

A real-time hypertext transfer protocol intrusion detection system on web server

Agus Tedyyana¹, Osman Ghazali², Onno W. Purbo³

¹Department of Informatic Engineering, Politeknik Negeri Bengkalis, Riau, Indonesia

²School of Computing, University Utara Malaysia, Sintok, Kedah, Malaysia

³Department of Informatic, Institute Technology Tangerang Selatan, Tangerang Selatan, Indonesia

Article Info

Article history:

Received Jun 05, 2021

Revised Mar 31, 2022

Accepted Jun 01, 2022

Keywords:

Cyber attack
HTTP IDS
Intrusion
OWASP ZAP
Telegram Bot

ABSTRACT

Behind the rapid development of the internet in today's era, various types of crime are also targeting vital players in the internet industry. With many online crime types rampant, an antidote is also needed to suppress internet crime. Therefore, the researcher proposes a solution in the form of a Keris, namely the hypertext transfer protocol (HTTP) intrusion detection system (IDS) that runs on the server where our website is running. Keris works by detecting rampant intrusions that attack servers or websites. When an intrusion is detected, Keris will notify the system admin using the Keris Telegram chatbot or an alternative Keris mobile application with firebase cloud messaging (FCM) technology. The research was conducted by comparing the results of one-way delay (OWD) between Telegram Webhook and FCM with the help of the open web application security project (OWASP) zed attack proxy (ZAP) test tool. From the results of the tests, OWD against directory brute force attacks on Telegram Webhook for 0.72 seconds and on FCM for 0.44 seconds. In this case, FCM is more suitable for real-time notifications if we need a very responsive notification.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Agus Tedyyana
Department of Informatics Engineering, Politeknik Negeri Bengkalis
Bengkalis, Riau, Indonesia
Email: agustedyyana@polbeng.ac.id

1. INTRODUCTION

With the rapid growth of the internet lately, the internet has become a much-needed thing and shifts old habits to be fast and easy. As time goes on, it is not only a communication tool or research tool, but now the government is also using the internet in e-government to provide services to its citizens, and this trend is running smoothly. Governments continue to use the internet to become more transparent about governance data. Then educational institutions and researchers also use the internet as a tool for solving problems and as a platform to collaborate and disseminate their findings and research with others. Meanwhile, companies use the internet to exchange information with business units, partners, suppliers, and customers efficiently and seamlessly. With such adoption, it is becoming clear that many elements are increasingly dependent on the internet and network-based information systems [1].

Various types of attacks are used as tools to commit cybercrime. These attacks may be classified into several categories, such as:

- Common web attack is a criminal attack that is generally done against web-based applications. The most common are path/directory traversal [2], cross-site scripting (XSS) [3], local file inclusion (LFI) [4], SQL injection (SQLi) [5], and distributed denial of service (DDoS) [6].

- b) Common vulnerabilities and exposures (CVE) is a system that gives many reference methods for commonly known vulnerabilities and exposure information in the form of data sets. This system can be used as a reference to protecting existing systems but can also be a reference for attackers to attack systems [7].
- c) Bad crawler is a bot used for criminal acts, such as illegally collecting (memo/crawler) website content or taking email addresses listed on websites to send spam to victims. In addition, there are also types of bots used to perform DDoS actions, which use up bandwidth and hurt website owners.
- d) Directory brute force is a frequent attack on web servers and websites.

A system that can detect suspicious activity in a system or network is known as intrusion detection [8]. Intrusion detection systems (IDS) are security tools that use sophisticated techniques to secure computer networks from illegal access or ongoing intrusions that have already occurred. Therefore, cybersecurity requires a robust IDS design, which can overcome ethical testing in real-time and large-scale hacking [9]. IDS consists of various types, such as network-based IDS (NIDS), WirelessIDS, and host-based IDS (HIDS), and there is also a hybrid IDS that combines various IDS [10].

In intrusion detection, web-based applications typically use weblogs from a web server to detect intrusion. In the research by Tanaka *et al.* [11], they created a model to detect bots that enter the website by utilizing the user-agent and user behavior recorded in the web server log. From the results of this web log analysis, admins can have insight into potential attack patterns that will occur in the future [12].

There are some significant challenges in building an IDS that runs on a web server. Intrusion detection methodologies are still immature in the web application security domain [13]. Tracking systems are mainly used as network security tools. However, IDS web design requires a different approach from traditional network IDS to address the complexities associated with web-based applications.

In this research, the researcher proposes an IDS in the form of hypertext transfer protocol (HTTP) IDS, namely Keris. Keris is a web log-based alerts IDS that runs on the terminal with resources collected and detects intrusion, and, as soon, returns the alert to server admin using notification. Analyzing the ongoing weblog will alert the admin in real time if any suspicious activity is recorded in the weblog. The researchers in this study hope to get the results of intrusion notifications that can be sent in real-time with the shortest available delay time. This IDS was built using the Go programming language. One of the advantages when using the go language in programming is that it does not use a lot of random access memory (RAM) [14]. In this research, Keris still uses external resources to detect intrusions into the system. For example, for types of attacks such as common web attack using PHP-IDS [15] resources, CVE uses nuclei templates [16]. Wrong IP address and bad referrer is obtained from the Nginx Ultimate Bad Bot Blocker [17] collection, and bad crawler takes from project resources from crawler detect [18]. The last one is directory bruteforce, leveraging resources from dirsearch [19].

2. METHOD

As described in the previous section, HTTP-IDS processes logs to detect intrusions into the system. HTTP-IDS collects logs from web servers and analyzes them using algorithms that can detect patterns that are abnormal or do not comply with specified standards. If the pattern found matches the signs of an attack, the system will give a warning. This section will explain how logs on the web server are used in the process of infiltrating the system, how the Keris system works, and how the Keris system uses external resources to help detect whether intrusion has occurred or not.

2.1. Proposed mechanism design

The output obtained from this stage is the design of the workings of the IDS Keris to identify the types of threats and malicious activities to be detected. Once a threat or unwanted activity is detected, the IDS system will notify the admin or take automatic action such as blocking access. It then tests the IDS to ensure that it is functioning properly and is effective in detecting the type of threat of concern. Figure 1 shows the following flowchart which will explain how the intrusion process is detected and reported to the system admin.

The flow chart illustrates how the Keris HTTP IDS workflow developed in this study. The request from the user will go to the web server that is used as the hosting of the web application. When the HTTP request comes in, information on the activity carried out by the client will be recorded in the web-server log. The logs will then be matched with external resources to check whether the incoming request was classified as an intrusion. Suppose an incoming HTTP request is declared suspicious by IDS based on these resources. In that case, the system will send a warning notification to the system admin about an intrusion attempt on the web server.

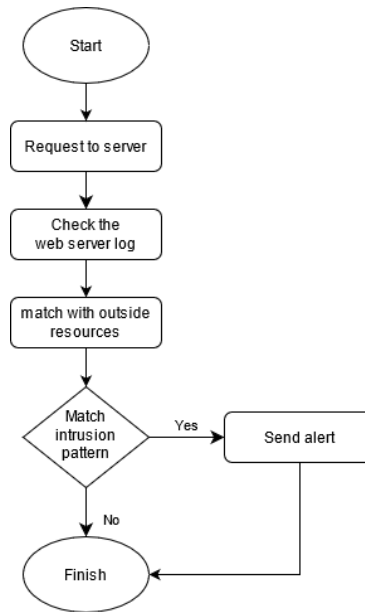


Figure 1. Flowchart of the Keris HTTP IDS

2.2. Use of web server logs on the intrusion process

Web servers include several parts that control how web users access hosted files. The minimum is an HTTP server. An HTTP server understands URLs (web addresses) and HTTP (protocols used by your browser to view web pages) [20], and a web server log file is written as an activity generated by a web server. Log files collect various data about information requests to a web server. This web server log data will then be analyzed using the Keris-IDS program with the help of resources used from outside, such as PHP-IDS, nuclei templates, and others.

2.3. Rule method used in the HTTP-IDS Keris

As explained in the previous section, getting system scripts to run by analyzing web server access log files. By examining past data, it is important to regularly review and analyze log files from IDS to ensure that systems are secure and to identify potential threats or vulnerabilities. Information security policies for web applications can be created and implemented properly [21]. In addition, further exploitation can be prevented in advance.

The proposed method can be described as a rule-based system. Unlike detection based on anomalies, the rules are static, i.e., using a blacklist and whitelist approach. A rule-based system is a detection based on a database of known intrusion or attack alerts crafted by humans based on their expert knowledge [22]. The database used here is an outside resource explained in the previous section. If IDS detects an intrusion, then according to the data from the database, immediate detection is categorized as an intrusion.

2.4. Intrusion warning mechanism

There are various ways to warn users if an intrusion is detected in the IDS. They were starting from short message service (SMS) gateway, email, or chatbot. Chatbots can be developed in any programming language. The backend receives the message, figures out what to answer, and returns a response to the user using a web application programming interface (API) from the chatbot service provider. In this case, the API is provided to developers in the public domain [23].

The use of chatbots as an intrusion warning medium is now widely used as its implementation is easy, as it is spoiled with documentation from existing developers and communities. For example, a Telegram Bot uses long polling and webhook to run a Telegram Bot. However, the long polling method has a higher (slower) response time than the webhook method [24].

In addition to the Telegram chatbot, there is another alternative as a delivery mechanism for intrusion detection, namely using, firebase cloud messaging (FCM) technology. The FCM architecture is presented in Figure 2. FCM supports notification and data messages [25]. The picture shows that FCM is the center that sends notification messages from application servers in the form of HTTP and extensible messaging and presence protocol (XMPP) and several functions that run in the cloud to client applications.

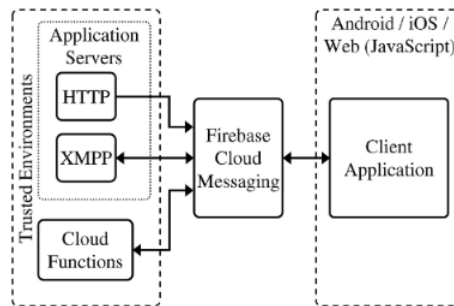


Figure 2. The FCM architecture of the Keris HTTP IDS

3. RESULTS AND DISCUSSION

The results of an IDS can vary depending on the type of IDS and the specific threats it is designed to detect. When the IDS detects a potential threat or malicious activity, it can issue an alert to the system administrator. The alert will typically include information about the nature of the threat and the actions taken by the IDS to mitigate it. This section describes the results of the research conducted along with the results of the application test analysis which are divided into three sub-sections.

3.1. Keris

The result of the Keris application is an HTTP-IDS built using the go programming language. Keris is installed on a server which is the base of the website. Keris is an application built without a graphical user interface, so to run it, the user has to use commands in the terminal. Keris can handle websites that are deployed on Apache and Nginx web servers. Figure 3 displays how Keris runs and reads the logs on the webserver to determine whether there is an intrusion.

```

KERIS
1094fd4
agustedyana@polbeng.ac.id
[INF] Analyzing...
[23/May/2021:21:42:27 +0800] [23.129.64.231] [Bad IP Address] 23.129.64.231
[23/May/2021:23:56:00 +0800] [203.186.25.21] [Bad Crawler] Go-http-client/1.1
[23/May/2021:23:15:38 +0800] [162.62.123.46] [Bad IP Address] 162.62.123.46
[24/May/2021:00:17:57 +0800] [193.118.53.210] [Bad IP Address] 193.118.53.210
[24/May/2021:00:20:43 +0800] [182.1.61.79] [Directory Bruteforce] /favicon.ico

```

Figure 3. Keris running on Nginx webserver

Figure 3 explains how the Keris looks when it is run through the terminal. When the user runs it for the first time, Keris will present the user with a logo of the application, and the contact user can address it. The below shows the analysis process done by Keris by reading the logs from the web server. The results of the analysis data look like the format in the figure, namely the date and time of the log are written, in this case, the time the intrusion occurred along with the time zone. Then there is the IP address of the client who intruded. Next is the type of intrusion carried out by the client. Moreover lastly, the data is a description of the information about the intrusion. For example, the bad IP address will undoubtedly display the bad IP address, and the bad crawler will display the type of crawler used for the intrusion.

In this study, researchers used the Nginx web server as test material. Other web-server options can be used, such as Apache. We only need to change the configuration file according to the web-server used. Configuration needs to be done, considering the format of the logs for each web-server is different. For example, Nginx uses the following format "\$ remote_addr \$ remote_user - [\$ time_local] "\$ request_method \$ request_uri \$ request_protocol" \$ status \$ body_bytes_sent "\$ http_referer" "\$ http_user_agent" while Apache has a slight difference in the use of "-", for example, "\$ remote_addr - \$ remote_user [\$ time_local] "\$ request_method \$ request_uri \$ request_protocol" \$ status \$ body_bytes_sent "\$ http_referer" "\$ http_user_agent". So we need a configuration according to the web-server before running the Keris application on the server.

3.2. Keris Telegram Bot

Not always web admins or system admins work in front of their computers to monitor intrusions from outside. Therefore, we also developed a Telegram Bot to meet the needs of monitoring for 24 hours with its real-time notification feature. First, we provide a template to present information that is easy to understand for the web admin. Then enter the token from the Telegram Bot that Telegram Bot created and the chat id from our account into the existing configuration file. Figure 4 shows how Keris Telegram Bot displays the information.



Figure 4. Keris Telegram Bot

This image illustrates how a bot will notify a web admin via an informative notification. Bots can send push notifications through the application or platform used by the admin, so that the admin can find out if there is a notification from the IDS system. The information sent includes the incoming intrusion type, HTTP request type, attacker's IP address, user-agent used, HTTP status response code from server to client, and bytes sent from server to client. The admin must plan the work logic of the bot, such as what conditions will cause the bot to send notifications, the message format to be received, and who will receive the notification.

3.3. Keris client app

Apart from using the Telegram chatbot, researchers also use FCM to examine which delay rate is lower between FCM and Telegram Webhooks. FCM is a messaging solution developed by Google that is used to send real-time messages to mobile and web applications. FCM allows developers to send messages to apps installed on a user's device without using an internet connection or mobile network as a notification recipient from FCM. FCM also allows sending messages to a single device, group of devices, or the entire application user base. An Android application is built that is useful as a client to display the contents of intrusion detection messages, as shown in Figure 5.

An IDS can generate reports that provide a summary of the detected threats and the actions taken to mitigate them. These reports are useful for monitoring the system's security over time and identifying trends or patterns in malicious activity. Figure 5 displays the application interface, which shows basic information such as the type of intrusion and the time of the intrusion down to the second format.



Figure 5. Keris client app

3.4. Performance metric

In order to analyze the real-time performance of the HTTP IDS used, the researcher uses the one-way delay (OWD) metric for the two types of intrusion-sending mechanisms used, namely Telegram Webhook and FCM. OWD is based on the time taken for a packet to be transmitted across a network from the source, which means the HTTP IDS to the destination, in this case, is the client. In testing the application in this research, the researcher used the open web application security project (OWASP) zed attack proxy (ZAP) tool to launch attacks on the website. The researcher chose OWASP ZAP because it has proven reliable in intrusion testing [26]. From the test results, the researcher then compares the time between attacks detected by Keris and the time the warning notification appears on the Telegram Bot to determine whether Keris could generate real-time alerts when attacking using the OWASP ZAP tool, as in Table 1.

Table 1. Results of OWD using Telegram Webhook

No	Attack type	Time of detect	Time of alert	Time difference(s)
1	Directory bruteforce	00:24:55	00:24:55	0
2	Directory bruteforce	00:24:55	00:24:55	0
3	Directory bruteforce	00:25:03	00:25:04	1
4	Directory bruteforce	00:25:11	00:25:11	0
5	Directory bruteforce	00:25:11	00:25:12	1
6	Directory bruteforce	00:25:12	00:25:12	0
7	Directory bruteforce	00:25:12	00:25:12	0
8	Directory bruteforce	00:25:12	00:25:13	1
9	Directory bruteforce	00:25:14	00:25:15	1
10	Directory bruteforce	00:25:14	00:25:15	1
11	Directory bruteforce	00:40:35	00:40:36	1
12	Directory bruteforce	00:40:42	00:40:43	1
13	Directory bruteforce	00:40:42	00:40:43	1
14	Directory bruteforce	00:40:43	00:40:44	1
15	Directory bruteforce	00:40:43	00:40:44	1
16	Directory bruteforce	00:40:43	00:40:44	1
17	Directory bruteforce	00:40:45	00:40:46	1
18	Directory bruteforce	00:40:45	00:40:46	1
The difference in time average				0.72

From the results in the previous table, it is found that of the last 18 attacks detected using OWASP ZAP, there is an average time difference of 0.72 seconds. From the results in the previous table, it was found that in the last 18 attacks detected using OWASP ZAP, the average time difference was 0.72 seconds. It can be interpreted that there is a delay of 0.72 to 1 second to send a notification message to Telegram since Keris detects an attack. After testing on the Telegram Webhook, the next researcher investigated using FCM. The results of OWD data on FCM are shown in Table 2, which shows that the delay rate on FCM is lower than on Telegram Webhook. Sama amount of data with the Telegram hook, the FCM has an average time difference of 0.44 seconds.

Table 2. Results of OWD using FCM

No	Attack type	Time of detect	Time of alert	Time difference(s)
1	Directory bruteforce	02:16:24	02:16:24	0
2	Directory bruteforce	02:17:31	02:17:32	1
3	Directory bruteforce	02:17:31	02:17:32	1
4	Directory bruteforce	02:17:32	02:17:32	0
5	Directory bruteforce	02:17:32	02:17:32	0
6	Directory bruteforce	02:18:53	02:18:54	1
7	Bad IP address	02:20:55	02:20:56	1
8	Directory bruteforce	02:24:43	02:24:43	0
9	Directory bruteforce	02:24:43	02:24:43	0
10	Directory bruteforce	02:24:43	02:24:43	0
11	Directory bruteforce	02:24:43	02:24:43	0
12	Directory bruteforce	02:26:40	02:26:41	1
13	Directory bruteforce	02:26:41	02:26:41	0
14	Directory bruteforce	02:26:41	02:26:41	0
15	Directory bruteforce	02:26:41	02:26:41	0
16	Directory bruteforce	02:28:15	02:28:16	1
17	Directory bruteforce	02:28:15	02:28:16	1
18	Directory bruteforce	02:28:15	02:28:16	1
The difference in time average				0.44




4. CONCLUSION

From the results and analysis of the application testing that has been carried out, it is found that the proposed HTTP-IDS Keris has successfully detected intrusions carried out by OWASP ZAP by utilizing external resources to detect the types of incoming attacks. From the OWASP ZAP test, the type of intrusion was directory bruteforce. Keris can also alert the system admin about the dangers of intrusion in real-time with a delay of 0.72 seconds using the Telegram Webhook and 0.44 seconds when using the FCM. This show that FCM is more suitable than Telegram Webhook when we want to use real-time notification. For future research, it is hoped that other technologies can be used as warning mechanisms to overcome the causes of delays and then reduce delay times. Using ensemble techniques can strengthen the performance of the detection system, by combining two or more detection methods, for example signature-based and anomaly-based methods. IDS is capable of regularly updating models and reviewing logs for potential problems detection. In addition, Research can also be carried out by utilizing the latest knowledge, to maximize the capabilities of IDS.




REFERENCES

- [1] S. Q. Mir, I. A. Mir, and B. M. Beigh, "Investigating the denial of service attack : A major threat to internet and the security of information," *JK Research Journal in Mathematics and Computer Sciences*, vol. 1, no. 1, pp. 121–131, 2018. [Online]. Available: <https://www.jkhighereducation.nic.in/jkrjcms/issue1/17.pdf>
- [2] I. -C. Potintue and R. Varga, "Detecting Injection Attacks using Long Short Term Memory," *2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2020, pp. 163–169, doi: 10.1109/ICCP51029.2020.9266177.
- [3] S. Rathore, P. K. Sharma, and J. H. Park, "XSSClassifier: An efficient XSS attack detection approach based on machine learning classifier on SNSs," *Journal of Information Processing Systems*, vol. 13, no. 4, pp. 1014–1028, 2017, doi: 10.3745/JIPS.03.0079.
- [4] M. A. Rahman, M. Amjad, B. Ahmed, and M. S. Siddik, "Analyzing web application vulnerabilities: An empirical study on e-commerce sector in Bangladesh," *ICCA 2020: Proceedings of the International Conference on Computing Advancements*, pp. 1–6, 2020, doi: 10.1145/3377049.3377107.
- [5] B. A. Pham and V. H. Subburaj, "An Experimental setup for Detecting SQLI Attacks using Machine Learning Algorithms," *Journal of The Colloquium for Information Systems Security Education*, vol. 8, no. 1, 2020, [Online] Available: <https://cisse.info/journal/index.php/cisse/article/view/124/124>
- [6] S. Haider *et al.*, "A Deep CNN Ensemble Framework for Efficient DDoS Attack Detection in Software Defined Networks," in *IEEE Access*, vol. 8, pp. 53972–53983, 2020, doi: 10.1109/ACCESS.2020.2976908.
- [7] J. Fan, Y. Li, S. Wang, and T. N. Nguyen, "A C/C++ Code Vulnerability Dataset with Code Changes and CVE Summaries," *MSR '20: Proceedings of the 17th International Conference on Mining Software Repositories*, pp. 508–512, 2020, doi: 10.1145/3379597.3387501.
- [8] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017, doi: 10.1016/j.jnca.2017.02.009.
- [9] A. S. Sohal, R. Sandhu, S. K. Sood, and V. Chang, "A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments," *Computers & Security*, vol. 74, pp. 340–354, 2018, doi: 10.1016/j.cose.2017.08.016.
- [10] S. Jose, D. Malathi, B. Reddy, and D. Jayaseeli, "A Survey on Anomaly Based Host Intrusion Detection System," *Journal of Physics: Conference Series*, 2018, vol. 1000, doi: 10.1088/1742-6596/1000/1/012049.
- [11] T. Tanaka, H. Niibori, S. Li, S. Nomura, H. Kawashima, and K. Tsuda, "Bot detection model using user agent and user behavior for web log analysis," *Procedia Computer Science*, vol. 176, pp. 1621–1625, 2020, doi: 10.1016/j.procs.2020.09.185.
- [12] K. Ma, R. Jiang, M. Dong, Y. Jia, and A. Li, "Neural Network Based Web Log Analysis for Web Intrusion Detection," *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, 2017, pp. 194–204, doi: 10.1007/978-3-319-72395-2_19.
- [13] N. Agarwal and S. Z. Hussain, "A Closer Look at Intrusion Detection System for Web Applications," *Hindawi: Security and Communication Networks*, vol. 2018, pp. 1–27, 2018, doi: 10.1155/2018/9601357.
- [14] P. Dymora and A. Paszkiewicz, "Performance analysis of selected programming languages in the context of supporting decision-making processes for industry 4.0," *Applied Sciences*, vol. 10, no. 23, 2020, doi: 10.3390/app10238521.
- [15] M. Chora and R. Kozik, "Machine Learning Techniques for Threat Modeling and Detection," *Security and Resilience in Intelligent Data-Centric Systems and Communication Networks*, Academic Press, pp. 179–192, 2018, doi: 10.1016/B978-0-12-811373-8.00008-2.
- [16] "Nuclei - Community Powered Vulnerability Scanner." Nuclei Project. <https://nuclei.projectdiscovery.io/templating-guide/> (accessed May 05, 2021).
- [17] M. Krogza. "Nginx Ultimate Bad Bot and User-Agent Blocker." GitHub. <https://github.com/mitchellkrogza/nginx-ultimate-bad-bot-blocker> (accessed May 05, 2021).
- [18] J. Bizzle. "Crawler Detect." GitHub. <https://github.com/JayBizzle/Crawler-Detect> (accessed May 06, 2021).
- [19] M. Soria. "Dirsearch." GitHub. <https://github.com/maurosoria/dirsearch> (accessed May 06, 2021).
- [20] G. Suchacka, A. Cabri, S. Rovetta, and F. Masulli, "Efficient on-the-fly Web bot detection," *Knowledge-Based Systems*, vol. 223, 2021, doi: 10.1016/j.knosys.2021.107074.
- [21] M. B. Seyyar, F. Ö. Çatak, and E. Gül, "Detection of attack-targeted scans from the Apache HTTP Server access logs," *Applied Computing and Informatics*, vol. 14, no. 1, pp. 28–36, 2018, doi: 10.1016/j.aci.2017.04.002.
- [22] F. Sadikin, T. V. Deursen, and S. Kumar, "A ZigBee Intrusion Detection System for IoT using Secure and Efficient Data Collection," *Internet of Things*, vol. 12, 2020, doi: 10.1016/j.iot.2020.100306.
- [23] Kozhevnikov V. A., Sabinin O. Y., and Shats J. E., "Library development for creating bots on slack, telegram and facebook messengers," *International scientific journal : Theoretical & Applied Science*, vol. 50, no. 06, pp. 59–62, 2017, doi: 10.15863/TAS.2017.06.50.4.
- [24] H. Candra, Rino, and Riki, "Designing a Chatbot Application for Student Information Centers on Telegram Messenger Using Fulltext Search Boolean Mode," *Journal Bit-Tech*, vol. 2, no. 2, pp. 71–80, 2020, doi: 10.32877/bt.v2i2.106.
- [25] G. Albertengo, F. G. Debele, W. Hassan, and D. Stramandino, "On the performance of web services, google cloud messaging and firebase cloud messaging," *Digital Communications and Networks*, vol. 6, no. 1, pp. 31–37, 2020, doi: 10.1016/j.dcan.2019.02.002.
- [26] K. Kritikos, K. Magoutis, M. Papoutsakis, and S. Ioannidis, "A survey on vulnerability assessment tools and databases for cloud-based web applications," *Array*, vol. 3–4, 2019, doi: 10.1016/j.array.2019.100011.




BIOGRAPHIES OF AUTHORS

Agus Tedyyana    is a senior lecturer at the Politeknik Negeri Bengkalis, Bengkalis, Riau, Indonesia. He has an educational background in computer science. He has worked in education as a lecturer since 2014. He has been continuing his Doctoral (Ph.D.) studies at the Universiti Utara Malaysia (UUM) Campus in Kedah Darul Aman, Malaysia, since early 2020. His research interests are in computer security. He can be contacted at email: agustedyyana@polbeng.ac.id.



Osman Ghazali    Dr. Osman Ghazali is an Associate Professor and the Deputy Dean of the School of Computing, Universiti Utara Malaysia. Osman holds a Ph.D. in Information Technology (Networking) from Awang Had Salleh Graduate School, Universiti Utara Malaysia (AHS GS). He was a visiting research fellow at the School of Engineering & Applied Science, Aston University (EAS), in 2012. In 2011, Osman was the Head of the Computer Science Department School of Computing, Universiti Utara Malaysia. Before that, from 2009 to 2011, he was the Technical Chairperson at the University Teaching and Learning Center, Universiti Utara Malaysia. Dr. Osman's research interest is Internetworking, Cloud Computing, and Information Security. He has more than 100 publications as refereed book chapters and refereed technical papers in journals and conferences. He is a senior member of the Internetworks Research Laboratory (IRL). He is also a member of the IEEE and the ACM. He can be contacted at email: osman@uum.edu.my.



Onno W. Purbo    Onno W. Purbo graduated from the Department of Electrical Engineering, Bandung Institute of Technology, in 1987. In 1989, he completed his postgraduate education at McMaster University, Canada, in the field of Semi-Conductor Laser. Five years later, he received his Ph.D. from the University of Waterloo, Canada, in the field of Integrated Circuit Technology for satellites. In November 2020, he received the Postel Service Award from the Internet Society. Postel Service Award was given to Onno for his outstanding contribution to the development of Internet technology in Indonesia. He can be contacted at email: onno@indo.net.id.