INTEGRATING GNN WITH ELK STACK FOR ENHANCED SECURITY LOG ANALYSIS

[1]Dr. J. R. Panchal, [2]Anupama Rajeevan, [3]Yogeshwari Bagul,[4]Rutuja Bhosale, [5]Susovan Bhowmik

[1]Professor, Dr. D. Y. Patil College of Engineering and Innovation, Talegaon, Pune, India,
[2]Students, Dr. D. Y. Patil College of Engineering and Innovation, Talegaon, Pune, India ,
[3]Students, Dr. D. Y. Patil College of Engineering and Innovation, Talegaon, Pune, India
[4]Students, Dr. D. Y. Patil College of Engineering and Innovation, Talegaon, Pune, India ,
[5]Students, Dr. D. Y. Patil College of Engineering and Innovation, Talegaon, Pune, India
[1]jagrutirpanchal2022@gmail.com, [2]anupamarajeevan54@gmail.com,
[3]bagulyogeshwari@gmail.com, [4]rutujabhosale203@gmail.com, [5]susovanbhowmik418@gmail.com

***Abstract*— Enhancing threat detection with SIEM tool using GNN aims to enhance the threat detection system by using GNN (Graph neural network) with integration of SIEM tool. By using real-time data of security events through GNN which represents complex relationships and patterns between the data in the form of nodes and edges. The project involves integrating GNN models into existing SIEM frameworks to enhance the threat detection, optimizing them for scalability, accuracy and effectiveness. The system provides a real time alerts when suspicious activity is detected. This project highlights the effectiveness of combining GNN with SIEM tool to boost cybersecurity defenses.**

***Index Terms*— GNN, SIEM Tool, Nodes, Edges, Alerts, Threat Detection.**

I. INTRODUCTION (HEADING 1)

In today's increasingly digital world, where technology is used in nearly every aspect of our life, cyber security has become more crucial than ever. Businesses, Governments and individuals relies heavily on online platform and connected devices. The landscape has been continuously evolving making cyber-attacks more frequent. Cyber criminals are constantly finding new ways vulnerabilities. So safe guarding sensitive information, ensuring the integrity of systems and maintaining privacy are fundamental priorities [2][5][6].

Application security is a critical concern for organizations as the threat landscape continues to evolve. Security information and event management (SIEM) tool plays a pivotal role in bolstering application security by offering real time monitoring, detection and response to security incidence across the IT environment.

SIEM is a comprehensive security threat management solution that aggregates, analyses and co-relates security event data from a variety of sources such as firewalls, servers, end points and application. It helps organizations detect anomalies, identify potential threats and takes action before breaches or attacks escalate[12]. But in the rapidly evolving cyber-crimes traditional SIEM systems are often challenged by volume, complexity and interconnected nature of security event.

By leveraging GNNs within the existing SIEM ecosystem, it's possible to address the limitations of conventional systems, providing security teams with a smarter, more adaptable, and less burdensome way to detect real threats while reducing false positives[1].

By using GNN in SIEM application like threat detection, user-entity behavior analytics, attack path prediction can be done[4][7][10]. System gives alerts to the admin about the threats so that the admin can take action according to the threat.

II. MATERIALS AND METHODS:

1. Data Sources

a. Log Data Collection:

- Source: Windows Security Logs, Sysmon logs (Event ID 1, 3, 10, 11, etc.)
- Tools Used:
  - Winlogbeat: For forwarding Windows event logs to Logstash.
  - Elasticsearch: Central log storage.
  - Logstash: Pipeline for ingesting, parsing, and filtering logs.

2. Data Preprocessing

a. Feature Extraction:

- Extracted fields: source IP, destination IP, user ID, process name, timestamp, event ID, command line.
- Categorical encoding and normalization performed using scikit-learn.

b. Labeling:

- Events were labeled as benign or malicious based on known attacks or dataset annotations.

3. Graph Construction[16]

a. Graph Representation:

- Nodes: Users, Processes, Hosts, Files, IP Addresses.
- Edges: Relationships such as logon, process creation, file access, and network connection.

b. Temporal Graphs:

- Graphs were constructed over a fixed time window (e.g., 5 minutes) to reflect session-based behavior.
- Each graph snapshot captures temporal and relational patterns.

c. Graph Tools:

- NetworkX: Initial graph visualization and structure verification.
- PyTorch Geometric / DGL: For dynamic graph construction and feeding into GNNs.

4. Model Architecture

a. Graph Neural Network (GNN):

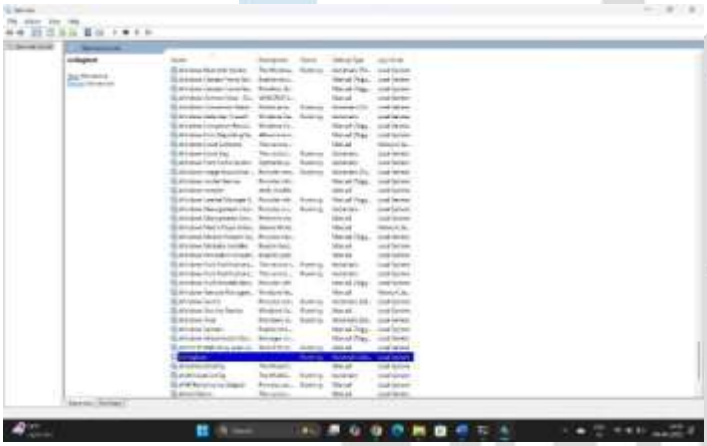
- Model Types Evaluated:
  - ➢ Graph Convolutional Network (GCN)
- Input: Node feature matrix and edge index list.
- Output: Binary classification logits (Malicious / Benign)

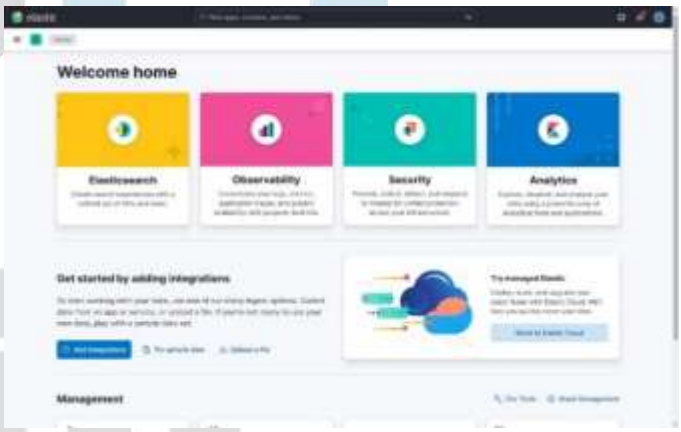
5. Real-Time Inference Pipeline

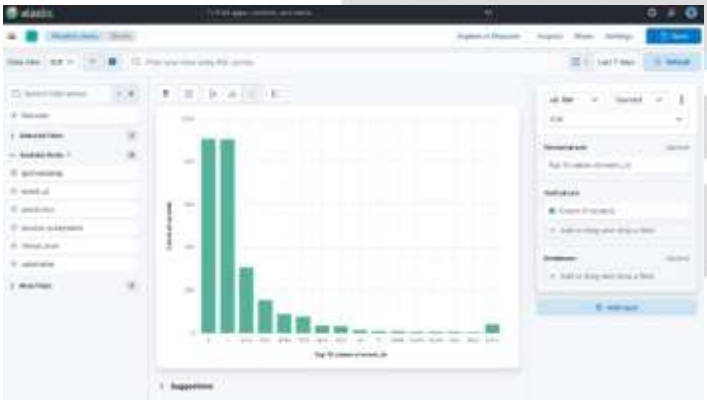
a. Architecture:

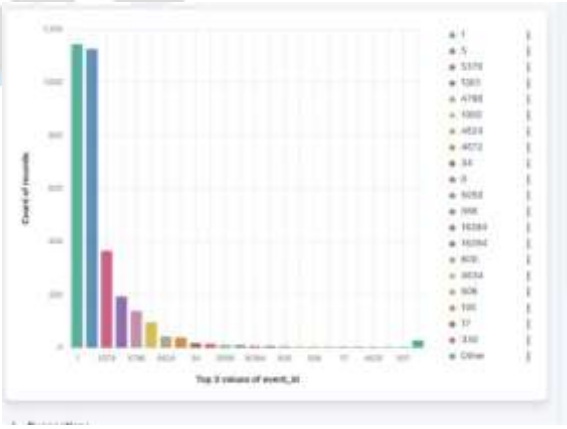
- A Flask API receives logs, transforms them into graphs, and returns predictions.
- The output is forwarded back to Elasticsearch for indexing and visualization in Kibana.

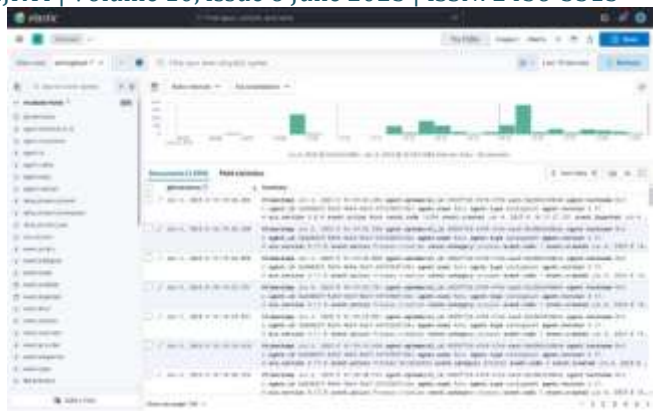
III. RESULTS:











IV. DISCUSSION:

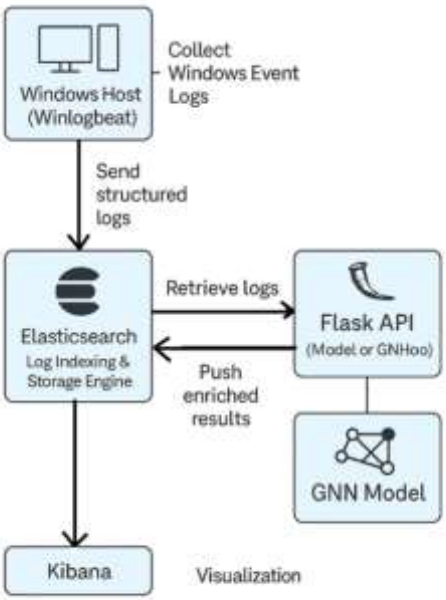
The project focused on building a centralized Windows Event Log monitoring and analytics system using the **ELK Stack (Elasticsearch, Logstash, Kibana)** integrated with **Winlogbeat** and a custom **Flask API**. The goal was to collect, process, analyze, and visualize Windows logs in real time to enhance system monitoring and threat detection capabilities. **Winlogbeat** was installed on Windows endpoints to collect logs from various sources, including **Security**, **System**, and **Application** events. These logs were shipped to **Logstash**, where they were parsed and filtered before being indexed into **Elasticsearch**.

A lightweight **Flask-based API service** was developed to extend the system's functionality. This API served as a bridge between external log analysis tools or machine learning models and the ELK stack. For example, the API could fetch recent events from Elasticsearch, process them and then return results or post alerts back into Elasticsearch for visualization in Kibana. This modular design made it possible to plug in future components like a **Graph Neural Network (GNN)** for advanced threat detection.

Kibana was used to build rich dashboards that visualized key metrics, such as user login activity (both successful and failed attempts), unauthorized access attempts, and service startup/shutdown events. Real-time streaming views and filtered tables allowed system administrators to monitor events as they happened.

The project was completed successfully, achieving improved visibility into Windows system activity, enhanced real-time threat monitoring, and a scalable architecture. The use of a **Flask API** added modularity and integration flexibility, preparing the system for advanced analytics and automation pipelines.

V. SYSTEM ARCHITECTURE:



1. Windows Host (Winlogbeat) Purpose: Collect raw logs from the Windows operating system.

- Component: Winlogbeat agent is installed on the host.
- Action: It collects Windows Event Logs (e.g., login attempts, process creations).
- Output: Sends structured logs to Logstash in real-time.

2. Logstash (Parsing & Filtering) Purpose: Acts as a log processing pipeline.

- Functionality:
  - Parses incoming logs.
  - Applies filters (e.g., grok, mutate) to clean and structure data.
- Two-way Interaction:
  - Retrieves logs from the source (Winlogbeat).
  - Sends logs to the Flask API for enrichment using GNN.
  - Receives enriched logs from Flask API and pushes them to Elasticsearch.

3. Flask API (Model or GNN Hook) Purpose: Acts as an interface to communicate between Logstash and the GNN model[14][15].

- Role:
  - Accepts raw or structured logs from Logstash.
  - Passes them to the backend GNN Model for context-aware analysis.
  - Collects enriched logs (with threat scores or alerts).
  - Returns them back to Logstash for indexing.

4. GNN Model (Graph Neural Network) Purpose: Perform intelligent threat detection using relationships between events.

- Working:
  - Converts logs into a graph structure (nodes = users, IPs, processes; edges = connections/events).
  - Learns complex behavior patterns over time.
  - Detects anomalies, suspicious clusters, or insider threats using graph-based inference.
  - Outputs enriched results with additional labels or risk scores[11].

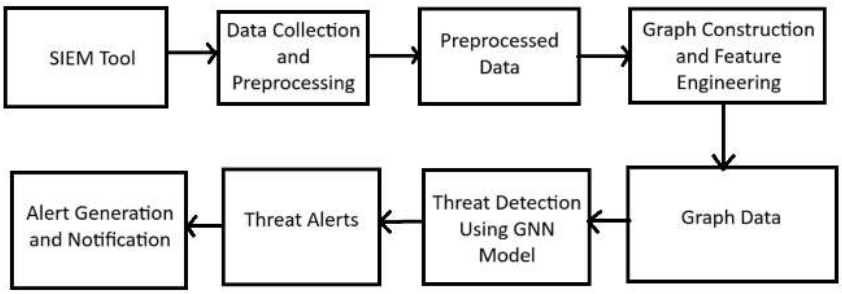
5. Elasticsearch (Log Indexing & Storage Engine) Purpose: Stores and indexes logs for fast searching and querying[13].

- Input: Receives enriched logs from Logstash (with GNN insights).
- Output: Supports real-time access for dashboards or queries.

6. Kibana (Dashboard & Visualization)

- Purpose: Provides a graphical interface to analyze logs and alerts.
- Features:
  - Visual dashboards for security monitoring.
  - Search, filter, and query enriched data.
  - Track anomalies or threats flagged by the GNN model.

VI. FLOW DIAGRAM:



1. SIEM Tool

Collects and stores security event logs from various sources (e.g., Windows logs, network devices).

2. Data Collection and Preprocessing

Extracts relevant log fields, cleans, normalizes, and converts them (e.g., JSON to CSV).

3. Pre-processed Data

Structured log data ready for transformation into graph format.

4. Graph Construction and Feature Engineering

Transforms tabular data into graph structures (nodes: users, IPs; edges: interactions) and extracts features.

5. Graph Data

Graph-formatted data used as input for the GNN model.

6. Threat Detection Using GNN Model

Applies a trained GNN to detect malicious patterns or anomalies in the graph.

7. Threat Alerts

Outputs classified alerts (e.g., benign or malicious) based on detected threats.

8. Alert Generation and Notification

Sends enriched alerts back to the SIEM or notifies analysts via dashboards or alerts.

VII. Conclusion:

In conclusion, integrating Graph Neural Networks (GNNs) with Security Information and Event Management (SIEM) tools is a significant step forward in cybersecurity. This project addresses the limitations of traditional SIEM systems in detecting complex and evolving threats. By leveraging GNNs' ability to analyze intricate relationships within security data, the project enhances threat detection accuracy, reduces false positives, and improves the overall efficiency of security operations. This advancement bridges the gap between cutting edge research and practical application, offering a robust solution to the increasing challenges in cybersecurity.

VIII. References :

[1] M. Thilagavathi , R. Saranyadevi , N. Vijayakumar, K. Selvi , L. Anitha, K. Sudharson: "AI-Driven Fraud Detection in Financial Transactions with Graph Neural Networks and Anomaly Detection". (2024)

[2] Vishva Gandhi, Tirthesh Gajjar: "Enhancing fraud detection in financial transactions through cyber security measures". (2024)

[3] Farhana Reza: "DDos-Net : Classifying DDoS attacks in wireless sensor networks with hybrid deep learning". (2024).

[4] Yenlik Begimbayeva, Oleksandr Gurko, Hanna Doroshenko, Serik Joldasbayev, Olena Fridman, Bakytzhan Kulambayev, Vitalina Baenko, Serhii Neronov: "Detection and Classification of Threats and Vulnerabilities on Hackers Forums Based on ML". (2024)

[5] Sudarshan Gaikar, Ajay Bichukale, Deep Barvekar: “Cyber Attack Detection with QR (Vol. 9, 4 April 2024)”.

[6] Dingari Janhavi , Mona A, Sandeep Pulata, Sasank Sami, Bharadwaj Vakamullu, Bharathi Mohan G: "Robust Hybrid Machine Learning Model for Financial Fraud Detection in Credit Card Transactions". (2023)

[7] Mangayarkarasi Ramaiah, C. Vanmathi, Mohammad Zubair Khan, M. Vanitha, M. Deepa: "An Efficient Intrusion Detection System to Combat Cyber Threats using Deep Neural Network Model (Vol.17,No.3, 2023)".

[8] Wasia Ashraf, Aamir Salaam Ahanger, Faheem Syeed Masoodi: "Enhancing Intrusion Detection using Supervised Machine Learning Algorithms".

[9] Ghazia Qaiser, Siva Chandrasekaran, Jinchuan Zheng, Rifai Chai: "A Hybrid ABNB Model for detecting malicious attacks for IIoS". (2023)

[10] Talla Yashwanth, K. Ashwini, Gandla Shiva Chaithanya, Arshiya Tabassum: “Network Intrusion Detection using Auto-encoder Neural Networks and MLP”. (2023)

[11] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner and G. Monfardini, "The Graph Neural Network Model," in *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61-80, Jan. 2009, doi: 10.1109/TNN.2008.2005605

[12] A. Vazão, L. Santos, M. B. Piedade and C. Rabadão, "SIEM Open Source Solutions: A Comparative Study," *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, Coimbra, Portugal, 2019, pp. 1-5, doi: 10.23919/CISTI.2019.8760980.

[13] P. Sankar, D. E. George and A. S. N. S, "Social media monitoring using ELK Stack," *2022 IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems (SPICES)*, THIRUVANANTHAPURAM, India, 2022, pp. 231-235, doi: 10.1109/SPICES52834.2022.9774273.

[14] Albesher, Luluh & Alfayez, Reem. (2024). An Observational Study on Flask Web Framework Questions on Stack Overflow (SO). IET Software. 2024. 10.1049/sfw2/1905538.

[15] Smyth, Patrick. (2018). Creating Web APIs with Python and Flask. The Programming Historian. 10.46430/phen0072.

[16] Chung, Wonyong & Moon, Jaeuk & Kim, Dongjun & Hwang, Eenjun. (2023). Graph Construction Method for GNN-Based Multivariate Time-Series Forecasting. Computers, Materials & Continua. 75. 5817-5836. 10.32604/cmc.2023.036830.