JOIV : Int. J. Inform. Visualization, 5(3) - September 2021 327-332

INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Teler Real-time HTTP Intrusion Detection at Website with Nginx Web Server

Agus Tedyyana[a,*], Osman Ghazali[b]

[a] *Department of Informatics Engineering, Politeknik Negeri Bengkalis, Bengkalis, 28711, Indonesia*
[b] *School of Computing, University Utara Malaysia, Sintok, Kedah, Malaysia*
Corresponding author: [*]agustedyyana@polbeng.ac.id

***Abstract*— Web servers and web-based applications are now widely used, but in this case, the crime rate in cyberspace has also increased. Crime in cyberspace can occur due to the exploitation of how a system works. For example, the way HTTP works are exploited to weaken the webserver. Various tools for attacking the internet are also starting to be easy to find, but so are the tools to detect these attacks. One of the useful tools for detecting attacks and sending warnings against threats is based on the weblogs on the webserver. Many have not reviewed Teler as an intrusion detection system on HTTP on web servers because the existing tools are relatively new. Teler detecting the weblog and run on the terminal with rule resources collected from the community. So here, the researcher tries to implement the use of Teler in detecting HTTP intrusions on a Nginx-based web server. Intrusion is carried out in attacks commonly used by attackers, for example, port scanning and directory brute force using the Nmap and OWASP ZAP tools. Then the detection results will be sent via the Telegram bot to the server admin. From the results of the experiments conducted, it has been found that Teler is still classified as being able to send warning notifications with a delay between the time of detection and the time when the alert is received, no more than 3 seconds.**

***Keywords*— HTTP; intrusion detection; teler; web server.**

Manuscript received 23 Dec. 2020; revised 8 May 2021; accepted 9 Jul. 2021. Date of publication 30 Sep. 2021.

I. INTRODUCTION

Web servers and web-based applications are now widely used in various organizations and business actors, and they are often the target of various attacks from the internet network that can cause damage to existing systems. To reduce the risk of web application attacks, a web application developer needs to write secure applications to prevent known attacks. Attack detection is essential for responding to incidents, limiting damage from attacks, preventing other attacks, and preventing future attacks.

According to a report by the National Cyber Agency's Cyber Operations Security Center and State Code, during 2019, the eagle observation point system had around 290.3 million cyber-attacks (intrusions) into Indonesia's internet network. The largest of these were data test attacks, followed by attacks using malware methods. Compared to many cyber-attacks, the number of complaints from the public regarding the incident that occurred is relatively small. Cyberattacks surged in September, October and decreased sharply in November. In November and December, this figure is still much higher than the first six months of the 2019s. One of the fire's causes was that the incident would involve many people in October to coincide with the appointment of the President and Vice President of Indonesia for the new 2019-2024 period [1]. Intrusion can be defined as a collection of events and threats that threaten the confidentiality and integrity of information or data on network resources such as user accounts, system files, and kernel systems. Attacks on computer networks are a radical threat to any network because they are threatened every hour of the day and with the development of various attacks that are extremely fast. The problem of detecting attacks on networks has increased much fold after the increased use of botnets and distributed attacks. Two common network attacks are Denial of Service (DoS) and Port Scan [2]. To deal with intrusion on a computer network, an Intrusion Detection System (IDS) is used which is installed on the server. IDS tracks network activity and system work with attacks and malicious activity exposed to the network and then sends reports to system security administrators [3].

IDS consists of various categories: Host-based IDS (HIDS), Network-based IDS (NIDS), and WirelessIDS. There is also

a Hybrid IDS that combines various IDS categories into one. Host-based IDS performs single host activity and if any malicious activity occurs on the server. HIDS policy in the process and policy policies system files, system logs, and registry keys from outside network intrusion. A host-based intrusion detection system that runs on the system includes techniques to analyze and analyze information on the system [4]. There is various software that can be used to wait for intrusion on the server. For example, Snort, used in Hambali and Nurmiati's research [5], implements IDS on server security against data flooding attacks using Snort. Snort is software to check for intruders. It can analyze packets that trace the network in real-time traffic, enter the database, and handle various attacks originating from within the network and outside the network [5]. In another study by Utomo *et al.* [6], Suricata software can also be used to check the detection of intrusions on a PC server. Suricata is a high-performance NIDS and network security monitoring engine. Suricata is classified as an open-source IDS and is owned by a non-profit community, namely the Open Information Security Foundation (OISF). Suricata is a machine that has a multi-threaded capability [6].

In intrusion detection, web-based applications typically use weblogs from a web server to detect intrusion. It is useful that the web administrator can recover or at least find out the cause of the web server failure. By analyzing these log files, we were able to mine some potential web attack patterns [7].

In this research, researchers used different software, namely Teler. Teler is a real-time intrusion detection tool and web log-based alerts that run on the terminal with resources collected and made available by the community. Teler is designed to be a fast terminal-based threat analyzer. The core idea is to quickly analyze and hunt threats to prevent impending dangers sooner [8]. The use of weblogs to detect intrusions has also been carried out. In the research of Seyyar *et al.* [9], they studied web vulnerability scans detection through access log files of Apache web servers in addition to detection of XSS and SQLI attacks using a rule-based methodology [9].

Port Scanning is the process of seeing which ports are open on a particular server or all servers on the network. The first thing an intruder does is find out what services are currently running on the network. Once an intruder finds this information, the attacker will look for known vulnerabilities for existing services. The preprocessor port scan is designed to perform port monitoring. The preprocessor can be used to log scanning activity to a specific location other than a standard log file. Hackers can use some port scanning, for example, Nmap as a port scanning tool [10].

In research by Suroto [11], which has examined reviews in protecting various web servers from Slow HTTP Attack attacks, explains that one of the threats that occur on web servers is the exploitation of the workings of the HTTP protocol. The way the HTTP protocol works requires that the server fully accept every client request before the request is processed. If the HTTP request is incomplete or the transfer rate is very low, the server remains busy waiting for more data. If the server keeps too many busy resources, there is a denial of service. Internet users can exploit these vulnerabilities, intentionally send incomplete data packets, and repeatedly ask [11].

Devi and Kumar [12] has conducted a study by analyzing web-based application vulnerabilities using Ethical Hacking. The main purpose of conducting a vulnerability analysis is to identify gaps and weaknesses in networks and web applications using penetration testing to protect various parties from threats in cyberspace. In the testing process, various tools were used, such as Nikto, OWASP ZAP, Netcraft, Sparta, and Nmap. From the analysis, the ZAP tool can find low-level attacks. However, from the comparison between Nikto and Zap tools, Nikto is more able to identify vulnerabilities than ZAP [12].

Telegram is a secure instant messaging application that is used to send and receive text and multimedia messages from fellow users. Telegram was chosen because of its security features and reliability in data encryption as well as its ability to make chatbots easily use several functions through a programming language using the Telegram Bot API. Telegram Bot API is used to help build Telegram Bot dynamically using various programming languages [13].

From previous studies, many have not reviewed the Teler to detect intrusion on HTTP on a web server because the existing tool is still relatively new. Therefore, the researchers attempt to implement the use of Teler in detecting HTTP intrusion on a Nginx-based web server. The intrusion is carried out in attacks commonly used by attackers, for example, port scanning and directory brute force using Nmap and OWASP ZAP tools. The purpose of this study is to test whether Teler can detect intrusions and then report them in real-time alerts to the telegram bot**.**

II. MATERIALS AND METHOD

The methodology carried out at this stage is to achieve the previous objectives.

A. Designing Attacking Scenario

Before conducting an attack detection experiment using Teler, the first thing to prepare is to design how the scenario flow of the attack experiment will be carried out. The flow to be carried out is, as shown in Figure 1 below.

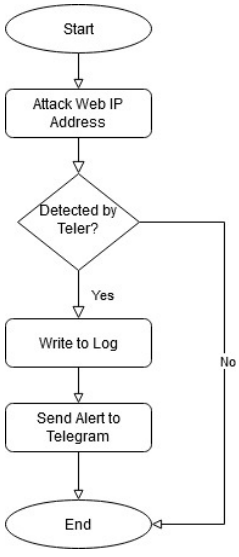

Fig. 1 Bot telegram of Teler real-time HTTP intrusion

From the previous figure, we get an overview of how Teler works in detecting an attack on the HTTP protocol. The scenario starts with attacking the target IP address using tools such as Nmap[14] and OWASP ZAP. Teler will later detect this intrusion by seeing changes to the logs on the webserver. If suspicious activity is detected written on the webserver log, Teler will write to the log first and then send a warning message to the researcher via telegram. If no suspicious activity is found, Teler will continue to monitor changes to the webserver log.

B. Preparing Environment for Testing (Sandbox)

To implement Teler, we first prepare a safe test environment for attempted attacks. The testing environment uses a cloud computing service with the Ubuntu 18.04 server operating system to install the required package as a web server, namely Nginx. Nginx is a lightweight web server as well as a reverse proxy server. It is well known for its less memory usage and reliable concurrency [10]. Because of the reliability, of course, the Nginx web server will be very suitable for Teler, which can be run with high concurrency values.

C. Make a Telegram Bot

Furthermore, to get real-time alerts, researchers created a telegram bot that will send messages whenever Teler detected the intrusion on the server. Telegram bot is created using @bot_father, which has been provided by telegram. The bot that has been completed will provide a token that will be used in the Teler configuration process. In addition to tokens, chat id from telegram is also needed so that the bot that is created can later send messages to the appropriate chat id recipient. The chat id is obtained by using the telegram @get_id_bot bot. The bot that has been created is in Figure 2 below.

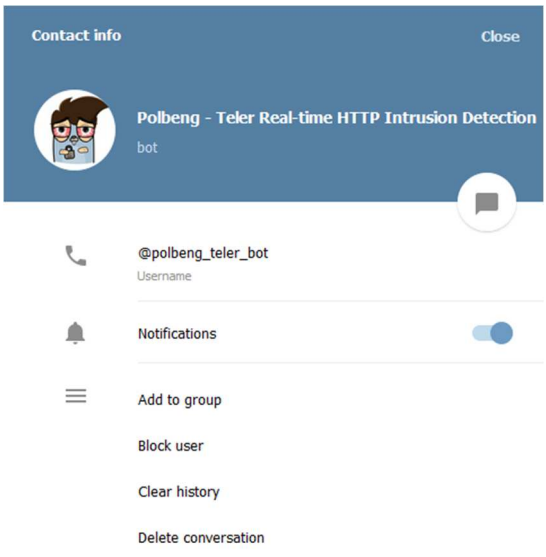

Fig. 2 Bot telegram of Teler real-time HTTP intrusion detection

D. Teler Installation and Configuration

The next stage is to install and install Teler on an existing server. The installation is done by cloning the source which has been provided open-source and is free to use on Github at the https://github.com/kitabisa/teler. The installation process requires the Go programming language package to be able to run Teler because this application is written in the Go language. Golang supports concurrency in programming systems very well, and its application into code is also fairly easy [15].

After installation, we must configure Teler to make it run according to the web servers being used. Several web servers can be used: Apache, Nginx, Amazon S3, Elastic LB, CloudFront, and Nginx Ingress. Each log server has a different log variable format. For example, the Nginx log format should be set, as shown in Figure 3 below.

Fig. 3 Teler log format for Nginx web server

In addition to the log format, we also have to apply it to the alert section. Three types of bots can be used, namely, Telegram, Discord, and Slack. Telegram was chosen because it was angry that the application was relatively easy and was more commonly used by ordinary people than the other two applications. Telegram is also widely used as a chatbot which is cost-effective and can be said to be minimal in terms of software infrastructure [16]. After selecting the chatbot application used, enter the telegram bot token and the idea we chat about after configuring it as in Figure 4. The researcher also customized a little telegram alert template to make it easier to understand, as in Figure 5.

```
alert:
  active: true
  provider: "telegram"

notifications:
  slack:
    token: "xoxo-...."
    color: "#ffd21a"
    channel: "G30SPKI"

  telegram:
    token: "1150410614:AAEQx2xRPCk-PdR0ehRFnzlCHQw_EITcySM"
    chat_id: "-612930935"
```

Fig. 4 Teler alert configuration for real-time notification

```
*teler Alert*
Peringatan: *{{ .category }}*

Request: `{{ .request_method }} {{ .request_uri }} {{ .request_protocol }}`
Tanggal: `{{ .time_local }}`
Alamat IP: `{{ .remote_addr }}`
User\-Agent: `{{ .http_user_agent }}`
Referrer: `{{ .http_referer }}`
Status code: `{{ .status }}`
Bytes sent: `{{ .body_bytes_sent }}`

Dikirim oleh Polbeng Teler Bot - Real-time HTTP Intrusion Detection
```

Fig. 5 Telegram chatbot template

E. Attack Testing Nmap and OWASP ZAP

To find Teler reliability in detecting intrusion, the researchers conducted an attack experiment with port scanning techniques using Nmap and OWASP ZAP tools. Before testing, first, we run Teler with the tail command to monitor file changes in real-time. Figure 6 below shows how the Teler process when run.

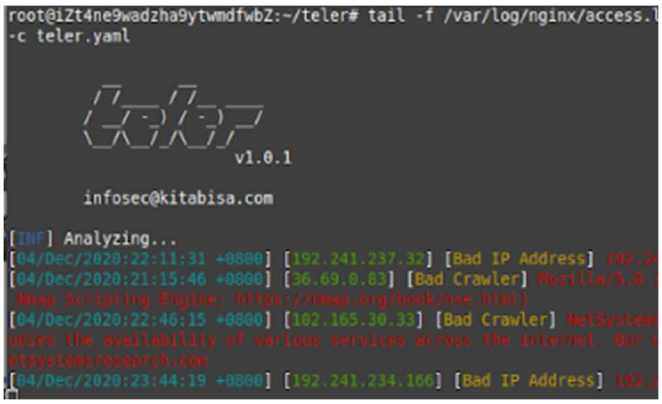

Fig. 6 Running Teler

Nmap and OWASP ZAP are then executed with the target, namely the IP address of the sandbox that has been provided. Researchers run commands like in Figure 7 below to perform port scanning using Nmap.

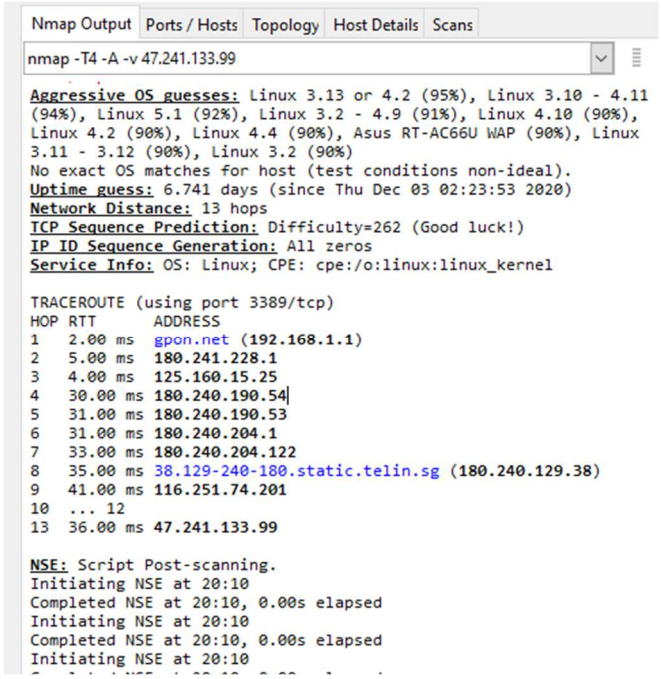

Fig. 7 Attacking using Nmap

OWASPs ZAP is used for security scanning of web applications. OWASP ZAP has various modules such as Proxy for recording and capturing packets, Fuzzer for identifying vulnerabilities, Spider for finding web applications, Scanner for active and passive attacks, and Dictionary method for accessing files [17]. In Figure 8 below are the attack results using the OWASP ZAP tool, where the researcher got 5 alerts with the highest alert on medium priority level.

Alerts are potential vulnerabilities and have been categorized as high, medium, low, and informational, indicating the degree of associated risks. A high-priority alert means that a problem in this category is more serious than other priority alerts. Also, medium priority, low priority, and informational priority alerts are becoming less serious. Alert categories are indicated by different colored flags [18].

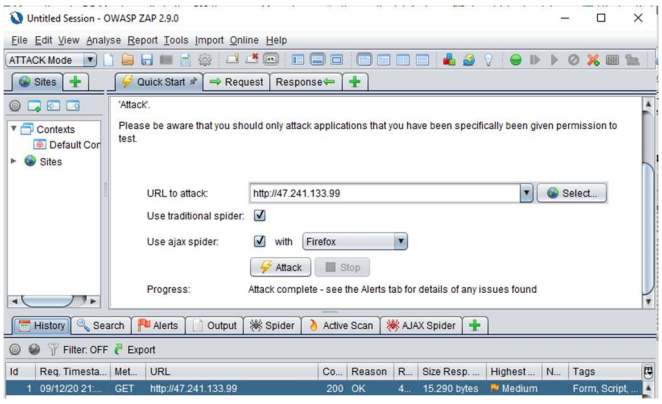

Fig. 8 Attacking using OWASP ZAP.

III. Result and Discussion

In this section, data analysis is carried out from the results obtained after carrying out the attack. The data in this experiment were collected from the time of the incoming message sent by the Telegram Bot and the time recorded in the log on the webserver. The two data will then be compared to find out whether Teler can send real-time notifications reliably.

A. Log Timestamp at Attack Using Nmap

In an attack using Nmap Teler can recognize what tools are used in the attack. As in Figure 9 below, Teler detected that the Nmap script carried out an attack with a Bad Crawler pattern.

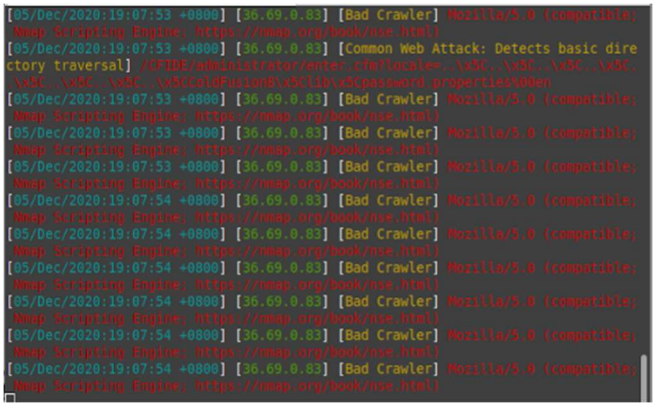

Fig. 9 Teler Log when attacking using Nmap

When a Web Crawler visits a Web page, it reads the visible text, hyperlinks, and the content of various tags used on the site, such as keyword-rich Meta tags. To use the information gathered from crawlers, search engines determine the site and index the information. Finally, all the text and metadata that define the Web documents scanned by crawlers are stored in the Search engine database. Bad bots (crawlers) use CDN (Content Delivery Network) bandwidth, use server resources and steal valuable content for abuse. Furthermore, by downloading full content from a Web server, a bad WebCrawler can affect server performance. [19].

The log data then compares with the time in the alert chat sent by the Telegram bot. In Figure 10 below, it can be seen that the bot is able to send real-time attack notifications by Nmap to the user. A little explanation beforehand that alerts follow the server's time zone, which is the GMT +08 zone,

while researchers are in the GMT +07 time zone, so there is a time difference of 1 hour, but alerts are still sent in real-time.

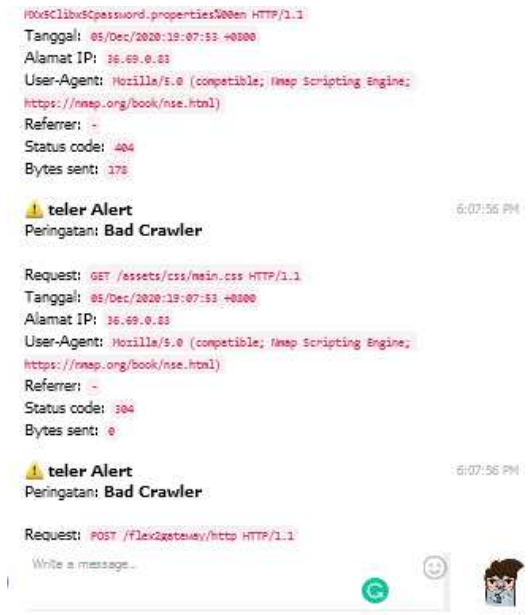

Fig. 10 Alert notification by Telegram bot of Nmap attack

B. Log Timestamp at Attack Using OWASP ZAP

In an attempted attack using OWASP ZAP, Teler was unable to detect what tools were used to attack. But still able to detect attacks that occur. For example, in the log in Figure 11 below, when OWASP performs vulnerability scanning techniques by scanning directories on the web, Teler can detect attacks that occur and send real-time warning notifications.

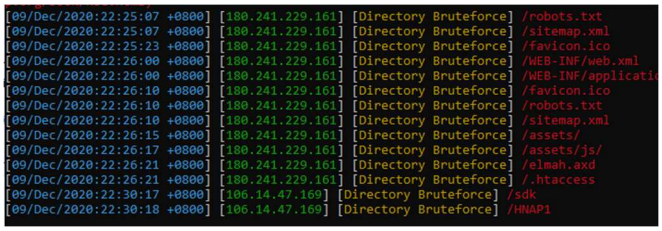

Fig. 11 Teler Log when attacking using OWASP ZAP

A brute force directory guessing attack is a very commonly used attack against websites and web servers. They are used to find hidden and often overlooked directories on the site to try to compromise. Attackers generally focus on directories (folders) that are likely to contain out-of-date or unsafe software. Directory guessing attacks are often vociferous and generate thousands of 404 (not found) errors in the logs. If we monitor and supervise our logs, we should identify them easily and block attackers' IP addresses [20].

C. Detection Accuracy Results

From the results of the timestamp on log earlier, the researcher then compared the timestamp between the time the attack was detected and the time the alert was received on telegram to determine whether Teler was able to generate real-time alerts when attacking using Nmap tool as in table 1 below.

TABLE I
RESULTS OF DETECTION TIME DIFFERENCE AND ALERT TIME USING NMAP

No	Attack Type	Time of Detect	Time of Alert	Time Difference(s)
1	Bad Crawler	19:07:53	19:07:56	3
2	Common Web Attack	19:07:53	19:07:56	3
3	Bad Crawler	19:07:53	19:07:56	3
4	Bad Crawler	19:07:53	19:07:56	3
5	Bad Crawler	19:07:53	19:07:56	3
6	Bad Crawler	19:07:53	19:07:56	3
7	Common Web Attack	19:07:53	19:07:56	3
8	Bad Crawler	19:07:53	19:07:56	3
9	Bad Crawler	19:07:53	19:07:56	3
10	Bad Crawler	19:07:54	19:07:56	2
The difference in time average				2.9

Based on the results of the table above, it is found that of the last ten types of attacks detected using Nmap, there is an average time difference of 2.9 seconds. This means a delay of about 2.9 seconds to send a warning to the telegram since Teler detected the attack. Next is the time difference between the time the attack was detected and the warning sent when the attack was carried out using OWASP ZAP in table 2 below.

TABLE II
RESULTS OF DETECTION TIME DIFFERENCE AND ALERT TIME USING OWASP ZAP

No	Attack Type	Time of Detect	Time of Alert	Time Difference(s)
1	Directory Bruteforce	01:05:21	01:05:21	0
2	Directory Bruteforce	01:05:21	01:05:21	0
3	Directory Bruteforce	01:05:30	01:05:31	1
4	Directory Bruteforce	01:05:30	01:05:31	1
5	Directory Bruteforce	01:05:38	01:05:39	1
6	Directory Bruteforce	01:05:38	01:05:39	1
7	Directory Bruteforce	01:05:42	01:05:43	1
8	Directory Bruteforce	01:05:43	01:05:44	1
9	Directory Bruteforce	01:05:48	01:05:48	0
10	Directory Bruteforce	01:05:48	01:05:48	0
The difference in time average				0.6

If observed in Table 2, the time difference obtained is not very small, around 0.6 seconds. This is faster than when Teler detected an attack from Nmap. From the two comparisons, it is found that the time lag required for Teler to send warning notifications to Telegram is not greater than 3 seconds.

IV. CONCLUSION

From the research results that have been done, the researcher concludes that from the attempted attack using the Nmap and OWASP ZAP attack tools against the webserver,

Teler has detected quite well. Teler can detect what type of attack is being carried out on the server, what IP address is attacking, and the user agent used by the user and then sends the warning results in real-time to the Telegram bot with a time lag of not more than 3 seconds. Future research may be possible to develop more features at Teler, such as implementing a prevention function against future attacks by utilizing data generated by Teler. For example, it implements machine learning to automatically blacklist certain IPs that are detected trying to attack the server.

REFERENCES

[1] BSSN, "BSSN'S Measure to Detect Cyberthreats," Jakarta, 2019. doi: 10.1017/CBO9781107415324.004.

[2] A. Gupta and L. Sen Sharma, "Detecting attacks in high-speed networks: Issues and solutions," *Inf. Secur. J.*, vol. 29, no. 2, pp. 51–61, 2020, doi: 10.1080/19393555.2020.1722296.

[3] M. Tabash, M. A. Allah, and B. Tawfik, "Intrusion detection model using naive bayes and deep learning technique," *Int. Arab J. Inf. Technol.*, vol. 17, no. 2, pp. 215–224, 2020, doi: 10.34028/iajit/17/2/9.

[4] S. Jose, D. Malathi, B. Reddy, and D. Jayaseeli, "A Survey on Anomaly Based Host Intrusion Detection System," *J. Phys. Conf. Ser.*, vol. 1000, no. 1, 2018, doi: 10.1088/1742-6596/1000/1/012049.

[5] A. H. Hambali and S. Nurmiati, "Implementasi Intrusion Detection System (IDS) Pada Keamanan PC Server Terhadap Serangan Flooding Data," *Sainstech J. Penelit. dan Pengkaj. Sains dan Teknol.*, vol. 28, no. 1, pp. 35–43, 2018, doi: 10.37277/stch.v28i1.267.

[6] D. Utomo, M. Sholeh, and A. Avorizano, "Membangun Sistem Mobile Monitoring Keamanan Web Aplikasi Menggunakan Suricata dan Bot Telegram Channel," *Semin. Nas. Teknoka*, vol. 2, no. 2502, pp. 1–7, 2017.

[7] K. Ma, R. Jiang, D. Mianxiong, Y. Jia, and A. Li, "Neural Network Based Web Log Analysis for Web Intrusion Detection," *Springer Int. Publ.*, pp. 194–204, 2017, doi: 10.1007/978-3-319-72395-2.

[8] D. Siswanto, "Teler Real-time HTTP Intrusion Detection," 2020. https://github.com/kitabisa/teler (accessed Dec. 05, 2020).

[9] M. Baş 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. Elsevier B.V., pp. 28–36, Jan. 01, 2018, doi: 10.1016/j.aci.2017.04.002.

[10] A. Erlansari, F. F. Coastera, and A. Husamudin, "Early Intrusion Detection System (IDS) using Snort and Telegram approach," *Sisforma*, vol. 7, no. 1, p. 21, 2020, doi: 10.24167/sisforma.v7i1.2629.

[11] S. Suroto, "A Review of Defense Against Slow HTTP Attack," *JOIV Int. J. Informatics Vis.*, vol. 1, no. 4, p. 127, 2017, doi: 10.30630/joiv.1.4.51.

[12] R. S. Devi and M. M. Kumar, "Testing for Security Weakness of Web Applications using Ethical Hacking," no. Icoei, pp. 354–361, 2020.

[13] J. C. de Oliveira, D. H. Santos, and M.́ario P. Neto, "Chatting with Arduino Platformthrough Telegram Bot," in *2016 IEEE International Symposium on Consumer Electronics*, 2016, pp. 1–2.

[14] K. Coffey, R. Smith, L. Maglaras, and H. Janicke, "Vulnerability Analysis of Network Scanning on SCADA Systems," *Secur. Commun. Networks*, vol. 2018, 2018, doi: 10.1155/2018/3794603.

[15] A. A. Kristanto, Y. Harjoseputro, and J. E. Samodra, "Implementasi Golang dan New Simple Queue pada Sistem Sandbox Pihak Ketiga Berbasis REST API," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 4, pp. 745–750, 2020.

[16] M. Barthelmäs, M. Killinger, and J. Keller, "Using a Telegram chatbot as cost-effective software infrastructure for ambulatory assessment studies with iOS and Android devices," *Behav. Res. Methods*, 2020, doi: 10.3758/s13428-020-01475-4.

[17] F. Holik and S. Neradova, "Vulnerabilities of modern web applications," *2017 40th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2017 - Proc.*, pp. 1256–1261, 2017, doi: 10.23919/MIPRO.2017.7973616.

[18] S. K. Paudel, "Vulnerable Web Applications and How To Audit Them," Oulu University of Applied Sciences, 2016.

[19] G. C. Deka, *NoSQL Web Crawler Application*, 1st ed., vol. 109. Elsevier Inc., 2018.

[20] Sucurisecurity, "Directory Guessing Brute Force Attacks," 2019. https://docs.sucuri.net/definitions/attacks/brute-force/directory-guessing-brute-force-attacks/ (accessed Dec. 12, 2020).