Runtime security monitoring by an interplay between rule matching and deep learning-based anomaly detection on logs

Jan Antić
XLAB
Ljubljana, Slovenia
0000-0003-3266-7064

Joao Pita Costa
XLAB
Ljubljana, Slovenia
0000-0001-5745-1302

Aleš Černivec
XLAB
Ljubljana, Slovenia
0000-0003-0011-8482

Matija Cankar
XLAB
Ljubljana, Slovenia
0000-0002-5805-0217

Tomaž Martinčič
XLAB
Ljubljana, Slovenia
0000-0001-6069-8373

Aljaž Potočnik
Hrvoje Ratkajec
XLAB
Ljubljana, Slovenia

Gorka Benguria Elguezabal
TECNALIA
Bilbao, Spain
0000-0003-1375-6731

Nelly Leligou
Alexandra Lakka
Synelixis
Athina, Greece

Ismael Torres Boigues
Eliseo Villanueva Morte
PRODEVELOP
Valencia, Spain

***Abstract*—In the era of digital transformation the increasing vulnerability of infrastructure and applications is often tied to the lack of technical capability and the improved intelligence of the attackers. In this paper, we discuss the complementarity between static security monitoring of rule matching and an application of self-supervised machine-learning to cybersecurity. Moreover, we analyse the context and challenges of supply chain resilience and smart logistics. Furthermore, we put this interplay between the two complementary methods in the context of a self-learning and self-healing approach.**

***Index Terms*—runtime, security monitoring, supply chain resilience, smart logistics, deep learning, natural language processing, anomaly detection, masked language modelling, self learning, self healing**

I. Introduction

In recent years, security attacks have become increasingly frequent and sophisticated, with attackers using advanced techniques to gain unauthorized access to computer systems, steal sensitive data, and cause other forms of damage. These attacks are not only becoming more frequent, but also more automated, making it harder for security teams to keep up. To combat this growing threat, organizations are turning to artificial intelligence (AI) for help. AI can be used to rapidly identify and respond to security threats, analyze large amounts of data to detect patterns and anomalies, and provide insights into potential vulnerabilities. AI-powered security systems can also learn from previous attacks, making them more effective at protecting against future attacks. Despite the great potential of AI for improving cybersecurity, it has its limitations and vulnerabilities. It is important that organizations use AI in conjunction with other security measures, such as firewalls, antivirus software, user training, and insider threats mitigation strategies to create a comprehensive holistic security strategy that addresses all potential threats.

In this paper, we discuss the interplay between the static rule matching approach and the dynamic self-supervised machine learning (ML) approach for anomaly detection based on logs. According to the AI taxonomy of the European Union Agency for Cybersecurity (ENISA), the technology presented in this paper is in the domain of ”NLP & Speech processing” [8] acting on the text data of the logs of infrastructure and applications, utilizing the paradigma of ”unsupervised learning”. With the traction gained by AI in the recent years facilitating intelligence and automation of decision-making, it is regarded as an emerging approach undoubtedly beneficial, but also presenting some threats in itself [9]. In this paper we will not address the latter but we will discuss how it can be used in pair with traditional rule matching approaches also in the context of ENISA’s 2022 Threat Landscape [7], considering the increasing data compromise and the current advances in cybersecurity defences and mitigation strategies.

The static approach is based on the open source security monitoring technology Wazuh (wazuh.com), allowing for the customisation of threat detection, integrity monitoring, incident response and basic compliance monitoring. This technology can be deployed on-premises or in hybrid and cloud environments, ensuring a wide range of applications across several industrial domains, on different target (also edge, lightweight) environments. Vulnerability assessment represents a critical component of the vulnerability management and information technology (IT) risk management lifecycles in the companies of any size, protecting systems and data from any unauthorized access and data breaches; and improving overall security of companies’ systems. The static security monitoring approach discussed in this paper is addressing problems related to the detection of malicious activities on the network level within specific organisations using the detection and protection

978-1-6654-7598-3/23/$31.00 ©2023 IEEE

components that trigger specific actions.

The dynamic part of the log monitoring approach presented in this paper is based on the anomaly detection utilizing machine learning. This is done through a novel LOg MOnitoring System (LOMOS), presented in this paper, with the log data being collected by Wazuh-based agents. The monitoring system is itself a passive component, providing a second layer of analysis, not requiring deployment in the user infrastructure. It is able to recheck metrics and events gathered by appliance tools and to assign them an anomaly score. Raw logs are processed by a Log Parser component, which is based on the Drain method [6]. A sequence of log templates is the input into Masked Language Modeling (MLM), a common self-supervised Natural Language Processing (NLP) methodology to predict masked log templates in the sequence. Moreover, we use Hypersphere Volume Minimization (HVM) assuming that 'normal' samples can be mapped to close representations. This method requires only data from 'normal' operating conditions, without manual labeling of the samples, extending LogBERT method [5].

This work has been developed in the context of the FISHY (fishy-project.eu project), contributing to the cyber resilience of supply chain systems with a coordinated framework facilitating adaptive system reconfigurations and reacting to and defy the effects of cyber attacks in real time in the ICT supply chain end-to-end. It is further put into a DevSecOps context [10] over the project PIACERE (https://piacere-project.eu through an open source platform offering tools, techniques and methods to allow organisations to develop and operate IaC through DevOps practices as they would do with traditional code, where security is defined both at design time and runtime.

The related work includes the anomaly detection analysis of large-scale system log data using a BERT-based pre-training method for the self-supervised learning-based anomaly detection based on a highly efficient natural language processing performance [3] [4], and using the masked language modeling loss function per log keyword during the inference process [14]. Recent research includes also the combination of supervised and unsupervised machine learning with domain knowledge, reducing the number of alerts by predicting anomalous log events based on that domain expertise [12]. It also includes the refinement of the algorithms from the perspective of anomaly scoring and anomaly decision using self-attention neural networks and data augmentation [13], and the improvement of the model within the training data selection, data grouping, class distribution, data noise, and early detection ability [11].

II. Methodology

A. Static Rule-based Security Monitoring

In complex systems, verification is not only a matter concerning the pre-operation steps, but it also encompasses the operation of the system. In particular, it is of paramount importance to make sure that the system is undergoing a continuous runtime verification for any security concerns or violations. Such violations can be recognized by monitoring various security metrics (e.g., file integrity, network configuration changes, usage of software reported as vulnerable, malware detection) and integration with the monitoring component to recognize violations of defined security policies and alert DevSecOps teams to address and eliminate threats as fast as possible. Through security verification and threat detection on multiple levels, the static approach presented in this paper empowers deployed applications by helping to prevent abuse and leakage of data. If any erroneous or anomalous log entries appear, the Wazuh-based technology can alert the user according to the configured rules. In figure 1 we can see FISHY's implementation of Wazuh extended capabilities detailing the rule matched, the agent collected and the time of collection, but also the rule level and ID.

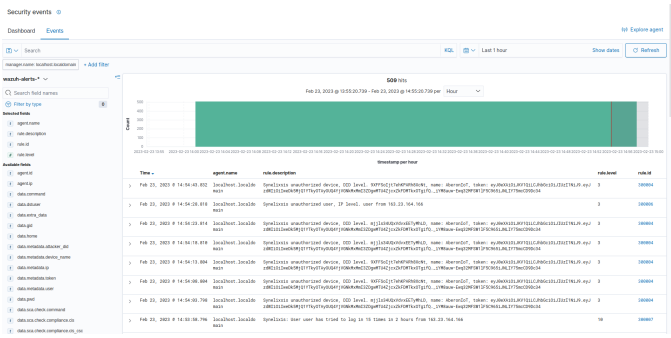

Fig. 1. A screenshot of the dashboard for the rule-based static security monitoring of logs showcasing potential vulnerabilities

Unlike some other evidence gathering tools, Wazuh is not primarily connected to the cloud interfaces, but its agents are installed directly on the (virtual) machines of the monitored infrastructure. The agents can run on many different platforms, such as Windows, Linux, Mac OS X, AIX, Solaris, and HP-UX. The deployment of this technology consists of several Wazuh agents, programs installed on the monitored machines, and a Wazuh server that gathers data from the agents and acts as their Orchestrator. The Wazuh server also contains an Elasticsearch database with a modified Kibana user interface for easier analytics. The system components connect to Wazuh's APIs to examine the configurations and possible alerts detected and based on this data generates evidence about the fulfilment of respective metrics.

Wazuh includes several modules that each support their respective detection capability. For each of the modules, specific rules are defined that include internal metrics and thresholds to trigger events or alerts. When an alert is produced based on some detected event(s), additional actions can be triggered to notify a user or another component about it. With this capability, certain events (e.g., malware detected, Wazuh agent shutdown), can trigger changes of values for specific metrics and event-driven generation of evidence.

B. Dynamic ML-based Security Monitoring

The system log generated in a computer system refers to large-scale data that are collected simultaneously and used as

the basic data for determining simple errors and detecting external adversarial intrusion or the abnormal behaviors of insiders. The aim of system log anomaly detection is to promptly identify anomalies while minimizing human intervention, which is a critical problem in the industry. Previous studies performed anomaly detection through algorithms after converting various forms of log data into a standardized template using a parser. These methods involved generating a template for refining the log key. Software-intensive systems produce logs for troubleshooting purposes. Recently, many deep learning models have been proposed to automatically detect system anomalies based on log data. These models typically claim very high detection accuracy. Logs are primary information resource for fault diagnosis and anomaly detection in large-scale computer systems, but it is hard to classify anomalies from system logs. AI-automated anomaly detection is becoming increasingly important for the dependability and serviceability of IT services across different industries, from the food sector to the maritime sector.

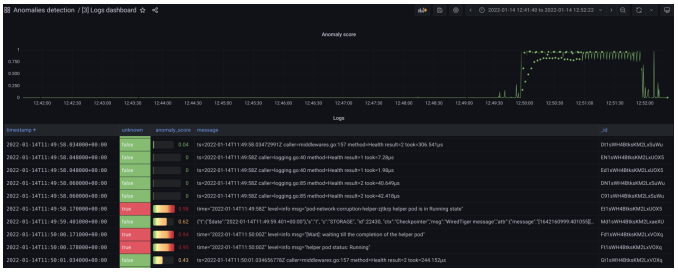

Fig. 2. A screenshot of the dashboard for the self-supervised anomaly detection from logs showcasing threat levels and characteristics

Traditional log monitoring solutions are limited to rule-based (manual) analysis of time series data. In contrast, LOMOS makes use of state-of-the-art Natural Language Processing architectures to model log streams and capture their normal operating conditions. This enables the implementation of a monitoring system that does not depend on any manually defined rules or human intervention, but that relies on that behavioural model to automatically detect deviations that would represent any kind of abnormal situations, including potential security threats. Our approach automatically analyses system or application logs and provides valuable insights regarding the current and past status of the monitored assets. The technology uses deep learning techniques, to compute anomaly score on sequences of log templates, applying NLP models to the IDs defined for the templates.

Without any manual preprocessing of raw logs from unstructured data, LOMOS aims at learning patterns in logs and identifying anomalous behaviour. To that aim, LOMOS tries to get some structure by identifying what are the log-templates that would match the log. The algorithm behind this tries to identify parameters (ids, services, ports, etc.) transforming the unstructured logs to structured log templates broken down according to the tree structure (with wildcards for parameters changing from log to log). Then, LOMOS observes the sequence of templates trying to learn what is the normal behavior, and provides an anomaly score (including aggregation and specific counts) that should be low for normal logs. In figure 2 we can see the LOMOS dashboard showcasing the dated identified threats, their ranking and the call for action (green/red).

There is a training period and a monitoring period ensuring a rapid response, based on the extension of LogBERT algorithms [5], using Drain [6] for log template parsing, and OpenSearch (opensearch.org) or Grafana (grafana.com) to setup alerts and send them to specific services. Real-time anomaly detection would require incremental learning (learn sample by sample and get immediate updates to the ML model) but in this case it does not make sense as we cannot update the model at every log.

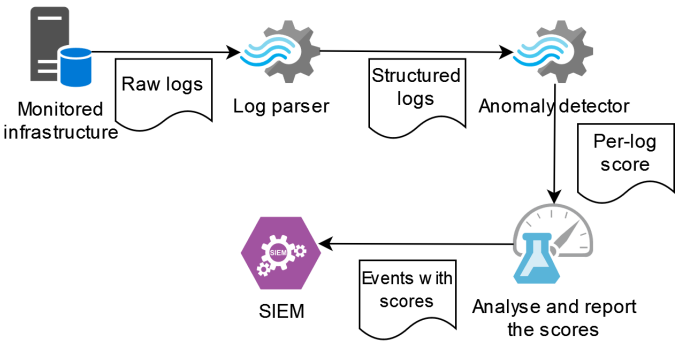

Fig. 3. Anomaly detection workflow from logs in this paper's approach

III. Contributing to the Challenges of a Smart Supply Chain Security

Ensuring the security of supply chains has become an increasingly challenging task in recent years. The global nature of supply chains means that products can pass through multiple countries and involve numerous companies (as nodes across the supply chain), making them vulnerable to a range of security risks. Cyber threats have been causing increasingly disruptive impacts, resulting in financial losses and reputation damage, highlighting the need for increased resilience and adaptability.

This includes log data streams from virtual machines running across multiple actors' premises along the supply chain system. To validate the proposed approach, we have focused on smart supply chain IT system which aggregates information from the producer, the transporter and the warehouse. Although the number of security threats and attacks for the supply chain are numerous and proliferate, we focus on the following as indicative examples: (i) unauthorized users; (ii) unauthorised devices and (iii) attacks to the blockchain functionality employed nowadays to improve the security levels of traditional supply chain systems. Logs passing a threshold will be stored in a central threat repository and the users (most probably the administrators of the relevant systems) are notified to take action, and/or an automated policy is recommended and (if allowed) enforced.

The ML-based anomaly detection approach discussed in this paper is used in the context of supply chain resilience to provide a second layer of analysis of gathered data and metrics. While most tools and services employed by the FISHY project form their own chain and process of gathering and analysing data, LOMOS is employed in a more passive, observational role. Instead of deploying its own data collectors and agents in the monitored infrastructure, it taps into the data stream of already collected metrics flowing to the FISHY platform and establishes a baseline for normal activity and searches for anomalous behaviour that a more specialised, rule-matching solution may miss.

LOMOS consumes raw logs, that are fed into its Log parser module, which structures the logs into a format ready for analysis. This part of the process is unsupervised learning, meaning the incoming logs do not need to be in a known format or require any pre-processing, also known as "cooking". The structured logs output by the log parser are then analysed by the anomaly detector module, which produces an individual, per-log anomaly score. Thus, LOMOS provides a second layer of analysis. Logs with an anomaly score that surpasses a threshold are persisted in the Central Repository and since data coming into the FISHY platform is always labelled by its source, the alerts generated by LOMOS allow system administrators to drill down into the sequence of events flagged as out-of-the-ordinary.

In the context of smart logistics, the opportunity lies on the efficiency of a continuous deployment of a Port Community System (PCS)., which is a complex big data solution with lots of configurations and parameterizations. Each installation requires a new project with much customization, and each region/ country has different laws, procedures and security requirements and regulations (e.g. ISO/IEC 27001). Different deployment typologies and providers are considered (cloud or on-premise), and cybersecurity challenges can be impactful (e.g. shutdown operations, human injures, etc.).

The technology proposed in this paper improves the operational security of the IaC, notifying about security threats according to the policies defined [2], guaranteeing security and privacy when using tools/data in virtualized environments, and accomplishing the SLAs. In addition, the proposed solution can detect security and performance problems and correct them automatically thanks to self-healing and self-learning capabilities. This is ensuring security and compliance with NFRs defined with clients in order to achieve greater client satisfaction by providing Quality of Service (QoS). With it is possible to reach more potential clients that have not much interest in the core Industrial Internet of Things (IIoT) technology but in specific applications or they already have an IIoT solution but decide to substitute or integrate with a PCS suite of tools such as, e.g., Posidonia (prodevelop.es/en/posidonia-pcs).

IV. A SELF LEARNING AND SELF HEALING CONTEXT

With the increasing complexity and scale of IT systems, the traditional approach of manually managing and troubleshooting them has become increasingly challenging. The concepts of deployment and monitoring are evolving from focusing on single servers to complex ecosystems of networks and servers. This shift demands a different paradigm for protecting against smarter and more efficient security attacks. To address this, self-learning and self-healing approaches are emerging as potential solutions. Self-learning IT systems use machine learning algorithms to monitor and analyze the system's behavior, identify patterns and anomalies, and predict potential issues before they occur. This allows the system to learn from its past behavior and continuously improve its performance (in terms of capability to defend against security attacks) over time. Self-healing IT systems take this step further by automatically detecting and resolving issues without human intervention. This may include tasks such as network reconfiguration, banning of IP or of preventing access from specific devices, rebooting a server, rerouting traffic to avoid a network bottleneck, or even halting the entire deployment. While self-learning and self-healing IT systems are still in their early stages, they have the potential to reduce downtime, improve reliability, and ultimately provide a more efficient and reliable IT infrastructure.

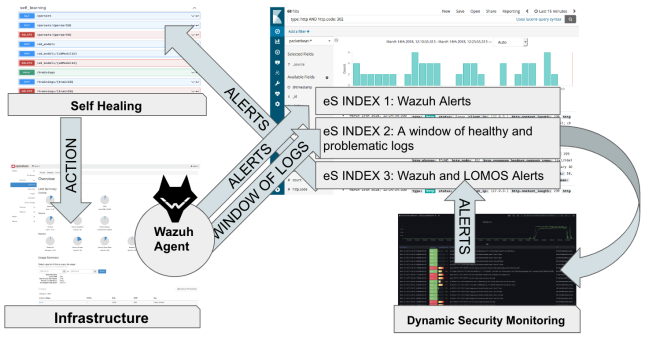

Fig. 4. Overall system functionality of static and dynamic security monitoring, aligning with self learning and self healing mechanisms

In order to always ensure the business continuity of the Infrastructure as Code (IaC) with respect to the pre-selected Non-Functional Requirements (NFRs), a monitoring component is proposed to be put into place. This monitoring component will not only ensure that the conditions are met, but also that a failure or a non-compliance of a NFRs is not likely to occur. This component will consume the data monitored and stored in a time-series database to create discriminative complex statistical variables and train a predictor which will learn potential failure patterns in order to prevent the system from falling into an NFR violation situation. Due to the critical nature of the environment, the performance of the failure prediction is guaranteed, always with the highest quality, by advanced learning strategies such as Concept Drift and Incremental-Learning in charge of keeping the optimal precision in whatever conditions. Once the situation becomes risky for the accomplishment of the defined NFRs, an alert to the DevSecOps teams will be sent and (automatically) the most appropriate IaC deployment configuration will be sought

and selected by the optimizer. PIACERE at runtime is capable of self-learning and self-healing and can tackle unexpected situations that may affect the correct performance of IaC and its underlying environment (i.e., infrastructure failures, deterioration in the response time, etc.). The IaC DevSecOps can therefore ensure that their infrastructural code is always conforming to the Service Level Agreements committed with the end-user even if the environmental situation changes.

This approach is able to automatically deploy security monitoring agents, integrated into the monitoring mechanisms and notify about security threats according to the policies, and react by predefined actions. It is addressing the need for monitoring stack for the runtime conditions so that the self-learning and self-healing mechanisms can operate. It does so through a fully automated monitoring system capable of detecting security-related events and incidents in the deployed application's environment. The Self-healing suggestion messages are received as in figure 5, presenting their id and timestamp but also the event types and the edit options.

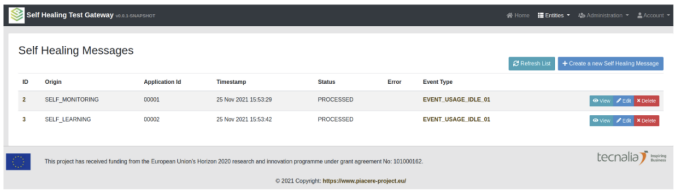

Fig. 5. Messages received in the Self-Healing component of PIACERE [1]

Conclusions and Future Work

The paper discusses the combination of a static rule matching approach with dynamic self-supervised machine learning to detect anomalies on logs. While the static approach is based on open-source security monitoring technology allowing for customized threat detection and incident response, the dynamic approach utilizes machine learning to perform anomaly detection and is implemented through a monitoring system that extracts log templates using state-of-the-art language models allowing for the transformation of unstructured logs into structured ones and consequent improvement of anomaly detection. These complementary approaches have achieved results in supply chain resilience in the context of the FISHY project, and in the application of a novel DevSecOps framework in the context of smart logistics through the PIACERE project.

One key research question in this field is whether identified anomalies can be transformed into Security Information and Event Management (SIEM) rules (e.g., Wazuh), and how the LOMOS tool can contribute to this process. This is done in the context of the interaction with the overall system's SIEM. Validating LOMOS and integrating its results directly into the SIEM could improve the its ability to learn and adapt to new threats. Additionally, the ability to aggregate logs from different applications and nodes could provide valuable insights and enable the creation of more accurate and comprehensive events. The steps towards better leveraging the interaction between the static and dynamic approaches discussed in this paper could include their validation, assessment and comparison in different supply chain IT systems. In particular, this includes comparing and tagging anomalies, moving them to a training set, and converting them into events that can be integrated into the SIEM. As this research progresses, the potential for creating more robust and efficient supply chain cybersecurity continues to increase.

Furthermore, the field of self-learning and self-healing IT systems is rapidly evolving, and it offers great potential for improving the reliability and efficiency of IT infrastructure. With the increasing complexity of IT systems and the growing threat of security risks, there is a need for innovative approaches that can adapt and learn in real-time.

Acknowledgment

This project has received funding from the European Union's Horizon 2020 research and innovation programmes under Grant Agreements No. 101000162 (PIACERE), 952644 (FISHY) and MEDINA (952633)

References

[1] J. Alonso and G. Benguria. *PIACERE run-time monitoring and self-learning, self-healing platform - v1*. PIACERE Project, 2021.

[2] M. Cankar, N. Petrovic, J. P. Costa, A. Cernivec, J. Antic, T. Martincic, and D. Stepec. Security in devsecops: Applying tools and machine learning to verifications and monitoring steps. In *FASTContinuum Workshop 2023, ICPE '23 Companion, April 15–19, 2023, Coimbra, Portugal*, pages 1–4, 11 2022.

[3] S. Chen and H. Liao. Bert-log: Anomaly detection for system logs based on pre-trained language model. *Applied Artificial Intelligence*, 36(1):2145642, 2022.

[4] A. FarzadT and A. Gulliver. Unsupervised log message anomaly detection. *ICT Express*, 6(3):229–237, 2020.

[5] H. Guo, S. Yuan, and X. Wu. Logbert: Log anomaly detection via bert, 2021.

[6] P. He, J. Zhu, Z. Zheng, and M. R. Lyu. Drain: An online log parsing approach with fixed depth tree. In *2017 IEEE international conference on web services (ICWS)*, pages 33–40. IEEE, 2017.

[7] I. Lella, E. Tsekmezoglou, and et al. *ENISA Threat Landscape 2022*. ENISA, 2021.

[8] A. Malatras, I. Agrafiotis, and M. Adamczyk. *Securing Machine Learning Algorithms*. ENISA, 2021.

[9] A. Malatras and G. Dede. *Artificial Intelligence Cybersecurity Challenges*. ENISA, 2020.

[10] N. Petrovic, M. Cankar, and A. Luzar. Automated approach to iac code inspection using python-based devsecops tool. In *30th Telecommunications Forum (TELFOR)*, pages 1–4, 11 2022.

[11] P. Raut, A. Mishra, S. Rao, S. Kawoor, S. Shelke, M. Deore, and V. Kumar. *Review on Log-Based Anomaly Detection Techniques*, pages 893–906. 02 2022.

[12] A. Shah, D. Pasha, E. Zadeh, and S. Konur. *Automated Log Analysis and Anomaly Detection Using Machine Learning*. 10 2022.

[13] T. Wittkopp, A. Acker, S. Nedelkoski, J. Bogatinovski, D. Scheinert, and W. F. O. Kao. A2log: Attentive augmented log anomaly detection. In *Proceedings of the Hawaii International Conference on System Sciences*, 2021.

[14] P. K. Yukyung Lee, J Kim. Lanobert: System log anomaly detection based on bert masked language model. *arXiv preprint*, 211109564, 2021.