Elastic Stack Ability Test Monitoring Slowloris Attack on Digital Ocean Server

Is Mardianto[1], Dedy Sugiarto[2], Krisna Aditama Ashari[3]
Informatics Engineering, Faculty of Industrial Technology, Universitas Trisakti, Jakarta, Indonesia
[1]mardianto@trisakti.ac.id, [2]dedy@trisakti.ac.id, [3]krisna064001904003@std.trisakti.ac.id

Accepted 31 July 2021
Approved 12 September 2021

***Abstract*— Servers have a central role in computer network. The server is in charge of serving user requests with various types of services. Every server activity in handling these things will generate different types of logs. Information from this large amount of logs is often ignored and has not been widely used as material for analyzing the performance of the server itself. In this study, Elastic Stack is functioned as a system that handles upstream to downstream processes starting from collection, transformation, and storage as well as graphical visualization of the Nginx web server given an attack scenario in the form of massive incoming connection requests and server login access attempts. The Elastic Stack components used as log collectors are Filebeat and Metricbeat for system metric data. For testing attacks using the Slowloris tool which will consume web server resources. The results of the research that have been carried out are when there are 500 incoming connections, the web server can serve requests normally, at 1000 connections there are some packets that are not served, the server becomes unable to access when it reaches a total of 2000 incoming connections. Metric data in the form of CPU Usage and Memory Usage are affected, although not significantly. Identification of IP Address shows the source of the attack comes from Singapore, according to the domicile of the attacker's computer. All access data in the form of username, time, origin of region trying to enter the server are recorded by the system.**

***Index Terms*—availability; cloud computing; filebeat; log; metricbeat.**

I. Introduction

Log is a file containing a list of events that occur on a computer system [1]. This log file is also owned on a server that runs various types of services on it. As a system administrator, log files on these servers can provide useful information in monitoring server performance. CPU performance, memory usage, disk, network I/O, as well as the ability to detect disturbances from inside and outside the system are some important indicators in the availabilty or sustainability of server performance in running its services. Thus, a centralized logging system is needed to be able to transmit existing system status data for later reporting that is easy to analyze [2]. Some of the capabilities that today's modern log management systems must have include: Aggregation, namely the ability to collect and transmit logs from various data sources. Processing to convert log messages into meaningful data. Data storage for a long time to allow monitoring, trend analysis, and security, as well as Analysis, which is to sort data by performing queries and making visualizations [3].

Elastic Stack or previously known as ELK Stack is a collection of open source software developed by Elastic which is useful for searching, analyzing, and visualizing logs generated from any source in any format [4].

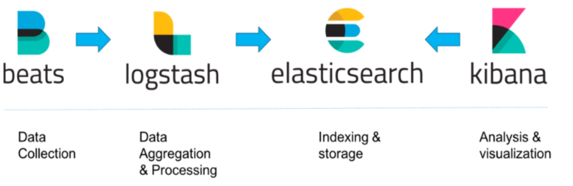


Fig. 1. Elastic Stack Components

The main components of the Elastic Stack are divided into 4 parts as shown in Fig. 1, Elasticsearch is a place where log data is stored and indexed, Logstash functions to process various sources and types of data into more structured information, and Kibana which is designed as a visualization platform, providing a web-based interface to search, view, and analyze data stored in Elasticsearch clusters [5].

Another important component is Beats. Beats is a platform that handles sending data from various sources [6]. Filebeat and Metricbeat are part of Beats. Filebeat is a data transmission platform that collects and transmits data from various sources and then forwards the data to the Elastic Stack [7]. Filebeat can be installed on almost any operating system, including as a Docker container, and it also comes with built-in modules for certain platforms (such as Apache, MySQL, Docker, MariaDB, Percona, Kafka, etc.). Metricbeat will periodically pass metric data from the operating system and service statistics running on the system to Elasticsearch or Logstash [8].

The use of Elastic Stack on the server is the right solution in log management because it is able to display information to administrators about statistics, trends, and anomalies that arise [9]. These are evidenced in several previous studies regarding the use of the Elastic Stack, including research on monitoring the apache web server in handling user requests [1], generating graphs of web server performance against time, utilization and error, other studies also collect logs from campus network infrastructure from various devices [2], generates a collection of log data by category such as warnings, errors, information and notifications. In addition, Elastic Stack can also be used as a detection system and information collection related to attempts to access the server and delete log data by the client [10]. Related research to analyze the types of slow HTTP attacks and their impact on virtual machines using the DSTAT tool, the results obtained indicate an increase in server resource consumption such as CPU and memory usage [11]. The difference between this research and previous research is in the use of Digital Ocean's cloud computing servers, as well as measurement and analysis of server performance in running the Nginx service against Slowloris attacks using Elastic Stack.

Nginx is an open-source web server software, created by Igor Sysoev and released to the public in October 2004. At its release, the makers assured the public that Nginx can solve web server performance problems in handling large numbers of active connections simultaneously. Nginx provides lower memory usage than other web servers, and also has the following features: reverse proxy, IPv6, load balancing, FastCGI support, web sockets, static file processing, TLS/SSL [12].

Slowloris is one such attack tool by opening a connection, then sending HTTP headers, adding them but never completing the request. Thousands of HTTP POST connections are made and HTTP headers are sent very slowly to force the target web server to keep the connection open. Slowloris will take all the resources of the target web server, thereby blocking requests from legitimate clients or clients who want to access the web server [13]. This attack belongs to the category of availability attacks where the server is not available when needed.

How to measure the impact on the system, and Nginx service when experiencing a Slowloris attack and analyze the source early is the purpose of this study. Therefore, this study aims to use the Elastic Stack as a system that helps collect and process various system log data with examples of Slowloris attacks.

II. METHOD

The method used in this study consists of several stages as shown in Fig. 2.

A. Preparation

This stage is in the form of designing and preparing a cloud server that will function as a log server and web server. The infrastructure uses the services of a cloud computing service provider, namely DigitalOcean. The operating system used is Ubuntu version 20.04. The amount of cloud computing costs varies depending on the required server specifications, in this study using server specifications with details as shown in Table I.

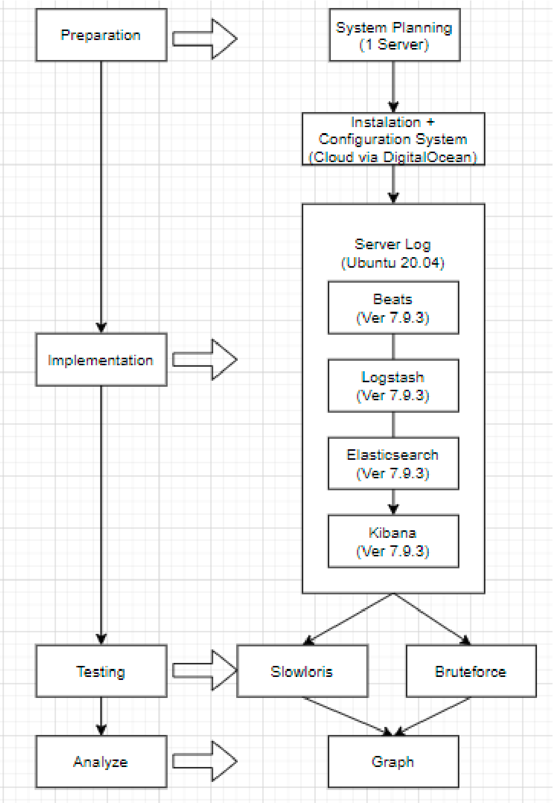
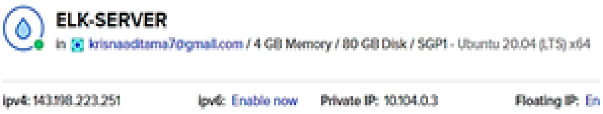

Preparation
System Planning (1 Server)
Instalation + Configuration System (Cloud via DigitalOcean)
Server Log (Ubuntu 20.04)
Beats (Ver 7.9.3)
Logstash (Ver 7.9.3)
Elasticsearch (Ver 7.9.3)
Kibana (Ver 7.9.3)
Implementation
Testing
Slowloris
Bruteforce
Analyze
Graph

Fig. 2. Methodology

TABLE I. SERVER SPECIFICATIONS

No	Component	Description
1	Processor	2 CPU
2	RAM	4 GB
3	HDD	80 GB
4	Transfer	4 TB
5	Location	Singapore

After the process of selecting specifications and filling in data about the created server has been completed, the server is ready for use as shown in Fig. 3


ELK-SERVER
in krisnaaditama7@gmail.com / 4 GB Memory / 80 GB Disk / SGP1 - Ubuntu 20.04 (LTS) x64

ipv4: 143.198.223.251 ipv6: Enable now Private IP: 10.104.0.3 Floating IP: En

Fig. 3. Server on DigitalOcean

To perform an attack simulation, an additional PC is needed that functions to run Slowloris. The 2 additional PCs use the infrastructure of Google Cloud Platform as shown in Table II.

TABLE II. ATTACKER SERVER

No	Name	Zone	External IP

1	krs1	asia-southeast1-a	35.240.175.118
2	krs2	asia-southeast2-a	34.101.189.102

B. *Implementation*

The second stage includes the implementation and configuration of the Elastic Stack packages, namely Beats, Logstash, Elasticsearch and Kibana as the system that will manage the logs. And the use of Nginx as a web server.

1) Elastic Stack and Nginx

Because the server uses a cloud system, interaction with the server is carried out remotely through the PuTTY application using the Public IP that has been provided by the cloud provider, as well as the password that was previously set when creating the server. To be able to use the Elastic Stack on the server, it requires the java package to be pre-installed by adding the repository from the Elastic Stack web (https://artifacts.elastic.co).

The first package to create is Elasticsearch. Elasticsearch has the ability to index logs and can instantly search for specific records from billions of log records [14]. In Elasticsearch the configuration is done in the /etc/elasticsearch/elasticsearch.yml file, remove the hash mark or uncomment the network.host and http.port lines, set the ip on network.host to 0.0.0.0 and because it only runs on single server then add the line discovery.type: single-node. The next package needed is Logstash, Logstash acts as a data collector, forwarder, and processes log data into JSON format [15], so that the log data from filebeat can be processed later, it is necessary to create a configuration file. The logstash.conf file is stored in the /etc/logstash/conf.d/ folder. The logstash configuration file is saved under the name nginx.conf:

```
nginx.conf
Input: beats
Output: index

  input {
    beats {
      port => 5044
    }
  }
  output {
    if [@metadata][pipeline] {
      elasticsearch {
        hosts => localhost
        manage_template => false
        index => "%{[@metadata][beat]}-
        %{[@metadata][version]}-
        %{+YYYY.MM.dd}"
        pipeline                           =>
"%{[@metadata][pipeline]}"
      }
    } else {
      elasticsearch {
        hosts => localhost
        manage_template => false
        index => "%{[@metadata][beat]}-
        %{[@metadata][version]}-
        %{+YYYY.MM.dd}"
      }
    }
  }
```

In the configuration file, Logstash processes log input from Beats and produces an output in the form of an index per day based on the type of log data provided by Beats to be saved to Elasticsearch. To run the configuration use sudo /usr/share/logstash/bin/logstash -f /etc/logstash/conf.d/nginx.conf.

The last Elastic Stack package is Kibana. Kibana is an open-source tool for visualizing Elasticsearch data. Present users with various types of visual dashboards, such as bar charts, pie charts, time charts, histograms, heat maps, map visualizations, etc [16]. The configuration file that has been changed is /etc/kibana/kibana.yml by removing the hash mark / uncomment on the server.port: 5601 and server.host: 0.0.0.0 lines. After all packages are installed and configured, the service is run with the command sudo service service-name start. The service-name is changed according to the package name (elasticsearch / logstash / kibana), because the web server uses Nginx, use the default configuration for the service.

2) Beats

To be able to get log data it is necessary to install Beats. Beats is an agent for single-purpose data shippers. It sends various types of data (such as metrics, files, and networks) to Logstash [17]. To get the log collection functions of Nginx and the system running, add the system and nginx modules with the command sudo filebeat modules enable nginx system. Next run the command filebeat setup --pipelines --modules nginx,system. This module features an embedded fileset such as a source directory of logs to be collected, an ingest node pipeline for parsing data as well as providing a sample Kibana dashboard for graphical visualization of several frequently processed log formats. In the process of parsing the data through the ingest node of the Elasticsearch pipeline, transformations are made from the log lines such as 180.248.120.169 - - [03/Jul/2021:01:45:56 +0000] "GET /krisna HTTP/1.1" 404 197 " -" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4472.101 Safari/537.36" into a more structured form of data in JSON format.

In this ingest pipeline there is also a process of enriching information from the logs obtained with the help of the processor inside. Examples of processors used are GeoIP and User agent, where GeoIP provides detailed information regarding the country of origin, city and coordinates of the location of the incoming request, while the User agent will provide information related to the browser and operating system used by the user. Log the JSON form with additional information such as the following:

```
JSON Format Log

  "nginx": {
        "access": {
          "remote_ip_list": [
            "180.248.228.125"
```

```
     ]
    }
   },
   "log": {
    "file": {
     "path": "/var/log/nginx/access.log"
    },
    "offset": 563087
   },
   "source": {
    "geo": {
     "continent_name": "Asia",
     "region_iso_code": "ID-KI",
     "city_name": "Balikpapan",
     "country_iso_code": "ID",
     "region_name": "East Kalimantan",
     "location": {
      "lon": 116.8428,
      "lat": -1.2551
     }
    },
    "as": {
     "number": 7713,
     "organization": {
      "name": "PT Telekomunikasi
Indonesia"
     }
   }
```

In the /etc/filebeat/filebeat.yml file, uncomment the output.logstash and hosts: [“localhost:5044”] lines so that the logs obtained will be forwarded to Logstash for further processing.

To install the metricbeat package, use sudo apt-get install metricbeat. By default, metricbeat will collect system data such as CPU performance, memory usage, network traffic and other data on the system module. Finally, load all the configurations that have been made earlier with the command filebeat setup and metricbeat setup.

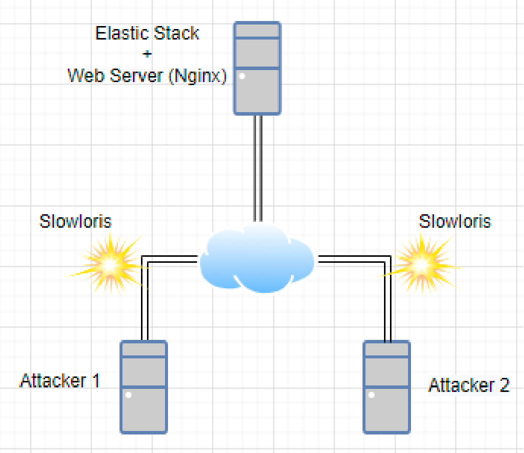


Fig. 4. Slowloris Attack Scheme

C. Testing

This testing phase includes testing the web server's ability to handle the maximum request load and can track the origin of the attack and record server login access attempts from outside parties. In the experiment, 2 PC servers were prepared as a source of attack that were installed using a cloud computing system, an illustration as shown in Fig. 4.

The test aims to obtain data on the impact of attacks recorded on the Elastic Stack, when several attack scenarios are carried out on the server as shown in Table III.

TABLE III. ATTACK SIMULATION

Scenario	Number of PCs	Attack (/PC)	Duration
1	1	500 Connection	20 Minutes
2	2	500 Connection	20 Minutes
3	2	1000 Connection	20 Minutes

In launching an attack on the web server, the Slowloris application was used which was obtained from the GitHub link, at the address https://github.com/gkbrk/slowloris.git, the attacker PC uses Git and clones the Slowloris file from the GitHub address above using the command git clone https://github.com/gkbrk/slowloris.git

To run Slowloris the command used is python3 slowloris.py <<Domain / IP Target>> -p 80 -s <<Number of Connections>> --sleeptime <<Attack lag time>>

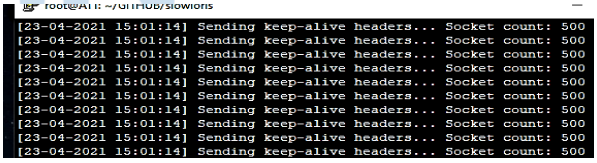
D. Analysis

The last stage is viewing the logs that have been processed into graphic form, to access the Kibana service on the browser by opening the address http://IP-Public:5601. Various types of visualizations have been provided by the Beats modules, this certainly makes it easier to analyze server performance and identify problems.

III. RESULT AND DISCUSSION

A. Attack Impact Testing

In the testing scenario 1 as shown in Fig. 5, the web server is loaded with request traffic of 500 connections originating from 1 PC using Slowloris for 20 minutes.



```
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
[23-04-2021 15:01:14] Sending keep-alive headers... Socket count: 500
```

Fig. 5. Slowloris attack 500 connections from 1 PC

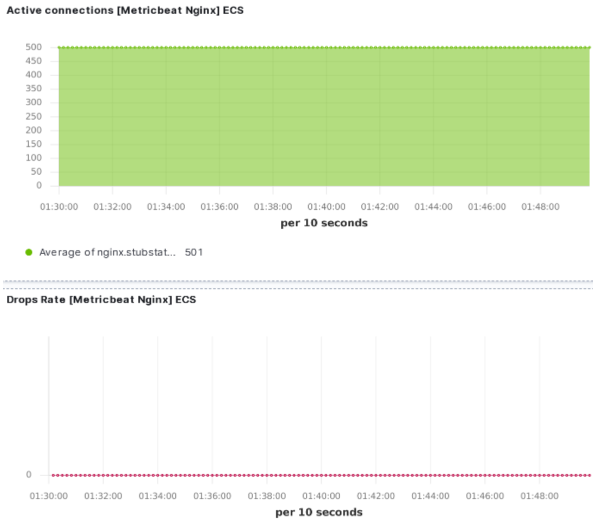


Fig. 6. Web Server Log Results in Scenario 1

Fig. 6 is the Nginx status chart in Kibana, show that the service is running normally where all 500 incoming connections can be served with a stable drop rate indicator at 0.

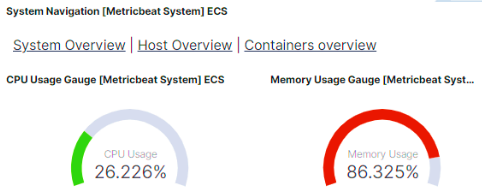
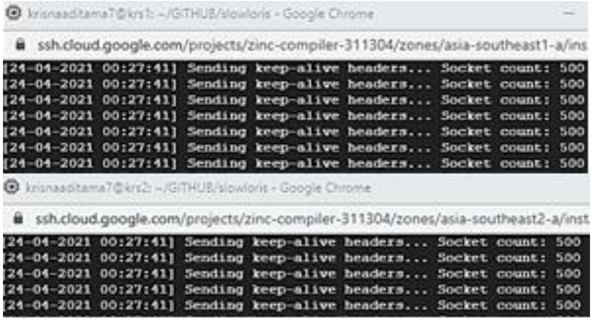

System Navigation [Metricbeat System] ECS
System Overview | Host Overview | Containers overview
CPU Usage Gauge [Metricbeat System] ECS
Memory Usage Gauge [Metricbeat Syst...
CPU Usage
26.226%
Memory Usage
86.325%

Fig. 7. CPU Usage and Memory Usage in Scenario 1

The results of the log observations as show on Fig. 7, CPU Usage indicator recorded during the attack was 26.226% and Memory Usage 86.325%. From the data obtained, it shows that the web server resource is still sufficient to serve all incoming requests. This is because the number of incoming attacks is still below the limits of the CPU and memory capabilities in handling incoming processes, so the attacks are not yet at the stage of disrupting the running of the computer system.

In testing scenario 2 as shown in Fig. 8, the web server is loaded with request traffic of 500 connections originating from 2 PCs using Slowloris with the same duration of 20 minutes.


krisnaaditama7@krs1: ~/GITHUB/slowloris - Google Chrome
ssh.cloud.google.com/projects/zinc-compiler-311304/zones/asia-southeast1-a/ins

```
[24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
[24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
[24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
[24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
[24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
[24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
[24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
```

krisnaaditama7@krs2: ~/GITHUB/slowloris - Google Chrome
ssh.cloud.google.com/projects/zinc-compiler-311304/zones/asia-southeast2-a/inst

```
24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
24-04-2021 00:27:41] Sending keep-alive headers... Socket count: 500
```

Fig. 8. Slowloris attack 500 connections from 2 PCs

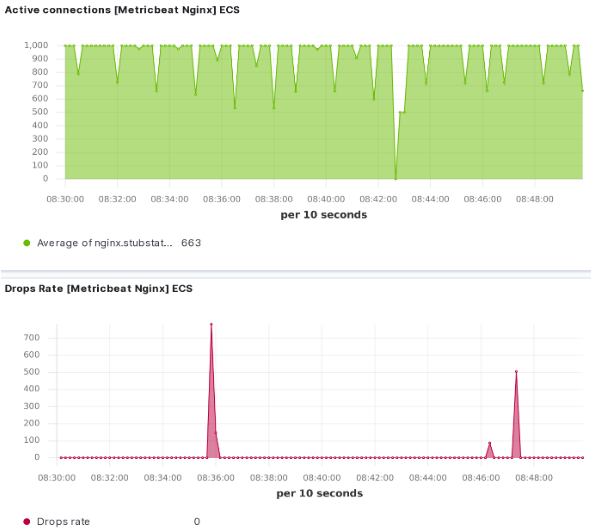


Fig. 9. Web Server Log Results in Scenario 2

The results of observations as shown in Fig. 9 show that there were several times the server failed to handle requests, with various drop rate indicators (100, 400, 700).

In the information shown in Fig. 10, CPU Usage and Memory Usage has increased compared to the first attack scenario to 31,861% and 87,241%. In this attack scenario with a total of 1000 incoming connections, the web server is still able to serve requests even though it terminates incoming connections several times due to resource limitations.

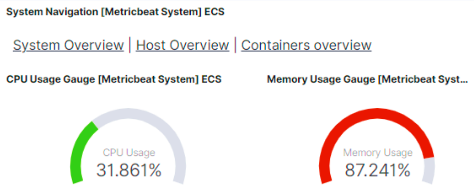
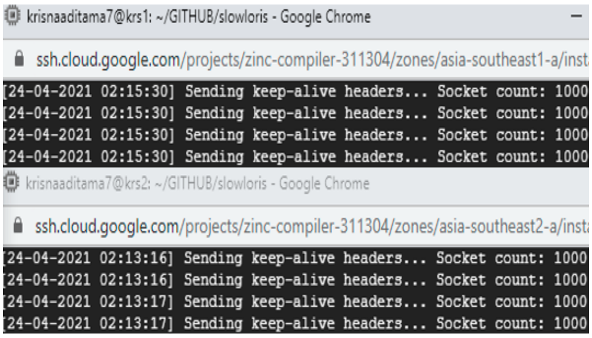

System Navigation [Metricbeat System] ECS
System Overview | Host Overview | Containers overview
CPU Usage Gauge [Metricbeat System] ECS
Memory Usage Gauge [Metricbeat Syst...
CPU Usage
31.861%
Memory Usage
87.241%

Fig. 10. CPU Usage and Memory Usage in Scenario 2

In testing scenario 3 as shown in Fig. 11, the web server is loaded with request traffic of 1000 connections originating from 2 PCs using Slowloris with a duration of 20 minutes. In total there are 2000 connections that make concurrent requests to the web server.


krisnaaditama7@krs1: ~/GITHUB/slowloris - Google Chrome
ssh.cloud.google.com/projects/zinc-compiler-311304/zones/asia-southeast1-a/inst

```
[24-04-2021 02:15:30] Sending keep-alive headers... Socket count: 1000
[24-04-2021 02:15:30] Sending keep-alive headers... Socket count: 1000
[24-04-2021 02:15:30] Sending keep-alive headers... Socket count: 1000
[24-04-2021 02:15:30] Sending keep-alive headers... Socket count: 1000
```

krisnaaditama7@krs2: ~/GITHUB/slowloris - Google Chrome
ssh.cloud.google.com/projects/zinc-compiler-311304/zones/asia-southeast2-a/inst

```
[24-04-2021 02:13:16] Sending keep-alive headers... Socket count: 1000
[24-04-2021 02:13:16] Sending keep-alive headers... Socket count: 1000
[24-04-2021 02:13:17] Sending keep-alive headers... Socket count: 1000
[24-04-2021 02:13:17] Sending keep-alive headers... Socket count: 1000
```

Fig. 11. Slowloris attack with 1000 connections from 2 PCs

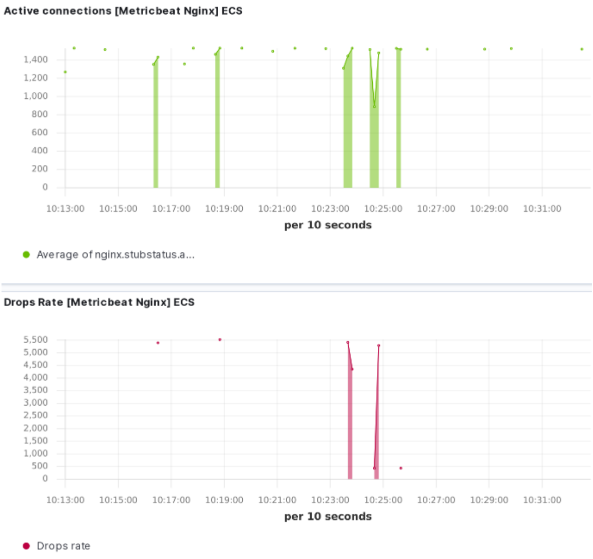
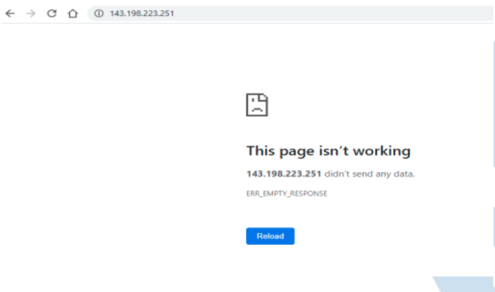


Fig. 12. Web Server Log Results in Scenario 3


143.198.223.251

This page isn't working

143.198.223.251 didn't send any data.

ERR_EMPTY_RESPONSE

Fig. 13. Inaccessible Web page

From the observations shown in Fig. 12, it can be seen that the web server has difficulty handling the many incoming requests collectively so that the service does not run well, the log graph indicator is not recorded normally and the web page display is not available as shown in Fig. 13

In the last attack scenario, although the total attack was 2 times larger than the previous one, CPU Usage and Memory Usage did not increase significantly, at 29.462% and 88.445% as shown in Fig. 14, this is due to the web server service that is not running to handles the number of other incoming connections. So that there is no visible increase in the number of cpu and memory usage even though the system is experiencing problems, where web pages cannot be accessed.

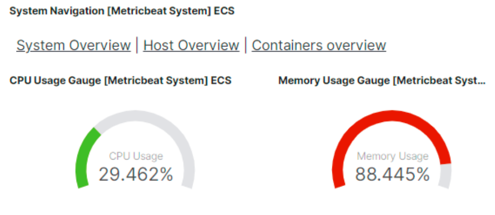

System Navigation [Metricbeat System] ECS
System Overview | Host Overview | Containers overview
CPU Usage Gauge [Metricbeat System] ECS
Memory Usage Gauge [Metricbeat Syst...
CPU Usage
29.462%
Memory Usage
88.445%

Fig. 14. CPU Usage and Memory Usage in Scenario 3

Based on Table IV, the web server test shows that the web service is still running and accessed normally when the number of incoming requests is 500 connections, when the number of requests given reaches 1000 connections most of the connection requests are still able to be served even though there are several times the request fails to be handled, due to limited resources. In testing with a total of 2000 requests, then the web server was unable to serve its services, resulting in the web page not functioning at all. This means that the Nginx web server resource by default is only able to handle requests for no more than 1000 simultaneous incoming connections, in the experimental conditions on the server specifications that are made as in Table I. In general Nginx as a web server is better able to cope with types of attacks such as Slowloris, but further configuration is needed to be able to serve more connections to the server.

TABLE IV. ATTACK IMPACT

Scenario	Total	Impact of Web Server
1	500 Connection	Service running normally
2	1000 Connection	Several times the service stopped
3	2000 Connection	Service stopped running

B. Attack Origin Testing

The second test aims to see the log data of the detected attack origin, in the form of the country of origin of the attack and attempts to access server logins from outside parties. The [Filebeat Nginx] section of the Elastic Common Schema (ECS) Overview, is capable of displaying a source map and details of the origin of the attack. From the resulting display as shown in Fig. 15, it was identified that the attack that was launched came from the Singapore area, this is in accordance with the experimental scenario where the attacker's PC used were from that region.

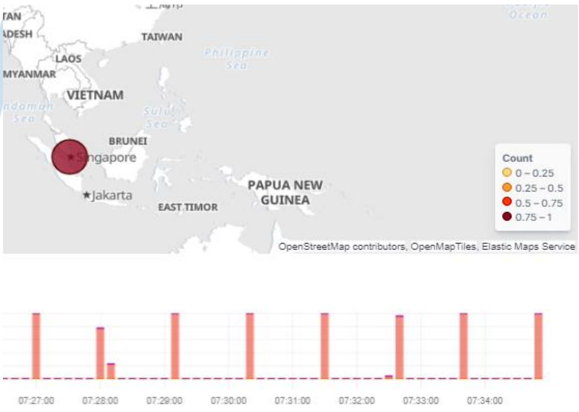


Fig. 15. Map of the origin of attack

Testing of login attempts to the server were carried out by entering a username and password at random to see if the Elastic Stack can record these attempt, statistical results can be seen in the [Filebeat System] section of ECS SSH login attempts. From the graphic shown in Fig. 16, there are a lot of attacks from outside parties trying to enter the system, whose IP addresses were detected from various countries. It can be seen that the largest number of attacks came from China territory. This proves earlier allegations that many of the attacks originated in that region.

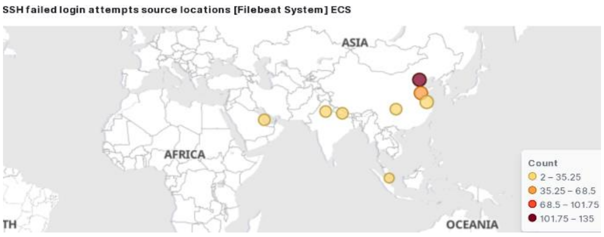


Fig. 16. Map of access for server logins

Several login attempts were recorded every time with various random user combinations, as the sample shown in Table V, these records are always updated on the [Filebeat System] SSH login attempts ECS.

TABLE V. SSH LOGIN ATTEMPTS

Time	user.name	source.ip
Apr 24, 2021@ 10:14:07.000	admin	161.97.183.201
Apr 24, 2021@ 10:14:10.000	camera	103.150.136.128
Apr 24, 2021@ 10:14:15.000	bertrand	42.51.9.162
Apr 24, 2021@ 10:14:16.000	user1	14.02.74.99
Apr 24, 2021@ 10:14:18.000	bertrand	42.51.9.162

IV. CONCLUSION

The results obtained are the process of measuring and analyzing server capabilities is made easier by using Elastic Stack in log management, with statistics and visuals generated. Based on the test results, a server with specifications of 2 CPUs and 4GB of memory is able to handle requests of 500 to 1000 connections, but the service will have difficulty until it stops running when the total incoming requests reaches 2000 connections. The experimental results show that the CPU Usage obtained when the attack was carried out with scenarios of 500 and 1000 connections was 26% and 31%, respectively. However, when the attack is carried out with a scenario of 2000 connections, the web server service is unable to serve requests and the CPU Usage is 29%. As for Memory Usage, every increase in connection requests results in an increase in memory consumption from 86% to 88%. Another result is that the source of the attack is identified according to the domicile of the attacker's simulated PC, namely Singapore. In addition, there is also a list of login access, both internal and external, using random users and passwords to enter the server system. The addition of the login authentication feature on the Elastic Stack and the use of Nginx as a web server is highly recommended to minimize the impact of the Slowloris attack.

REFERENCES

[1] C. Tarigan and D. Angela, “Sistem Pengawasan Kinerja Jaringan Server Web Apache dengan Log Management System ELK (Elasticsearch , Logstash , Kibana),” *J. Telemat. Ed. Ind. Eng. Semin. Call Pap.*, vol. 1, no. 1, pp. 7–14, 2018.

[2] A. F. Rochim, M. A. Aziz, and A. Fauzi, “Design Log Management System of Computer Network Devices Infrastructures Based on ELK Stack,” *ICECOS 2019 - 3rd Int. Conf. Electr. Eng. Comput. Sci. Proceeding*, no. May 2020, pp. 338–342, 2019, doi: 10.1109/ICECOS47637.2019.8984494.

[3] D. Berman, “The Complete Guide to the ELK Stack,” 2020. https://logz.io/learn/complete-guide-elk-stack/.

[4] J. Ellingwood and V. Kalsin, “How To Install Elasticsearch, Logstash, and Kibana (Elastic Stack) on Ubuntu 18.04,” 2018. https://www.digitalocean.com/community/tutorials/how-to-install-elasticsearch-logstash-and-kibana-elastic-stack-on-ubuntu-18-04.

[5] K. Lotz, “Integrating the Elastic Stack into ExplorViz to Collect Logs and Runtime Metrics,” 2019.

[6] T. Diotalevi *et al.*, “Collection and harmonization of system logs and prototypal Analytics services with the Elastic (ELK) suite at the INFN-CNAF computing centre,” *Proc. Sci.*, vol. 351, pp. 0–14, 2019, doi: 10.22323/1.351.0027.

[7] S. Alghfeli, Z. Alhadrami, M. Alghfeli, N. Albloushi, and A. Alfaresi, “Bayyinah, A Log Analysis Forensics Tool,” *Proc. - 2019 Amity Int. Conf. Artif. Intell. AICAI 2019*, no. Ii, pp. 845–849, 2019, doi: 10.1109/AICAI.2019.8701405.

[8] M. Harikanth and P. Rajarajeswari, “Malicious event detection using ELK stack through cyber threat intelligence,” *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 7, pp. 882–886, 2019.

[9] W. Sholihah, S. Pripambudi, and A. Mardiyono, “Log Event Management Server Menggunakan Elastic Search Logstash Kibana (ELK Stack),” *JTIM J. Teknol. Inf. dan Multimed.*, vol. 2, no. 1, pp. 12–20, 2020, doi: 10.35746/jtim.v2i1.79.

[10] M. Bajer, “Building an IoT data hub with elasticsearch, Logstash and Kibana,” *Proc. - 2017 5th Int. Conf. Futur. Internet Things Cloud Work. W-FiCloud 2017*, vol. 2017-Janua, no. August 2017, pp. 63–68, 2017, doi: 10.1109/FiCloudW.2017.101.

[11] S. M. Helalat, “An Investigation of the Impact of the Slow HTTP DOS and DDOS attacks on the Cloud environment,” Blekinge Institute of Technology, 2017.

[12] A. Y. Chandra, “Analisis Performansi Antara Apache & Nginx Web Server Dalam Menangani Client Request,” *J. Sist. dan Inform.*, vol. 14, no. 1, pp. 48–56, 2019, doi: 10.30864/jsi.v14i1.248.

[13] M. Arman, “Metode Pertahanan Web Server Terhadap Distributed Slow HTTP DoS Attack,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 7, no. 1, pp. 56–70, 2020, doi: 10.35957/jatisi.v7i1.284.

[14] Y. Li *et al.*, “A cloud-based framework for large-scale log mining through Apache Spark And Elasticsearch3,” *Appl. Sci.*, vol. 9, no. 6, 2019, doi: 10.3390/app9061114.

[15] F. Abd. Hadi, M. Ahyar, and I. Syamsuddin, “ELK : Teknologi Mesin Pencari Big Data Terdistribusi,” *Semin. Nas. Tek. Elektro dan Inform. 2018*, no. September, pp. 333–338, 2018.

[16] V. K. Et. al., “Twego Trending: Data Analytics Based Search Engine Using Elasticsearch,” *Turkish J. Comput. Math. Educ.*, vol. 12, no. 1S, pp. 246–251, 2021, doi: 10.17762/turcomat.v12i1s.1764.

[17] K. Yamnual, P. Phunchongharn, and T. Achalakul, “Failure detection through monitoring of the scientific distributed system,” *Proc. 2017 IEEE Int. Conf. Appl. Syst. Innov. Appl. Syst. Innov. Mod. Technol. ICASI 2017*, pp. 568–571, 2017, doi: 10.1109/ICASI.2017.7988485.