest gratitude to Tadulako University, especially the Merdeka Center for Learning at Merdeka Campus (MBKM) and the Informatics Engineering Study Program. Thanks to the support of Tadulako University through the MBKM program, researchers can complete this research as well as possible. The opportunity to participate in the MBKM Independent Study program also provided the researchers with valuable experiences that increased their knowledge and insight in the field of computer science.

References

[1]. R. Riswandi, K. Kasim, and M. F. Raharjo, "Evaluasi Kinerja Web Server Apache menggunakan Protokol HTTP2," J. Tek. dan Sist. Inform., vol. 2, no. 1, pp. 19-31, 2020, doi: 10.36079/lamintang.jetas-0201.92.

[2]. Riska and H. Alamsyah, "Penerapan Sistem Keamanan Web Menggunakan Metode Web Application Firewall," J. Amplif. : J. Ilm. Bid. Tek. Elekt. dan Komput., vol. 11, no. 1, pp. 37-42, 2021, doi: 10.33369/jamplifier.v11i1.16683.

[3]. A. Susanto, "Pengaturan Keamanan Squid Proxy Pada Jaringan Lokal Dan Internet Menggunakan Openldap," 2022, vol. 2.

[4]. B. Wiguna, A. P. W., and R. Ananda, "Implementasi Web Application Firewall Dalam Mencegah Serangan SQL Injection Pada Website," Digital Zone, vol. 11, no. 2, pp. 245-256, 2020, doi: 10.31849/digitalzone.v11i2.4867.

[5]. T. A. F. Anugrah, S. Ikhwan, and A. G. J. Gusti, "Implementasi Intrusion Prevention System (IPS) Menggunakan Suricata Untuk Serangan SQL Injection," Techne, vol. 21, no. 2, pp. 199-210, 2022, doi: 10
[6]. Zain AR, Matin IMM. "Analisis Implementasi Modsecurity dan Reverse Proxy Untuk Pencegahan Serangan Keamanan DDoS pada Web Server." 2023;2(1).
[7]. Syani M. "IMPLEMENTASI INTRUSION DETECTION SYSTEM (IDS) MENGGUNAKAN SURICATA PADA LINUX DEBIAN 9 BERBASIS CLOUD VIRTUAL PRIVATE SERVERS (VPS)." JurnalInkofar. 2020;1(1). doi:10.46846/jurnalinkofar.v1i1.155
[8]. Suwanto R, Ruslianto I, Diponegoro M. "IMPLEMENTASI INTRUSION PREVENTION SYSTEM (IPS) MENGGUNAKAN SNORT DAN IPTABLE PADA MONITORING JARINGAN LOKAL BERBASIS WEBSITE." 2019;07(1).
[9]. Sutabri T. "Analisis Cyber Crime handling pada Aplikasi Web dengan WAF ModSecurity." 2023;16(1).
[10]. Ullah MU, Hassan A, Asif M, Farooq MS, Saleem M, Ullah U. "Intelligent Intrusion Detection System for Apache Web Server Empowered with Machine Learning Approaches." 2022;1.
[11]. Nasir A, Lordianto RL. "IMPLEMENTASI PROXY SERVER UNTUK OPTIMALISASI MANAJEMEN BANDWIDTH JARINGAN KOMPUTER PADA UNIVERSITAS XYZ." JTech. 2023;11(1):16-23. doi:10.30869/jtech.v11i1.1164
[12]. Muntaha MF, Trisnawan PH, Primananda R. "Implementasi Intrusion Prevention System (IPS) berbasis Athena untuk Mencegah Serangan DDoS pada Arsitektur Software-Defined Network (SDN).".
[13]. Muqorobin M, Hisyam Z, Mashuri M, Hanafi H, Setiyantara Y. "Implementasi Network Intrusion Detection System (NIDS) Dalam Sistem Keamanan Open Cloud Computing." MIBJ. 2019;17(2):1-9. doi:10.33489/mibj.v17i2.205
[14]. Universitas Methodist Indonesia, Gea A, Purba MJ, Putra AA, Jamaluddin J, Siringoringo R. "IMPLEMENTASI METODE ACCESS CONTROL LIST UNTUK MEMONITORING AKSES JARINGAN MENGGUNAKAN SQUID PROXY." jmika. 2022;6(1):79-84. doi:10.46880/jmika.Vol6No1.pp79-84
[15] R. A. Muzaki, O. C. Briliyant, M. A. Hasditama, and H. Ritchie, "Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall," in *2020 International Workshop on Big Data and Information Security (IWBIS)*, Depok, Indonesia: IEEE, Oct. 2020, pp. 85–90. doi: 10.1109/IWBIS50925.2020.9255601.
[16] R. A. Muzaki, Y. A. Wilar, K. Yuliawan, and A. A. Natsir, "ANALISIS KEAMANAN SISTEM MANAJEMEN INFORMASI RUMAH SAKIT UMUM DAERAH NABIRE," MAHESA: MALAHAYATI HEALTH STUDENT JOURNAL, vol. 3, no. 10, pp. 3365-3374, 2023, doi: 10.33024/mahesa.v3i10.11246.
[17] I. Stepheng, C. A. Sari, E. H. Rachmawanto, and F. O. Isinkaye, "A Combination of Vigenere Cipher and Advanced Encryption Standard for Image Security," ASSET: International Journal of Advanced Science and Engineering Technology, vol. 5, no. 3, pp. 0230305-01~0230305-08, Oct. 2023, doi: 10.26877/asset.v5i3.17150.
[18] A. S. Joel, F. Abdussalaam, dan Y. Yunengsih, "Tata Kelola Rekam Medis Berbasis Teknologi Informasi dalam Penanganan Kerahasiaan dan Keamanan Data Pasien dengan Metode Kriptografi," Jurnal Ilmiah Manajemen Informatika dan Komunikasi, vol. 4, no. 3, pp. 12-24, Sep. 2023, doi: 10.35870/jimik.v4i3.287.
[19] H. Amri, A. A. Haryada, K. Abdi, dan A. Ikhwan, "Manajemen Resiko Keamanan Aset Informasi Pada Puskesmas Pancur Batu Tuntungan," Jurnal Informatika dan Teknologi, vol. 3, no. 1, pp. 10-19, Mar. 2023.
[20] R. S. A. Gusni, Kraugusteeliana, dan I. W. W. Pradnyana, "Analisis Tata Kelola Keamanan Sistem Informasi Rumah Sakit XYZ Menggunakan Cobit 2019 (Studi Kasus pada Rumah Sakit XYZ)," Jurnal Informatika dan Teknologi, vol. 5, no. 1, pp. 6-11, Jun. 2021.

[21] J. Wijaya, A. Megafitri, K. Khotimah, dan R. Astriratma, "Analisis dan Manajemen Risiko Keamanan Informasi pada Rumah Sakit Menggunakan Metode Octave Allegro (Studi Kasus: Rumah Sakit Umum Daerah Cengkareng)," Jurnal Informatika dan Teknologi, vol. 2, no. 2, pp. 13-25, Sept. 2021.
[22] A. T. Kurniawan, B. S. WA, dan A. Nasiri, "Analisis Architecture Teknologi Menggunakan SABSA untuk Meningkatkan Keamanan di Rumah Sakit Queen Latifa," Jurnal Informatika dan Teknologi, vol. 3, no. 2, pp. 6-11, Des. 2022.