Enhancing Security Monitoring with AI-Enabled Log Collection and NLP Modules on a Unified Open Source Platform

1st Yehor Safonov
Department of Telecommunications
FEEC, Brno University of Technology
Brno, Czech Republic,
0000-0002-3549-2178
yehor.safonov@vutbr.cz

2nd Michal Zernovic
Department of Telecommunications
FEEC, Brno University of Technology
Brno, Czech Republic
xzerno00@vutbr.cz

***Abstract*—The number of computer attacks continues to increase daily, posing significant challenges to modern security administrators to provide security in their organizations. With the rise of sophisticated cyber threats, it is becoming increasingly difficult to detect and prevent attacks using traditional security measures. As a result, security monitoring solutions such as *Security Information and Event Management* (SIEM) have become a critical component of modern security infrastructures. However, these solutions still face limitations, and administrators are constantly seeking ways to enhance their capabilities to effectively protect their cyber units. This paper explores how advanced deep learning techniques can help boost security monitoring capabilities by utilizing them throughout all stages of log processing. The presented platform has the potential to fundamentally transform and bring about a significant change in the field of security monitoring with advanced AI capabilities. The study includes a detailed comparison of modern log collection platforms, with the goal of determining the most effective approach. The key benefits of the proposed solution are its scalability and multipurpose nature. The platform integrates an open source solution and allows the organization to connect any event log sources or the entire SIEM solution, normalize and filter data, and use this data to train and deploy different AI models to perform different security monitoring tasks more efficiently.**

***Index Terms*—Artificial intelligence, deep learning, Fluentd, log collection, log processing, Logstash, security monitoring, SIEM.**

I. Introduction

In today's world, it is impossible to imagine any modern field without a computer infrastructure [1]. Given its ubiquity, ensuring uninterrupted and secure operation of these systems has become a top priority [2], [3]. Various hardware and software solutions are utilised inside cyberspaces to achieve their smooth functioning. When it comes to computer infrastructure, there are four main categories of elements that could be found: *Networking Devices* (such as routers, switches, and firewalls), *Operating Systems* (including Unix-based and Windows systems), *Security Applications* (such as Next Generation Firewalls (NGFW), Vulnerability Management Systems (VMS), Endpoint Detection and Response (EDR), Data Los Protection (DLP), Privileged Access Management (PAM), etc.) and *Other Applications* (such as database systems, customer applications, cloud applications, etc.) [3]. All of the above elements help to ensure the smooth running of the organisation and enable all organisational processes to be fulfilled.

These devices, along with end devices and applications, generate records of their traffic and usage, known as logs [2], [4]. They are crucial for evaluating the network's status, security, and events that occur on it. They can provide insight into potential attacks on the infrastructure or can be used to analyse previous incidents and take appropriate action [1]. The log analysis process is essential for discovering relevant information from these records. From a security perspective, all log records must be centrally analysed, normalised, and stored for the long term to ensure effective security monitoring [3]. The main challenge of this issue is the lack of standardisation on the LES (*Log Event Source*) level due to the existence of different log formats and protocols for their transmission, see Fig. 1. This complicates their central processing.

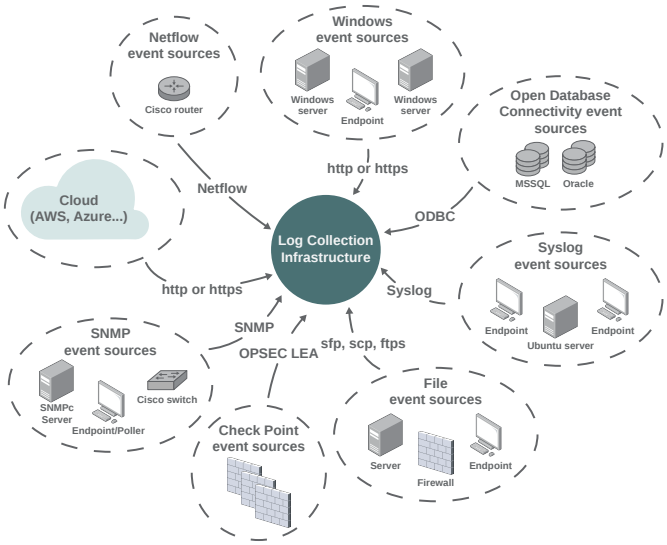

Fig. 1. Schema of centralized log collection in computing infrastructures.

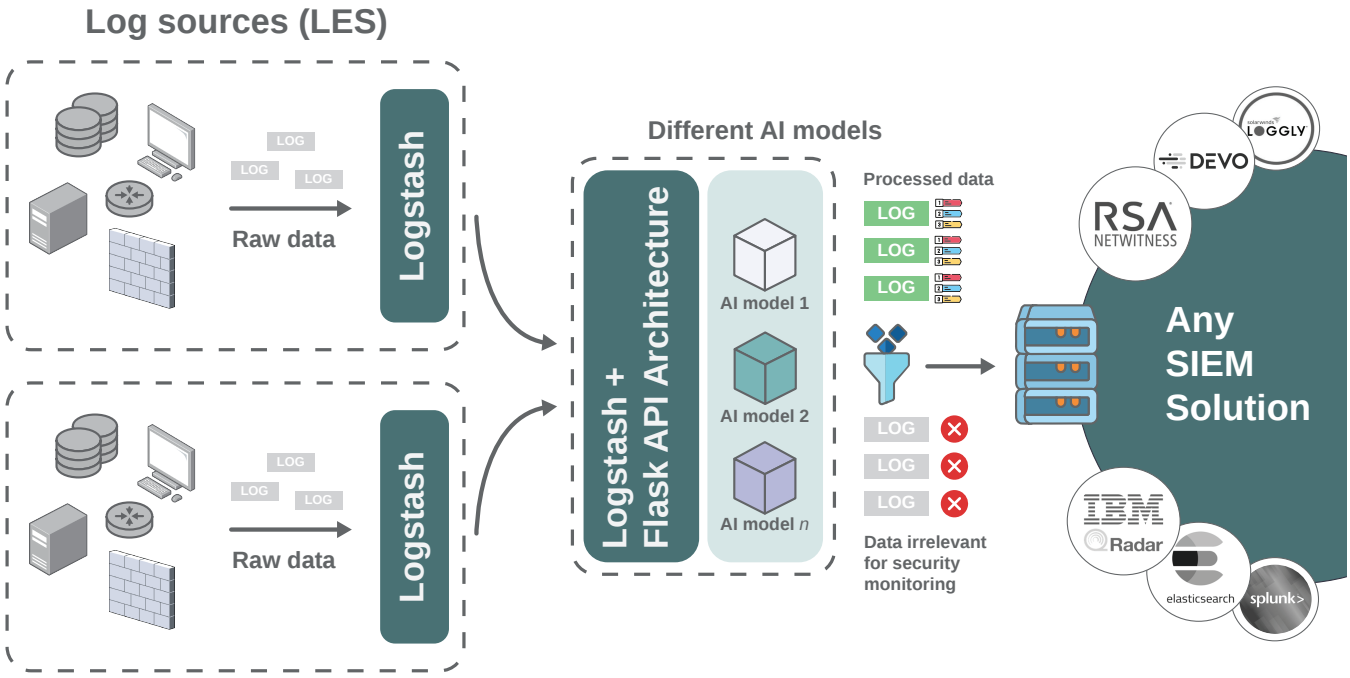

Fig. 2. A way of integration purposed platform into log processing to cooperate with SIEM systems, while also demonstrating its scalability.

In general, there are two distinct methods for providing security monitoring. Simple analysis techniques, such as manual commands to retrieve records, may be suitable for smaller infrastructures, but are not effective for larger systems [4]. In this day and age, advanced analysis techniques are offered within SIEM systems, which also allow responding to security incidents, creating rules, and notifying and preventing attacks. SIEM offers capabilities such as advanced log analysis, log storage, and other advanced features [2].

One way to enrich log processing is to embed a neural network into it. Although current proprietary SIEM solutions use neural networks to solve correlation tasks, the presented approach performs a wide range of security tasks and allows SIEM systems to be more efficient.

II. On current systems for security monitoring

SIEM is a platform or a system that is responsible for collecting security-relevant data in a centralised environment to detect threats and incidents. This system consists of two concepts: SIM is used to collect security-relevant data and generate reports. SEM provides security incident analysis, incident correlation, and real-time alerting. Together, these two categories constitute a SIEM. [2]

Before the records can be analyzed, a series of steps encompassing various SIEM functions must be completed within the SIEM tools: [3]

Log collection – can be done in different ways. Some log sources are based on a push technique (e.g. Syslog), where the user on the end device configures the destination IP address and port. The user then configures a listener on the desired device and receives the logs. Some sources in turn work on a pull technique (e.g. Windows event log), where it is necessary to connect to the machine and download the logs. [3], [5], [6]

Normalization – all log forms require specific connectors. They convert different log file formats from different devices, versions to a common SIEM understandable format. [1]

Aggregation – a process of removing duplicates. Logs are aggregated according to similar characteristics, thus opening up the possibility to discard unnecessary data. [3]

Correlation and analysis – a great advantage of these solutions are the correlation rules. A constant stream of logs from different connected devices should flow into the SIEM. Correlation rules tell the SIEM which sequence of events indicates a potential incident or vulnerability. [3]

To illustrate the concept, the following example is useful: alert administrators if the same IP address attempts to log on to a particular device on the network. It fails five times with different login credentials and then successfully connects to a device on the network. All events occurred within a fifteen-minute time window. This sequence of events may be an indication that a potential attacker has gained authentication into the network and could signal an escalation of privilege, so a potential security incident occurred. [7]

When setting up a SIEM system, the rules in question must be reprogrammed as needed. They are inserted manually, and therefore, it is necessary to know which anomalies do not make sense to report and which should trigger an alarm. [4]

Notification and response – In the event of a security breach, SIEM alerts the personnel responsible for network protection in real time. It also selects an adequate response for the optimal protection of assets. [3]

It can be observed that the SIEM is a complex system with a lot of functionality. The entire operation of receiving and processing logs can be enriched by integrating a neural network to perform various security tasks.

As seen in Fig. 2, the way it would work is as follows: Different locations contain a certain infrastructure that produces many logs, which can be considered for this purpose as ”raw data”. The infrastructure would forward these data to a Logstash instance, Logstash being the software to collect these logs. Log collectors allow for scalability, which is useful for not only distributing the load but also creating logical units by separating locations or infrastructures. Logstash then sends these data to a primary Logstash instance, which directly communicates with the platform. The platform would then be responsible for routing logs into the AI model, which performs various advanced security tasks. This makes it much easier to deploy this type of infrastructure, and the AI networks add many advantages to the logging process.

III. The advantages of AI in log processing

Technological advancements constantly push the boundaries of what is possible, with new inventions and innovations emerging on a daily basis. Throughout history, transformative breakthroughs such as the steam engine, electricity, computers, and the Internet have revolutionised society and propelled human progress forward. These advances have fundamentally transformed existing processes and pushed society to new heights on the evolutionary spiral [8].

The fourth industrial revolution has been driven primarily by advances in artificial intelligence and machine learning, which enable intelligent automation and data-driven decision-making [8], [9]. Artificial intelligence has been used in different fields to improve existing processes and uncover previously unknown relationships between data sets. Artificial intelligence techniques are proving to be highly promising for security monitoring, especially in log processing [9].

Log records which are processed by SIEM solution are represented as unstructured text data, as there are a multitude of log event sources and formats. Since log records often contain a large amount of semantic information that cannot be captured by traditional statistical methods, it is possible to use *natural language processing* (NLP) algorithms to tackle various security tasks [8]. There are a variety of options in the context of processing log data effectively and provide security monitoring. These include, but are not limited to, log correlation to identify patterns, log anonymisation to protect sensitive information, log parsing to extract useful data, and log filtering to remove irrelevant data with the motivation to optimise SIEM licence usage. In order to mitigate this problem, deep artificial networks can be used [8], [10]. However, it is worth noting that the latter may have a constraint on the maximum token size for the input sequence, despite achieving the state-of-the-art results for various NLP tasks such as Question Answering, Classification, Categorisation, and more.

Moreover, the popularity of AI algorithms is further bolstered by the limitations of modern SIEM solutions. Due to the limited ability of these systems to respond automatically to cybersecurity incidents, without human intervention. Correlation rules must be mainly written manually, making it impractical to address every possible security incident. In addition, there may be a lack of the data needed to process and detect all security incidents effectively. The usage of archived data during incident investigation can also pose a challenge. [11]

One of the main challenges in using AI techniques for log processing is the requirement for fast data processing. This is because modern SIEM solutions operate on real-time log data and must evaluate potential attack vectors as quickly as possible. It is important to know that modern SIEM solutions typically have a data flow rate of more then 5000 EPS[1], which is based on the number of connected LESes. [12]

IV. Comparison of log collection approaches

Multiple approaches can be taken when creating a platform which supports the integration of AI models into log processing. These approaches vary in difficulty and effectiveness.

One of them is to create a log collector integrated within the platform. A simple collector can be made using a programming language like Python and its various libraries, which allow the manipulation of logs. Unfortunately, making a reliable and safe collector that is capable of pulling logs from many different sources would be a very lengthy and difficult process. There are already log collectors that can be used for this purpose and connect into the platform.

This section contains a comparison between three popular choices when it comes to log collecting. Each of them has its own advantages and fits differently to different architectures.

A. *NXLog*

This collector is supported on many different operating systems, which is its biggest advantage when compared to other collectors. It is the only application in this section that also has a paid version [13]. A configuration file is what is used to control the flow of logs into the application. The file consists of directives that are similar to XML tags. [14]

The process of creating a file consists of defining the constants, setting the global directives, input, output and route configurations [14]. This is a lengthier process when compared to other mentioned collectors.

B. *Fluentd*

An open source collector, which has many plugins to support a wider range of protocols and offers different options to work with LESes. There are 9 types of plugins: Input, Parser, Filter, Output, Formatter, Storage, Service Discovery, Buffer, and Metrics. [15]

The heart of this software is a configuration file, where the main parts are the input and output plugins. Plugins are defined in the file, and within them there are tags that specify information such as ports, protocol types, and other configuration fields. [15]

[1]EPS (*Events per second*) refers to the rate at which events (such as log entries) are generated and processed by the system. The EPS rate is a measure of how much log data a SIEM solution can handle and process in real-time.

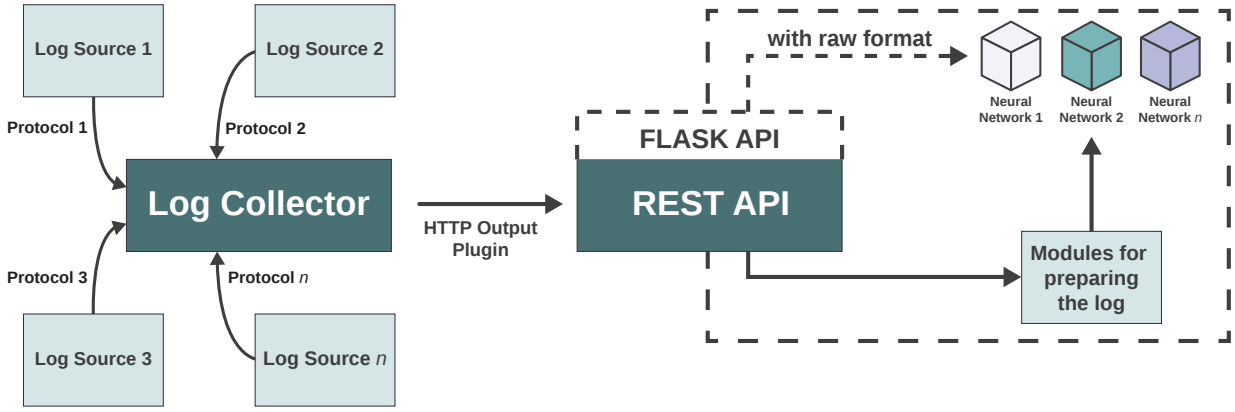

Fig. 3. Simplified view of the architecture of the proposed platform.

C. *Logstash*

Logstash is an open source application which is part of the Elastic stack. Such a program allows for the collection of logs from different sources using inputs in the program configuration file, which is called a logstash pipeline. Pipeline can contain multiple fields, such as input, filter, and output, but only needs input and output to function. [16]

The input field contains the type of protocol used for receiving logs. It usually specifies the port on which the transfer will take place. [17]

When it comes to output, multiple formats are supported. Logstash supports many different log protocols, for example, syslog, file, redis, exec, jdbc, http and many others. [18]

Logstash is a good choice for this infrastructure, considering it is open source, has options for scalability (instances can forward data to other Logstash instances) and the configuration file is quick to set up. It also has a large community, which opens up options for support. It also makes getting certain logs easier by supporting an application called Beats. It is a part of the Elastic suite of software, and makes particular systems able to send logs automatically, without having to pull them manually. [19]

D. *Final comparison of selected log collectors*

There are a few metrics for the comparison of log collectors. **Supported platforms** – Fluentd and Logstash are supported on devices with Windows and Linux operating systems. NXLog is supported on both platforms and also on macOS and many others, such as Oracle Solaris. [13], [19]

Event routing – Fluentd relies on tags for routing. Logstash, on the other hand, uses logical expressions and elements. NXLog uses blocks similar to XML tags. At its core, the syntax and main idea of the configuration are similar across the three applications. [19]

Support for extensions – while with Logstash the effort is to keep all plug-ins centralised under a single GitHub repository, Fluentd offers only 11 input plugins in its official repository; all the others are created outside of it and, therefore, have decentralised access. NXLog offers plug-ins, extensions, and, by default, modules; a list of which is available in the corresponding documentation. Table I lists some of the standard plugins supported in the compared applications. The number of supported plugins can be increased by using the repositories and extensions mentioned above. [19]

Transport and Reliability – in terms of data retention after reboot and a transfer of data, Logstash requires Redis, which serves as an external queue for data preservation. Fluentd by default includes options to configure the system for caching. NXLog directly supports flow control and a log queue that stores data on disk in case of failure. [19], [20]

Performance – Fluentd uses only 40 MB of RAM compared to Logstash, which uses 120 MB. The recommended size of RAM for NXLog is 250 MB. This difference can be negligible in a broader network. Logstash solves the load by installing Beats, which are often used on endpoints to send logs to a central Logstash ELK station. [19]

TABLE I
A COMPARISON OF COMMON PROTOCOLS USED IN LOG COLLECTORS.

Fluentd	Logstash	NXLog
Tail (*File*)	File	File
Syslog	Syslog	Syslog
UDP	UDP	UDP
TCP	TCP	TCP
HTTP	HTTP	HTTP
Windows Event Log	Beats	Windows Event Forwarding
Unix	Unix	Unix Domain Socket
Exec	Exec	Exec
Sample	Generator	Testgen
Monitor agent	–	–
Forward	–	Internal
–	Redis	Redis
–	Kafka	Kafka
–	Github	–
–	JDBC (*DB*)	ODBC (*DB*)
–	–	macOS Unified Logging System
...	...	...

V. Results and discussion

The aim of the project is to use an open-source solution for log collection that supports scalability and a wide array of log sources. The inclusion of a reliable and free-to-use log collector is crucial to creating manageable pipelines and zones for data to flow through. The nature of logs and different protocols makes it necessary to use software that is capable of collecting and forwarding the data. After analysis, it was determined that Logstash is an application that meets the project requirements, considering its popularity, support, log types, and additional software. The next step is to create an API that can connect different AI models for different purposes. It is made to work in two modes, learning and real deployment. AI models are only as good as their input data, which means that collecting data for learning is crucial. Fig. 3 shows the design of the application. The log sources are configured to send logs to the Log Collector, which forwards the logs to the application programming interface via HTTP. It consists of modules to modify the format of the logs so that they are suitable for usage by the neural network. If the API receives the logs in a raw format, they can be directly pushed to the neural network without the need for processing modules to interfere with them. As shown in Fig. 2, the system is scalable and multiple Logstash instances can be connected to each other. In the case of real-time log processing, it is important to ensure that AI models can process data as quickly as possible and can handle a large number of EPS. The platform is not yet designed to be fully deployed into the cloud due to high data protection requirements.

VI. Conclusion and future work

The main goal of this paper was to analyse and enhance existing security monitoring processes using deep learning techniques. For example, modern deep neural networks like XLNet, BERT, and GPT2 can be transfer-learned and utilised to machinally understand the context of incoming log records and perform various NLP tasks. Discussed AI techniques revolutionised the field of deep learning by enabling parallel processing of input text data and avoiding recursion [21]. The main challenge of the modern security monitoring domain lies in the difficulty of accessing raw log source data and the lack of a platform that can seamlessly integrate normalised data with modern SIEM and Log Manager solutions. One of the significant contributions of this study is the creation of an open-source parallel platform based on Logstash that can seamlessly connect all modern log sources, normalise their outputs, and enable efficient training and deployment of any modern deep neural networks within the organisation's infrastructure. The Logstash solution was chosen after analysis of current log collection platforms, presented in section IV. Moreover, the presented solution can be successfully used for scientific purposes in the security monitoring domain. In the future, it is planned to optimise the discussed platform, deploy a prototype in real world infrastructures, and address the challenge of cloud deployment.

References

[1] BHATT, Sandeep, Pratyusa MANADHATA a Loai ZOMLOT. *The Operational Role of Security Information and Event Management Systems* [online]. 12. 2014 [cit. 26. 02. 23]. ISSN 1558-4046. Available at: https://dx.doi.org/10.1109/MSP.2014.103.

[2] Security Information and Event Management (SIEM). In: VIELBERTH, Manfred. *Encyclopedia of Cryptography, Security and Privacy* [online]. Springer: Springer, 2021, s. 1-3 [cit. 25. 02. 23]. ISBN 978-3-642-27739-9. Available at: https://doi.org/10.1007/978-3-642-27739-9_1681-1.

[3] HRISTOV, Marian, Maria NENOVA, Georgi ILIEV a Dimiter AVRESKY. Integration of Splunk Enterprise SIEM for DDoS Attack Detection in IoT. In: *2021 IEEE 20th International Symposium on Network Computing and Applications (NCA)*. Boston: IEEE, 2021, s. 1-5. ISBN 978-1-6654-9550-9. ISSN 2643-7929. Available at: http://dx.doi.org/10.1109/NCA53618.2021.9685977.

[4] MARTINASEK, Zdenek. Logging problematics, IDS and IPS systems [presentation]. ICT Security 2. Brno University of Technology, 2022.

[5] Installation and configuration for Windows Remote Management. *Microsoft* [online]. Redmond: Microsoft Corporation, © 2022, 10 september 2021 [cit. 26. 02. 23]. Available at: https://learn.microsoft.com/en-us/windows/win32/winrm/installation-and-configuration-for-windows-remote-management.

[6] Configuring rsyslog on a Logging Server. *Red Hat Customer Portal* [online]. Raleigh: Red Hat, © 2022 [cit. 26. 02. 23]. Available at: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/6/html/deployment_guide.

[7] CRAWLEY, Kim. *How SIEM Correlation Rules Work*. AT&T Cybersecurity [online]. Dallas: AT&T, © 2022, 20 February 2018 [cit. 27. 02. 23]. Available at: https://cybersecurity.att.com/blogs/security-essentials/how-siem-correlation-rules-work.

[8] CHOLLET, Francois. *Deep learning with Python*. Shelter Island, New York: Manning Publications Co. [2018]. ISBN 1617294438.

[9] MINSKY, Martin. *Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind*. 2006. ISBN 0743276639.

[10] RUDD, Ethan a Ahmed ABDALLAH. *Training Transformers for Information Security Tasks: A Case Study on Malicious URL Prediction*. [online]. [cit. 11. 03. 23]. Available at: https://arxiv.org/pdf/2011.03040.pdf.

[11] GONZALES-GRANADILLO, Gustavo, Susana GONZALES-ZARZOSA a Rodrigo DIAZ. *Security Information and Event Management (SIEM): Analysis, Trends, and Usage in Critical Infrastructures*. [online]. Madrid: Cybersecurity Unit, Atos Research & Innovation, 2021 [cit. 11. 03. 23]. ISSN 1424-8220. Available at: https://www.mdpi.com/1424-8220/21/14/4759.

[12] CONSTANTINE, Conrad. *What to log in a SIEM: SIEM and security logging best practices explained*. [online]. [cit. 10. 03. 23]. Available at: https://cybersecurity.att.com/blogs/security-essentials/what-kind-of-logs-for-effective-siem-implementation.

[13] OS Support. *NXLog* [online]. Newark: NXLog, © 2022 [cit. 02. 03. 23]. Available at: https://docs.nxlog.co/userguide/os/index.html.

[14] Configuration overview. *NXLog* [online]. Newark: NXLog, © 2022 [cit. 02. 03. 23]. Available at: https://docs.nxlog.co/userguide/configure/overview.html.

[15] Overview. *Fluentd* [online]. Fluentd Project, © 2022 [cit. 03. 03. 23]. Available at: https://docs.fluentd.org/quickstart.

[16] Support Matrix. *Elastic* [online]. Mountain View: Elasticsearch B.V., © 2022 [cit. 04. 03. 23]. Available at: https://www.elastic.co/support/matrix.

[17] Creating a Logstash pipeline. *Elastic* [online]. Mountain View: Elasticsearch B.V., © 2022 [cit. 04. 03. 23]. Available at: https://www.elastic.co/guide/en/logstash/current/configuration.html.

[18] Input plugins. *Elastic* [online]. Mountain View: Elasticsearch B.V., © 2022 [cit. 04. 03. 23]. Available at: https://www.elastic.co/guide/en/logstash/current/input-plugins.html.

[19] PERI, Noni. Fluentd vs Logstash: A Comparison of Log Collectors. *Logz.io* [online]. Tel Aviv: Logshero, © 2015-2022 [cit. 07. 03. 23]. Available at: https://logz.io/blog/fluentd-logstash/.

[20] Buffering and flow control. *NXLog* [online]. Newark: NXLog, © 2022 [cit. 02. 03. 23]. Available at: https://docs.nxlog.co/userguide/intro/buffering-and-flow-control.html.

[21] VASWANI, Ashish, Noam SHAZEER, Niki PARMAR, Jakob USZKOREIT, Llion JONES, Aidan N. GOMEZ, Lukasz KAISER a Illia POLOSUKHIN. *Attention Is All You Need* [online]. [cit. 3. 3. 2023]. Available at: https://arxiv.org/pdf/1706.03762.pdf.