PERFORMANCE EVALUATION OF ELK STACK VERSUS GRAYLOG AS OPEN-SOURCE LOG MANAGEMENT TOOLS

Roxana-Daniela ȘUȘTIC[1], Alexandra MORARU[2], Andrei-Bogdan RUS[1], Virgil DOBROTA[1]
[1]*Communications Department, Technical University of Cluj-Napoca, Romania*
[2]*Frequentis Romania, Cluj-Napoca, Romania*
Corresponding author: Virgil Dobrota (e-mail: Virgil.Dobrota@com.utcluj.ro)

Abstract: This paper proposes a comparison between two log management tools, namely ELK Stack and Graylog, used as proactive measures for keeping track of changes in a network environment. Several experiments were based on performance metrics such as response time, CPU and memory usage, as well as on testing the security, alerting, monitoring, and data visualization capabilities. Performance-wise, ELK was the best solution due to its stability, faster response time and reliability under stress conditions. Graylog, on the other hand, was an easier to maintain, fitting the needs of a centralized log management, being more suitable for security and alerting purposes.

Keywords: *Containerization; Docker; ELK Stack; Graylog; log management; Virtualization.*

I. INTRODUCTION

In the so-called Zettabyte Era, Internet usage continues to grow and technology starts playing an even more important role in our lives. It is crucial to keep tabs of relevant events occurring online or locally on our computers. All software applications and the infrastructures on which they run generate log files which include information on every action that took place in the system. From databases and enterprise applications to firewalls and routers, wireless access points and Voice over Internet Protocol (VoIP) gateways, logs are being spewed forth at an ever-increasing pace.

Immense volumes of logging data are generated daily, resulting in a need to store, filter, and search all this. Reviewing the information needs to happen both proactively, in search of potential risks and future problems and reactively, following incidents that require immediate intervention [1]. Therefore, instead of considering log files as separate, unrelated entities, the solution is using a management tool to centralize them and get visibility into the health of the applications.

Currently, there are multiple open-source and commercial logging solutions present on the market. For a company that manages a fair number of recorded information per day, there are so many options to be considered. Thus it becomes a tedious task to choose which one suits the best. There is also the possibility that the implemented log management tool may not have the desired functionalities or has so many extra functionalities it becomes difficult to work with. Therefore, when choosing the right tool the consumer should figure out whether the platform is more suited for a small organization looking to get the basic data out of their logs, or they plan to upgrade to enterprise which will require more powerful and efficient tools to tackle large scale.

The rest of the paper is organized as follows: Section II discusses the related work, followed by the implementation. Section IV presents the experimental results. Last section includes conclusions and future work.

II. RELATED WORK

Paper [2] describes ELK Stack, the acronym referring to three open source projects: Elasticsearch, Logstash, and Kibana. As a centralized log management system it allows users to collect structured or unstructured data from any source, and analyze, search and monitor that data in real-time. Due to its wide variety of plugins, ELK Stack can be used for monitoring a large selection of data, apart from logs. For instance in [3] the sentiment analysis of social networking data was performed, whilst in [4] Internet of Things data was integrated into an Elasticsearch based implementation for processing. A different open-source logging solution is presented in [5], where Graylog is used to store and analyze log information of Linux endpoints in order to gain better visibility of the clients. There are also commercial tools, such as Splunk, presented in [6] and used to facilitate real-time data collection from IoT devices installed in a greenhouse. Paper [7] discusses about a commercial logging platform called Logentries to proactively monitor cloud performance.

A recent survey [13], regarding the best log management and analysis tools in 2022, placed Graylog as 4th seed of the top, being offered both as virtual appliance or as Software-as-a-Service (SaaS) platform. It was a surprise from our point of view that Logstash and Kibana (parts of ELK Stack) appeared on the 14th and respectively the 15th position.

Our vision is that choosing the most suitable solution becomes dependent on project requirements and intended

Manuscript received March 13, 2022; revised June 20, 2022

purpose. Also, the "technical culture" of the company could influence the decision. Therefore our approach herein was a bit unusual as we implemented in parallel both solutions, allowing them to run simultaneously (see Figure 1). We do not pretend that this is suitable for all projects, but in our IoT environment we wanted to take benefits from both ELK Stack and Graylog, as they proved to be complementary. Also we used each of them to validate the results provided by the other one.

III. IMPLEMENTATION

A log management tool is a game-changer when it comes to keeping track of all parameter modifications in the IT environment. By providing real-time system monitoring and data visualization capabilities, it facilitates the early detection of malicious behavior and it provides better insights into the processes that take place. Setting up such a tool is a proactive measure that any business should consider. However, some basic planning is required before choosing the logging solution. The type of logs generated by the infrastructure and their overall volume are some of the things to take into account. An enterprise should also set a budget, evaluate how much time they are willing to allocate for the setup and maintenance, and establish a log management policy. Nonetheless, there are so many open-source log management tools present on the market that businesses may face difficulties when choosing which one suits the logging needs the most. The motivation behind this paper consisted of solving these difficulties by providing a pertinent comparison between two of the most popular open-source logging solutions.

Due to the comparative nature of the paper, the two open-source log management tools, ELK Stack and Graylog were implemented in parallel, by providing a better simultaneous overview of the system. Overall this approach improved the visibility of the processes running in the background. The testbed scenario is presented in Figure 1.

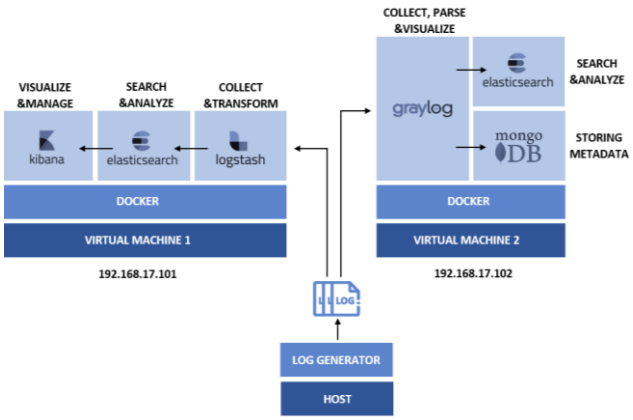

Figure 1. Testbed scenario with two log management tools working simultaneously

A. Building the virtual network

The implementation includes two Linux-based virtual machines (VM1 and VM2) and one host, communicating with each other and having Internet access. The installation and configuration of the virtual machines was done using VMware Workstation 15 Player, a virtualization software running on a 64-bit Windows 10 host. Community Enterprise Operating System (CentOS) 7 was chosen due to the compatibility with its upstream source Red Hat Enterprise Linux (RHEL), preferred by most of the software companies for development and production servers. Both virtual machines were configured with 20GB hard disk capacity, 1GB RAM, 1 CPU, two network adapters. Because it enables Internet access, the Network Address Translation (NAT) network adapter was used for software installation purposes. The problem with NAT is that the virtual machine does not have its own IP address on the external network, making it impossible to be accessed remotely. To solve this issue, an additional Host-only network adapter was used to allocate a static IP to the virtual machine. After configuring the IP addresses of the two virtual machines, remote access was achieved using an open-source remote connections manager called mRemoteNG.

B. Setting up the Docker platform

Docker platform was installed to maximize the use of the host machine's physical resources. The installation process of Docker engine was carried in parallel for both virtual machines and consisted of setting up the Docker repository. Docker is an open-source containerization platform that streamlines the software development lifecycle, allowing developers to package applications into containers, fully packaged and portable computing environments that share the kernel of the host operating system. The Docker engine uses a client-server architecture containing a server represented by a background process called daemon, a command line interface (CLI) client, and a REST API interface used by the CLI to communicate with the daemon. The Docker daemon manages Docker objects such as images, containers, networks, and volumes and listens for API requests from the client. Docker images are read-only files containing instructions for the creation of Docker containers. Images can be pulled from or pushed to a public registry called Docker Hub or the configured registry of choice. The containers are the run-time instances of the Docker images. They can be created, started, stopped, moved, or deleted using the CLI, being thoroughly isolated from one another and the host machine [8].

C. Installing and configuring ELK Stack

Following the setup of Docker engine, ELK Stack was installed by pulling the official Elastic Docker images of Elasticsearch, Logstash, and Kibana from the Docker hub repository and deploying them as containers on one of the virtual machines (see Figure 2).

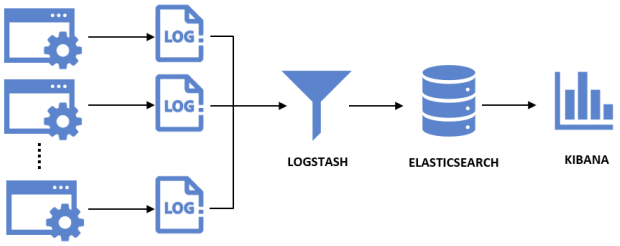

Figure 2. ELK Stack architecture

ELK Stack is one of the most popular open-source log management tools designed to collect, centralize, analyze, and visualize data of any format, from any type of source Elasticsearch is an open-source distributed search engine based on Apache Lucene, providing real-time analysis for both structured and unstructured data. Complex data structures are stored in it as schema-less JSON documents, being indexed and fully searchable in near real-time [9]. Due to its distributed nature, Elasticsearch is capable of scaling and adapting to any ingested volume of data. Generally, its configuration comprises a cluster with one or multiple nodes, where each node refers to a single instance of Elasticsearch. As nodes are added or removed from the configuration, the cluster reorganizes itself to spread the data evenly [10]. Elasticsearch is also equipped with a comprehensive REST API acessible through ports 9200 and 9300 for client communication. For the installation the Elasticsearch v6.8.5 Docker image was pulled from the Docker Hub and ran as a container. As it comes with good defaults, Elasticsearch configuration was not required for the implementation.

Kibana is an open-source web interface used for visualizing and exploring Elasticsearch data in a user-friendly manner. It also represents a useful tool for monitoring and managing the entire ELK Stack. By default, Kibana's browser-based interface is reachable over port 5601. Its installation was performed by pulling the Kibana v6.8.5 Docker image from the repository and deploying it in a container linked to the Elasticsearch container. Kibana also required very little configuration, mainly for removing unnecessary User Interface (UI) plugins.

Logstash is an open-source data collector with real-time pipelining capabilities that centralizes logs and other types of events from disparate sources, parses the collected data, and distributes it to the destinations of choice. Events are processed in a three-stage pipeline – input, filter and output. Input plugins enable Logstash to read events from specific sources such as text-based files, standard input, TCP/UDP sockets and HTTP API endpoints. Filter plugins perform the intermediary processing of events, parsing and transforming the data received from the input. Output plugins represent the final stage in the event pipeline, enabling Logstash to send the parsed log data to specific destinations. Usually, events are sent to Elasticsearch but data can also be forwarded to archiving tools, monitoring solutions, alerting systems, or databases. For its installation, the Logstash v6.8.5 Docker image was pulled from the repository, deployed as a container and linked with Elasticsearch. Port 9500 was forwarded for TCP connections, later being used as the input port for the incoming log stream. A `logstash.conf` configuration file was created for collecting, parsing and forwarding the incoming stream of logs. The input of the configuration file used a tcp plugin to collect data received on port 9500 and assign it a type. In the filter part, there are several plugins: (1) a comma-separated values (CSV) one parses the raw text and separates it into columns; (2) a mutate one eliminates the spaces that appear in the columns; and (3) a date one translates the date and time as the timestamp of the event. After the logs are parsed, Logstash forwards them to Elasticsearch via port 9200.

D. Installing and configuring Graylog

To install the Graylog log management tool, Docker images of MongoDB, Elasticsearch, and Graylog were used in a similar manner to the ELK Stack installation. Graylog is an open-source log management solution used for collecting, processing, and visualizing both structured and unstructured data from a wide variety of sources. It uses Elasticsearch for storing, indexing, and searching purposes and a MongoDB database for managing metadata such as user, settings, and configuration data. Figure 3 presents the minimal setup of a Graylog architecture.

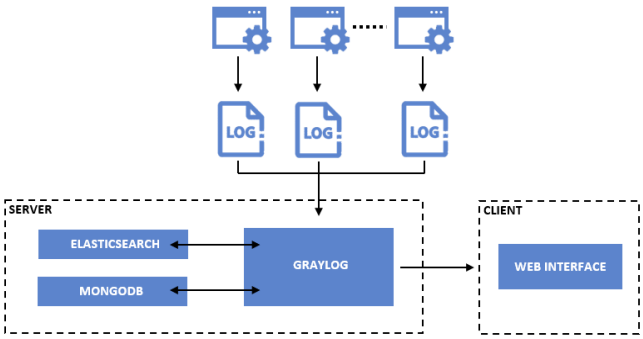

Figure 3. Graylog architecture

Graylog consists of the main server that receives data and a web interface used for querying, analyzing, and visualizing indexed data as well as configuring the environment. The web interface is reachable through port 9000 by default, fetching data through HTTP requests from the REST API of the Graylog server [11].

After pulling the MongoDB v3 and Elasticsearch v6.8.5 Docker images from Docker Hub, the latest version of Graylog v3.2 was installed and deployed in a container having the Mongo and Elasticsearch containers as dependencies. Besides its default ports 9000 for accessing the web interface, 12201 for GELF input and 1514 for Syslog input, port 5555 was forwarded for the TCP input. Because the web interface is accessed remotely from the host's browser the server IP was defined as the virtual machine's static IP address.

The configuration of Graylog is mainly performed on its web interface, however, a `graylog.conf` configuration file is also used for more advanced settings. The first step in the

configuration process was launching a Raw/Plaintext TCP input on the current node for receiving the stream of logs. After starting the input, the logs are coming through as raw, unprocessed message blocks. For processing, a pipeline rule was configured as well. The configured pipeline splits the unprocessed message block of the logs into individual fields separated by the `|` separator and replaces the timestamp generated by Graylog with the timestamp of the event.

E. Developing the logging application

For generating the stream of logs, a Java application was developed based on the Apache Log4j2 logging framework. Given the importance of logging for both audit and debugging purposes, choosing a reliable logging library is essential. Log4j2 is the most popular logging framework used by IT companies for their Java applications. The easiest way of logging with Log4j2 is by using an Apache Maven build. The Integrated Development Environment (IDE) used for building the logging application is IntelliJ IDEA Community Edition 2020.1, a free and open-source Apache platform for developing Java applications. Four appenders were defined in the logging framework configuration file, namely a Console appender for appending log events to the IDE console, a File appender for writing the logs in a `.log` file, and two Socket appenders for sending the logs via TCP to the log management tools. Logs are generated in an infinite loop having the frequency set by the user. An UUID was added to the message body for distinguishing between two or more logs having the same severity level and message.

IV. EXPERIMENTAL RESULTS

As presented in the previous chapter, a parallel approach was the best solution for performing the comparative analysis of the two open-source log management tools. Several tests were conducted on both logging solutions for measuring their durability, performance, and overall usability as well as testing features such as security, alerting, and visualization capabilities. For obtaining a pertinent conclusion, the comparison aimed solely the out-of-the-box functionalities of the two log management tools with no additional software being used besides the default components. Both front-end and back-end functionalities were taken into consideration when performing the comparative analysis.

A. CPU and memory usage testing

To measure the CPU and memory usage of the evaluated log management tools, two throughputs were considered – namely, the average throughput of a medium-sized enterprise ingesting roughly around 100GB of logs per day and a maximum throughput of 1TB/day for stress testing the systems. The throughputs are adjusted by modifying the frequency with which the logs are generated, defined by the `Thread.sleep()` instruction in the logging application.

For this experiment, the `docker stats` command was used to measure the ELK Stack and Graylog containers performance metrics. The command returns the Docker containers CPU and memory usage as a percentage of the hosting base CPU and memory capacity which is one i7-10510U CPU @1.80GHz and 2GB RAM for each virtual machine. The following snippet was executed to write the output of the docker stats command in a .txt file every one minute:

```
while true; do docker stats --no-stream --
format "table
{{.Name}}\t{{.CPUPerc}}\t{{.MemPerc}}" |
tee  --append cpumem.txt; sleep 60; done
```

In a testing period of one hour, 60 samples were collected and the average of five samples per five minutes was calculated for obtaining the results. At 100GB/day, Figure 4 and Figure 5 display the graphical representation of the CPU and memory usage, respectively.

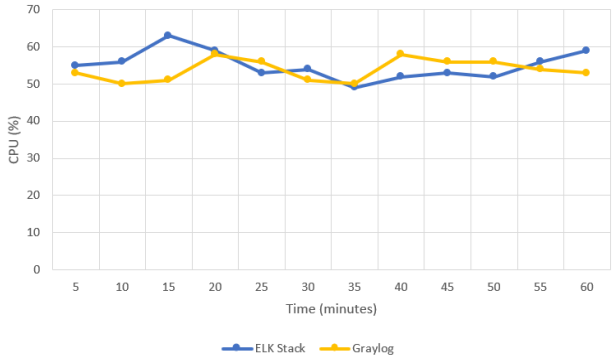

Figure 4. CPU usage of both log management tools at 100GB/day

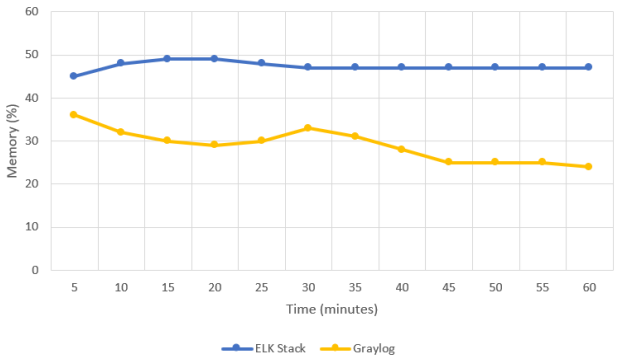

Figure 5. Memory usage of both log management tools at 100GB/day

Regarding CPU usage both ELK Stack and Graylog have similar performances, maintaining an average between 50% and 60%. Regarding memory usage, the ELK Stack is 1.6 times more memory-consuming than Graylog with Elasticsearch maintaining an average usage of 50% out of the total 2GB RAM. For stress conditions, approximately three millions of logs were injected into both log management solutions for a testing period of one hour. The graphical representations of the CPU and memory usage for

this situation are in Figure 6 and, respectively, in Figure 7.

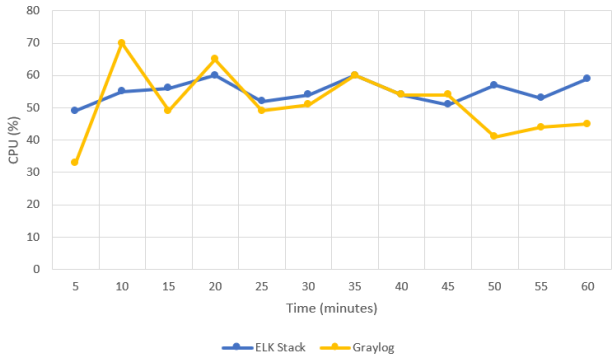

Figure 6. CPU usage of both log management tools at 1TB/day

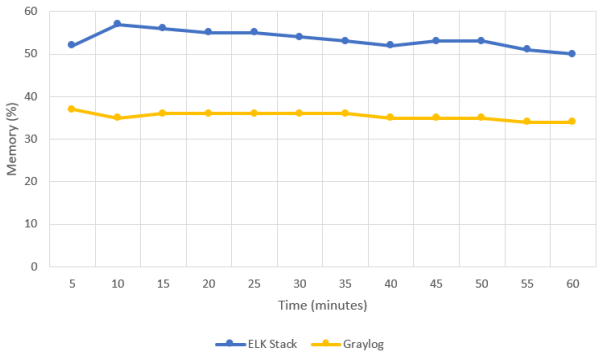

Figure 7. Memory usage of both log management tools at 1TB/day

Following the experiment, it can be observed that ELK Stack has a steadier CPU usage curve, maintaining its 50%-60% average while Graylog is quite unstable, alternating between low and high values. Due to the significantly larger quantity of ingested logs, Graylog also experienced web interface timeouts and a data visualization latency of one hour. Regarding memory usage, both log management tools sustained a 10% growth. Although ELK Stack remains 1.6 times more memory-consuming, it is more stable and has an overall better performance providing real-time visualization of data.

B. Response time testing

Another important factor in testing the performance of a log management tool is the response time. The response time measures how long the system takes to respond to a received request. To measure the REST APIs response time of both ELK Stack and Graylog, a Maven application was developed that uses the REST-assured library, a Java DSL for testing REST services. The idea behind the response time test is sending a log with a specific UUID and the message "This is a test message", submitting a query request for that UUID and measuring the time it takes to fetch the corresponding JSON that contains the given message. The test passes if the response time is faster than 10 milliseconds and fails otherwise, returning an error containing the exact value of the response time. To obtain results closer to real-case scenarios, the test was conducted in parallel with the logging application configured to generate the medium throughput of 100GB/day. For this experiment, an average of 60 samples was calculated for each of the log management tools. The results demonstrate that ELK Stack is 1.5 times faster than Graylog, measuring a response time of 73 milliseconds, while Graylog averages around 110 milliseconds.

C. Security functionality testing

As logs often contain sensitive data or confidential information about the infrastructure of a software company, securing the log management tool is necessary to restrict unauthorized access and prevent data breaches. Password protection, role-based access control and data encryption are some of the methods used to secure a logging solution.

Regarding security functionality, both log management tools allow password protecting sensitive data and customizing users and roles for personalized access. ELK Stack basic license provides security features through the X-pack extension, requiring manual activation and extensive configuration while Graylog comes with authentication credentials by default. Both log management tools share some configuration steps as they use the same search and analytics engine, however, Graylog is the better choice because the setup is faster and most configurations are performed on the user interface.

D. Alerting functionality testing

Alerting is an important part of any log management tool for detecting abnormal activities such as malicious attacks or failures in the system. Alerts are background tasks that run periodically to detect specific conditions. When a condition is met, the alert triggers an action that may involve integrations with third-party applications. The most common type of notification triggering is by sending an email.

Although ELK Stack is an open-source log management tool, alerting features are enabled only for the paid subscriptions. There is a possibility to integrate third-party plugins like ElastAlert for alerting purposes, however, this alternative was not taken into consideration as the comparison was intended to aim solely the out-of-the-box functionalities of the two log management tools. A different approach consisted in adapting Logstash to trigger notifications by configuring an email output, but this method is far-fetched and not recommended outside testing scenarios as it produces email flooding. In this context, Graylog is the best solution as it comes with alerting capabilities by default and configuration is done mainly on the web interface. Event definitions can be created in the Alerts page of the web interface, running periodically and searching for the imposed condition, triggering an email notification to the specified email address if the condition is met.

E. Monitoring functionality testing

Monitoring is vital for keeping up with changes in the system, observing its overall health and following important indicators that reveal whether or not a vertical or horizontal scale out is needed. ELK Stack provides extensive monitoring capabilities by means of an interactive user interface that offers in-depth details about the runtime metrics of the stack. Graylog, on the other hand, lacks the centralized monitoring a user interface can provide and relies on bash commands to access internal metrics. Taking everything into account, ELK Stack seems to be a better option for monitoring purposes.

F. Data visualization functionality testing

Visualization of data is probably one of the most important aspects of a log management tool. Analyzing queries and understanding the results requires a certain degree of domain knowledge. By using visualizations, specialists can define the search query once and display the results in an easy to understand way for non-technical departments like sales or marketing.

ELK Stack provides impressive data visualization features through Kibana. Basic charts, data tables, metrics, maps, time series visualizations and markdown widgets are just some of the available visualization types. Dashboards are interactive, allowing filtering to drilldown through the layers of detail. Persistence is a key feature of Kibana as dashboards and their corresponding visualizations can be imported and exported, a very important aspect when upgrading or reinstalling the stack.

Graylog also supports a large variety of visualizations called widgets, although not as many as ELK Stack. Dashboards are customizable but lacking persistence as import and export is not supported and they can only be shared among users in the Graylog environment. Also, the overall process of creating and managing a dashboard is not as user-friendly and interactive as it is in ELK Stack case. An interesting feature present in Graylog that ELK Stack lacks is storing the time with the dashboard. All things considered, both log management tool have great data visualization capabilities but ELK Stack is better due to its wide variety of visualizations, flexibility and persistence.

V. CONCLUSIONS AND FUTURE WORK

Both log management tools analyzed herein, ELK Stack and Graylog, have a similar set of basic features. Choosing the right solution is completely based on the requirements and logging needs of the consumer. ELK Stack is a reliable solution, providing striking real-time visualizations, complex query capabilities and scalability for matching any volume of data. On the other hand, Graylog is an easy to maintain solution that fits the needs of centralized log management, providing out-of-the-box functionalities that require little configuration.

Future work may involve a deeper analysis of features not discussed in this paper, such as encryption of data, archiving capabilities or index management. Also, multiple input sources can be integrated for researching the processing capabilities of the log management tools for a wider variety of logs. For gaining insights closer to real-case scenarios, the virtual network can be expanded and more Elasticsearch clusters can be added to simulate the enterprise IT environment. Further experiments may target newer versions of the log management solutions and explore the broad spectrum of community-built add-ons.

ACKNOWLEDGMENT

An initial expanded version of this work was presented by R.D. Sustic as B.Sc. thesis in Telecommunications Technologies and Systems at Technical University of Cluj-Napoca in 17 July 2020. We acknowledge the support provided by Frequentis Romania to perform the experiments.

REFERENCES

[1] B. Williams and C. Anton, "Logging and monitoring in depth," in “CI Compliance: Understand and Implement Effective PCI Data Security Standard Compliance”, Edition 4, Syngress, 2014, p. 386.
[2] F. Ahmed, U. Jahangir, H. Rahim, K. Ali and D. -e. -S. Agha, "Centralized Log Management Using Elasticsearch, Logstash and Kibana," 2020 International Conference on Information Science and Communication Technology (ICISCT), 2020, pp. 1-7, doi: 10.1109/ICISCT49550.2020.9080053.
[3] D. Bhatnagar, R. J. SubaLakshmi and C. Vanmathi, "Twitter Sentiment Analysis Using Elasticsearch, LOGSTASH And KIBANA," 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), 2020, pp. 1-5, doi: 10.1109/ic-ETITE47903.2020.351.
[4] M. Bajer, "Building an IoT Data Hub with Elasticsearch, Logstash and Kibana," 2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), 2017, pp. 63-68, doi: 10.1109/FiCloudW.2017.101.
[5] T. Talikka, Implementing Linux support and monitoring in a workplace, Helsinki, Finland: Metropolia University of Applied Sciences - Bachelor’s Thesis, 2019.
[6] Y. Chen and H. Chien, "IoT-based green house system with splunk data analysis," 2017 IEEE 8th International Conference on Awareness Science and Technology (iCAST), 2017, pp. 260-263, doi: 10.1109/ICAwST.2017.8256458.
[7] B. Siniarski, C. Olariu, P. Perry, T. Parsons and J. Murphy, "Real-time monitoring of SDN networks using non-invasive cloud-based logging platforms," 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016, pp. 1-6, doi: 10.1109/PIMRC.2016.7794973.
[8] "Docker online documentation: Docker overview", Docker 2022, [Online], Available: https://docs.docker.com/get-started/overview/.
[9] Elastic, "Elasticsearch Documentation," [Online] Available: https://www.elastic.co/guide/index.html.
[10] C. Gormley and Z. Tong, Elasticsearch - The definitive guide: A Distributed Real-Time Search and Analytics Engine, USA: O'Reilly Media, 2015.
[11] "Graylog Documentation" Graylog 2022, [Online], Available: https://docs.graylog.org/en/3.3/index.html.
[12] R. Sustic, “ELK Stack vs. Graylog - A comparative analysis of two open-source log management tools”, B.Sc. Thesis, Technical University of Cluj-Napoca, 17 July 2020.
[13] S. Cooper, “17 Best Log Management & Analysis Tools”, Comparitech 2022, [Online], Available: https://www.comparitech.com/net-admin/log-management-tools/.