Buletin Ilmiah Sarjana Teknik Elektro
Vol. 5, No. 3 September 2023, pp. 351-358
ISSN: 2685-9572, DOI: 10.12928/biste.v5i3.7719

Vulnerability Analysis and Prevention on Software as a Service (SaaS) of Archive Websites

Mushlihudin, Danis Faisal
Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

ARTICLE INFORMATION

Article History:

Submitted 17 February 2023
Revised 11 August 2023
Accepted 17 August 2023

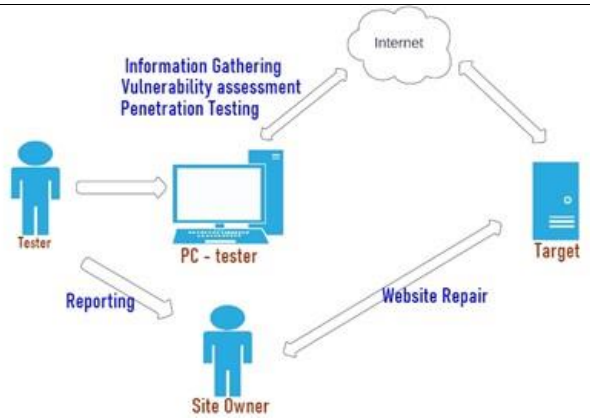
Keywords:

Cyber Security;
Penetration Testing;
Software as a Service;
Vulnerability Assessment;
Archive Website

Corresponding Author:

Mushlihudin,
Department of Informatics,
Universitas Ahmad Dahlan,
Yogyakarta, Indonesia.
Email:
mushlihudin@tif.uad.ac.id

ABSTRACT



Web Archive is a SaaS service that has an important role in providing better document storage and management. Good document management has a positive impact on optimizing business operations, increasing collaboration, reducing costs, and protecting sensitive information. Cybercrime, which has an increasingly high intensity, is a serious threat to the security of data stored in web archives. This research aims to improve data security on web archives by conducting ongoing testing. Testing was carried out on a server with a Linux operating system and web archives managed by a file manager system. This study tests the attack using the OWASP application method, and an XSS attack on a web archive with a Linux server and using a file management application. The testing phase includes Information Gathering, Vulnerability Assessment, Exploiting, and Reporting. Based on the results of the research, it was obtained that the first vulnerability test contained 9 vulnerabilities in 9 categories. The second vulnerability test obtained 7 vulnerabilities and the third test found no vulnerabilities. At the end of each test, recommendations for improvements to the web archive are made to the web archive manager and a re-testing process for vulnerabilities is carried out. This process is carried out repeatedly with continuous improvement. Testing the attack and repair of the web archive was carried out repeatedly and managed to get a vulnerability level of Level 0.1-3.9 points with Low status.

Document Citation:

M. Mushlihudin and D. Faisal, "Vulnerability Analysis and Prevention on Software as a Service (SaaS) of Archive Websites," *Buletin Ilmiah Sarjana Teknik Elektro*, vol. 5, no. 3, pp. 351-358, 2023, DOI: 10.12928/biste.v3i1.7719

1. INTRODUCTION

Web Archive is one of the SaaS services that has an important role in progress for organizations and companies in the digital era [1]. Web Archive helps manage documents and information in a more structured and efficient manner by category, date, or other attributes. Store documents in digital format, reducing the need for physical storage space. Besides that, with the Web Archive service, it can be accessed from anywhere and anytime, as long as it is connected to the internet [2].

The Web Archive service also makes it easy for users to share documents and collaborate effectively with fellow teams without geographical boundaries. In special circumstances, digital documents are equipped with strong security features, such as data encryption and access settings. This helps protect sensitive information from unauthorized access. For auditing purposes, Web Archive Services can provide audit trails, enabling organizations to track who accessed or changed documents.With fast access and regular document management, Web Archive services can improve operational efficiency in various aspects of the business.

Software as a Service is a model of providing software to users via the Internet. In this case, the user does not need to own the software or physically download and install the software on a computer or other device but can access and use the application through a web browser. SaaS aims to provide easy access to various types of software without the need for complicated installations. Likewise, users do not need to provide their own physical devices and operational costs, so that expenses can be reduced. Additionally, the SaaS Provider will be responsible for maintaining and updating the software.

The ease and efficiency that are obtained by using SaaS, there is a problem, namely the SaaS Provider manages user data, so there is a risk that sensitive data is stored by third parties. Security breaches by providers can result in information leaks. Reliance on the SaaS provider is also a problem if the service is suddenly unavailable or has a technical problem. Cybercrime, which has an increasingly high intensity, is a serious threat to the security of data stored in web archives.

Reducing the impact of cybercrime on data security requires penetration testing on SaaS to ensure the security and performance of the applications provided [7]. Several vulnerability tests can be carried out, including Vulnerability Scanning, Penetration Testing, Red Team Testing, Blue Team Testing, Third-Party Testing, Code Review, SQL injection, Cross-Site Scripting (XSS), and Social Engineering Testing [9][10] [11]. Penetration testing is an attempt to legally exploit a computer system with the aim of making the system secure [7]. Well-performed penetration testing can generate recommendations for addressing and fixing problems found during testing. Meanwhile, Penetration Testing is a series of processes containing procedures and techniques for evaluating the security of a computer system or network by carrying out attack simulations to find out where the vulnerability gaps are in the system so that these gaps are then closed or repaired.

Cross-Site Scripting (XSS) itself is the most common web application flaw [9]. An XSS flaw occurs when an application includes user-supplied data in a page sent to the browser, without validating or filtering the content. Cross-Site Scripting (XSS) attacks will be very uncomfortable for website visitors so that website visitors will be much reduced [10][11]. Therefore, improving the security of the website is very important to maintain the integrity of the website address.

The range of attacks based on XSS is almost unlimited, but usually includes sending private data, such as cookies or other session information, to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user's machine [12].

This study aims to improve data security on web archives by conducting ongoing testing. Testing was carried out on a server with a Linux operating system and web archives managed by a file manager system. Testing attacks using the OWASP application method, and XSS attacks on a web archive with a Linux server and using a file management application. The testing phase includes Information Gathering, Vulnerability Assessment, Exploiting, and Reporting [3].

2. METHODS

The Penetration Testing method is used for vulnerability testing [13], as well as to support data collection using the Literature Study, Observation, and Experiment methods. The literature study method aims to collect data by searching for various kinds of documents such as books and articles as well as from the literature from the final project that is interrelated with website security testing. The Observation Method carries out the process of observing, which is followed by recording the results of observations in an orderly manner. It consists of several elements that appear in the phenomena of the object under study. The results of the process are reported in a systematic report and in accordance with applicable rules.

This research went through several stages of research. The stages carried out in this study can be seen in the flowchart as shown in Figure 1.

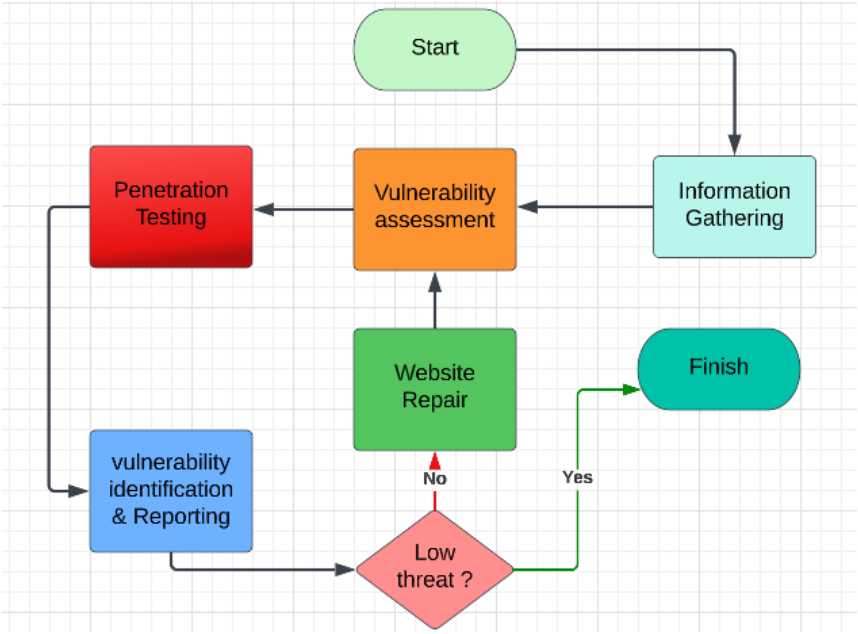


Figure 1. Stages of vulnerability testing and repair

2.1. Information Gathering

In Figure 1 all information related to the target system is collected. In the data collection process, information is divided into 2 parts, namely Passive Information Gathering and Active Information Gathering [14]. Information collection using passive information gathering techniques can use the WHOIS service [15], DNS [16], Qualys SSL Lab [16], Search Engine (Google), Website Security Analyst (Netcraft) [17] and can also use several tools such as Maltego [18], metabolic and traceroute. As for active information-gathering procedures, hackers usually use Port Scanning, Banner Grab, Fingerprinting, Network Mapping, and ARP Poisoning techniques [14].

2.2. Vulnerability Assessment

This process is a continuation of gathering. At this stage, analysis is carried out, identification of weaknesses that might be used for target exploitation. This process can use website vulnerability databases such as CVE Database [19], Inesecure, cve.mitre.org, xforce.iss.net, or you can also use software tools such as Nessus, Qualys, Fuzzing (Vunelrability Scanning Web Based), Nikto (Vunelrability Web Based Scanning), and WebInspect (Web Based Vulnerability Scanning).

2.3. Penetration Testing

The results of the previous stage that provide vulnerability then enter the next stage, namely the exploitation stage. At this process or stage, an attack attempt is carried out on the target. This stage looks for system security holes to be exploited further.

2.4. Vulnerability Identification

This process is a continuation of Information Gathering. At this stage, analysis is carried out, identification of weaknesses that might be used for target exploitation. This process can use database vulnerabilities websites such as CVE Database, Inesecure, cve.mitre.org, xforce.iss.net, or you can also use software tools such as Nessus, Qualys, Fuzzing (Web Based Vulnerability Scanning), Nikto (Vulnerability Scanning Web Based), and WebInspect (Web Based Vulnerability Scanning).

2.5. Reporting

The last is the Reporting stage, this stage is the last step on Penetration Testing. This stage contains a report on work steps that have been completed do beforehand on Exploiting, security hole reports and suggestions repair. Then next is the follow-up to improve the system.

2.6. Decision making

Reporting results will be evaluated whether it has a low threat or not. If it has a low threat, it will end the process. However, if the vulnerability is high, recommendations will be given for improving the website.

2.7. Test Scenario

The stages of Planning and Preparation are to prepare what is needed for research Figure 2. At this stage, the Cross-Site Scripting attack planning scenario is carried out manually and automatically using the Penetration Testing method or other tools that will be needed in the testing.

In the testing process that has been planned, an attack scenario was made against the Website which had become a test target that has been installed on the web server. An attacker is a tester who performs testing using the Penetration Testing method and Cross-Site Scripting attacks on targets. The attacker scans the web which is the object, then if gaps and threats are found, the attacker then conducts testing by sending certain scripts to the web browser which aims to provide interference. If the interference sent by the attacker is successful, then the web has a vulnerability.

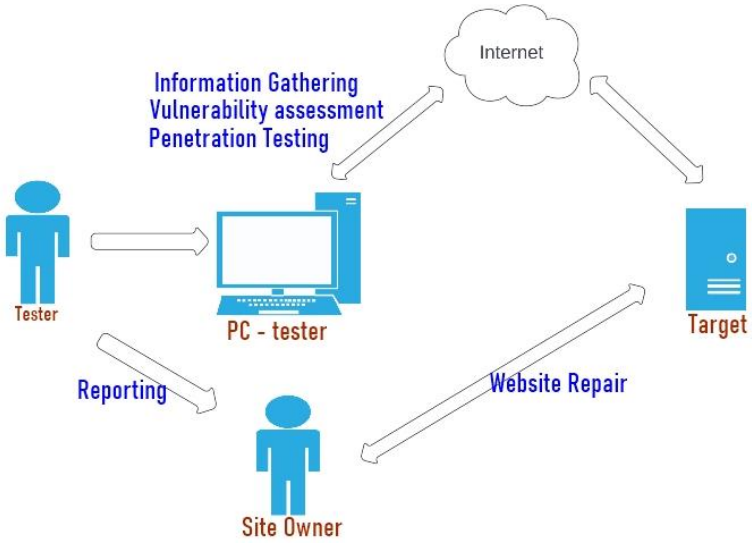


Figure 2. Test scenario

3. RESULT AND DISCUSSION

3.1. Information Gathering

The first stage in this research is information gathering. The author uses web applications such as Qualys SSL Lab and WhoIs to collect this information. Table 1 provides information on the results of the information gathering carried out.

Testing Qualys SSL Lab, the text states that as a result of the scanning process, it has been determined that the website is already implementing SSL security. SSL (Secure Sockets Layer) is a protocol that provides secure communication over the internet by encrypting data transmitted between a user's browser and the website's server. The text further mentions that the website is utilizing the SHA-256 with RSA signature algorithm. This is a cryptographic algorithm used for digital signatures and certificates. It ensures data integrity and authenticity by creating a unique hash value of the data and encrypting it with a private key.

WhoIs testing, the results of scanning on the WhoIs application show that the website arsip.bromonic.com is associated with the main domain, which is bromonic.com. The registration date for the domain is April 22, 2020, and the expiration date is April 22, 2023. The server's name associated with the domain is Centralops.net.

Table 1. Information *g*athering

Testing	Results
Qualys SSL Lab	In the results of the scanning that has been carried out, the result is that the website already uses SSL security. The website is also noted to have the SHA25withRSA signature algorithm.
WhoIs	The results of scanning on the WhoIs application show that the arsip.bromonic.com website has the main domain, namely bromonic.com, the registration date is 2020-04-22, and the expiration date is 2023-04-22, and the server name is Centralops.net.

3.2. Vulnerability Assessment

The next stage involves conducting a Vulnerability Assessment on the web that will be tested. In this test, the author utilizes the OWASP ZAP application. The results obtained from scanning the web using the OWASP ZAP application presented in Table 2. The scan results have been compiled into a table, as depicted in Figure 2, which lists threat vulnerabilities alongside the corresponding vulnerability levels for each threat.

Table 2. Threat classification and vulnerability levels

Classification	Risk Levels	Sum
CSP: Wildcard Directive	Medium	2
CSP: Script-src unsafe-inline	Medium	2
CSP: Style-src unsafe-inline	Medium	2
X-Content-Type-Option Header Missing	Low	8
Information Disclosure – Suspicious Comment	Informational	5
Re-examine Cache-control Directives	Informational	2

In Table 2 it can be explained, The vulnerability pertains to the Content Security Policy (CSP) with a wildcard directive. It has a medium risk level, indicating that it could potentially lead to security issues. The "2" suggests that there are two instances of this vulnerability. The points to a CSP vulnerability where the "script-src" directive allows unsafe inline scripting. Again, it has a medium risk level and two instances of occurrence. Similar to the previous one, the vulnerability relates to the "style-src" directive allowing unsafe inline styles. It also carries a medium risk level and has appeared twice. The vulnerability is about the missing X-Content-Type-Options header, which is considered a low-risk issue. The number "8" suggests eight instances of this vulnerability, making it more prevalent. The classification an informational finding rather than a critical vulnerability. It refers to potential information disclosure due to suspicious comments in the code. It's rated as "Informational" and has occurred five times. Finally, Classification, the need for re-evaluating Cache-Control directives. It is informational in nature and has been identified twice.

3.3. Penetration Testing

After completing a vulnerability assessment on the web, the subsequent step involves penetration testing. This testing phase is executed on the website, based on the findings obtained from the previous scanning process. The threats identified in the threat categories are subjected to rigorous testing to determine if there are vulnerabilities present. The outcomes of these tests are documented and can be observed in Table 3.

In the testing across various categories, including CSP directives and content-related headers, the outcomes indicate that the applied measures were not successful in addressing the identified vulnerabilities. This highlights the need for further attention and corrective actions to fortify the system's security posture.

Table 3. Penetration Testing Result

Categories	Testing Result
CSP: Wildcard Directive	Not successful
CSP: Script-src unsafe-inline	Not successful
CSP: Style-src unsafe-inline	Not successful
X-Content-Type-Option Header Missing	Not successful
Information Disclosure – Suspicius Comment	Not successful
Re-examine Cache-control Directives	Not successful

3.4. Reporting

The results obtained after testing are compiled into a report that includes the test outcomes, along with solutions and recommendations offered by the testing team for the previously assessed web application [20]. The report is presented in Table 4.

In Table 4 it can be explained, the solution involves ensuring proper configuration of the website server, application server, and related components to establish accurate Content Security Policy (CSP) headers. Additionally, it's crucial to activate the CSP feature on the hosting provider's cPanel. This helps mitigate potential security risks associated with wildcards in CSP.

For CSP, Script-src unsafe-inline, similar to the previous category, the recommended approach includes proper configuration of the website and application servers for accurate CSP headers. Activating the CSP feature on the hosting provider's cPanel is also emphasized. These measures collectively enhance security by addressing unsafe inline scripting risks.

In the X-Content-Type-Option Header Missing category, the recommended course of action is to ensure proper configuration of the application or web server's Content-Type header and the X-Content-Type-Options header. Specifically, the X-Content-Type-Options header should be set to 'nosniff' for all web pages. Additionally, if feasible, encouraging users to utilize modern, standards-compliant web browsers that do not engage in MIME-sniffing, or instructing the web application not to perform MIME-sniffing, can provide an additional layer of security.

In the Information Disclosure – Suspicious Comment category, the recommended approach involves removing all comments within scripts that expose information potentially beneficial to attackers. Additionally, it is vital to address the underlying issues to which these comments allude. By eliminating comments that could

potentially reveal sensitive details and by resolving the fundamental problems they indicate, the overall security posture of the web application is strengthened.

In the context of Re-examine Cache-control Directives, it is advisable to establish the cache-control HTTP header as 'no-cache, no-store, must-revalidate' for content necessitating heightened security. When assets need to be cached, it is prudent to contemplate employing the 'public, max-age, immutable' directive. These cache-control configurations collectively play a pivotal role in fortifying the protection of sensitive content and elevating the overall security stance.

Table 4. Reporting

Category	Solutions and Recommendations
CSP: Wildcard Directive	Make sure the website server, application server etc. Configured properly to set Content Security Policy headers. Also make sure that the CSP feature on the hosting provider's cPanel is Active.
CSP: Script-src unsafe-inline	Make sure the website server, application server etc. Configured properly to set Content Security Policy headers. Also make sure that the CSP feature on the hosting provider's cPanel is Active.
X-Content-Type-Option Header Missing	Make sure the application/web server sets the Content-Type header correctly, and sets the X-Content-Type-Options header to 'nosniff' for all web pages. If possible, ensure that the end user is using a modern, standards-compliant web browser that doesn't perform MIME-sniffing at all, or that the web application/webserver can instruct not to perform MIME-sniffing.
Information Disclosure – Suspicious Comment	Remove all comments on scripts that return information that could help an attacker and fix the underlying problem they refer to.
Re-examine Cache-control Directives	For safe content, make sure the cache-control HTTP header is set "no-cache, no-store, must-revalidate". If an asset must be cached, consider setting a "public, max-age, immutable" directive.

The report findings are communicated to the owner of the archive website, and subsequent to implementing enhancements, a reevaluation is conducted using the same protocol. The testing and scanning process is iterated, employing OWASP TOP 10-2017 as a benchmark. Table 5 presents a comprehensive compilation of vulnerabilities alongside corresponding test outcomes.

Table 5. Vulnerabilities and test results

Category	Vulnerability	Test Results
A1:2017-Injection	Not found	Not successful
A2:2017-Broken Authentication	Not found	Not successful
A3:2017-Sensitive Data Exposure	Not found	-
A4:2017-XML External Entities (XXE)	Not found	-
A5:2017-Broken Access Control	Not found	-
A6:2017-Security Misconfiguration	Not found	-
A7:2017-Cross-Site Scripting (XSS)	Not found	Not successful
A8:2017-Insecure Deserialization	Not found	-
A9:2017-Using Components with Known Vulnerabilities	Not found	-
A10:2017-Insufficient Logging & Monitoring	Not found	-

The Table 5 provides an overview of the outcomes of the assessment, indicating the presence or absence of vulnerabilities according to the OWASP TOP 10-2017 criteria. "Not Successful" denotes instances where vulnerabilities were detected.

4. CONCLUSIONS

Conducting Penetration Testing on the Archive website utilizing the OWASP Top-10 2017 guidelines as a testing parameter unveiled noteworthy insights regarding the site's vulnerabilities. The vulnerabilities observed on the Archive website encompassed multiple threat categories, primarily manifested as relatively low-risk vulnerabilities across six distinct categories. Moreover, the presence of an Information Disclosure vulnerability was identified, with disclosed comments within scripts potentially aiding malicious actors. Notably, the evaluation using a Cross-Site Scripting attack did not reveal any vulnerabilities within the web archive. The Penetration Test approach, featuring its comprehensive framework of experimentation, emerges as a valuable method for uncovering and addressing vulnerabilities in web applications, enhancing overall security resilience. Numerous opportunities for enhancing the Archive website's security ahead, encompassing the exploration of advanced testing scenarios, the implementation of mitigation strategies, continuous

monitoring and remediation practices, the incorporation of automation, and the development of a comprehensive threat model tailored specifically to the Archive website. These potential collectively foster a proactive security approach, adept at anticipating potential threats and vulnerabilities.

REFERENCES

[1] S. Aheleroff, N. Mostashiri, X. Xu, and R. Y. Zhong, "Mass personalisation as a service in industry 4.0: A resilient response case study," *Advanced Engineering Informatics*, vol. 50, p. 101438, 2021, https://doi.org/10.1016/j.aei.2021.101438.

[2] N. Bruno and R. Roncella, "HBIM for conservation: A new proposal for information modeling," *Remote Sensing*, vol. 11, no. 15, p. 1751, 2019, https://doi.org/10.3390/rs11151751.

[3] V. K. Malviya, S. Rai, and A. Gupta, "Development of web browser prototype with embedded classification capability for mitigating Cross-Site Scripting attacks," *Applied Soft Computing*, vol. 102, p. 106873, 2021, https://doi.org/10.1016/j.asoc.2020.106873.

[4] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," *Journal of Big data*, vol. 6, no. 1, pp. 1-21, 2019, https://doi.org/10.1186/s40537-019-0268-2.

[5] A. Ghasempour, "Internet of things in smart grid: Architecture, applications, services, key technologies, and challenges," *Inventions*, vol. 4, no. 1, p. 22, 2019, https://doi.org/10.3390/inventions4010022.

[6] R. Huang, A. Tlili, T. W. Chang, X. Zhang, F. Nascimbeni, and D. Burgos, "Disrupted classes, undisrupted learning during COVID-19 outbreak in China: application of open educational practices and resources," *Smart Learning Environments*, vol. 7, pp. 1-15, 2020, https://doi.org/10.1186/s40561-020-00125-8.

[7] M. Agreindra Helmiawan, E. Firmansyah, I. Fadil, Y. Sofivan, F. Mahardika, and A. Guntara, "Analysis of Web Security Using Open Web Application Security Project 10," *2020 8th International Conference on Cyber and IT Service Management (CITSM)*, pp. 1-5, 2020, https://doi.org/10.1109/CITSM50537.2020.9268856.

[8] M. Singh, P. Singh, and P. Kumar, "An Analytical Study on Cross-Site Scripting," *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, pp. 1-6, 2020, https://doi.org/10.1109/ICCSEA49143.2020.9132894.

[9] A. A. R. Farea, C. Wang, E. Farea, and A. Ba Alawi, "Cross-Site Scripting (XSS) and SQL Injection Attacks Multi-classification Using Bidirectional LSTM Recurrent Neural Network," *2021 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pp. 358-363, 2021, https://doi.org/10.1109/PIC53636.2021.9687064.

[10] F. A. Mereani and J. M. Howe, "Detecting cross-site scripting attacks using machine learning," In *International conference on advanced machine learning technologies and applications*, pp. 200-210, 2018, https://doi.org/10.1007/978-3-319-74690-6_20.

[11] G. E. Rodríguez, J. G. Torres, P. Flores, and D. E. Benavides, "Cross-site scripting (XSS) attacks and mitigation: A survey," *Computer Networks*, vol. 166, p. 106960, 2020, https://doi.org/10.1016/j.comnet.2019.106960.

[12] D. Korać, B. Damjanović, D. Simić, and K. K. R. Choo, "A hybrid XSS attack (HYXSSA) based on fusion approach: Challenges, threats and implications in cybersecurity," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 9284-9300, 2022, https://doi.org/10.1016/j.jksuci.2022.09.008.

[13] S. Siboni *et al*., "Security Testbed for Internet-of-Things Devices," in *IEEE Transactions on Reliability*, vol. 68, no. 1, pp. 23-44, 2019, https://doi.org/10.1109/TR.2018.2864536.

[14] A. Arora, P. M. Furlong, R. Fitch, S. Sukkarieh, and T. Fong, "Multi-modal active perception for information gathering in science missions," *Autonomous Robots*, vol. 43, pp. 1827-1853, 2019, https://doi.org/10.1007/s10514-019-09836-5.

[15] Y. Cheng, Y. Liu, G. Piao, Z. Zhang, N. Li, and M. Qiu, "Disclosing Features and Characteristics of. COM Registrars' WHOIS Servers," *2022 2nd Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*, pp. 52-56, 2022, https://doi.org/10.1109/ACCTCS53867.2022.00018.

[16] W. Zhang, J. Chen, Y. Kuo, and Y. Zhou, "Transmit Beamforming for Layered Physical Layer Security," in *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9747-9760, 2019, https://doi.org/10.1109/TVT.2019.2932753.

[17] P. Ranade, A. Piplai, S. Mittal, A. Joshi, and T. Finin, "Generating Fake Cyber Threat Intelligence Using Transformer-Based Models," *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-9, 2021, https://doi.org/10.1109/IJCNN52387.2021.9534192.

[18] B. A. Mozzaquatro, C. Agostinho, D. Goncalves, J. Martins, and R. Jardim-Goncalves, "An ontology-based cybersecurity framework for the internet of things," *Sensors*, vol. 18, no, 9, p. 3053, 2018, https://doi.org/10.3390/s18093053.

[19] G. J. Blinowski and P. Piotrowski, "CVE based classification of vulnerable IoT systems," In *Theory and Applications of Dependable Computer Systems: Proceedings of the Fifteenth International Conference on Dependability of Computer Systems DepCoS-RELCOMEX, June 29–July 3, 2020, Brunów, Poland 15*, pp. 82-93, 2020, https://doi.org/10.1007/978-3-030-48256-5_9.

[20] S. S. Malladi and H. C. Subramanian, "Bug Bounty Programs for Cybersecurity: Practices, Issues, and Recommendations," in *IEEE Software*, vol. 37, no. 1, pp. 31-39, 2020, https://doi.org/10.1109/MS.2018.2880508.

AUTHOR BIOGRAPHY

Mushlihudin is a Lecturer at Department of Informatic, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia.

Danis Faisal is a student from Department of Informatic, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia.