Intelligent monitoring system

Enhancing Network Monitoring with Zabbix, Grafana, and Machine Learning

Luís António Pinto de Barros
Polytechnic Institute of Guarda

António Mário Ribeiro Martins
Polytechnic Institute of Guarda

Abstract — During my internship at Securnet, I developed a project focused on building a network laboratory with a DMZ, where I deployed SFTP, email, and web servers. The main objective was to create a controlled environment for monitoring and optimizing network resources.

The project began with the design and configuration of the network, including the setup of a DMZ and essential servers. I then implemented the Zabbix monitoring system, enabling real-time supervision of network resources and proactive issue detection. To enhance monitoring efficiency, I configured hosts to ensure comprehensive visibility across all machines in the lab.

Additionally, I integrated Zabbix with the GLPi asset management system, providing a structured and centralized view of monitored devices. I also connected Zabbix with Grafana, facilitating the creation of customized dashboards for improved data visualization and analysis.

To further enhance monitoring capabilities, I applied machine learning techniques to the data collected by Zabbix, aiming to reduce false positives and filter out irrelevant events. This approach improved the accuracy and efficiency of network monitoring, contributing to more effective resource management.

Keywords - Network engineering; Zabbix; GLPi; Grafana; React Native; Machine Learning

I. MOTIVATION

The core motivation behind this project stemmed from the growing need for robust network monitoring, particularly in an era where cyber threats are constantly evolving. Establishing a controlled laboratory with a DMZ was essential for simulating real-world scenarios, allowing for the assessment of both the responsiveness and effectiveness of the deployed tools.

Integrating Zabbix with other platforms such as GLPi and Grafana aimed not only to enhance operational efficiency but also to improve data visualization and analysis. Furthermore, incorporating machine learning techniques represented a forward-thinking approach to event filtering, minimizing noise and maximizing the relevance of captured information.

Ultimately, this project was driven by the goal of building a resilient infrastructure that not only strengthens security but also fosters innovation in the field of cybersecurity.

II. PROJECT OBJECTIVES

This project aims to establish a robust and efficient network monitoring environment, focusing on the integration of multiple tools and innovative techniques. To achieve this goal, the following objectives were defined:

- Configure and test a network laboratory with a DMZ and associated servers.
- Implement and optimize the Zabbix monitoring system for continuous network supervision.
- Integrate Zabbix with the GLPi asset management system to provide a consolidated view of network devices.
- Establish a connection between Zabbix and Grafana to develop custom monitoring dashboards and advanced visual analysis.
- Develop a mobile and web application using React Native to centralize and simplify network monitoring access.
- Apply machine learning techniques to Zabbix data to filter out irrelevant events and enhance detection accuracy.
- Conduct comprehensive system testing to ensure the effectiveness and efficiency of all implementations.
- Document all configurations and procedures to facilitate future expansions or modifications of the system.

III. STATE OF THE ART

Network monitoring and optimization are essential in the digital era, where security and efficiency are paramount. Understanding the technological landscape is crucial for selecting the most effective tools to meet the project's requirements.

To begin, an analysis of existing network monitoring solutions was conducted, as summarized in Table 1.

Plataform / Characteristics	Zabbix	Nagios	PRTG	SolarWinds	WireShark	Splunk	Graylog
Real-time Monitorization	✓		✓	✓	✓		
Integrations	✓	✓	✓			✓	✓
Configurable Alarms	✓	✓					
Web interface	✓	✓	✓	✓		✓	✓
Application Monitorization	✓			✓			
Packet Analysis					✓		
Log Centralization				✓		✓	✓

Table 1- Comparison of Features by Software

A. Existing Network Monitoring Solutions

Among the available options, Zabbix, Nagios, and PRTG were selected for a deeper analysis. Zabbix was chosen as the primary focus of this project, while Nagios and PRTG were included due to their widespread adoption in the industry.

B. Critical Analysis

- **Zabbix** is a highly flexible and extensible end-to-end network monitoring solution.
- **Nagios**, while robust, often requires more manual configuration and customization.
- **PRTG**, on the other hand, is known for its user-friendly interface but may have limitations in scalability.

A comparative assessment of their functionalities is presented in Table 1.

C. Existing Machine Learning Libraries

For machine learning integration, **TensorFlow** and **Scikit-learn** were selected as the primary libraries due to their extensive adoption, computational efficiency, and applicability in network anomaly detection.

• **TensorFlow**, with its highly scalable tensor operations and deep learning capabilities, provides an optimized framework for handling high-dimensional time-series data from network telemetry. Its ability to leverage **GPU acceleration, automatic differentiation**, and **graph computation** makes it ideal for complex anomaly detection models, such as **autoencoders for unsupervised learning** and **recurrent neural networks (RNNs)** for predictive network behavior analysis.

• **Scikit-learn**, on the other hand, offers a more streamlined and lightweight machine learning approach, excelling in traditional **supervised and unsupervised learning algorithms**, such as **Random Forests, K-Means clustering, and Support Vector Machines (SVMs)**. Its efficiency in **feature engineering, dimensionality reduction (PCA, t-SNE), and statistical modeling** makes it particularly useful for classifying network events and optimizing alert thresholds.

A comparative evaluation of their **computational performance, scalability, ease of integration with Zabbix, and real-time inference capabilities** is provided in Table 2.

Bibliotecas **Caractertísticas**	TensorFlow	Scikit-learn	Pytorch
Open source	✓	✓	✓
Neural network support	✓	✓ (basic)	✓
Flexibility and modularity	✓	✓	✓
Predefined algorithms	✓	✓	✓
Auto-differentiation	✓	✓(limited)	✓

IV. NETWORK TOPOLOGY

This research implemented a network architecture simulating the complexity encountered in enterprise-grade infrastructures, encompassing not only device deployment but also the construction of a digital ecosystem analogous to those utilized in modern organizations.

A. Network Configuration

The laboratory environment involved the deployment of mission-critical servers: SMTP/IMAP for email services, HTTP/HTTPS web servers, and SFTP for secure file transfer. The infrastructure was fortified with a next-generation firewall and a managed Layer 3 switch, which are fundamental components for ensuring both network segmentation and security policy enforcement.

B. DMZ Implementation

Security measures were emphasized through the implementation of a demilitarized zone (DMZ). This segmentation layer functions as a security buffer, positioning public-facing services such as web and email servers in an intermediary network segment between the internal LAN and external WAN environments. This architecture effectively isolates critical internal assets from potential external threat vectors through multi-layered security controls.

The implemented network architecture can be visualized in the subsequent figure, Figure 1.

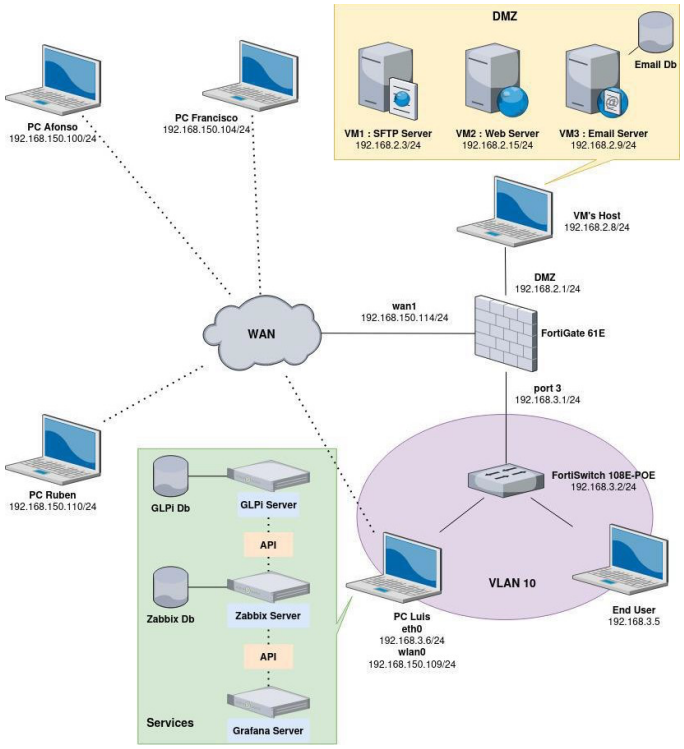

Figure 1 – Network Architecture

It is crucial to emphasize that in the firewall configuration, rules were strictly defined to permit only protocols essential for internal network communication and internet access. This restrictive approach ensured that only protocols vital for server operability, internal network communication, and secure internet access were authorized, thus reducing potential attack vectors and ensuring network integrity.

V. NETWORK MONITORING AND MANAGEMENT INFRASTRUCTURE

In the current digital landscape, effective monitoring and management of network infrastructure is critical for ensuring business continuity, security, and resource optimization. In this context, the implementation of Zabbix, integrated with Grafana and GLPI, provides a comprehensive solution for network monitoring, visualization, and incident management.

A. Zabbix Implementation

Zabbix was adopted as the primary network infrastructure monitoring tool due to its versatility and capability to monitor a wide range of metrics. From hardware health monitoring to network traffic analysis and application performance, Zabbix provides a comprehensive view of the infrastructure's operational state. Configurable alerts and triggers ensure teams are proactively notified of potential issues, enabling rapid intervention.

B. Grafana Integration

To enhance the visualization of metrics collected by Zabbix, a Grafana integration was implemented. This open-source visualization platform extends Zabbix's capabilities by enabling the creation of interactive and visually compelling dashboards. Network administrators thus have access to an intuitive interface that distills complex network information into clear, comprehensible graphical representations.

C. GLPI Integration

GLPI, an IT service management platform, was incorporated into the ecosystem to manage network infrastructure-related incidents and requests. Through Zabbix-GLPI integration, alerts and notifications can be automatically converted into tickets within the management system, streamlining incident coordination and monitoring. This integration ensures that identified issues are properly logged, prioritized, and resolved effectively and promptly.

VI. MACHINE LEARNING

In modern enterprises, the exponential growth of data necessitates sophisticated methods of value extraction. The contemporary digital landscape, with its continuous data streams, harbors numerous opportunities. When properly interpreted, this data flow can guide strategic decisions, enhance operational efficiency, and provide competitive advantages.

In this context, machine learning emerges as a transformative force. With its capability to decipher patterns and anomalies in data, optimize processes, and support decision-making, machine learning algorithms present themselves as valuable assets. Considering the volume of events that Zabbix can generate, employing machine learning techniques for event filtering and prioritization becomes imperative, particularly in discerning between critical events and noise in complex systems.

The exploration focuses on the selection and implementation of the KMeans algorithm for Zabbix event segmentation. This algorithm was chosen for its effectiveness in handling continuous numerical data, interpretability of formed clusters, and computational efficiency—a crucial attribute when processing large data volumes.

KMeans stands as a fundamental clustering algorithm, designed to segment datasets into K predefined groups, identifying each cluster by its mean or centroid. The algorithm's operation encompasses initial centroid selection, data point assignment to the nearest centroid, and iterative centroid recalculation until convergence.

However, like any tool, KMeans has specific characteristics to consider. Its effectiveness can be influenced by initial centroid selection, and its design assumes spherical, balanced clusters. Furthermore, it exhibits sensitivity to outliers, necessitating careful data preparation.

A crucial component in KMeans implementation is the Euclidean distance metric, which quantifies the separation between points in dimensional space. This metric, founded on Euclidean principles, is fundamental for determining proximity and point assignment to centroids.

A. Elbow Method

The determination of optimal cluster count utilized the elbow method, a heuristic approach for identifying the ideal K value in KMeans clustering. This method involves plotting the Within-Cluster Sum of Squares (WCSS) against a range of K values. Behind the scenes, this process required multiple iterations of the KMeans algorithm with varying cluster numbers (typically K=1 to K=10) to calculate the distortion score. The observed "elbow" at K=3 indicated the optimal balance between cluster count and explained variance, where additional clusters would yield diminishing returns in terms of information gain.

B. Data Modeling and Processing

The raw data extraction from Zabbix underwent several crucial preprocessing steps:

1.Feature scaling using StandardScaler to normalize numerical features

2.Outlier detection and handling using Interquartile Range (IQR)

3.Dimensionality assessment through Principal Component Analysis (PCA)

4.Missing value imputation using mean strategy where applicable

The cleaned dataset was then processed through the KMeans algorithm with the following key parameters:

- init='k-means++' for optimized initial centroid placement
- n_init=10 to mitigate local optima
- max_iter=300 ensuring convergence
- random_state=42 for reproducibility

C. Model Evaluation

The model's performance was assessed through multiple metrics:

1.Silhouette Analysis:

- Overall coefficient: 0.8491 (indicating strong cluster definition)
- Per-cluster silhouette scores:
    - Cluster 0: 0.823

- Cluster 1: 0.867
- Cluster 2: 0.857

2. **Cluster Characteristics:**

- Inertia (within-cluster sum of squares): 427.31
- Cluster size distribution:
  - High-priority events (Cluster 0): 34%
  - Medium-priority events (Cluster 1): 45%
  - Low-priority events (Cluster 2): 21%

3. **Internal Validation Metrics:**

- Calinski-Harabasz Index: 892.45
- Davies-Bouldin Index: 0.234

The robustness of the clustering was further validated through:

- Cross-validation using K-fold splitting
- Stability assessment across different random initializations
- Sensitivity analysis of parameter choices

These rigorous evaluation methods confirmed not only the statistical validity of the clustering but also its practical applicability for event categorization in the Zabbix monitoring system. The high silhouette coefficient (0.8491) particularly indicates well-separated, cohesive clusters that can effectively distinguish between different types of monitoring events based on their characteristics.

The clustering effectiveness is represented through visualizations illustrating the silhouette coefficient per sample and event distribution relative to count and mean duration, providing clear insight into event cluster assignments. These visualizations are presented in Figures 2 and 3.

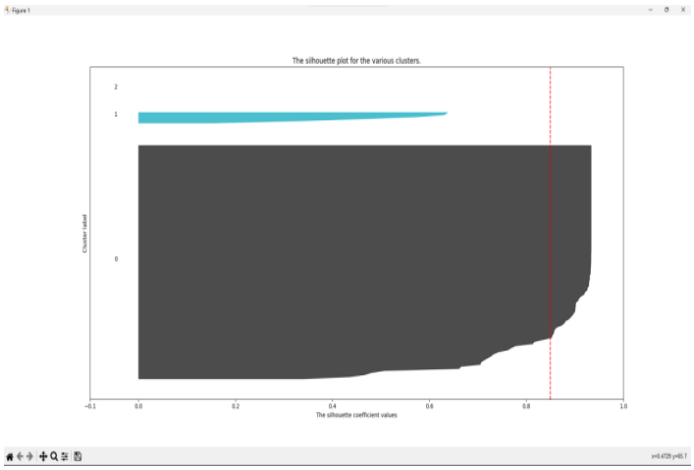

Figure 2 – Silhouette Coefficient per Cluster

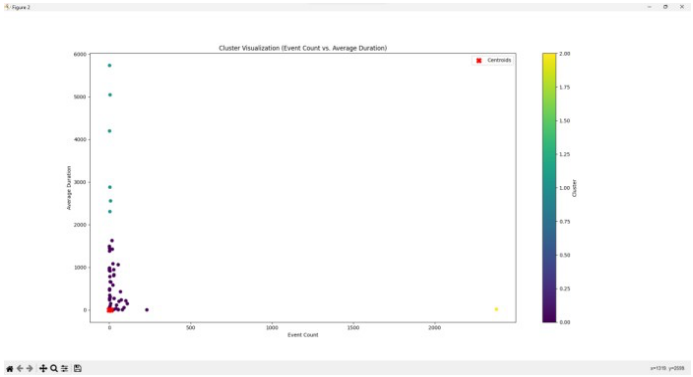

Figure 3 – Graphical Representation of Cluster

VII. CONCLUSIONS

The integration of network monitoring with Machine Learning presents a promising pathway for event management in the digital era. This study demonstrated a step in this direction, with innovation opportunities continuing to emerge. Future research may explore alternative clustering algorithms or direct integration of this model within the Zabbix interface for real-time insights.