PROJECT

Multi-tenant cybersecurity operations centre designed
as an open cloud service with machine learning elements

Linked to work package KPB1 – Design of the virtualization solution

Linked to output
V3 – Description of the selected solution meeting the specified criteria

type – documentation

Analysis III

Analysis of log collection tools for the needs of a NOC and SOC centre as applied to a smaller company

CONTENTS

1 Analysis 4
1.1 Description of the topic 4
1.2 Specification of the problem 4
2 Objectives of the work 5
2.1 Objectives of the solution 5
3 Results of the work 6
3.1 Familiarization with the operating principles of the SOC and NOC 6
3.1.1 Security Operation Center (SOC) 6
3.1.2 Network Operation Center (NOC) 7
3.2 analysis of logging tools 8
3.2.1 Syslog-ng 9
3.2.2 Graylog 9
3.2.3 Logstash 9
3.2.4 Loki 10
3.3 Definition of criteria for comparing the individual tools 10
3.3.1 Performance 11
3.3.2 Integration 11
3.3.3 Customization 11
3.3.4 Security 11
3.3.5 Monitoring and analysis 11
3.3.6 Data retention and management 12
3.3.7 User experience 12
3.3.8 Installation 12
3.3.9 Maintenance 12
3.3.10 Updates and support 12
3.3.11 Costs and licensing 13
3.4 Comparison of the tools 14
3.4.1 Scoring 15
Conclusion 17

1 ANALYSIS

1.1 DESCRIPTION OF THE TOPIC

Modern IT infrastructures, even in smaller organizations, consist of a growing number of interconnected components: network devices, servers, operating systems and applications. Each of these elements continuously generates records of its activity – logs. These logs contain valuable information about the system state, performance, errors, user activity and potential security threats. Without a systematic approach to their collection and analysis, however, organizations face a fundamental problem: a lack of overview of and visibility into their own infrastructure.

To address these challenges and ensure proactive IT management, specialized teams are being established: the Network Operations Center (NOC) and the Security Operations Center (SOC). The NOC focuses primarily on ensuring the operational stability, availability and performance of IT services. Its aim is to monitor the infrastructure, rapidly identify and resolve operational problems and optimize performance. The SOC, by contrast, focuses on cybersecurity. It monitors systems for the presence of threats, detects suspicious activity, analyses security events and coordinates incident response with the aim of minimizing damage.

In order for the NOC and SOC to fulfil their mission effectively, they need a continuous supply of accurate and timely information about what is happening in the infrastructure. This is precisely where log management systems play a key role. Their primary function is to centralize the collection of logs from diverse sources (networks, servers, applications) in a single place.

1.2 SPECIFICATION OF THE PROBLEM

The department has historically operated or trialled several log systems. Recently it has begun building NOC and SOC centres, for which a decision must be made as to which solution to select in the end. The aim of the project is to systematically analyse, classify and compare the suitability of selected solutions for the purpose of creating a system for the collection, archiving and analysis of logs from network and server entities at KIS FRI UNIZA based on the requirements. It is assumed that logs will be collected from various devices and various operating systems and sent to a central logging server, on which the administrators and operators of the NOC and SOC centre will work. The selected solutions analysed include:

- Syslog-ng
- Graylog
- Logstash
- Loki environment

2 OBJECTIVES OF THE WORK

2.1 OBJECTIVES OF THE SOLUTION

Within the project, we pursued several objectives, which include:

- Familiarization with the operating principles of SOC and NOC centres and definition of the requirements for these centres. It was also necessary to study the existing solutions that had been operated at the faculty, as described in the diploma theses written by Ing. Drožďík and Ing. Dočár
- Research into logging tools, study of official documentation, reviews and available studies. The main purpose was to understand how these tools work, what their functions are and any limitations. These tools include:
  - Syslog-ng
  - Graylog
  - Logstash
  - Loki environment
- Definition of criteria for comparing the individual tools, so that we could compare these tools and select a suitable tool for our purpose
- Testing and configuration of the individual tools in a test environment for a better understanding of their operation and implementation.
- The actual comparison of the individual tools based on the criteria we defined, and the final evaluation

3 RESULTS OF THE WORK

3.1 FAMILIARIZATION WITH THE OPERATING PRINCIPLES OF THE SOC AND NOC

3.1.1 Security Operation Center (SOC)

With a SOC (Security Operation Center), an organization can monitor and manage network traffic and look for threats – it is focused more on security. In addition to monitoring, the SOC also focuses on incident handling, searching for potential threats and preventing potential attacks.

Main tasks of the SOC:

- Monitoring the behaviour of network traffic
- Maintaining records of activities and communications in the traffic
- Assessing the risk/severity of incidents
- Incident response
- Investigating the root cause of an incident

Tools for the SOC

For network monitoring, the SOC uses various tools that serve to analyse data from network traffic across the entire organization. These tools fall into the following categories:

- SIEM systems (Security Information and Event Management)
- IDS (Intrusion detection systems)
- Endpoint detection and response (EDR) solutions

SIEM systems are used for log management and are a combination of Security information management (SIM) and Security event management (SEM). These tools guarantee real-time monitoring and analysis of network security events.

The most widely used open-source SIEM tools include:

- OpenSearch
- OSSEC
- SecurityOnion
- Wazuh
- OSSIM
- Prelude

IDS are network security tools that monitor network traffic and devices in the network and detect malicious and suspicious activities and security policy violations. They are usually software applications that may reside on end devices, on devices dedicated to network monitoring or in the cloud. When suspicious traffic is detected, they send alerts. They also keep records of security incidents or send these records to a SIEM system in the form of logs.

IDS are not standalone tools but are part of other security systems:

- IDS and SIEM
- IDS and IPS (Intrusion Prevention Systems)
- IDS and firewalls

EDR are network security solutions that monitor user devices and detect and address cyber threats such as ransomware or malware. They serve as the first line of defence thanks to advanced antivirus and anti-malware protection. Unlike a conventional antivirus, which has access to a database of already discovered threats, EDR can detect as yet unknown threats.

Popular open-source EDR tools include:

- OSSEC
- TheHive Project
- osQuery
- SNORT

3.1.2 Network Operation Center (NOC)

The NOC (Network Operation Center) is used by network administrators to monitor the state of the network and its devices – it is focused more on the operation of the network. The main task of the SOC is to monitor the resources used in the network and to take care of the operation of the network, address network outages and efficiency.

Main tasks of the NOC:

- Updating, troubleshooting and installing software on systems connected to the network
- Management of IT infrastructure and devices
- Data backup and ensuring its accessibility in the network
- Monitoring firewalls and network security software
- Monitoring the state of the network, including network state analysis, performance reporting and network optimization
- Network recovery after a possible failure

Tools for the NOC

The NOC uses tools for monitoring the network, identifying bottlenecks, managing configurations and automating solutions to common network traffic problems. These are tools from the following categories:

- Network monitoring tools
- Network performance analyzers
- Configuration management databases (CMDB)

Network monitoring tools serve to ensure the smooth operation and performance of the network. Thanks to these tools, we can detect outages and problems in the network in good time and monitor network performance, such as server load or response time. They are also used for network analysis and display statistics on network performance. To a certain extent they can also monitor the

security aspects of the network, but this is mostly the job of SOC tools.

Popular open-source network monitoring tools are:

- Zabbix
- Nagios
- Icinga
- Prometheus
- OpenNMS
- LibreNMS

Network performance analyzers are tools for analysing network performance. They focus mainly on analysing the quality of network connections and throughput. They allow a detailed analysis of how efficiently the network operates and detect performance degradation.

These are tools from the following categories:

- Packet Sniffers
  - Wireshark
- Network Mapping
  - Nmap
- Bandwidth analysis
  - Wireshark
  - SolarWinds NetFlow
- Performance testing
  - iPerf
- Configuration managers
  - Ansible
  - Puppet
- Security audits

CMDB is a database that stores information about the hardware and software configuration of the IT components used in an organization. It tracks details about hardware, software, network devices and their interconnections.

Popular tools are:

- GLPI
- Ralph
- DataGerry
- I-doit
- CMDBuild

3.2 ANALYSIS OF LOGGING TOOLS

Logging systems are used to collect logs from various devices. These logs

can be sent to a single centralized logging server, where we have an overall overview of what is happening in the network.

The logging systems we selected include:

- Syslog-ng
- Graylog
- Logstash
- Loki enviroment

3.2.1 Syslog-ng

Syslog-ng is a tool for collecting logs on Linux or UNIX devices. This tool is capable of logging almost any activity on a device. It can store these logs locally or send them to a centralized logging server.

Functions:

- Support for the Syslog protocol and other input protocols (e.g. TCP, UDP, RELP).
- Filtering, processing and sorting of logs.
- Export of logs to various formats (JSON, CSV, Elasticsearch).
- Encryption and signing of logs to ensure integrity.

3.2.2 Graylog

Graylog is an open-source platform for log management and analysis that helps organizations collect, store and analyse log data from various sources. It offers powerful search and visualization capabilities, which makes it useful for a broad range of use cases.

Functions:

- Collection of logs from various sources
- Real-time log processing
- Normalization and parsing of logs
- Advanced log search
- Alerts and notifications for specified events
- Data visualization
- Log storage
- Security and access control allows only authorized users to search logs
- Many plugins

Graylog uses the Elasticsearch tool to store and search logs. It also uses a MongoDB database to store configurations, metadata and user information.

3.2.3 Logstash

Logstash is an open-source data processing tool that receives, transforms and sends data from various sources to various target systems. It is part of the Elastic Stack, which also includes Elasticsearch and Kibana. It supports real-time data collection and transformation using a modular architecture with plugins for input, filters and output. It is commonly used for centralized logging, data

transformation and real-time analysis. Logstash makes it possible to create workflows and gain immediate insights by sending data to systems such as Elasticsearch or PubNub Iluminate for analysis.

Main functions:

- Data collection: Logstash can collect and aggregate data from multiple sources in real time. It supports a wide range of input sources, including log files, databases, message queues and various cloud services.
- Transformation: Once data have been obtained, Logstash allows them to be processed and transformed using various filters. You can use these filters to clean, enrich and modify data before they are sent to their final destination. Common transformations include parsing unstructured log data, adding geographical information or anonymizing sensitive information.
- Output: After processing, Logstash can send data to various target systems, such as Elasticsearch (for storage and search), to various databases or to other systems and services.

3.2.4 Loki

Loki is a log management tool developed by Grafana Labs. It helps collect, store and search logs from your applications, but it does so somewhat differently from traditional logging systems. Most log management tools, such as Elasticsearch, index the individual log data. This can be very demanding on system resources, especially when a large volume of logs is coming in.

Loki indexes only the labels (metadata) of logs—things such as the application name, environment or other tags. This means that Loki does not index the entire content of logs, which saves a lot of storage space and makes it much more efficient.

Log processing procedure:

- Log collection – using e.g. Promtail
- Indexing using tags
- Storing logs in chunks to save space
- Searching logs using tags

Loki is usually used in combination with **Promtail** for automatic collection of logs from the environment and with the **Grafana** tool for efficient display and visualization of logs.

3.3 DEFINITION OF CRITERIA FOR COMPARING THE INDIVIDUAL TOOLS

One of the objectives was also research into the parameters and criteria by which we can compare the logging systems mentioned. Overall, we agreed on 12 parameters, with each parameter divided into several smaller parts that need to be taken into account.

The selected parameters include:

3.3.1 Performance

Parameters focused on data processing.

1. **Scalability -** A property that determines whether the logging system can handle the expected volume of data from the network and from how many sources the system can process logs.
2. **Data processing –** the ways in which the system can process data, enrich them with new information, filter, parse, remove sensitive information from the data; overall flexibility of data processing.
3. **Supported inputs** - What data sources can the system accept? For example: text logs, JSON, syslog, databases or cloud services.

3.3.2 Integration

Parameters relating to integration and communication with other applications.

1. **Integration with other tools -** Can the system also cooperate with other tools, such as SIEM tools? Or rather, can it make data available to them?
2. **API support -** Does the system provide an API for integration with other applications or process automation?

3.3.3 Customization

Parameters relating to customization

1. **Extensions and custom modules** - Does the system support extensions or plugins? Can an organization create its own tailor-made extensions?
2. **Automation and Scripting** - Support for automating certain actions based on thresholds or other parameters

3.3.4 Security

Parameters relating to the security of log retention, access and transmission.

1. **Secure transmission** - Does the system support encrypted data transmission (e.g. TLS/SSL)?
2. **Encryption of logs at rest** - Can stored data be encrypted to prevent unauthorized access?
3. **Access control** - Does the system offer robust access control options (e.g. role-based access control, user authentication)?

3.3.5 Monitoring and analysis

Parameters relating to monitoring and data overview.

1. **Alerts** - Real-time alerting function
2. **Event** – Issuing alerts based on specified requirements
3. **Reporting** - Can the system generate clear reports on the state of the network and on incidents?

4. **Data analysis** - Does the system offer data visualization tools, such as charts and dashboards? Is external software required for this functionality?

3.3.6 Data retention and management

Parameters relating to the customization and storage of data and their management.

1. **Data storage** - What methods does the system use to retain data ?
2. **Data compression** - Does the system support compression techniques to save storage space?
3. **Data archiving** - Does it provide the option of long-term data storage?
4. **Data rotation** - What options does the system offer for automatic log rotation based on time or size?
5. **Data search** - How efficient and flexible is data search (e.g. full-text search, support for regular expressions)?

3.3.7 User experience

Parameters relating to assistance for users in managing the system.

1. **User interface and UX** - Does the system have a built-in user interface? Is it intuitive and clear? What functions does the interface offer for system management?

– **Documentation** - Is the system documentation detailed, up to date and easily accessible?

3.3.8 Installation

Parameters relating to the complexity and options of installation.

1. **Ease of installation** - How difficult is it to install the given system ?
2. **Installation in VM/Kubernetes** - Does the system support deployment in virtualized environments or container platforms such as Kubernetes?

3.3.9 Maintenance

Parameters that determine how demanding it is to look after the system.

1. **Upgrade** - How difficult is it to perform an update to a newer version?
2. **Ease of configuration** - How complex is it to set up the system? Does it offer tools to simplify configuration?

3.3.10 Updates and support

Parameters relating to the support of the system by its developers

1. **Currency of the solution** - How often are updates released? Are improvements and bug fixes provided regularly?

3.3.11 Costs and licensing

Parameters relating to the costs of operating the system

1. **Licence** - What are the licensing terms (open-source, commercial licence, freemium)?
2. **Price** - What are the direct and indirect costs of implementing, operating and maintaining the system?

3.4 COMPARISON OF THE TOOLS

After testing and gathering information, I created a list of our criteria for the individual tools and assigned them points. I assigned each criterion points on a scale from 1 to 10, where 1 represents the lowest possible rating of how suitable the parameter is compared with the other logging systems and, conversely, 10 is the most suitable. I also assigned each parameter a weight from 1 to 10, which represents a value determining how important the parameter is for us (1 least, 10 most).

Parameter	Weight	Graylog	Logstash	Loki	Syslog-ng
		Rating			
Hardware requirements	8	8	7	9	10
Scalability	7	9	8	6	7
Data processing	6	8	9	6	8
Supported inputs	9	8	9	7	9
Integration with other tools	7	8	9	8	7
API support	6	8	7	8	7
Extensions and custom modules	6	8	9	6	8
Automation and Scripting	5	8	7	7	7
Secure transmission	8	9	9	9	9
Encryption of logs at rest	5	3	3	3	6
Access control	5	8	5	5	5
Notifications	1	8	1	1	1
Event	1	8	1	1	1
Reporting	1	6	1	1	1
Data analysis	1	8	1	1	1
Data storage	9	7	6	9	7
Data compression	8	1	1	9	5
Data archiving	7	2	1	9	2
Data rotation	7	1	1	9	1
Data search	6	8	6	10	1
User interface and UX	1	9	1	9	1
Ease of installation	5	7	7	8	8
Installation in VM/Kubernetes	4	8	8	10	7
Upgrade	7	6	6	8	8
Ease of configuration	7	7	6	7	7
Currency of the solution	10	9	8	9	8
Licence	10	7	9	10	8
Price	10	5	9	10	8
Total		**11.13**	**10.81**	**13.27**	**11.07**

3.4.1 Scoring

- **Scalability**: Graylog/Loki have strong clustering. Logstash is scalable but may require multiple instances. Syslog-ng handles many sources.
- **Data processing**: Logstash has the richest options (plugins). Graylog/Syslog-ng are also strong. Loki focuses less on transformation and more on indexing.
- **Supported inputs**: Logstash and Syslog-ng are very flexible. Graylog also has broad support. Loki depends more on agents (Promtail, Fluentd).
- **Integration with other tools**: Logstash excels in the Elastic Stack. Graylog has good integrations (SIEM, Slack). Loki closely with Grafana. Syslog-ng with SIEM.
- **API support**: Graylog/Loki have a REST API for logs/configuration. Logstash for pipeline management/monitoring. Syslog-ng for automation.
- **Extensions and custom modules**: Logstash and Syslog-ng are very strong in plugins/modules. Graylog supports plugins. Loki via agent configuration.
- **Automation and Scripting**: Graylog (streams, pipelines). Logstash (pipeline logic). Loki (LogQL rules). Syslog-ng (rules, scripts).
- **Secure transmission**: All support TLS
- **Encryption of logs at rest**: Mostly not natively (Graylog/Loki/Logstash require external tools/disk). Syslog-ng supports GPG.
- **Access control**: Graylog has RBAC. The others often depend on external systems (Elastic Stack, Grafana) or basic OS/network methods.
- **Alerts**: Graylog has them natively. Loki via Grafana. Logstash/Syslog-ng require external tools (Kibana, Grafana, etc.).
- **Event**: Same as alerts
- **Reporting**: Graylog has basic reporting (better in the paid version). Loki via Grafana. Logstash/Syslog-ng need external tools.
- **Data analysis**: Graylog has strong capabilities. Loki via LogQL/Grafana. Logstash/Syslog-ng depend on the target storage/visualization (Elasticsearch/Kibana).
- **Data storage**: Loki efficient (chunks, cloud). Graylog (Elastic/OpenSearch). Syslog-ng (databases, files). Logstash only forwards.
- **Data compression**: Loki has native support. Graylog/Logstash depend on the database. Syslog-ng via external tools (gzip).
- **Data archiving**: Loki (cloud/chunk backup). Graylog (depends on the DB, paid version). Logstash/Syslog-ng handled externally (e.g. logrotate).
- **Data rotation**: Loki has policies. Graylog/Logstash depend on the DB. Syslog-ng externally.
- **Data search**: Graylog (full-text). Loki (LogQL via labels – efficient). Logstash (via the target storage). Syslog-ng (via external tools).

- **User interface and UX**: Graylog has a strong web UI. Loki via Grafana (very good). Logstash/Syslog-ng have no native UI (CLI/YAML/integrations).
- **Ease of installation**: Loki/Syslog-ng relatively simple (binaries, packages, Docker). Graylog requires a DB. Logstash (YAML).
- **Installation in VM/Kubernetes**: Loki is optimized for K8s. Graylog/Logstash well supported. Syslog-ng has support.
- **Upgrade**: Loki/Syslog-ng relatively simple. Graylog/Logstash may be more demanding with complex configurations/clusters.
- **Ease of configuration**: Graylog (web UI + files). Loki/Syslog-ng (YAML/config files – flexible). Logstash (YAML – can be complex).
- **Currency of the solution**: All are actively developed and regularly updated (Graylog very frequently).
- **Licence**: Loki (MIT) and Logstash (Open Source – Elastic Lic./Apache) are the most open. Syslog-ng (OSS + PE). Graylog (OSS + Enterprise).
- **Price**: Loki/Logstash basic version free. Syslog-ng basic version free. Graylog has a paid version for advanced features at a relatively high price.

CONCLUSION

Based on the criteria gathered and their ratings, we can conclude that Loki is a suitable candidate for log retention. It excels in particular in data storage and search, and for data analysis it integrates natively with the Grafana interface, which is an excellent visualization tool for logs and metrics. The entire solution is open-source, which is a major advantage.

In second place was the Graylog tool, which has a very clear interface and strong data analysis but is less flexible in storage encryption and, in the open version, offers only basic functions. Switching to the paid version, which includes a number of additional features, is very expensive.

Syslog came third. Nevertheless, it is an excellent tool for processing logs and sending them to central storage, which makes it very suitable for use together with Loki. It is very flexible and powerful in collecting, routing and filtering logs, has low resource requirements and good security features. However, it lacks a UI and advanced analysis.

Logstash came last. This tool excels in particular in data processing; however, it is mainly a processor/pipeline and is usually integrated within the ELK solution, which may be more demanding to maintain than the previous solutions.