PROJECT

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

Linked to work package KPB3 – Design of ML algorithms

Linked to output V7 result of the analysis of suitable algorithms

Funded by the European Union
NextGenerationEU

RECOVERY AND RESILIENCE PLAN

GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

Contents

1. Methodology 3
2. SUITABLE PUBLICATIONS 4
2.1. Enhancing intelligence SOC with big data tools 4
2.2. Building a Flexible and Resource-Light Monitoring Plat form for a WLCG-Tier2 4
2.3. Observability in Microservices: An In-Depth Exploration of Frameworks, Challenges, and Deployment Paradigms 5
2.4. An ensemble-based framework for user behaviour anomaly detection and classification for cybersecurity 6
2.5. SUESSA: Sustainable & Ultra-Elastic Stack Security Architecture for Securing IoT Networks of Future Smart Cities 7
2.6. Evaluating the Observability of Network Security Monitoring Strategies With TOMATO 8
2.7. EDGE ANALYTICS AND COMPLEX EVENT PROCESSING FOR REAL TIME AIR POLLUTION MONITORING AND CONTROL 8
2.8. Container-Based Honeypot Deployment for the Analysis of Malicious Activity 9
2.9. Self-Healing Cloud Services in Private Multi-Clouds 10
2.10. Flexible visualization of a 3rd party Intrusion Prevention (Security) tool: A use case with the ELK stack 10
2.11. An ML Based Anomaly Detection System in real time data streams 11
2.12. Anomaly Detection in POSTFIX mail log using Principal Component Analysis 12
2.13. Cyberattacks Detection and Analysis in a Network Log System Using XGBoost with ELK Stack 13
2.14. Improving State-of-the-art Compression Techniques for Log Management Tools 13
2.15. SDGen: A Scalable, Reproducible and Flexible Approach to Generate Real World Cyber Security Datasets 15
List of sources 16

1. METHODOLOGY

The literature search for suitable use cases of ML/AI in the detection of cyber anomalies, or rather the search for applications of AI/ML for the SOC, was carried out by searching for suitable publications on the Web of Science platform (WoS for short) for access to citation data from scientific journals and conference proceedings. A string of keywords, or rather a combination of individual keywords, was used for the search, with the static element being the use of either the keyword “Elastic Stack / ELK Stack” or “OpenSearch”, as these are open data analytics platforms for security, or rather systems for collecting, indexing, searching, visualizing and analysing logs and events, which support the detection of cyber threats and anomalies by means of both rules and machine learning. The search on WoS was carried out by combining these main keywords supplemented with the following additional keywords:

A. AI
B. ML
C. SOC
D. ANOMALY DETECTION
E. CYBER SECURITY
F. SECURITY OPERATIONS
G. SECURITY MONITORING
H. ALERTING

The result of the analysis when using the keyword “Elastic Stack”, or “ELK Stack”, is 31 publications (table 1).

Table 1 Number of publications by keywords in combination with “Elastic Stack/ ELK Stack”

	A	B	C	D	E	F	G	H
Count	3	1	3	12	5	1	4	2

The largest number of publications, i.e. 12 publications, could be found using the keyword string “Elastic Stack/ ELK Stack” AND “ANOMALY DETECTION”.

The result of the analysis when using the keyword “OpenSearch” in combination with the specified additional keywords is equal to two publications (table 2).

Table 2 Number of publications by keywords in combination with “OpenSearch”

	A	B	C	D	E	F	G	H
Count	0	0	0	0	0	0	1	1

Thus, when searching for suitable publications in the context of the OpenSearch analytics platform, only two publications were found, namely with the addition of the keywords “SECURITY ALERTING” and “ALERTING”-

The 33 publications found were subsequently analysed on the basis of two criteria, namely their availability, i.e. the possibility of tracing the full text of the publication, and the language of the publication (English). Of the 33 publications, 15 publications were identified that meet the given criteria, i.e. they are published in English and are also open source.

2. SUITABLE PUBLICATIONS

This chapter contains a brief description of the individual publications, or rather their abstracts in the original wording (English), together with their Slovak translations. Folino et al. (2023) dealt directly with the detection of cyber threats. Kulshrestha & Durbha (2020) created a real-time anomaly detection system. Kyriakou & Sklavos (2018) investigated how to capture new threats in the field of cybersecurity. Poat et al. (2023) sought to achieve automated real-time intrusion prevention across various systems, with a focus on an immediate overview of what was blocked and from where, and who was blocked, while also being able to inform the community in the event of access being revoked, without the need to create “tickets”, which may be handled with a delay. Jose et al. (2021) focused on more advanced approaches of network anomaly detection systems, with an emphasis not only on a robust anomaly detection algorithm but also on a real-time data transfer mechanism that minimizes response delays. The research focuses on the implementation of a real-time streaming system for network anomaly detection capable of functioning even with a high volume of data traffic. Tran & Tran (2018) proposed a method for identifying anomalies in a data set that can detect an anomaly at the time an attack occurs via spam e-mails. Yang et al. (2021) investigated the use of artificial intelligence and machine learning methods in cyberattacks, focusing in particular on defence methods against these attacks by detecting and identifying them through observation of logging data and analysis of whether they contain abnormal behaviour.

2.1. Enhancing intelligence SOC with big data tools

Abstract: Traditional security tools are based on predetermined signatures or rules that show rigidity in the face of the dynamics of the interconnections that occur in organizations. The inclusion of analytic in security systems extends the field of vision of security analysts for proactive threat detection. This allows SOC specialists to make decisions in real time and focus on the protection of critical assets of the organization. Our contribution in this work is to analyze the applicability of the use of Big Data as a complementary tool for the detection of security events in a real CSIRT environment, validating the architecture, configuration and visualization using the ELK stack as a Big Data platform. (Andrade & Torres, 2018)

keywords: security operation center; cognitive security; big data; cybersecurity

Abstract: Traditional security tools are based on predetermined signatures or rules, which are relatively rigid and inflexible with respect to the dynamics of interconnections in organizations. The inclusion of analytical tools in security systems broadens the field of view of security analysts for proactive threat detection. This enables SOC specialists to make decisions in real time and focus on protecting the critical assets of the organization. The contribution of this work is an analysis of the applicability of Big Data as a complementary tool for detecting security events in a real CSIRT environment, and validation of the architecture, configuration and visualization using the ELK stack as a Big Data platform. (Andrade & Torres, 2018)

2.2. Building a Flexible and Resource-Light Monitoring Plat form for a WLCG-Tier2

Abstract: Software development projects at Edinburgh identified a desire to build and manage our own monitoring platform. This better allows us to support the developing and varied physics and computing interests of our Experimental Particle Physics group. This production platform enables oversight of international experimental data management, local software development projects and active monitoring of lab facilities within our research group.

Larger sites such as CERN have access to many resources to support generalpurpose centralised monitoring solutions such as MONIT. At a WLCG Tier2 we only have access to a fraction of these resources and manpower. Recycling nodes from grid storage and borrowed capacity from our Tier2 Hypervisors has enabled us to build a reliable monitoring infrastructure. This also contributes back to our Tier2 management improving our operational and security monitoring. Shared experiences from larger sites gave us a head-start in building our own service monitoring (FluentD) and multi-protocol (AMQP/STOMP/UDP datagram) messaging frameworks atop both our Elasticsearch and OpenSearch clusters. This has been built with minimal hardware and software complexity, maximising maintainability, and reducing manpower costs. A secondary design goal has also been the ability to migrate and upgrade individual components with minimal service interruption. To achieve this, we made heavy use of different layers of containerisation (Podman/Docker), virtualization and NGINX web proxies. This presentation details our experiences in developing this platform from scratch with a focus on minimal resource use. This includes lessons learnt in deploying and comparing both an Elasticsearch and OpenSearch clusters, as well as designing various levels of automation and resiliency for our monitoring framework. This has culminated in us effectively indexing, parsing and storing >200GB of logging and monitoring data per day. (Currie & Yuan, 2024)

Abstract: Software development projects in Edinburgh identified the need to build and manage their own monitoring platform. This initiative makes it possible to support more effectively the dynamically developing and diverse interests in physics and computing within the experimental particle physics group. The production platform provides oversight of international experimental data management and local software development projects, and ensures active monitoring of laboratory facilities in the research infrastructure. Large institutions such as CERN have extensive resources to implement centralized monitoring solutions such as the MONIT system. At the WLCG Tier2 level, however, only a limited volume of resources and work capacity is available. Recycling nodes from grid storage and using borrowed capacity from the Tier2 hypervisors made it possible to build a robust and reliable monitoring infrastructure. This approach also brings added value for the management of the Tier2 environment itself by improving operational and security monitoring. Experiences shared by larger centres provided a significant advantage in the design and implementation of their own service monitoring system (FluentD) and multi-protocol messaging frameworks (AMQP/STOMP/UDP datagram) on top of the Elasticsearch and OpenSearch clusters. The entire infrastructure was designed with an emphasis on minimal hardware and software complexity, which increases its maintainability and reduces labour costs. A secondary design goal was the ability to migrate and upgrade individual components with minimal impact on service availability. To achieve this goal, multiple layers of containerization (Podman/Docker), virtualization and NGINX web proxy servers were used extensively. This presentation summarizes the experience of building the platform from scratch, with an emphasis on the efficient use of available resources. It also includes knowledge gained in deploying and comparing Elasticsearch and OpenSearch clusters, as well as in designing various levels of automation and resilience of the monitoring framework. The result is the ability to efficiently index, parse and store more than 200 GB of logging and monitoring data per day. (Currie & Yuan, 2024)

2.3. Observability in Microservices: An In-Depth Exploration of Frameworks, Challenges, and Deployment Paradigms

Abstract: The widespread adoption of microservices-based architectures in native cloud systems has amplified the need for robust observability strategies to ensure system reliability and performance. Microservices, while enabling scalability and flexibility, introduce complexities such as distributed failure points, performance bottlenecks, and resource

mismanagement. This paper provides a comprehensive review of state-of-the-art observability frameworks and tools designed for microservices architecture. It focuses on key aspects of observability at the system, service, and network levels, utilizing logs, traces, and metrics as core data sources. A thematic taxonomy is proposed, classifying the diverse approaches to monitoring microservices based on implementation paradigms, deployment platforms, and architecture patterns. The study compares existing solutions in terms of their capabilities, limitations, and effectiveness in addressing observability challenges in cloud, edge, and fog environments. Furthermore, open research issues are identified, and practical recommendations are provided to optimize observability in complex, distributed microservices ecosystems. This work aims to serve as a valuable resource for system architects, DevOps engineers, and researchers striving to enhance the reliability and performance of modern cloudnative systems. (Ummay Faseeha et al., 2025)

keywords: Microservices observability; Cloud-based microservices; System reliability; Containerized computing environments; Real-time performance monitoring

Abstract: The widespread adoption of microservices-based architectures in cloud-native systems has highlighted the need for robust observability strategies to ensure the reliability and performance of systems. Although microservices enable scalability and flexibility, they also introduce complexity in the form of distributed points of failure, performance bottlenecks and inefficient resource management. This work provides a comprehensive review of state-of-the-art observability frameworks and tools designed for microservices architecture. It focuses on key aspects of observability at the system, service and network levels, using logs, traces and metrics as core data sources. A thematic taxonomy is proposed that classifies the various approaches to monitoring microservices based on implementation paradigms, deployment platforms and architectural patterns. The study compares existing solutions in terms of their capabilities, limitations and effectiveness in addressing observability challenges in cloud, edge computing and fog computing environments. In addition, it identifies open research problems and provides practical recommendations for optimizing observability in complex distributed microservices ecosystems. The aim of this work is to serve as a valuable resource for system architects, DevOps engineers and researchers striving to increase the reliability and performance of modern cloud-native systems. (Ummay Faseeha et al., 2025)

2.4. An ensemble-based framework for user behaviour anomaly detection and classification for cybersecurity

Abstract: Nowadays, the speed of the user and application logs is so quick that it is almost impossible to analyse them in real time without using high-performance systems and platforms. In cybersecurity, human behaviour is responsible directly or indirectly for the most common attacks (i.e. ransomware and phishing). To monitor user behaviour, it is necessary to process fast user logs coming from different and heterogeneous sources, having part of the data or some entire sources missing. A framework based on the elastic stack (ELK) to process and store log data in real time from different users and applications is proposed for this aim. This system generates an ensemble of models to classify user behaviour and detect anomalies in real time, exploiting the advantages of the ELK-based software architecture and of the Kubernetes platform. In addition, a distributed evolutionary algorithm is used to classify the users by exploiting their digital footprints derived from many data sources. Experiments conducted on two real-life data sets verify the approach's goodness in detecting anomalies in user behaviour, coping with missing data and lowering the number of false alarms. (Folino et al., 2023)

keywords: Cybersecurity · Ensemble learning · Anomaly detection · Behavior analytics

Abstract: Nowadays, the rate at which user and application logs are generated is so high that analysing them in real time is practically impossible without the use of high-performance systems and platforms. In the field of cybersecurity, human behaviour is directly or indirectly responsible for the most common types of attacks (e.g. ransomware and phishing). To monitor user behaviour, it is necessary to process rapidly arriving logs from various heterogeneous sources, where part of the data or even entire sources may be missing. For this purpose, a framework based on the Elastic stack (ELK) is proposed, which makes it possible to process and store logging data in real time from various user and application sources. The system generates an ensemble of models to classify user behaviour and detect anomalies in real time, exploiting the advantages of the ELK-based software architecture and the Kubernetes platform. In addition, a distributed evolutionary algorithm is used to classify users based on their digital footprints derived from multiple data sources. Experiments conducted on two real-world data sets confirm the effectiveness of the proposed approach in detecting anomalies in user behaviour, in working with incomplete data and in reducing the number of false alarms. (Folino et al., 2023)

2.5. SUESSA: Sustainable & Ultra-Elastic Stack Security Architecture for Securing IoT Networks of Future Smart Cities

Abstract: Recently, the concept of a super smart city has received special attention and has become the most popular agenda among city planners, policy makers and IT industries around the globe. In this paper, we proposed a novel Sustainable Ultra-Elastic Stack Security Architecture (SUESSA) which can address various future security challenges of artificial intelligence (AI), big data and communication networks. Our architecture employs the holistic approach based on sustainability of network security which is different than that of the prevailing security practices. We propose to apply the sustainable approach in the Network security architecture through augmentation of cloud, networks and end-point level, our approach provides a robust and ultra-elastic mechanism of security that surpasses the prevailing architecture. We evaluated the feasibility and performance of the security stack containing a machine learning component, Elastic search engine, Kibana and Logstash as the building blocks of the architecture. Based on our survey, SUESSA architecture overcomes the disadvantages of current architecture specifically solving the problem of sustainability strongly demanded in security architecture of the future smart cities. (Gautam & Norio, 2020)

keywords: Sustainable & Elastic Security Architecture, Holistic Approach, Super Smart City, Future Cities, Artificial Intelligence

Abstract: Recently, the concept of a super smart city has received special attention and has become one of the most popular topics among city planners, policy makers and the IT industry worldwide. This article proposes a new architecture, the Sustainable Ultra-Elastic Stack Security Architecture (SUESSA), which can address various future security challenges associated with artificial intelligence (AI), big data and communication networks. This architecture employs a holistic approach based on the sustainability of network security, which differs from traditional security practices. It is proposed to apply sustainability principles to the network security architecture through augmentation at the cloud, network and end-point levels. This approach provides a robust and ultra-elastic security mechanism that surpasses existing architectures. The feasibility and performance of the security stack, whose basic components are machine learning, the Elasticsearch search engine, the Kibana visualization tool and the Logstash log processor, were evaluated. Based on this research, the SUESSA architecture overcomes the disadvantages of current solutions, specifically solving the problem of sustainability, which is critically demanded in the security architectures of future smart cities. (Gautam & Norio, 2020)

2.6. Evaluating the Observability of Network Security Monitoring Strategies With TOMATO

Abstract: Monitoring systems for malicious behavior increasingly requires aggregating and analyzing data from various sources, such as network flows, host logs, and end-point monitoring platforms. However, there's currently a lack of metrics and methodologies to compute the observability and efficiency of a security monitoring strategy. This manuscript introduces TOMATO (Threat Observability & Monitoring Assessment Tool), which is a platform to evaluate the effectiveness of a security monitoring strategy by exploring both the number of known adversarial techniques that can be detected within a network, along with evaluating the number of false-positives produced by the monitoring strategy. The output produces both an observability score and efficiency score of a set of deployed monitoring techniques, which are evaluated based on the data from the environment, and simulated attacks generated from MITRE ATT&CK. The proposed approach is then integrated into an ELK stack and evaluated on real SCADA devices within the WSU Smart City Testbed. (Halvorsen et al., 2019)

keywords: Computer security, network security, monitoring.

Abstract: Monitoring systems aimed at detecting malicious behaviour increasingly require the aggregation and analysis of data from various sources, such as network flows, host logs and end-point monitoring platforms. Currently, however, there is a lack of metrics and methodologies for evaluating the observability and efficiency of a security monitoring strategy. This article introduces TOMATO (Threat Observability & Monitoring Assessment Tool), a platform designed to evaluate the effectiveness of a security monitoring strategy. The tool evaluates both the number of known adversarial techniques that can be detected in a network and the number of false positive findings generated by the monitoring strategy. The output consists of two metrics, an observability score and an efficiency score, for a set of deployed monitoring techniques. These metrics are calculated based on data from the examined environment and simulated attacks created according to the MITRE ATT&CK framework. The proposed approach is subsequently integrated into an ELK stack and evaluated on real SCADA devices within the WSU Smart City Testbed. (Halvorsen et al., 2019)

2.7. EDGE ANALYTICS AND COMPLEX EVENT PROCESSING FOR REAL TIME AIR POLLUTION MONITORING AND CONTROL

Abstract: The advent of the Internet of Things (IoT) has led to the generation of tremendous amounts of data from various sources. Cloud based systems are effective in storage and application of machine learning algorithms on such datasets. However, in some cases it is important to enable real time processing for making immediate decisions. There are many applications which require instantaneous analysis of the generated data for remedies in event of an anomaly. Data associated with such use cases remains significant only for a short duration of time. Various electronic sensors, e.g. Temperature, Moisture, Air Quality, Pressure, Wind Velocity etc. present in a Wireless Sensor Network generate streams of values. It can be processed using pipelines which provide prompt and quick analysis for decision making. Stream Processing Systems can be helpful in such cases as they analyse data streams within a few milliseconds to a few seconds. In this paper, we discuss an event based processing of streaming data from air pollution sensors to create a real time anomaly detection system. To reduce the delays associated with the generation of alarms in our pipeline, Apache Foundation's Stream Processing Tools, Kafka and Flink were used for operations on our streams. To further accelerate the process, all the analysis is conducted on an embedded edge computing gateway device rather than sending data to the cloud for batch processing. The results are obtained in the form of a geographical map visualization using ELK stack. The map

highlights the coordinates of the location with an unhealthy air quality index in real time. (Kulshrestha & Durbha, 2020)

keywords: Internet of Things, Complex Event Processing, Air Pollution Sensors, Stream Processing, LoRaWAN, Geospatial Big Data Analysis

Abstract: The advent of the Internet of Things (IoT) has led to the massive generation of data from various sources. Cloud systems are efficient at storing such datasets and applying machine learning algorithms to them. In some cases, however, it is important to enable real-time data processing so that decisions can be made immediately. There are many applications that require immediate analysis of the generated data in order to take action when anomalies occur. The data associated with such cases are relevant only for a short period of time. Various electronic sensors, for example temperature, humidity, air quality, pressure and wind speed sensors, etc., which form part of a wireless sensor network, generate streams of values. These can be processed using data pipelines that provide rapid and immediate analysis to support decision-making. Stream Processing Systems are useful in such cases, as they can analyse data streams on a time horizon ranging from a few milliseconds to a few seconds. This article discusses event-driven processing of data streams from air pollution sensors with the aim of creating a real-time anomaly detection system. To reduce the delays in generating alarms in the pipeline, the Apache Foundation tools for stream data processing, Kafka and Flink, were used. To further speed up the process, the entire analysis is performed on an embedded edge computing gateway device instead of sending the data to the cloud for batch processing. The results are then presented in the form of a geographic map visualization using the ELK stack, with the map highlighting in real time the coordinates of locations with an unhealthy air quality index. (Kulshrestha & Durbha, 2020)

2.8. Container-Based Honeypot Deployment for the Analysis of Malicious Activity

Abstract: In today's world, the field of cyber security is a fastpaced changing environment. New threats are continuously emerging, and the ability to capture and effectively analyze them is paramount. In our work, we are deploying multiple honeypot sensors in order to monitor and study the actions of the attackers. The selected honeypots are Cowrie, Dionaea and Glastopf, presented as a Linux host, a Windows host and a Web application respectively. This enables us to have a diverse and broad environment that can attract attackers aiming at different attack surfaces. The sensors are running on a containerization platform, Docker and in this way, they are lightweight, resilient and could be easily deployed and managed. Our goal is the creation of a single dashboard that can present the captured data effectively in real-time and both in macroscopic and microscopic levels. Thus, we are utilizing the Elastic Stack and we are enriching our data sources using Virus Total's analysis engine. The proposed system ran for a three-month period and provided numerous data points, from which instantaneous useful conclusions were drawn for the behavior and nature of the malicious users. (Kyriakou & Sklavos, 2018)

keywords: honeypot, Docker, virtualization, intrusion detection, malware, Cowrie, Dionaea, Glastopf, Elastic Stack

Abstract: In today's world, the field of cybersecurity is a dynamically changing environment. New threats are constantly emerging, and the ability to capture and effectively analyse them is crucial. In this work, multiple honeypot sensors are deployed in order to monitor and study the behaviour of attackers. The selected honeypots are Cowrie, Dionaea and Glastopf, representing a Linux host, a Windows host and a web application. This approach makes it possible to create a diverse and extensive environment capable of attracting attackers targeting different attack surfaces. The sensors run on the Docker platform, which makes them

lightweight, resilient and easy to deploy and manage. The goal is to create a single dashboard that can effectively present the captured data in real time, at both the macroscopic and the microscopic level. For this purpose, the Elastic Stack is used and the data sources are enriched using the VirusTotal analysis engine. The proposed system ran for a three-month period and provided numerous data points from which useful conclusions about the behaviour and nature of the malicious users could be drawn immediately. (Kyriakou & Sklavos, 2018)

2.9. Self-Healing Cloud Services in Private Multi-Clouds

Abstract: As clouds continue to dominate as the de facto means of consuming shared computing resources, cloud providers are required to ensure that their services always meet or exceed service level agreements in terms of availability, reliability, scalability, performance etc every hour, day and year. These challenging requirements are even exacerbated when it comes to multi-cloud environments. In the era of automation and continuous cost reductions, using humans to manually test and trigger corrective actions when needed in order to fulfill these service level agreements is considered cumbersome and too expensive. In order to thrive, cloud service providers need a solution which is efficient, cheaper to deploy and easy to operate. This paper introduces a distributed container based solution for testing and self-healing of cloud resources. The proposed solution is based on a containerized multi-agent architecture which leverages Elasticsearch, Logstash, Kibana (ELK) stack and uses a rule-based self-healing algorithm implemented using Drools rule engine and Spring Boot framework. Preliminary tests show that the solution meets its target objectives in terms of maintaining service level agreements, lowering deployment costs, high performance and operability. (Mfula & Nurminen, 2018)

keywords: self-healing; multi-cloud; cloud computing

Abstract: As cloud solutions continue to dominate as the de facto means of using shared computing resources, cloud providers must ensure that their services always meet or exceed the level of service level agreements (SLAs) in terms of availability, reliability, scalability, performance, etc., every hour, day and year. These demanding requirements become even more pronounced in multi-cloud infrastructure environments. In the era of automation and continuous cost reduction, using human labour to manually test and trigger corrective actions to fulfil SLAs is considered time-consuming and too costly. To operate successfully, cloud service providers need a solution that is efficient, cheaper to deploy and easy to operate. This article presents a distributed container-based solution for testing and self-healing mechanisms of cloud resources. The proposed solution is based on a containerized multi-agent architecture that uses the Elasticsearch, Logstash and Kibana (ELK) stack and applies a rule-driven self-healing algorithm implemented using the Drools rule engine and the Spring Boot framework. Preliminary tests show that the solution meets the set objectives in terms of maintaining SLAs, reducing deployment costs, achieving high performance and ease of operation. (Mfula & Nurminen, 2018)

2.10. Flexible visualization of a 3rd party Intrusion Prevention (Security) tool: A use case with the ELK stack

Abstract: A difficult aspect of cyber security is the ability to achieve automated real time intrusion prevention across various sets of systems. To this extent, several companies are offering comprehensive solutions that leverage an “accuracy of scale” and moving much of the intelligence and detection on the Cloud, relying on an ever-growing set of data and analytics to increase decision accuracy. Often, they provide tools to visualize the decision workflows in attack prevention (as well as tune the algorithm) but those solutions are not always practical as companies see the problem as “global” that is, from a unified Cyber-security standpoint.

However, a key to a successful Cyber-security program is transparency and trust: from an experimental team viewpoint, this specifically means having the ability to immediately see what and from where, who has been blocked and being able to inform the community in case of a revoked access without the need for filing a “ticket” (that may eventually be answered) – in other words, rapid response to their user-base is essential but solutions targeting “sub-groups” in an organization are not often available. We have come up with a versatile solution leveraging the ELK stack (Elasticsearch, Logstash, & Kibana) and an IPS (Intrusion Prevention System) based WAF (Web Application Firewall) from Signal Sciences. Signal Science allows the streaming of detailed logs in a Logstash format suitable for custom solutions for visualization. By combining these two tools, we have strengthened our security posture and enabled individual experiments to monitor their own traffic. Specifically, the IPS WAF provides unique data such as country of origin, protocol, response code, source IP, and paths accessed. In this contribution, we will show how we engineered a visualization solution so experiment groups could access a dashboard with predefined graphs but also, where they can create individual customizable dashboards used to display blocked traffic and troubleshoot latency issues. We will discuss the details and procedures for developing and configuring these tools and how it benefits cyber security postures across our scientific based environment. (Poat et al., 2023)

Abstract: One of the challenging aspects of cybersecurity is the ability to achieve automated real-time intrusion prevention across various systems. In this context, several companies offer comprehensive solutions that leverage the so-called "accuracy of scale" and move a large part of the intelligence and detection to the cloud, relying on an ever-growing volume of data and analytical tools to increase decision accuracy. They often also provide tools for visualizing decision workflows in attack prevention (as well as options for tuning the algorithms), but these solutions are not always practical, as companies perceive the problem from the global perspective of unified cybersecurity. The key to a successful cybersecurity programme, however, is transparency and trust. From the viewpoint of an experimental team, this means having an immediate overview of what was blocked and from where, and who was blocked, while also being able to inform the community in the event of access being revoked, without the need to create "tickets", which may be handled with a delay. In other words, rapid response to the user base is essential, but solutions targeting sub-groups within an organization are often not available. A versatile solution was therefore designed, based on the ELK stack (Elasticsearch, Logstash, Kibana) and a WAF-type (Web Application Firewall) IPS (Intrusion Prevention System) from Signal Sciences. Signal Sciences allows the streaming of detailed logs in the Logstash format, which are suitable for building custom visualization solutions. Combining these two tools strengthened the security architecture and enabled individual experiments to monitor their own traffic. Specifically, the IPS WAF provides unique data such as country of origin, protocol, response code, source IP address and paths accessed. This contribution shows how a visualization solution was designed that allows experimental groups to access a dashboard with predefined graphs, as well as to create individual customized dashboards for displaying blocked traffic and troubleshooting latency issues. The procedures for developing and configuring these tools are presented in detail, and it is shown how they contribute to strengthening cybersecurity in an environment oriented towards scientific research. (Poat et al., 2023)

2.11. An ML Based Anomaly Detection System in real time data streams

Abstract: Due to the advancements in machine learning and artificial intelligence applied fields, network anomaly detection systems have experienced an evolution from traditional signaturebased methods for intrusion detection. Nonetheless, as security measures evolve, more sophisticated attacks are also constantly being developed by hackers. Not only

a robust anomaly detection algorithm is needed, but also a real-time data feeding mechanism for minimizing the reaction-time impact is required. Moreover, DDoS attacks can flood the network data channels with more than thousands of packets per second with the latent effect of overloading most traditional monitoring systems that rely on data storage. Due to this, the research presented in this paper focuses its efforts on implementing a real-time data streaming system for network anomaly detection that can operate during a high volume of traffic data. The solution includes the deployment of a flow collector platform connected to Apache Kafka for receiving NetFlow data from network switches. Also, real-time big data processing techniques are applied through Apache Spark, where the ML anomaly detection is triggered. The detection of anomalies is performed by a combination of the unsupervised learning clustering algorithm k-means and the supervised learning classifier KNN (k- nearest neighbors). Finally, a monitoring system consisting of an ELK stack collects historical data for further evolution of the ML algorithms. (Jose et al., 2021)

keywords: Anomaly Detection, Machine Learning, Big Data, NetFlow, Data Streams

Abstract: Thanks to advances in machine learning and applied artificial intelligence, network anomaly detection systems have evolved from traditional signature-based intrusion detection methods to more advanced approaches. Nevertheless, as security measures improve, attackers are constantly developing more sophisticated attack techniques. It is therefore essential not only to deploy a robust anomaly detection algorithm, but also to create a real-time data transfer mechanism that minimizes response delays. Moreover, DDoS attacks can flood network data channels with thousands of packets per second, which leads to the overloading of most traditional monitoring systems that rely on data storage. For this reason, the research presented in this article focuses on implementing a real-time streaming system for network anomaly detection that is capable of operating even under a high volume of data traffic. The proposed solution includes the deployment of a flow collector platform connected to Apache Kafka for receiving NetFlow data from network switches. Subsequently, real-time big data processing techniques are applied via Apache Spark, where ML anomaly detection is triggered. Detection is performed by a combination of the unsupervised k-means clustering method and the supervised KNN (k-nearest neighbors) classifier. Finally, the solution also includes a monitoring system based on the ELK stack, which collects historical data for the further development of the ML algorithms. (Jose et al., 2021)

2.12. Anomaly Detection in POSTFIX mail log using Principal Component Analysis

Abstract: Anomalies in a dataset are the abnormal observations of the data, and they often represent the problems or defectives coming from some changes in the system operation. Detecting the anomalies in a data set is the first step from which further analysis can be done to locate the actual root cause in the data. In this paper, we propose a general method to pinpoint the anomalies from a set of Postfix mail log. This method utilizes the Elastic Stack to collect and process log data, then applies Principal Component Analysis to separate the abnormal data from the normal one. To demonstrate the feasibility of the solution, we test it against a collection of a million Postfix log messages. The result is promising under experimentation as it can detect an anomaly at the time a spam mail attack occurs. (Tran & Tran, 2018)

keywords: anomaly, elasticsearch, elastic stack, postfix, kibana, logstash, watcher

Abstract: Anomalies in a dataset are abnormal observations of the data, which often signal problems or faults caused by changes in system operation. Detecting anomalies in a dataset is the first step, followed by further analysis aimed at determining the actual root cause in the data. This article proposes a general method for identifying anomalies from a set of

Postfix mail server logs. The proposed method uses the Elastic Stack to collect and process log data and then applies principal component analysis (PCA – Principal Component Analysis) to separate abnormal data from normal data. To verify the feasibility of the solution, it was tested on a set containing one million Postfix log messages. The experimental results are promising, as the method is able to detect an anomaly at the time when an attack by means of spam e-mails occurs. (Tran & Tran, 2018)

2.13. Cyberattacks Detection and Analysis in a Network Log System Using XGBoost with ELK Stack

Abstract: The usage of artificial intelligence and machine learning methods on cyberattacks increasing significantly recently. For the defense method of cyberattacks, it is possible to detect and identify the attack event by observing the log data and analyzing whether it has abnormal behavior or not. This paper implemented the ELK Stack network log system (NetFlow Log) to visually analyze log data and present several network attack behavior characteristics for further analysis. Additionally, this system evaluated the extreme gradient enhancement (XGBoost), Recurrent Neural Network (RNN), and Deep Neural Network (DNN) model for machine learning methods. Keras was used as a deep learning framework for building a model to detect the attack event. From the experiments, it can be confirmed that the XGBoost model has an accuracy rate of 96.01% for potential threats. The full attack data set can achieve 96.26% accuracy, which is better than RNN and DNN models. (Yang et al., 2021)

keywords: Cyber Security , Machine Learning , ELK Stack , XGBoost , NetFlow Log

Abstract: The use of artificial intelligence (AI) and machine learning (ML) methods in cyberattacks has grown significantly in recent times. In terms of defence methods against these attacks, an attack can be detected and identified by observing log data and analysing whether they contain abnormal behaviour. In this article, an ELK Stack system for network logs (NetFlow Log) was implemented for the visual analysis of log data and the presentation of selected characteristics of network attack behaviour for further analysis. In addition, the system evaluated the machine learning methods Extreme Gradient Boosting (XGBoost), Recurrent Neural Network (RNN) and Deep Neural Network (DNN). The Keras deep learning framework was used to build the attack detection model. The experimental results confirm that the XGBoost model achieves an accuracy of 96.01% in identifying potential threats. When working with the complete attack dataset, an accuracy of 96.26% is achieved, which is a better result compared with the RNN and DNN models. (Yang et al., 2021)

2.14. Improving State-of-the-art Compression Techniques for Log Management Tools

Abstract: Log data records important runtime information about the running of a software system for different purposes including performance assurance, capacity planning, and anomaly detection. Log management tools such as ELK Stack and Splunk are widely adopted to manage and leverage log data in order to assist DevOps in real-time log analytics and decision making. To enable fast queries and to save storage space, such tools split log data into small blocks (e.g., 16KB), then index and compress each block separately. Previous log compression studies focus on improving the compression of either large-sized log files or

log streams, without considering improving the compression of small log blocks (the actual compression need by modern log management tools). The evaluation of four state-of-the-art compression approaches (e.g., Logzip, a variation of Logzip by pre-extracting log templates named Logzip-E, LogArchive and Cowic) indicates that these approaches do not perform well on small log blocks. In fact, the compressed blocks that are preprocessed using Logzip, Logzip-E, LogArchive or Cowic are even larger (on median 1.3 times, 1.5 times, 0.2 times or 6.6 times) than the compressed blocks without any preprocessing. Hence, we propose an approach named LogBlock to preprocess small log blocks before compressing them with a general compressor such as gzip, deflate and lz4, which are widely adopted by log management tools. LogBlock reduces the repetitiveness of logs by preprocessing the log headers and rearranging the log content leading to an improved compression ratio for a log file. Our evaluation on 16 log files shows that, for 16KB to 128KB block sizes, the compressed blocks by LogBlock are on median 5% to 21% smaller than the same compressed blocks without preprocessing (outperforming the state-of-the-art compression approaches). LogBlock achieves both a higher compression ratio (a median of 1.7 to 8.4 times, 1.9 to 10.0 times, 1.3 to 1.9 times and 6.2 to 11.4 times) and a faster compression speed (a median of 30.8 to 49.7 times, 42.6 to 53.8 times, 4.5 to 6.0 times and 2.5 to 4.0 times) than Logzip, Logzip-E, LogArchive and Cowic. LogBlock can help improve the storage efficiency of log management tools. (Yao et al., 2021)

keywords: Software log compression, Software logging, Log management tools

Abstract: Log data record important information about the running of a software system, which is used for various purposes, including performance assurance, capacity planning and anomaly detection. Log management tools such as ELK Stack and Splunk are widely used to manage and utilize log data in order to support DevOps in real-time log analysis and decision-making. To enable fast queries and save storage space, such tools split log data into small blocks (e.g. 16 KB), which they then index and compress separately. Previous studies on log compression focused on improving the compression of either large log files or log streams, without considering the compression of small log blocks, which is in fact the actual need of modern log management tools. An evaluation of four state-of-the-art compression approaches, Logzip, its variation Logzip-E (with pre-extracted log templates), LogArchive and Cowic, shows that these approaches do not perform well on small log blocks. In fact, it turns out that blocks processed using these approaches are even larger after compression (by a median of 1.3×, 1.5×, 0.2× or 6.6×) than blocks compressed without any preprocessing. Therefore, a new approach called LogBlock is proposed, which enables the preprocessing of small log blocks before their compression with general-purpose compressors such as gzip, deflate or lz4, which are widely used in log management tools. LogBlock reduces log redundancy by preprocessing the log headers and rearranging the content, which leads to a better compression ratio for log files. An evaluation on 16 log files shows that, for block sizes of 16 KB to 128 KB, blocks compressed with LogBlock are a median of 5% to 21% smaller than the same blocks compressed without preprocessing (while also outperforming existing approaches). LogBlock achieves not only a higher compression ratio (a median of 1.7–8.4×, 1.9–10.0×, 1.3–1.9× and 6.2–11.4×) but also a higher compression speed (a median of 30.8–49.7×, 42.6–53.8×, 4.5–6.0× and 2.5–4.0×) than Logzip, Logzip-E, LogArchive and Cowic. LogBlock thus represents an effective way to improve the storage efficiency of log management tools. (Yao et al., 2021)

2.15. SDGen: A Scalable, Reproducible and Flexible Approach to Generate Real World Cyber Security Datasets

Abstract: Real world cyber security datasets are essential for developing and evaluating new techniques to counter cyber attacks. Ideally, these datasets should represent modern network infrastructures with upto-date cyber attacks. However, existing datasets commonly used by researchers are either synthetic, unscalable or easily outdated due to the dynamic network infrastructure and evolving nature of cyber attacks. In this paper, we introduce a security dataset generator (SDGen) which focuses on a scalable, reproducible and flexible approach to generate real world datasets for detection and response against cyber attacks. We implement SDGen within a virtual environment using DetectionLab, ELK (Elasticsearch, Logstash, Kibana) stack with Beats and AttackIQ (a security control validation platform). This implementation in fact provides a proof-of-concept (POC) of SDGen to demonstrate the dataset generation of an organisation being compromised by several types of Ransomware. We showcase that our proposed dataset generator, SDGen, provides scalability, reproducibility and flexibility in generating cyber security datasets by modifying the configurations in DetectionLab, Vagrant-Files and launching different types of attacks in AttackIQ. (Abigail et al., 2022)

keywords: Dataset generation; MITRE ATT&CK framework; Application security

Abstract: Real-world cybersecurity datasets are essential for developing and evaluating new techniques to counter cyberattacks. Ideally, these datasets should represent modern network infrastructures with current types of attacks. However, the existing datasets commonly used by researchers are often synthetic, difficult to scale or quickly become outdated due to the dynamics of network infrastructure and the constantly evolving nature of cyberattacks. This article presents the Security Dataset Generator (SDGen), which focuses on a scalable, reproducible and flexible approach to creating real-world datasets intended for detection of and response to cyberattacks. SDGen is implemented in a virtual environment using DetectionLab, the ELK stack (Elasticsearch, Logstash, Kibana) with Beats, and the AttackIQ platform (a platform for validating security controls). This implementation represents a proof-of-concept (POC) of the SDGen tool, which demonstrates the generation of a dataset of an organization compromised by various types of ransomware. It is shown that the proposed SDGen dataset generator provides scalability, reproducibility and flexibility in creating cybersecurity datasets by modifying configurations in DetectionLab and Vagrant-Files and launching various types of attacks in AttackIQ. (Abigail et al., 2022)

LIST OF SOURCES

[1] Abigail, Xie, M., Ryan, Sterner, C., Choi, T., & Dong, N. (2022). SDGen: A Scalable, Reproducible and Flexible Approach to Generate Real World Cyber Security Datasets. *Communications in Computer and Information Science*, 102–115. https://doi.org/10.1007/978-981-19-0468-4_8.

[2] Andrade, R., & Torres, J. (2018). *Enhancing intelligence SOC with big data tools*. https://doi.org/10.1109/iemcon.2018.8614779.

[3] Currie, R., & Yuan, W. (2024). Building a Flexible and Resource-Light Monitoring Platform for a WLCG-Tier2. *EPJ Web of Conferences*, 295, 07008. https://doi.org/10.1051/epjconf/202429507008.

[4] Folino, G., Otranto Godano, C., & Pisani, F. S. (2023). An ensemble-based framework for user behaviour anomaly detection and classification for cybersecurity. *The Journal of Supercomputing.* https://doi.org/10.1007/s11227-023-05049-x.

[5] Gautam, B. P., & Norio, S. (2020). SUESSA: Sustainable & Ultra-Elastic Stack Security Architecture for Securing IoT Networks of Future Smart Cities. *2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW)*, 387–390. https://doi.org/10.1109/candarw51189.2020.00079.

[6] Halvorsen, J., Waite, J., & Hahn, A. (2019). Evaluating the Observability of Network Security Monitoring Strategies With TOMATO. *IEEE Access*, *7*, 108304–108315. https://doi.org/10.1109/ACCESS.2019.2933415.

[7] Jose, J., Talha Ahmed Khan, Akbar, W., Muhammad Afaq, & Song, W.-C. (2021). An ML Based Anomaly Detection System in real-time data streams. *2021 International Conference on Computational Science and Computational Intelligence (CSCI)*. https://doi.org/10.1109/csci54926.2021.00270.

[8] Kulshrestha, U., & Durbha, S. (2020, September 1). *Edge Analytics and Complex Event Processing for Real Time Air Pollution Monitoring and Control*. IEEE Xplore. https://doi.org/10.1109/IGARSS39084.2020.9323584.

[9] Kyriakou, A., & Sklavos, N. (2018, October 1). *Container-Based Honeypot Deployment for the Analysis of Malicious Activity*. IEEE Xplore. https://doi.org/10.1109/GIIS.2018.8635778.

[10] Mfula, H., & Nurminen, J. K. (2018). Self-Healing Cloud Services in Private Multi-Clouds. *2018 International Conference on High Performance Computing & Simulation (HPCS)*. https://doi.org/10.1109/hpcs.2018.00041.

[11] Poat, M. D., Lauret, J., & Fedele, D. (2023). Flexible visualization of a 3rd party Intrusion Prevention (Security) tool: A use case with the ELK stack. *Journal of Physics. Conference Series*, *2438*(1), 012040–012040. https://doi.org/10.1088/1742-6596/2438/1/012040.

[12] Tran, C.-P., & Tran, D.-K. (2018). Anomaly Detection in POSTFIX mail log using Principal Component Analysis. *2018 10th International Conference on Knowledge and Systems Engineering (KSE)*, 107–112. https://doi.org/10.1109/kse.2018.8573410.

[13] Ummay Faseeha, Syed, H. J., Samad, F., Sehar Zehra, & Ahmed, H. (2025). Observability in Microservices: An In-Depth Exploration of Frameworks, Challenges, and Deployment Paradigms. *IEEE Access*, 1–1. https://doi.org/10.1109/access.2025.3562125.

[14] Yang, C.-T., Chan, Y.-W., Liu, J.-C. L., Endah Kristiani, & Lai, C.-H. (2021, August 23). *Cyberattacks Detection and Analysis in a Network Log System Using XGBoost with ELK Stack*. https://doi.org/10.21203/rs.3.rs-838650/v1.

Funded by the European Union NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC

[15] Yao, K., Sayagh, M., Shang, W., & Hassan, A. E. (2021). Improving State-of-the-art Compression Techniques for Log Management Tools. *IEEE Transactions on Software Engineering*, 1–1. https://doi.org/10.1109/tse.2021.3069958.

Funded by the European Union
NextGenerationEU
RECOVERY AND RESILIENCE PLAN
GOVERNMENT OFFICE OF THE SLOVAK REPUBLIC