A PROJECT REPORT ON

NEXGEN ™ SIEM: MODERN SOC ARCHITECTURE FOR EVOLVING THREAT LANDSCAPE

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

(CYBERSECURITY)

BY

ANUGANTI THARUNADITYA (20P61A6206)

Under the esteemed guidance of

Mr. K. Ashok

Associate Professor, Dept. of CSE (CS)

Department of Computer Science and Engineering (Cybersecurity)

VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY

(Approved by AICTE, accredited by NBA, NAAC, Permanently Affiliated to JNTUH)

Aushapur (v), Ghatkesar (m), Medchal Dist, TELANGANA- 501 301

Academic Year 2023-2024

Disclaimer:

This project, titled '**NEXGEN ™ SIEM: MODERN SOC ARCHITECTURE FOR THE EVOLVING THREAT LANDSCAPE**,' including its contents, and code are protected under copyright laws (Copyright Act of 1957). No part of this project, including but not limited to its contents, code, and name, shall be copied, reproduced, or used for any purpose without explicit permission and proper credits given to the author. The word 'NexGen™' is a unique noun used in this project and is considered a trademark of ANUGANTI THARUNADITYA. It cannot be used or replicated in other projects without explicit permission. Any unauthorized use, copying, or posting of this project's content, including on social media, using the same name or any part of the title, is strictly prohibited and may result in legal action under the relevant intellectual property laws and regulations enforced by the Indian Copyrights Office."

Furthermore, certain portions of this project may be subject to the **MIT License**. Users are encouraged to review the specific license terms provided with such portions and comply with the terms and conditions outlined in the MIT License for usage.

If individuals wish to contribute to this project, they can refer to the contributions.md file for guidelines and instructions on contributing.

A PROJECT REPORT ON

NEXGEN SIEM: MODERN SOC ARCHITECTURE FOR EVOLVING THREAT LANDSCAPE

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

(CYBERSECURITY)

BY

ANUGANTI THARUNADITYA (20P61A6206)

Under the esteemed guidance of

Mr. K. Ashok

Associate Professor, Dept. of CSE (CS)

Department of Computer Science and Engineering (Cybersecurity)

VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY

(Approved by AICTE, accredited by NBA, NAAC, Permanently Affiliated to JNTUH)

Aushapur (v), Ghatkesar (m), Medchal Dist, TELANGANA- 501 301

Academic Year 2023-2024

VIGNANA BHARATHI Institute of Technology

AUTONOMOUS Under UGC

Accredited by NBA NATIONAL BOARD OF ACCREDITATION (ECE,EEE,CSE,IT,CE) NAAC NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL A Grade

AUSHAPUR (V), GHATKESAR (M), MEDCHAL. DIST-501 301

Department of Computer Science and Engineering

(Cybersecurity)

CERTIFICATE

This is to certify that the project entitled **"NEXGEN SIEM: MODERN SOC ARCHITECTURE FOR EVOLVING THREAT LANDSCAPE"** is being submitted by **ANUGANTI THARUNADITYA (20P61A6206)** in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in Computer Science and Engineering (Cybersecurity)** is a record of bonafide work carried out by them under my guidance and supervision during the academic year 2023-2024. The results embodied in this project report have not been submitted to any other University for the award of any degree or diploma.

Internal Guide
Mr. K. Ashok
Associate Professor
Department of CSE (CS)

Head of the Department
Dr. P. Sushma
Professor
Department of CSE (CS)

HOD, CSE(Cyber Security)
Vignana Bharathi Institute of Technology
Aushapur (V), Ghatkesar (M), Medchal (D).

Project Coordinator
Mr. K. Ashok
Associate Professor
Department of CSE (CS)

External Examiner

VIGNANA BHARATHI Institute of Technology
AUTONOMOUS Under UGC
Accredited by NBA NATIONAL BOARD of ACCREDITATION (ECE,EEE,CSE,IT,CE)
NAAC NATIONAL ASSESSMENT AND ACCREDITATION COUNCIL A Grade

AUSHAPUR (V), GHATKESAR (M), MEDCHAL .DIST-501 301

Department of Computer Science and Engineering (Cybersecurity)

DECLARATION

I, **ANUGANTI THARUNADITYA** bearing hall ticket number **20P61A6206,** hereby declare that the project report entitled **"NEXGEN SIEM: MODERN SOC ARCHITECTURE FOR EVOLVING THREAT LANDSCAPE**" under the guidance of **Mr. K. Ashok,** Asst. Professor, Department of Computer Science and Engineering (Cybersecurity), VBIT, Hyderabad, is submitted in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Department of Computer Science and Engineering (Cybersecurity)

This is a record of bonafide work carried out by us and the results embodied in this project have not beenreproduced or copied from any source. The results embodied in this project report have not been submitted to any other university or institute for the award of any other degree or diploma.

ANUGANTI THARUNADITYA (20P61A6206)

ACKNOWLEDGEMENT

First and foremost, we wish to express our gratitude towards the institution "**Vignana Bharathi Institute of Technology"** for fulfilling the most cherished goal of our life to do Bachelor of Technology.

It is great pleasure in expressing deep sense of gratitude to our **Internal guide, Mr. K. Ashok,** Assistant Professor, Department of Computer Science and Engineering (Cybersecurity) for her valuable guidance and freedom he gave us.

We also express our sincere thanks to **Mr. K. Ashok, Project Coordinator** for his encouragement and support throughout the project.

We are deeply grateful to **Dr. P. Sushma, Head of Department, Department of Computer Science and Engineering (Cybersecurity)** for granting us the opportunity to conduct this project.

We take immense pleasure in thanking **Dr. P.V.S. Srinivas Principal, Vignana Bharathi Institute of Technology**, **Ghatkesar** for having permitted us to carry out this project work.

Our utmost thanks also go to all the **FACULTY MEMBERS** and **NON-TEACHING STAFF** of the **Department of Computer Science and Engineering (Cybersecurity)** for their support throughout our project work

ANUGANTI THARUNADITYA (20P61A6206)

ABSTRACT

NexGen SIEM" is a project focused on enhancing Security Operations Centre (SOC) capabilities by integrating open-source technologies. This study presents advancements in modern SOC architectures, emphasizing improvements in agility, efficiency, and effectiveness for security analysts and SOC specialists. The project utilizes a range of open-source components to ensure accessibility and adaptability, facilitating seamless deployment and scalability across diverse organizational environments. Centralized data collection and normalization mechanisms are implemented to streamline security data monitoring processes, providing a consolidated view of the organization's security posture. Key features of "NexGen SIEM" include an automated incident management system, empowering SOC professionals to respond rapidly and effectively to cyber threats. Automated threat hunting and playbook creation contribute to proactive threat detection and mitigation, reducing response times and minimizing potential impact. Furthermore, "NexGen SIEM" promotes collaborative workflows and knowledge sharing among SOC teams, fostering a culture of collective intelligence and teamwork. Integration with open-source threat intelligence platforms enhances data feeds, strengthening the organization's defense against evolving cyber risks. This research underscores a commitment to continuous improvement and innovation within the dynamic Cybersecurity landscape. "NexGen SIEM" serves as a valuable contribution to SOC implementations, embodying a collaborative, adaptive, and resilient approach to addressing emerging challenges.

.

VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering (Cybersecurity)

COURSE OUTCOMES

Course: Major Project Class: IV B. Tech II

Semester AY: 2023-24

Course Outcomes

After completing the Projects, the student will be able to:

Code	Course Outcomes	Taxonomy
C424.1	Identify and state the problem precisely to prepare the abstract	Remember
C424.2	Analyze the existing system, and outlining the proposedmethodology for effective solution	Analyze
C424.3	Use various modern tools for designing applications based onspecified requirements	Apply
C424.4	Develop applications with adequate features and evaluate theapplication to ensure the quality	Create
C424.5	Prepare the document of the project as per the guidelines	Create

PROGRAM OUTCOMES (POs)

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals,and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. **Conduct investigations of complex problems**: Use research-based knowledge

and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern
engineering and IT tools including prediction and modeling to complex engineering activities withan understanding of the limitations.

6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms ofthe engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader indiverse teams, and in multidisciplinary settings.

10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage inindependent and life-long learning in the broadest context of technological change.

PROGRAM SPECIFIC OUTCOMES (PSO's)

PSO1 Simulate computer hardware and apply software engineering principles and techniques to develop various IT applications

PSO2 Analyze various networking concepts and also aware of how security policies, standards and practices are used for trouble-shooting.

PSO3 Design and maintain database for providing back-end support to software projects.

PSO4 Apply algorithms and programming paradigms to produce IT based solutions for thereal-world problems.

VIGNANA BHARATHI INSTITUTE OF TECHNOLOGY

Department of Computer Science and Engineering (Cybersecurity)

COs Mapping with PO/PSO

Project Title: NexGen SIEM: Modern SOC Architecture for evolving threat landscape.

Name of the Supervisor: Mr. K. ASHOK

Batch Details:

S.NO.	Regd. No.	Student Name	Technology
1	20P61A6206	A.THARUNADITYA	Security Information and Event Management
2	20P61A6214	CH. SNEHA	
3	20P61A6255	V. ADVAIT	

Note: Write your domain name in technology field (ex. ML, IOT, BC, Security, Cloud etc.)

CO-PO Mapping for Major Project:High -3 Medium -2 Low-1

PO / CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PS O1	PS O2	PS O3	PS O4
C424.1	3	3	2	2	-	2	-	3	3	3	3	3	2	3	3	2
C424.2	2	2	3	2	3	2	-	2	3	3	3	3	2	2	3	2
C424.3	2	2	3	2	3	2	-	2	3	3	3	3	2	2	3	3
C424.4	2	2	3	2	3	2	-	2	3	3	3	3	2	2	3	2
C424.5	2	2	2	2	3	2	-	2	3	3	3	3	2	2	3	3
AVG	2.2	2.2	2	2	2.4	2	-	2.2	3	3	3	3	2	2.2	3	2.2

INDEX

CONTENTS

List of Figures

List of Tables

CHAPTER 01

INTRODUCTION

1. INTRODUCTION

In today's ever-evolving digital landscape, Security Operations Centres (SOCs) serve as the cornerstone of organizational cybersecurity, tasked with defending against a myriad of sophisticated threats. This project represents a strategic initiative aimed at propelling SOC capabilities into the next generation through the seamless integration of cutting-edge open-source technologies.

The overarching objective is to drive substantial enhancements in SOC architectures, with a strong emphasis on agility, efficiency, and effectiveness for both security analysts and SOC specialists. Leveraging a meticulously curated suite of open-source components, this initiative facilitates smooth deployment and scalability across diverse organizational infrastructures.

A core aspect of this initiative is the implementation of centralized data collection and normalization mechanisms, which play a pivotal role in streamlining security data monitoring processes. By providing SOC teams with a consolidated and actionable view of security events and incidents, these mechanisms empower timely and informed decision-making.

Furthermore, the project introduces a suite of advanced features designed to automate key SOC workflows. These include an automated incident management system that expedites incident triage and resolution, proactive threat hunting capabilities for early threat detection, playbook creation tools to guide standardized response procedures, and collaborative workflows that foster knowledge sharing and collective intelligence among SOC professionals.

By fostering a culture of collaboration and continuous improvement within SOC teams, this initiative aims to maximize expertise and resilience against evolving cyber threats. Through ongoing innovation and refinement, it seeks to set new standards in SOC capabilities, serving as a catalyst for organizations seeking to navigate the complex cybersecurity landscape with confidence.

In summary, this project embodies a forward-thinking approach to SOC optimization, integrating advanced open-source technologies to fortify organizational defenses and ensure a more secure and resilient digital future.

1.1 MOTIVATION

The motivation behind the "NexGen SIEM" project stems from a deep understanding of the evolving cybersecurity landscape and the pressing need for SOC capabilities to adapt and excel in the face of increasingly sophisticated threats. Several key factors drive the urgency and significance of this initiative:

1. Rapidly Evolving Threat Landscape: Cyber threats continue to evolve in complexity and diversity, with malicious actors employing advanced tactics such as zero-day exploits, polymorphic malware, and AI-driven attacks. This dynamic threat landscape necessitates a proactive and adaptive approach to cybersecurity.

2. Increasing Volume and Velocity of Security Data: Organizations are inundated with vast amounts of security data from diverse sources, including network logs, endpoint telemetry, cloud environments, and threat intelligence feeds. Managing, correlating, and analyzing this data in real-time poses significant challenges for SOC teams.

3. Shortening Response Times: The window of opportunity to detect and respond to cyber threats is shrinking rapidly. SOC teams are under immense pressure to identify and mitigate threats swiftly to minimize the potential impact on organizational assets and operations.

4. Complexity of IT Environments: Modern IT infrastructures are characterized by hybrid cloud environments, IoT devices, remote workforces, and interconnected networks. Securing these complex environments requires a holistic and integrated approach that traditional SIEM solutions may struggle to provide.

5. Need for Automation and Orchestration: Manual SOC processes are labor-intensive, time-consuming, and prone to human error. Automating routine tasks, orchestrating response workflows, and leveraging AI-driven analytics are essential for enhancing SOC efficiency and effectiveness.

6. Continuous Improvement and Innovation: Cybersecurity is a constantly evolving domain, requiring SOC teams to stay ahead of emerging threats and adopt new technologies and best practices. Continuous improvement and innovation are essential for maintaining a robust security posture.

The "NexGen SIEM" project is motivated by these pressing challenges and opportunities. By harnessing the power of open-source technologies, automation, collaborative workflows, and advanced analytics, the project aims to address these key motivators and empower SOC teams to navigate the complexities of modern cybersecurity with confidence and resilience.

1.1.1 OVERVIEW OF EXISTING SYSTEM

The current state of Security Operations Centres (SOCs) and SIEM (Security Information and Event Management) systems reflects a landscape characterized by several key strengths and limitations:

1. Data Aggregation and Correlation: Existing SIEM systems excel in aggregating and correlating security data from diverse sources such as network devices, servers, applications, and endpoints. This capability provides SOC teams with a comprehensive view of security events and incidents across the organization.

2. Alert Prioritization and Incident Management: SIEM solutions offer alert prioritization mechanisms based on predefined rules and correlation logic. This helps SOC analysts focus on high-priority alerts and streamline incident management processes, improving response times and reducing false positives.

3. Compliance and Reporting: Many SIEM platforms include compliance management features that facilitate regulatory compliance assessments and reporting. These capabilities are crucial for organizations operating in regulated industries where adherence to compliance standards is mandatory.

4. Threat Intelligence Integration: SIEM systems often integrate with external threat intelligence feeds, enriching security data with contextual information about known threats, indicators of compromise (IOCs), and emerging attack patterns. This integration enhances threat detection and response capabilities.

5. Workflow Automation: Some advanced SIEM solutions offer workflow automation capabilities, allowing SOC teams to automate routine tasks, incident triage, and response actions. Automation reduces manual workload and improves SOC efficiency.

However, despite these strengths, traditional SIEM systems face several challenges and limitations:

1. Scalability and Performance: As the volume and velocity of security data continue to grow, traditional SIEM systems may struggle to scale effectively and maintain

optimal performance. This can lead to delays in data processing, alert fatigue, and missed detections.

2.Complexity and Customization: Configuring and customizing SIEM rules, correlations, and dashboards can be complex and time-consuming. SOC teams often require specialized expertise to fine-tune SIEM configurations according to their unique environment and security requirements.

3. Detection of Advanced Threats: Advanced and evasive threats, such as zero-day exploits, fileless malware, and insider threats, pose significant challenges for traditional SIEM solutions. Detecting and mitigating these threats require advanced analytics, behavioral monitoring, and threat hunting capabilities.

4. Response Orchestration: While SIEM systems provide alerting and incident management capabilities, orchestrating complex response workflows involving multiple security tools and processes may require manual intervention or integration with separate orchestration platforms.

5.Cost and Resource Constraints: Acquiring and maintaining enterprise-grade SIEM solutions can be costly in terms of licensing fees, infrastructure requirements, and ongoing maintenance. Resource constraints, including skilled SOC personnel and dedicated security budgets, can further limit the effectiveness of SIEM deployments.

In light of these observations, there is a clear opportunity and necessity for the "NexGen SIEM" project to address these existing system challenges and leverage emerging technologies and best practices to enhance SOC capabilities and resilience against evolving cyber threats.

1.1.2 OVERVIEW OF PROPOSED SYSTEM:

The proposed system aims to enhance Security Operations Center (SOC) capabilities through the integration of open-source technologies. Key features include:

1. Centralized Data Collection:
Utilizes Beats agents on user machines to collect logs and other security-related data, forwarding them to Log stash for processing.

2. Central Management with Fleet Server:
Implements Fleet Server for centralized management of Elastic Agents, allowing for policy application and configuration management.

3. Data Processing and Indexing:
Logstash processes data from Beats agents and other sources, preparing it for indexing into Elastic search for efficient storage and retrieval.

4. Real-time Monitoring and Analysis:
Elasticsearch enables real-time monitoring and analysis of indexed data, supporting rapid threat detection and response.

5. Visualization and Reporting:
Kibana provides data visualization tools, interactive dashboards, and reporting capabilities for SOC analysts and administrators.

6. Security Enhancement with Elastic Defend:
Integrates Elastic Defend with Kibana to enhance security posture, including threat detection, incident response, and security incident analysis.

7. Scalability and Flexibility:
Utilizes open-source components for scalability and adaptability, accommodating diverse data sources and evolving security requirements.

8. Collaborative Workflows:
Facilitates collaborative workflows within the SOC team, promoting collective intelligence, teamwork, and effective security operations.

Overall, the system aims to streamline security data monitoring, improve threat detection and response capabilities, and enhance overall SOC efficiency and effectiveness through advanced open-source technologies and integrations.

1.2 PROBLEM DIFINITION

The existing Security Operations Center (SOC) infrastructure faces challenges in efficiently collecting, processing, and analyzing security-related data from diverse sources. Manual processes, lack of centralized management, and limited scalability hinder the SOC's ability to detect and respond to threats effectively. Additionally, the absence of advanced security features and real-time monitoring capabilities contributes to delayed incident response and increased security risks.

Key Problems:
1. Fragmented Data Collection: Current methods of data collection from user machines and network devices lack centralized management and consistency, leading to fragmented data sources and incomplete visibility into security events.

2. Manual Configuration and Management: The manual configuration and management of security agents across distributed environments result in operational inefficiencies, configuration errors, and delayed response times to security incidents.

3. Limited Scalability: The existing SOC infrastructure struggles to scale with the growing volume and complexity of security data, leading to performance bottlenecks, resource constraints, and reduced effectiveness in threat detection and response.

4. Lack of Advanced Security Features: The absence of advanced security features such as threat intelligence integration, behavior analytics, and automated response mechanisms limits the SOC's ability to proactively identify and mitigate security threats.

5. Ineffective Monitoring and Analysis: The lack of real-time monitoring and comprehensive data analysis tools hinders the SOC's ability to detect and investigate security incidents promptly, resulting in prolonged exposure to potential threats and security breaches.

6. Compliance and Reporting Challenges: Manual compliance management and reporting processes are time-consuming, error-prone, and do not provide real-time insights into security posture, impacting regulatory compliance and risk management efforts.

Moreover, existing systems are not cost-efficient, as they require significant manual effort, maintenance, and resources, leading to higher operational costs and reduced ROI.

Pricing of SIEM Tools (approximate costs):
1. Splunk Enterprise Security: Starting from $2,500 per month.
2. IBM QRadar: Starting from $10,000 per year.
3. LogRhythm: Starting from $3,000 per month.
4. Elastic Security: Starting from $2,000 per month.

These pricing figures are approximate and can vary based on the specific features, deployment options, and licensing models offered by each SIEM vendor.

1.3 OBJECTIVE OF PROJECT

Our NexGen SIEM project serves a dual purpose: to enhance security operations within our organization and to provide a simulated Security Operations Center (SOC) environment for educational purposes. This initiative aims to offer students and institutions hands-on experience in cybersecurity and SOC operations, acting as a model to demonstrate how a SOC functions in a real-world scenario.

By setting up a SOC environment within our organization, we create a practical learning platform where students can gain valuable insights into security monitoring, incident response, threat detection, and mitigation strategies. This experiential learning opportunity allows students to apply theoretical knowledge to real-world scenarios, improving their skills and readiness for careers in cybersecurity.

Furthermore, our SOC environment serves as a model that showcases best practices, workflows, and technologies commonly used in SOC operations. This model demonstrates how various components, such as data collection agents, central management systems, data processing pipelines, analytics tools, and security enhancements, work together seamlessly to safeguard against cyber threats and protect organizational assets.

Through this project, we aim to bridge the gap between academic learning and practical cybersecurity skills, empower students and institutions with hands-on experience, and contribute to the development of skilled cybersecurity professionals equipped to address evolving cyber threats effectively.

1.4 SCOPE OF PROJECT

Our NexGen SIEM project is intricately aligned with the principles outlined in the NIST Cybersecurity Framework, fostering a holistic approach to cybersecurity management and operations. At its core, our project aims to identify, protect, detect, respond, and recover from cyber threats by leveraging advanced technologies and best practices.

In terms of identification, we meticulously categorize and analyze components within our NexGen SIEM infrastructure, such as Beats agents, Logstash, Elasticsearch, Kibana, Fleet Server, and potentially Elastic Defend. This thorough understanding allows us to assess vulnerabilities and risks effectively, aligning with NIST's emphasis on asset management and risk assessment.

Protection is a paramount aspect of our project, achieved through centralized management via Fleet Server. This centralized approach enables us to apply consistent security controls, policies, and configurations across our infrastructure, bolstering our defense against cyber threats. Integration of advanced security features, including threat intelligence and automated response mechanisms, further fortifies our protection measures.

Our project excels in detection capabilities, utilizing real-time monitoring and analysis tools like Elasticsearch and Kibana. These tools enable us to detect anomalies, suspicious activities, and security incidents promptly, aligning seamlessly with NIST's continuous monitoring and detection principles.

In terms of response, our NexGen SIEM project incorporates automated response mechanisms, allowing for swift and effective responses to identified security threats and vulnerabilities. An incident response plan is also developed, outlining procedures,

roles, responsibilities, and communication protocols, in line with NIST's incident response and management guidelines.

Furthermore, our project encompasses recovery measures, including backup and recovery processes for critical components of our infrastructure and simulated SOC environment. This ensures business continuity and resilience, supporting NIST's objectives related to recovery and resilience.

Additionally, our project's scope extends to creating a simulated SOC environment for educational purposes. This initiative provides hands-on training and experience in cybersecurity operations, contributing to cybersecurity workforce development and training, a key focus area of the NIST Cybersecurity Framework.

Overall, our NexGen SIEM project's comprehensive approach and alignment with NIST principles underscore our commitment to proactive cybersecurity management, rapid incident response, and continuous improvement in cybersecurity operations.

CHAPTER 02
LITERATURE SURVEY

2. LITERATURE SURVEY

Title of the paper: A SURVEY ON: "LOG ANALYSIS WITH ELK STACK TOOL"
Authors: Pranitha P. Bavaskar
University: Sandip University
Journal and year: IJRAR 2019

This paper thoroughly explores log analysis using the ELK Stack tool, integrating Elasticsearch, Logstash, and Kibana. It delves into the vital role log analysis plays in contemporary cybersecurity strategies, aiding in the detection and mitigation of various forms of malicious activities. The ELK Stack's strength lies in its adeptness at efficiently handling log data, creating essential indexes for data retrieval, seamlessly collecting logs from diverse sources via Logstash, and providing insightful data visualizations through Kibana. The survey meticulously examines the functionalities of each ELK component and their synergistic impact on diverse log analysis scenarios. It underscores the paramount importance of log analysis across different domains, including pinpointing and addressing malicious events, fortifying network security, identifying anomalies in system behaviour, and pre-emptively thwarting cybercrimes. By showcasing the ELK Stack's robust capabilities in processing, analysing, and presenting log data, the paper strongly advocates for its widespread adoption to fortify cybersecurity measures. Organizations stand to gain invaluable insights from leveraging the ELK Stack, enabling them to detect, analyse, and respond to potential security threats with enhanced efficacy and agility.

Title of the paper: Toward a Secure ELK Stack
Authors: Fatimetou Abdou VADHIL 1, Mohamed Lemine SALIHI 2, Mohamedade Farouk NANNE 3
University: Université de Nouakchott Alassriya. Nouakchott, Mauritanie
Journal and year: IJCSIS 2019

The paper "Toward a Secure ELK Stack" by Fatimetou Abdou VADHIL, Mohamed Lemine SALIHI, and Mohamedade Farouk NANNE from Université de Nouakchott Al-Aasriya discusses the evolving landscape of Big Data technology and the increasing complexity of securing data processing platforms like the ELK Stack (Elasticsearch, Logstash, Kibana). It highlights the importance of integrating security solutions into such platforms due to the growing concern for data privacy and protection. The authors emphasize the need for research and development in securing ELK Stack components, especially considering that security features represent a small fraction of current research efforts. They propose exploring alternative security solutions such as integrating ELK with a SIEM (Security Information and

Event Management) system, leveraging features like Security, Machine Learning, and Alerting from X-Pack to enhance the overall security posture. The paper encourages further exploration of combining ELK Stack capabilities with SIEM functionalities to create a robust and cost-effective security solution for organizations dealing with large-scale data analysis and visualization.

CHAPTER 03

SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 SYSTEM ARCHITECTURE:

3.1.1 ARCHITECTURE DIAGRAM

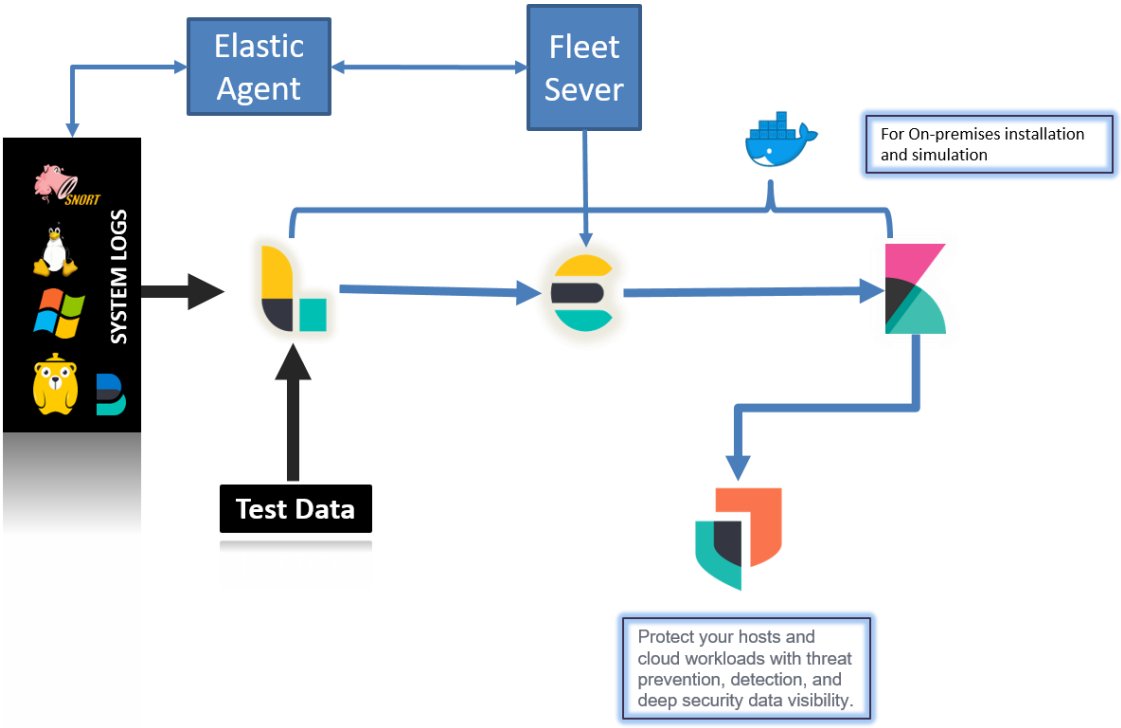

Fig 3.1.1 Architecture Diagram

The architecture diagram of your NexGen SIEM project illustrates a cohesive system where data flows seamlessly from collection to analysis and security enhancement. It starts with Beats Agents deployed on user machines, responsible for gathering various data types like logs and metrics. This data is then passed to Logstash, acting as a data processing pipeline, where it undergoes transformations and filters before being sent to Elasticsearch. Elasticsearch serves as the centralized storage and retrieval system, indexing the processed data for efficient search and analysis. Kibana, connected to Elasticsearch, provides a visual interface for data exploration, dashboards, and analytics, empowering SOC analysts with real-time insights and monitoring capabilities.Fleet Server plays a crucial role in central management, ensuring consistent policies and configurations across Elastic Agents (formerly Beats agents) deployed in

distributed environments. Optionally, Elastic Defend can be integrated, enhancing security with advanced features like threat intelligence, behaviour analytics, automated responses, and comprehensive incident analysis. This architecture promotes a robust security posture by facilitating streamlined data management, real-time monitoring, and advanced threat detection and response mechanisms within your NexGen SIEM environment.

3.1.2 VS Code

Visual Studio Code (VS Code) is a free, open-source code editor developed by Microsoft. It's designed to be lightweight, fast, and highly customizable, making it suitable for a wide range of programming tasks. VS Code supports various programming languages and frameworks through its extensive ecosystem of extensions, allowing developers to customize their codingenvironment according to their preferences and project requirements. Some key features of VS Code include built-in Git integration for version control, IntelliSense for code completion and intelligent suggestions, debugging capabilities, and a rich set of keyboard shortcuts for efficient coding. It's available on multiple platforms, including Windows, macOS, and Linux, making it a popular choice among developers across different operating systems. Overall, VS Code provides a modern and versatile coding experience with a focus on productivity and ease of use.

3.1.3 Logstash

Logstash is a crucial component of your NexGen SIEM project, serving as an open-source data processing pipeline. It acts as a connector between data sources, such as Beats agents, and Elasticsearch, the central storage system. Logstash excels in real-time data collection, processing, and transformation, handling various data types like logs and metrics. It applies customizable transformations, filters, and enrichments to the data, ensuring its accuracy and relevance. Logstash's versatility is evident in its support for a wide range of input, output, and filter plugins, enabling seamless integration with diverse data sources and destinations. This flexibility allows for efficient handling of complex data processing tasks, ensuring scalability and reliability. Logstash promotes data integrity through fault tolerance mechanisms, buffering, and parallel processing

capabilities. It plays a vital role in enhancing data analysis, visualization, and security operations within your organization's NexGen SIEM infrastructure.

3.1.4 Elastic Search

Elasticsearch is the core component of your NexGen SIEM infrastructure, serving as a distributed, RESTful search and analytics engine. It acts as the centralized storage and retrieval system for the vast amount of data collected and processed by Logstash and other components. Elasticsearch excels in indexing and searching structured and unstructured data in real-time, enabling fast and efficient data retrieval and analysis. Its scalability and performance make it ideal for handling large volumes of data, supporting rapid search and response capabilities. One of Elasticsearch's key strengths is its ability to handle complex queries, aggregations, and analytics, allowing for deep insights into your security-related data. It supports various search features, including full-text search, geo-search, and fuzzy search, enhancing the flexibility and accuracy of data retrieval. Elasticsearch's distributed nature ensures high availability, fault tolerance, and resilience, critical for maintaining uninterrupted operations and data integrity. Moreover, Elasticsearch integrates seamlessly with Kibana, providing a powerful combination for data visualization, exploration, and dashboarding. This integration enhances SOC operations by enabling interactive dashboards, visualizations, and analytics that facilitate real-time monitoring, trend analysis, and anomaly detection. Overall, Elasticsearch is a fundamental component that drives efficient data storage, retrieval, and analysis within your NexGen SIEM environment, empowering your organization to make data-driven security decisions and respond effectively to cyber threats.

3.1.5 Kibana

Kibana is a powerful data visualization and exploration tool that complements your NexGen SIEM project by providing intuitive interfaces and actionable insights. It connects seamlessly with Elasticsearch, enabling interactive dashboards, visualizations, and analytics for SOC analysts and administrators. Kibana's user-friendly interface allows for easy exploration and interpretation of data, empowering users to identify security trends, anomalies, and potential threats effectively. Its customizable

dashboards and visualizations enhance data visibility and understanding, facilitating informed decision-making and proactive threat detection. Kibana supports real-time monitoring, alerting, and reporting, enabling SOC teams to respond swiftly to security incidents and mitigate risks promptly. Additionally, Kibana's integration with Elasticsearch allows for advanced search capabilities, data filtering, and drill-down analysis, further enhancing data exploration and forensic investigations. Overall, Kibana plays a crucial role in enhancing the user experience, streamlining data visualization and analysis, and improving overall security posture within your NexGen SIEM environment.

3.1.6 Elastic Security

Elastic Security, also known as Elastic Defend, is an advanced security solution that enhances your NexGen SIEM project with comprehensive threat detection, response, and protection capabilities. It integrates seamlessly with Elasticsearch and Kibana, leveraging their robust search and analytics engine for enhanced security operations. One of Elastic Security's key features is threat intelligence integration, which allows you to incorporate threat feeds, indicators of compromise (IOCs), and security intelligence into your security monitoring and analysis processes. This enables proactive threat detection by correlating real-time data with known threat patterns and indicators. Behavior analytics is another critical aspect of Elastic Security, leveraging machine learning algorithms to detect anomalous behavior and suspicious activities within your IT environment. This helps in identifying potential security incidents and threats early, enabling rapid response and mitigation actions. Automated response mechanisms are a notable capability of Elastic Security, allowing you to automate response actions based on predefined security policies and rules.

This includes automated threat containment, quarantine, and remediation, reducing response times and minimizing the impact of security incidents. Furthermore, Elastic Security provides comprehensive security incident analysis tools, including advanced visualizations, forensic analysis capabilities, and timeline views. These tools facilitate in-depth investigation and root cause analysis of security incidents, aiding in understanding the full scope of an attack and improving incident response strategies.Overall, Elastic Security enhances your NexGen SIEM environment with

advanced threat detection, behavior analytics, automated response, and comprehensive security incident analysis capabilities, strengthening your organization's cybersecurity posture and resilience against evolving cyber threats.

3.1.7 Docker

Docker is a leading containerization platform that revolutionizes software development and deployment by encapsulating applications and their dependencies into lightweight, portable containers. Its key features include:

1. Containerization: Docker enables the creation and management of containers, which are isolated environments containing everything an application needs to run, including code, runtime, system tools, libraries, and settings.

2. Portability: Containers created with Docker are highly portable and can run consistently across different environments, from developer laptops to production servers, eliminating the "works on my machine" problem and streamlining deployment workflows.

3. Efficiency: Docker containers share the host operating system's kernel, making them lightweight and efficient in terms of resource utilization. Multiple containers can run on a single host without significant overhead.

4. Isolation: Containers offer a high degree of isolation, ensuring that applications and their dependencies are encapsulated and do not interfere with each other or the underlying host system.

5. Versatility: Docker supports a vast ecosystem of tools, services, and containerized applications, making it a versatile platform for micro services architecture, DevOps practices, continuous integration/continuous deployment (CI/CD), and cloud-native development.

Overall, Docker revolutionizes software development and deployment practices by providing a flexible, efficient, and portable containerization solution that accelerates application delivery and enhances scalability, reliability, and consistency across diverse IT environments.

Key Features of ELK stack include:

1. Elasticsearch:

Distributed Search and Analytics: Enables fast and efficient search and analysis of structured and unstructured data in real-time across distributed environments.

Scalability and Performance: Supports horizontal scaling to handle large volumes of data, ensuring high performance and responsiveness.

Advanced Query Capabilities: Offers powerful query and aggregation capabilities for complex data retrieval, including full-text search, geo-search, and fuzzy search.

High Availability and Resilience: Provides built-in fault tolerance, replication, and clustering features to ensure continuous availability and data integrity.

2. Logstash:

Data Collection and Processing: Collects data from multiple sources simultaneously, processes it in real-time, and applies transformations, filters, and enrichments.

Versatile Integration: Integrates seamlessly with various data sources, including logs, metrics, databases, and cloud platforms, using input, output, and filter plugins.

Scalability and Reliability: Supports parallel processing, buffering, and fault tolerance mechanisms for handling high-volume data processing tasks efficiently.

3. Kibana:

Data Visualization and Exploration: Provides interactive dashboards, visualizations, and analytics for exploring and interpreting data effectively.

Customizable Dashboards: Allows customization of dashboards and visualizations to suit specific use cases and data analysis requirements.

Real-time Monitoring and Alerting: Supports real-time monitoring, alerting, and reporting functionalities for proactive threat detection and response.

Integration with Elastic search: Seamlessly integrates with Elastic search, enabling powerful data search, filtering, and drill-down analysis capabilities.

Overall, the ELK stack offers a comprehensive and integrated solution for data collection, processing, storage, retrieval, visualization, and analysis, making it a versatile platform for log management, security monitoring, and operational insights across diverse IT environment.

3.2 OPERATING REQUIREMENTS

3.2.1 HARDWARE REQUIREMENTS

- Processor: Intel Core i5 or higher
- RAM: Minimum 16GB
- Storage: SSD with at least 500GB capacity
- Network: Gigabit Ethernet connectivity
- Monitor: Full HD (1920x1080) resolution

3.2.2 SOFTWARE REQUIREMENTS

- Operating System: Linux (Ubuntu 20.04 LTS recommended)
- Elasticsearch: Version 7.x or later
- Logstash: Latest stable version
- Kibana: Latest version compatible with Elasticsearch
- Fleet Server: Latest version for centralized management
- Beats Agents: Latest versions for data collection
- Docker: Latest version for containerization
- Python: Version 3.6+ for scripting and automation
- Security Tools: IDS/IPS, antivirus software
- Web Browser: Chrome, Firefox for Kibana access

CHAPTER 04

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard ismanaged, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of two major components: a Meta- model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

GOALS

1. The Primary goals in the design of the UML are as follows:
2. Provide users a ready-to-use, expressive visual modelling Language so thatthey can develop and exchange meaningful models.
3. Provide extendibility and specialization mechanisms to extend the core concepts.
4. Be independent of particular programming languages and development process.
5. Provide a formal basis for understanding the modelling language.
6. Encourage the growth of OO tools market.
7. Support higher level development concepts such as collaborations, frameworks, patterns and components

TYPES OF UML DIAGRAM

Each UML diagram is designed to let developers and customers view a software system from a different perspective and in varying degrees of abstraction. UML diagrams commonly created in visual modelling tools include:

1. Class Diagram: Shows the static structure of a system by depicting classes, attributes, operations, and relationships between classes.

2. Object Diagram: Represents instances of classes and their relationships at a specific point in time, providing a snapshot of the system.

3. Use Case Diagram: Illustrates the interactions between actors (users or external systems) and the system to capture functional requirements.

4. Sequence Diagram: Describes how objects interact in a particular scenario, showing the sequence of messages exchanged between objects.

5. Collaboration Diagram (Communication Diagram): Similar to a sequence diagram but focuses on the interactions between objects and their links.

6. Activity Diagram: Represents the flow of control or activities within a system, showing actions, decisions, and transitions between states.

7. State Diagram: Depicts the lifecycle of an object, specifying different states an object can be in and the events that trigger state transitions.

8. Component Diagram: Illustrates the components of a system and their dependencies, showing how the system is structured at a high level.

9. Deployment Diagram: Shows the physical deployment of software components on hardware nodes, including servers, devices, and networks.

10. Package Diagram: Organizes and shows dependencies between packages in a system, providing a high-level view of the system's structure.

These diagrams are used in software development to model, visualize, and document various aspects of a system, aiding in communication and design.

A. USE CASE DIAGRAM

The use case diagram for the NexGen SIEM project illustrates a comprehensive set of functionalities and interactions crucial for effective security operations. It includes actors such as "User Machines," representing devices with data collection agents, "Network Equipment," essential for data transmission, and the "Security Team," responsible for system management and security tasks.

Key use cases encompass various aspects of security management:

Login and Logout enables secure access to the system for the Security Team. Change Password facilitates password management for enhanced security. Manage Users allows the Security Team to administer user accounts efficiently. Query Logs (KQL) empowers the team to perform advanced log queries using the Kibana Query Language. Generate Threat Rules enables creation and management of threat detection rules. Generate Incident Guide assists in developing response protocols for security incidents. Raise Alerts triggers notifications based on predefined rules for immediate action. Monitor Threats provides real-time monitoring of security threats and anomalies. Collaborate with Platforms facilitates communication and collaboration with external platforms. Forward Logs to ELK stack ensures seamless data flow to the ELK stack for centralized storage and analysis. Visualize Logs and Metrics offers graphical visualization of logs and metrics for better insights. Kibana Dashboard provides a comprehensive dashboard interface for monitoring and analysis purposes.

These use cases cover a wide range of functionalities essential for security monitoring, threat detection, incident response, collaboration, and data visualization within the NexGen SIEM environment.

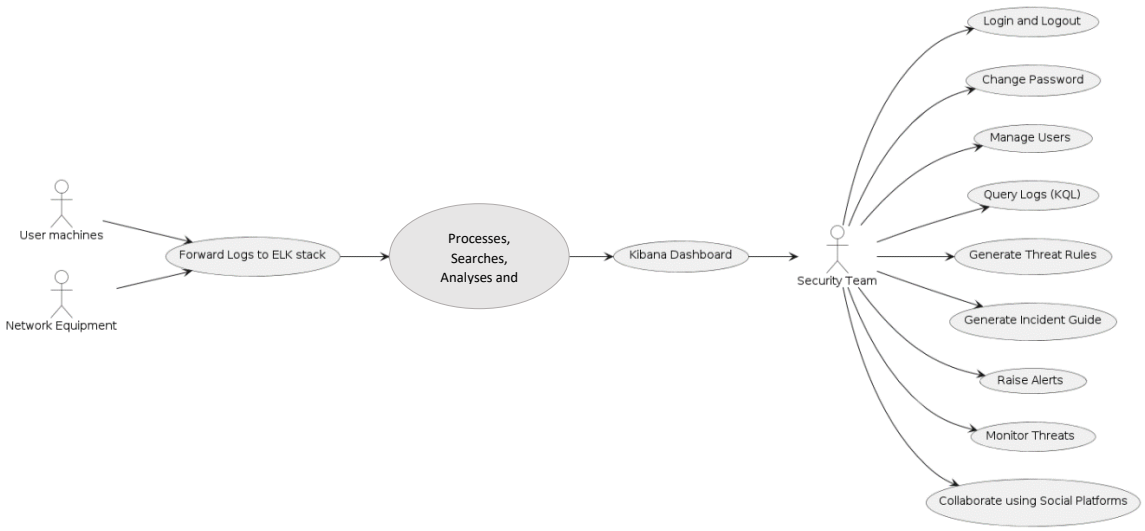

Fig 4.1 Use Case Diagram

B. CLASS DIAGRAM

The class diagram provided encapsulates the software architecture of the NexGen SIEM project, designed to enhance security management within organizational networks. The User class represents individuals interacting with the system, storing attributes like id, username, password, email, role, ipAddress, and osInfo. This class also includes methods for login, logout, changePassword, and updateProfile, crucial for user authentication and management. The Role class defines different user roles within the system, ensuring proper access control and privilege management. BeatsAgent facilitates log collection and forwarding to ELKStack, specifying destinations like hostKibana, hostElasticsearch, and hostLogstash. ELKStack, the core component, processes, indexes, and visualizes logs via Kibana. SecurityTeam, interacting with Kibana, handles security-related tasks such as login, logout, changePassword, manageUsers, queryLogs, generateThreatRules, generateIncidentGuide, raiseAlerts, monitorThreats, and collaboratePlatforms. The Log class captures log details crucial for analysis and auditing. This comprehensive architecture integrates essential functionalities for effective security monitoring, threat detection, incident response, and collaboration within a Security Operations Center (SOC) environment.

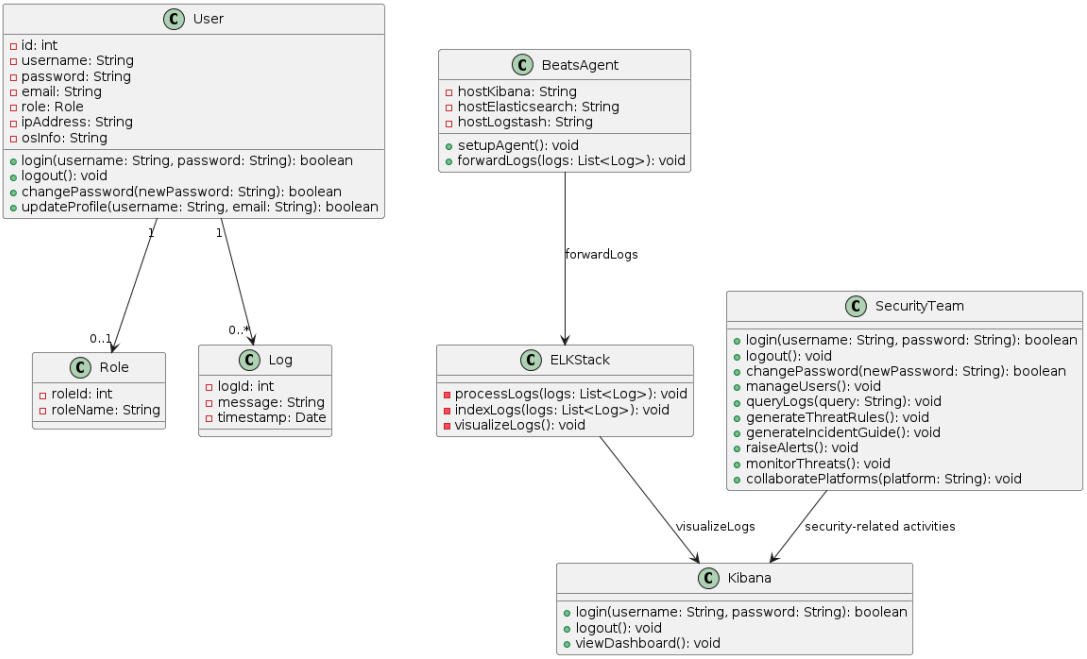

Fig 4.2 Class diagram

C. ACTIVITY DIAGRAM

The activity diagram illustrates the operational flow within the NexGen SIEM project. It begins with Employee machines and Network equipment generating logs independently. These logs are then forwarded to the ELK stack by both Beats Agent and Elastic Agents separately, indicating parallel processes.

Within the ELK stack, logs undergo crucial phases such as processing, indexing, and visualization, forming the core data handling and analysis stages. Subsequently, the Security teams access the ELK stack to monitor logs and detect potential threats effectively.

Upon detecting threats, the Security teams proceed to perform necessary security actions, which may include incident response, alerting, or collaboration for threat mitigation. This process encapsulates the holistic approach of log management, analysis, and security enforcement within the NexGen SIEM project, ensuring robust threat detection and response capabilities.

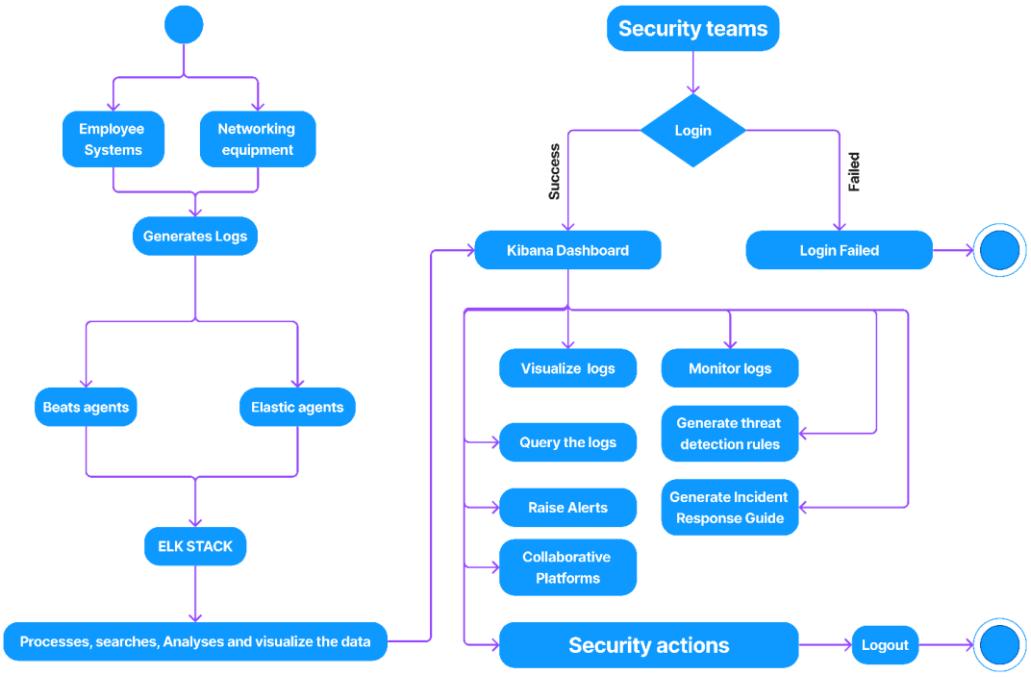

Fig 4.3 Activity diagram

D. SEQUENCE DIAGRAM

The sequence diagram depicts the flow of actions in the NexGen SIEM project. UserMachines initiate the process by sending logs to BeatsAgent, which then forwards them to Logstash for further handling. Logstash transfers the logs to Elasticsearch for storage and analysis purposes. Concurrently, UserMachines also send logs and information to ElasticAgent, which plays a role in data forwarding. The diagram shows the establishment of connections between Logstash and Elasticsearch, as well as bidirectional communication between FleetServer and ElasticAgent. FleetServer applies agent policies on ElasticAgent as part of the management process. Data is then transferred from FleetServer to Elasticsearch, while ElasticAgent directly sends data to Elasticsearch.

The final stages involve Elasticsearch connecting to Kibana for data visualization and analysis. Kibana integrates with ElasticDefend, enhancing the security features of the system. Overall, this sequence diagram illustrates the systematic flow of log processing, storage, visualization, and security integration in the NexGen SIEM project.

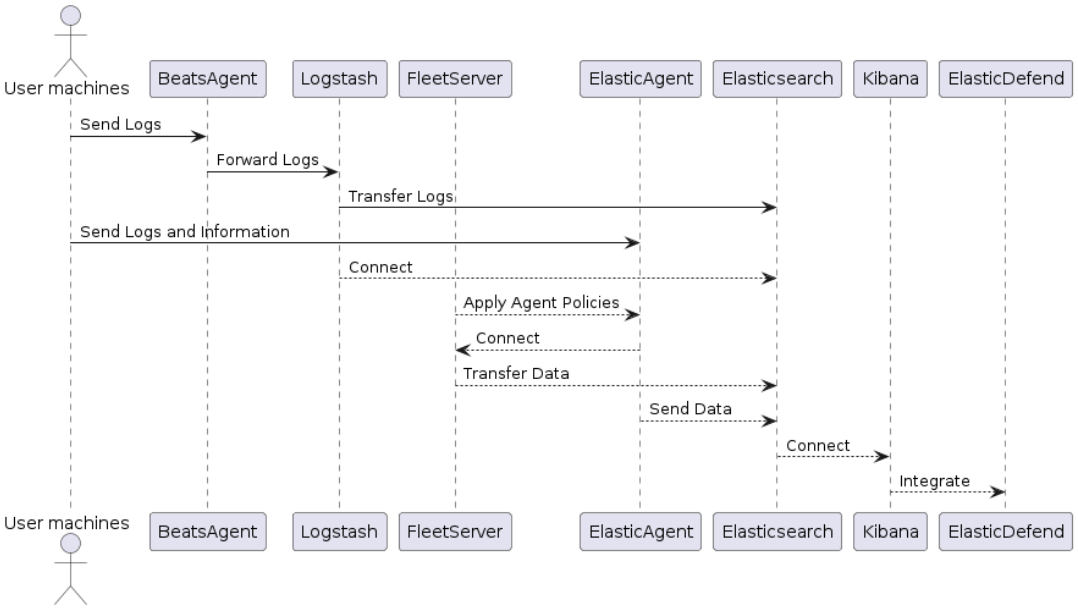

Fig 4.4 Sequence diagram

CHAPTER 05

IMPLEMENTATION

5. IMPLEMENTATION

5.1 CONFIGURATION FILES

5.1.1 docker-compose.yml

```
version: '3.7'

services:

  # The 'setup' service runs a one-off script which initializes user's inside
  # Elasticsearch — such as 'logstash_internal' and 'kibana_system' — with the
  # values of the passwords defined in the '.env' file. It also creates the
  # roles required by some of these users.
  #
  # This task only needs to be performed once, during the *initial* startup of
  # the stack. Any subsequent run will reset the passwords of existing users to
  # the values defined inside the '.env' file, and the built-in roles to their
  # default permissions.
  #
  # By default, it is excluded from the services started by 'docker compose up'
  # due to the non-default profile it belongs to. To run it, either provide the
  # '--profile=setup' CLI flag to Compose commands, or "up" the service by name
  # such as 'docker compose up setup'.
  setup:
    profiles:
      - setup
    build:
      context: setup/
      args:
        ELASTIC_VERSION: ${ELASTIC_VERSION}
    init: true
    volumes:
      - ./setup/entrypoint.sh:/entrypoint.sh:ro,Z
      - ./setup/lib.sh:/lib.sh:ro,Z
      - ./setup/roles:/roles:ro,Z
    environment:
      ELASTIC_PASSWORD: ${ELASTIC_PASSWORD:-}
      LOGSTASH_INTERNAL_PASSWORD: ${LOGSTASH_INTERNAL_PASSWORD:-}
      KIBANA_SYSTEM_PASSWORD: ${KIBANA_SYSTEM_PASSWORD:-}
      METRICBEAT_INTERNAL_PASSWORD: ${METRICBEAT_INTERNAL_PASSWORD:-
}
      FILEBEAT_INTERNAL_PASSWORD: ${FILEBEAT_INTERNAL_PASSWORD:-}
      HEARTBEAT_INTERNAL_PASSWORD: ${HEARTBEAT_INTERNAL_PASSWORD:-}
```

```
    MONITORING_INTERNAL_PASSWORD: ${MONITORING_INTERNAL_PASSWORD:-
}
    BEATS_SYSTEM_PASSWORD: ${BEATS_SYSTEM_PASSWORD:-}
  networks:
    - elk
  depends_on:
    - elasticsearch

 elasticsearch:
  build:
    context: elasticsearch/
    args:
      ELASTIC_VERSION: ${ELASTIC_VERSION}
  volumes:
    - ./elasticsearch/config/elasticsearch.yml:/usr/share/elasticsearch/config/elasticsearch.yml:ro,Z
    - elasticsearch:/usr/share/elasticsearch/data:Z
  ports:
    - 9200:9200
    - 9300:9300
  environment:
    node.name: elasticsearch
    ES_JAVA_OPTS: -Xms512m -Xmx512m
    # Bootstrap password.
    # Used to initialize the keystore during the initial startup of
    # Elasticsearch. Ignored on subsequent runs.
    ELASTIC_PASSWORD: ${ELASTIC_PASSWORD:-}
    # Use single node discovery in order to disable production mode and avoid bootstrap checks.
    # see: https://www.elastic.co/guide/en/elasticsearch/reference/current/bootstrap-checks.html
    discovery.type: single-node
  networks:
    - elk
  restart: unless-stopped

 logstash:
  build:
    context: logstash/
    args:
      ELASTIC_VERSION: ${ELASTIC_VERSION}
  volumes:
    - ./logstash/config/logstash.yml:/usr/share/logstash/config/logstash.yml:ro,Z
    - ./logstash/pipeline:/usr/share/logstash/pipeline:ro,Z
  ports:
    - 5044:5044
    - 50000:50000/tcp
    - 50000:50000/udp
    - 9600:9600
  environment:
    LS_JAVA_OPTS: -Xms256m -Xmx256m
```

```
    LOGSTASH_INTERNAL_PASSWORD: ${LOGSTASH_INTERNAL_PASSWORD:-}
  networks:
    - elk
  depends_on:
    - elasticsearch
  restart: unless-stopped

 kibana:
  build:
    context: kibana/
    args:
      ELASTIC_VERSION: ${ELASTIC_VERSION}
  volumes:
    - ./kibana/config/kibana.yml:/usr/share/kibana/config/kibana.yml:ro,Z
  ports:
    - 5601:5601
  environment:
    KIBANA_SYSTEM_PASSWORD: ${KIBANA_SYSTEM_PASSWORD:-}
  networks:
    - elk
  depends_on:
    - elasticsearch
  restart: unless-stopped

networks:
 elk:
  driver: bridge

volumes:
 elasticsearch:
```

5.1.2 Kibana.yml

```
---
## Default Kibana configuration from Kibana base image.
##
https://github.com/elastic/kibana/blob/main/src/dev/build/tasks/os_packages/docker_generator/templates/kibana_yml.template.ts
#
server.name: kibana
server.host: 0.0.0.0
elasticsearch.hosts: [ http://elasticsearch:9200 ]
```

```
monitoring.ui.container.elasticsearch.enabled: true
monitoring.ui.container.logstash.enabled: true

## X-Pack security credentials
#
elasticsearch.username: kibana_system
elasticsearch.password: ${KIBANA_SYSTEM_PASSWORD}

## Encryption keys (optional but highly recommended)
##
## Generate with either
##  $ docker container run --rm docker.elastic.co/kibana/kibana:8.6.2 bin/kibana-encryption-keys generate
##  $ openssl rand -hex 32
##
## https://www.elastic.co/guide/en/kibana/current/using-kibana-with-security.html
## https://www.elastic.co/guide/en/kibana/current/kibana-encryption-keys.html
#
#xpack.security.encryptionKey:
#xpack.encryptedSavedObjects.encryptionKey:
#xpack.reporting.encryptionKey:

## Fleet
## https://www.elastic.co/guide/en/kibana/current/fleet-settings-kb.html
#
xpack.fleet.agents.fleet_server.hosts: [ http://fleet-server:8220 ]

xpack.fleet.outputs:
  - id: fleet-default-output
    name: default
    type: elasticsearch
    hosts: [ http://elasticsearch:9200 ]
    is_default: true
    is_default_monitoring: true

xpack.fleet.packages:
  - name: fleet_server
    version: latest
  - name: system
    version: latest
  - name: elastic_agent
    version: latest
  - name: docker
    version: latest
  - name: apm
    version: latest

xpack.fleet.agentPolicies:
```

```
- name: Fleet Server Policy
  id: fleet-server-policy
  description: Static agent policy for Fleet Server
  monitoring_enabled:
    - logs
    - metrics
  package_policies:
    - name: fleet_server-1
      package:
        name: fleet_server
    - name: system-1
      package:
        name: system
    - name: elastic_agent-1
      package:
        name: elastic_agent
    - name: docker-1
      package:
        name: docker
- name: Agent Policy APM Server
  id: agent-policy-apm-server
  description: Static agent policy for the APM Server integration
  monitoring_enabled:
    - logs
    - metrics
  package_policies:
    - name: system-1
      package:
        name: system
    - name: elastic_agent-1
      package:
        name: elastic_agent
    - name: apm-1
      package:
        name: apm
      # See the APM package manifest for a list of possible inputs.
      # https://github.com/elastic/apm-server/blob/v8.5.0/apmpackage/apm/manifest.yml#L41-
L168
      inputs:
        - type: apm
          vars:
            - name: host
              value: 0.0.0.0:8200
            - name: url
              value: http://apm-server:8200
```

5.1.3 Elasticsearch

```
---
## Default Elasticsearch configuration from Elasticsearch base image.
##
https://github.com/elastic/elasticsearch/blob/main/distribution/docker/src/docker/config/elasticsearch.yml
#
cluster.name: docker-cluster
network.host: 0.0.0.0

## X-Pack settings
## see https://www.elastic.co/guide/en/elasticsearch/reference/current/security-settings.html
#
xpack.license.self_generated.type: trial
xpack.security.enabled: true
```

5.1.4 Logstash

```
---
## Default Logstash configuration from Logstash base image.
## https://github.com/elastic/logstash/blob/main/docker/data/logstash/config/logstash-full.yml
#
http.host: 0.0.0.0

node.name: logstash
```

5.1.5 DOCKER INSTALLATION

Step 1: Install Dependency packages

Start the installation by ensuring that all the packages used by docker as dependencies are installed.

```
sudo apt update && sudo apt -y full-upgrade
sudo apt install curl gnupg2 apt-transport-https software-properties-common ca-certificates
```

Check if a reboot is required after the upgrade:

```
[ -f /var/run/reboot-required ] && sudo reboot -f
```

Step 2: Import Docker GPG key

Import Docker GPG key used for signing Docker packages:

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/docker-archive-keyring.gpg
```

Step 3: Add the Docker repository to Kali Linux

Add Docker repository which contain the latest stable releases of Docker CE.

```
echo "deb [arch=amd64] https://download.docker.com/linux/debian bullseye stable" | sudo tee  /etc/apt/sources.list.d/docker.list
```

This command will add repository URL to */etc/apt/sources.list.d/docker.list*.

Step 4: Install Docker on Kali Linux

Update the apt package index.

```
$ sudo apt update

gn:1 http://dl.google.com/linux/chrome/deb stable InRelease

Get:3 https://download.docker.com/linux/debian bullseye InRelease [44.4 kB]

Hit:2 http://kali.download/kali kali-rolling InRelease

Hit:4 http://dl.google.com/linux/chrome/deb stable Release

Get:5 https://download.docker.com/linux/debian bullseye/stable amd64 Packages [10.9 kB]

Fetched 55.3 kB in 1s (45.2 kB/s)

Reading package lists... Done

Building dependency tree

Reading state information... Done

186 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

To install Docker CE on Kali Linux, run the command:

```
sudo apt install docker-ce docker-ce-cli containerd.io
```

Hit the **y** key to start installation of Docker.

```
Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

The following packages were automatically installed and are no longer required:

 fastjar fonts-roboto-slab gnome-desktop3-data jarwrapper kali-wallpapers-2021.4
libaom0 libcbor0 libcodec2-0.9 libdap27 libdapclient6v5 libdav1d4 libepsilon1
libfluidsynth2 libfmt7 libgdal28

 libgdal29 libgdk-pixbuf-xlib-2.0-0 libgdk-pixbuf2.0-0 libgeos-3.9.1 libgeos3.10.1
libgnome-desktop-3-19 libgupnp-1.2-0 libidn11 libigdgmm11 libnetcdf18 libntfs-3g883
libodbc1 libodbccr2

 libomp-11-dev libomp5-11 libproj19 libqhull8.0 liburcu6 liburing1 libvpx6 libwireshark14
libwiretap11 libwsutil12 libx265-192 libxkbregistry0 libyara4 linux-image-5.10.0-kali9-
amd64 maltego
```

```
odbcinst odbcinst1debian2 python3-editor python3-exif python3-ipython-genutils
python3-orjson python3-pylnk python3-stem ruby-atomic ruby-thread-safe starkiller
zaproxy

After this operation, 409 MB of additional disk space will be used.

Do you want to continue? [Y/n] y

Use 'sudo apt autoremove' to remove them.

The following additional packages will be installed:

 docker-ce-rootless-extras docker-scan-plugin libslirp0 pigz slirp4netns

Suggested packages:

 aufs-tools cgroupfs-mount | cgroup-lite

The following NEW packages will be installed:

 containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-scan-plugin
libslirp0 pigz slirp4netns

0 upgraded, 8 newly installed, 0 to remove and 16 not upgraded.

Need to get 97.1 MB of archives.
```

This installation will add **docker** group to the system without any users. Add your user account to the group to run docker commands as non-privileged user.

```
sudo usermod -aG docker $USER
newgrp docker
```

Check Docker version installed.

```
$ docker version
```

Install Docker compose

```
sudo apt update
sudo apt install -y curl wget
```

Once curl has been installed, download the latest Compose on your Linux machine.

```
curl -s https://api.github.com/repos/docker/compose/releases/latest | grep browser_download_url  | grep docker-compose-linux-x86_64 | cut -d '"' -f 4 | wget -qi -
```

```
chmod +x docker-compose-linux-x86_64
```

```
sudo mv docker-compose-linux-x86_64 /usr/local/bin/docker-compose
```

Docker compose version

```
$ docker-compose version
Docker Compose version v2.23.1
```

5.1.6 Installation setup

```
$ cd setup
docker-compose up setup
docker-compose up –d
```

We get the services running as follows

When booting a Docker Compose setup for the ELK (Elasticsearch, Logstash, Kibana) stack, several services start on different ports to facilitate log processing, storage, and visualization:

1. Elasticsearch:

Service Name: elasticsearch
Ports:
9200: HTTP REST API for data querying and management
9300: Elasticsearch Transport Protocol for node-to-node communication

2. Logstash:

Service Name: logstash

Port: 5044 for log ingestion and processing

3. Kibana:

Service Name: kibana

Port: 5601 for the web interface, allowing users to interact with Elastic search data through dashboards, visualizations, and queries

These services collectively form the ELK stack, providing comprehensive log management, analysis, and visualization capabilities.

Beats agent configuration file

Winlogbeat.yml

```
###################### Winlogbeat Configuration Example
#########################

# This file is an example configuration file highlighting only the most
common
# options. The winlogbeat.reference.yml file from the same directory
contains
```

```
# all the supported options with more comments. You can use it as a
reference.
#
# You can find the full configuration reference here:
# https://www.elastic.co/guide/en/beats/winlogbeat/index.html

# ======================== Winlogbeat specific options
=========================

# event_logs specifies a list of event logs to monitor as well as any
# accompanying options. The YAML data type of event_logs is a list of
# dictionaries.
#
# The supported keys are name, id, xml_query, tags, fields,
fields_under_root,
# forwarded, ignore_older, level, event_id, provider, and include_xml.
# The xml_query key requires an id and must not be used with the name,
# ignore_older, level, event_id, or provider keys. Please visit the
# documentation for the complete details of each option.
# https://go.es.io/WinlogbeatConfig

winlogbeat.event_logs:
  - name: Application
    ignore_older: 72h

  - name: System

  - name: Security

  - name: Microsoft-Windows-Sysmon/Operational

  - name: Windows PowerShell
    event_id: 400, 403, 600, 800

  - name: Microsoft-Windows-PowerShell/Operational
    event_id: 4103, 4104, 4105, 4106

  - name: ForwardedEvents
    tags: [forwarded]

# ====================== Elasticsearch template settings
======================

setup.template.settings:
  index.number_of_shards: 1
  #index.codec: best_compression
  #_source.enabled: false
```

```
# ================================== General
==================================

# The name of the shipper that publishes the network data. It can be
used to group
# all the transactions sent by a single shipper in the web interface.
#name:

# The tags of the shipper are included in their field with each
# transaction published.
#tags: ["service-X", "web-tier"]

# Optional fields that you can specify to add additional information to
the
# output.
#fields:
#  env: staging

# ================================= Dashboards
=================================
# These settings control loading the sample dashboards to the Kibana
index. Loading
# the dashboards is disabled by default and can be enabled either by
setting the
# options here or by using the `setup` command.
#setup.dashboards.enabled: false

# The URL from where to download the dashboard archive. By default,
this URL
# has a value that is computed based on the Beat name and version. For
released
# versions, this URL points to the dashboard archive on the
artifacts.elastic.co
# website.
#setup.dashboards.url:

# ================================== Kibana
==================================

# Starting with Beats version 6.0.0, the dashboards are loaded via the
Kibana API.
# This requires a Kibana endpoint configuration.
setup.kibana:

  # Kibana Host
  # Scheme and port can be left out and will be set to the default
(http and 5601)
```

```
  # In case you specify and additional path, the scheme is required:
http://localhost:5601/path
  # IPv6 addresses should always be defined as:
https://[2001:db8::1]:5601
  #host: "localhost:5601"

  # Kibana Space ID
  # ID of the Kibana Space into which the dashboards should be loaded.
By default,
  # the Default Space will be used.
  #space.id:

# ============================== Elastic Cloud
==============================

# These settings simplify using Winlogbeat with the Elastic Cloud
(https://cloud.elastic.co/).

# The cloud.id setting overwrites the `output.elasticsearch.hosts` and
# `setup.kibana.host` options.
# You can find the `cloud.id` in the Elastic Cloud web UI.
#cloud.id:

# The cloud.auth setting overwrites the `output.elasticsearch.username`
and
# `output.elasticsearch.password` settings. The format is
`<user>:<pass>`.
#cloud.auth:

# ================================== Outputs
===================================

# Configure what output to use when sending the data collected by the
beat.

# ---------------------------- Elasticsearch Output ------------------
---------
output.elasticsearch:
  # Array of hosts to connect to.
  hosts: ["localhost:9200"]

  # Protocol - either `http` (default) or `https`.
  #protocol: "https"

  # Authentication credentials - either API key or username/password.
  #api_key: "id:api_key"
  #username: "elastic"
  #password: "changeme"
```

```
  # Pipeline to route events to security, sysmon, or powershell
pipelines.
  pipeline: "winlogbeat-%{[agent.version]}-routing"

# ---------------------------- Logstash Output ----------------------
---------
#output.logstash:
  # The Logstash hosts
  #hosts: ["localhost:5044"]

  # Optional SSL. By default is off.
  # List of root certificates for HTTPS server verifications
  #ssl.certificate_authorities: ["/etc/pki/root/ca.pem"]

  # Certificate for SSL client authentication
  #ssl.certificate: "/etc/pki/client/cert.pem"

  # Client Certificate Key
  #ssl.key: "/etc/pki/client/cert.key"

# ================================= Processors
=================================
processors:
  - add_host_metadata:
      when.not.contains.tags: forwarded
  - add_cloud_metadata: ~

# ================================== Logging
==================================

# Sets log level. The default log level is info.
# Available log levels are: error, warning, info, debug
#logging.level: debug

# At debug level, you can selectively enable logging only for some
components.
# To enable all selectors, use ["*"]. Examples of other selectors are
"beat",
# "publisher", "service".
#logging.selectors: ["*"]

# ============================== X-Pack Monitoring
==============================
# Winlogbeat can export internal metrics to a central Elasticsearch
monitoring
# cluster.  This requires xpack monitoring to be enabled in
Elasticsearch.  The
```

```
# reporting is disabled by default.

# Set to true to enable the monitoring reporter.
#monitoring.enabled: false

# Sets the UUID of the Elasticsearch cluster under which monitoring
data for this
# Winlogbeat instance will appear in the Stack Monitoring UI. If
output.elasticsearch
# is enabled, the UUID is derived from the Elasticsearch cluster
referenced by output.elasticsearch.
#monitoring.cluster_uuid:

# Uncomment to send the metrics to Elasticsearch. Most settings from
the
# Elasticsearch outputs are accepted here as well.
# Note that the settings should point to your Elasticsearch
*monitoring* cluster.
# Any setting that is not set is automatically inherited from the
Elasticsearch
# output configuration, so if you have the Elasticsearch output
configured such
# that it is pointing to your Elasticsearch monitoring cluster, you can
simply
# uncomment the following line.
#monitoring.elasticsearch:

# ============================== Instrumentation
===============================

# Instrumentation support for the winlogbeat.
#instrumentation:
    # Set to true to enable instrumentation of winlogbeat.
    #enabled: false

    # Environment in which winlogbeat is running on (eg: staging,
production, etc.)
    #environment: ""

    # APM Server hosts to report instrumentation results to.
    #hosts:
    #  - http://localhost:8200

    # API Key for the APM Server(s).
    # If api_key is set then secret_token will be ignored.
    #api_key:

    # Secret token for the APM Server(s).
```

```
    #secret_token:


# ================================ Migration
=================================

# This allows to enable 6.7 migration aliases
#migration.6_to_7.enabled: true
```

CHAPTER 06

OUTPUT SCREENS

6. OUTPUT SCREENS

6.1 BEATS AGENT SERVICE (Starting)

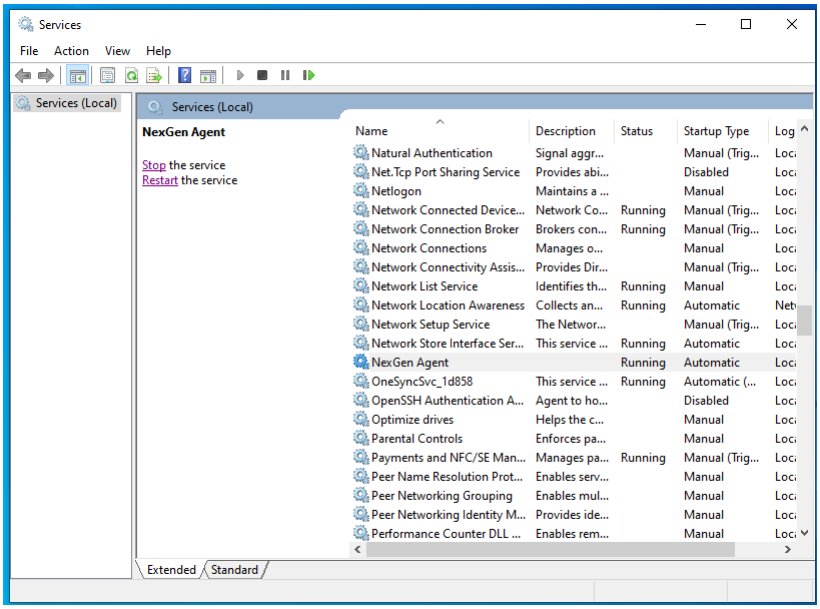

Fig. 6.1 NexGen agent

6.2 AGENT PROPERTIES

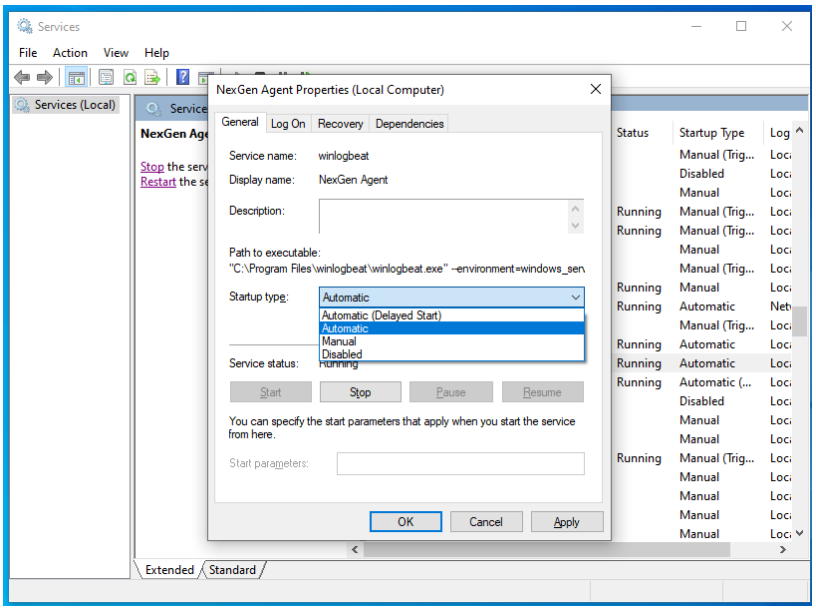

Fig. 6.2 NexGen agent properties

6.3 FLEET SERVER MANAGEMENT

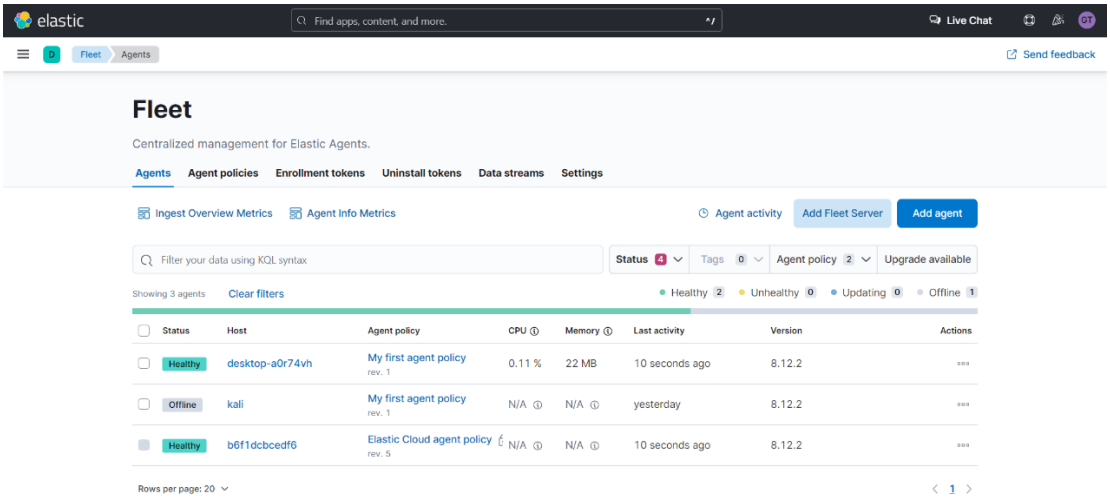

Fig. 6.3 Fleet server centralized management for Elastic Agents

6.4 DISCOVERING REAL TIME LOGS IN THE KIBANA

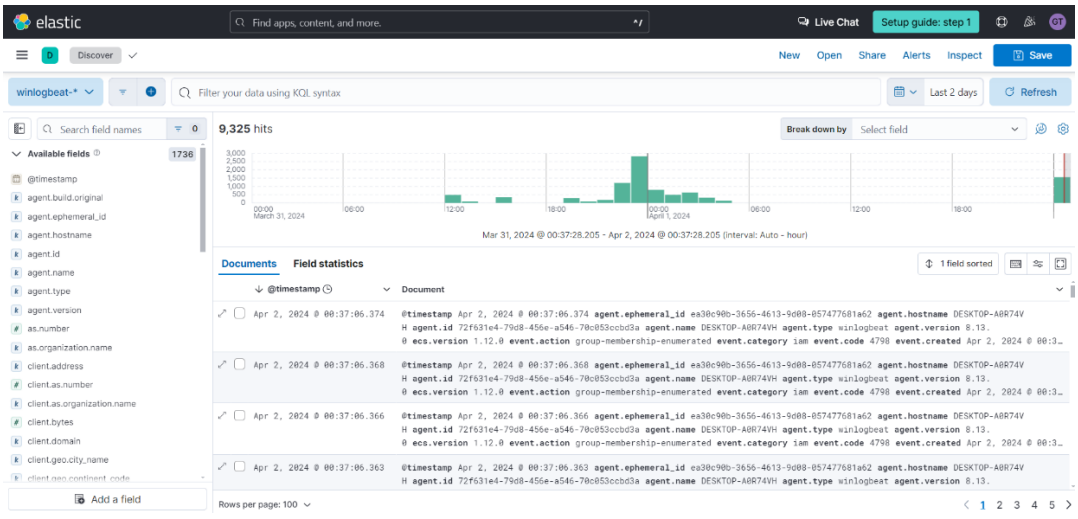

Fig. 6.4 Visualizing the logs and metrics in the Kibana

6.5 QUERYING A LOG RECORD IN KQL FILTER

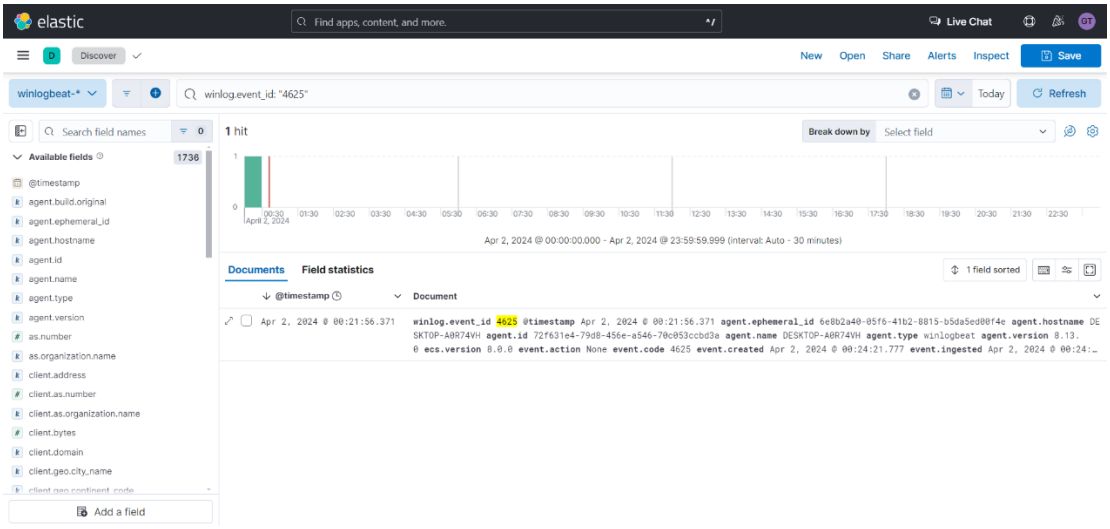

Fig. 6.5 Querying logs of Failed login attempts

6.6 GOOGLING THE LOG EVENT ID FOR MORE INFORMATION

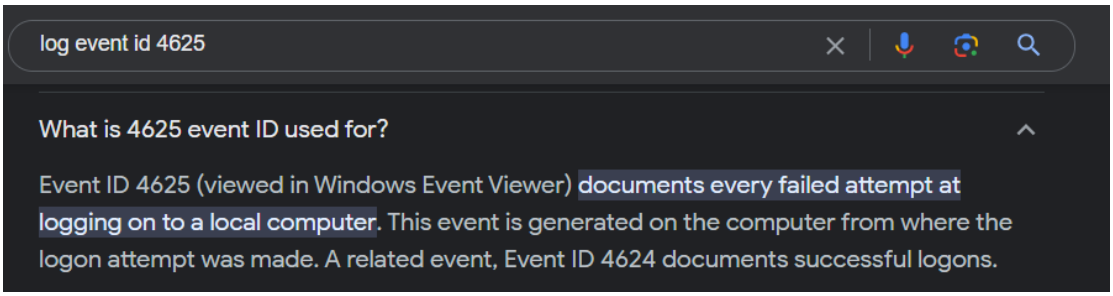

Fig. 6.6 Log event id 4625

6.7 INTEGRATING ELASTIC DEFEND FOR SECURITY ACTIONS

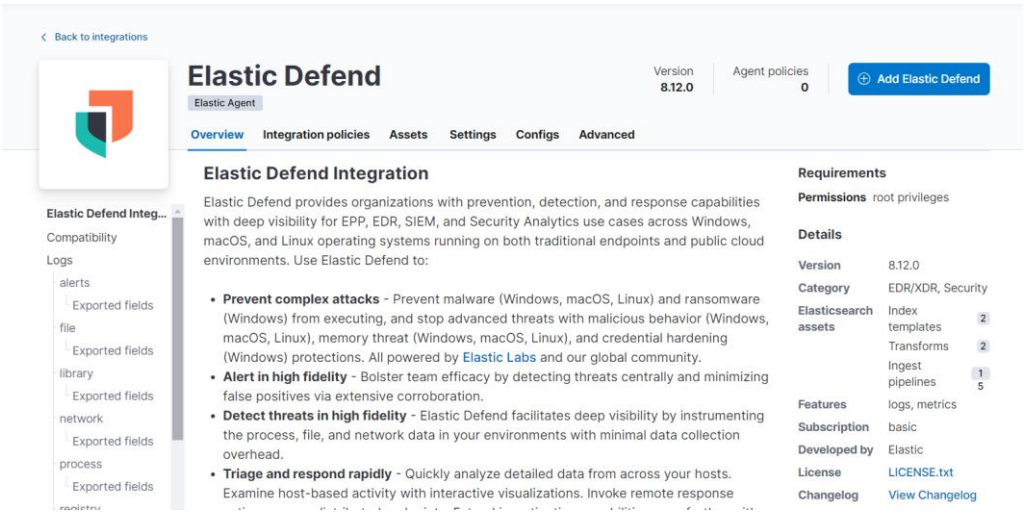

Fig. 6.7 Elastic Defend Integration

6.8 SECURITY OPTIONS AFTER INTEGRATING ELASTIC DEFEND

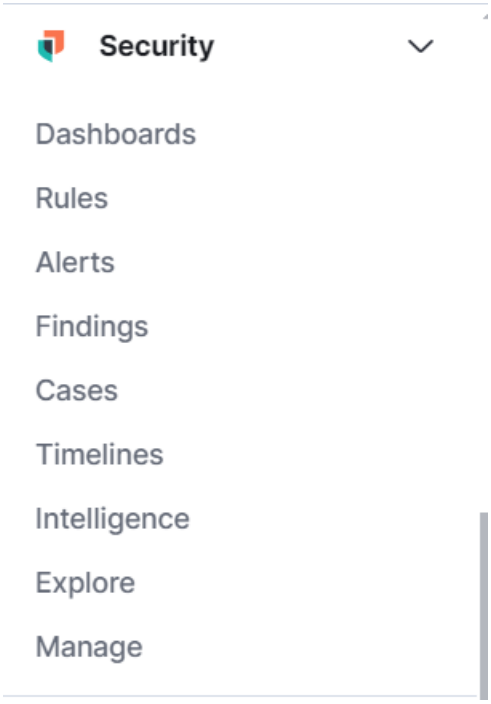

Fig. 6.8 Security options

6.9.1 DETECTION RULE SPECIFING

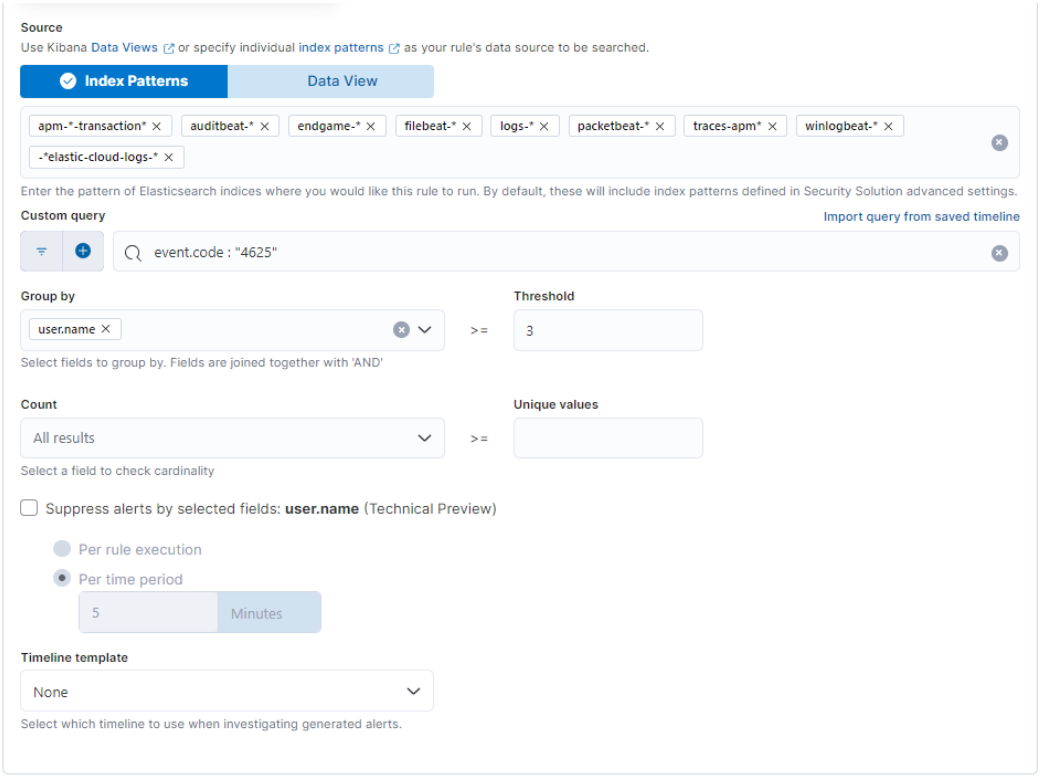

Fig. 6.9.1 Detection rule specification

6.9.2 DETECTION RULE ABOUT

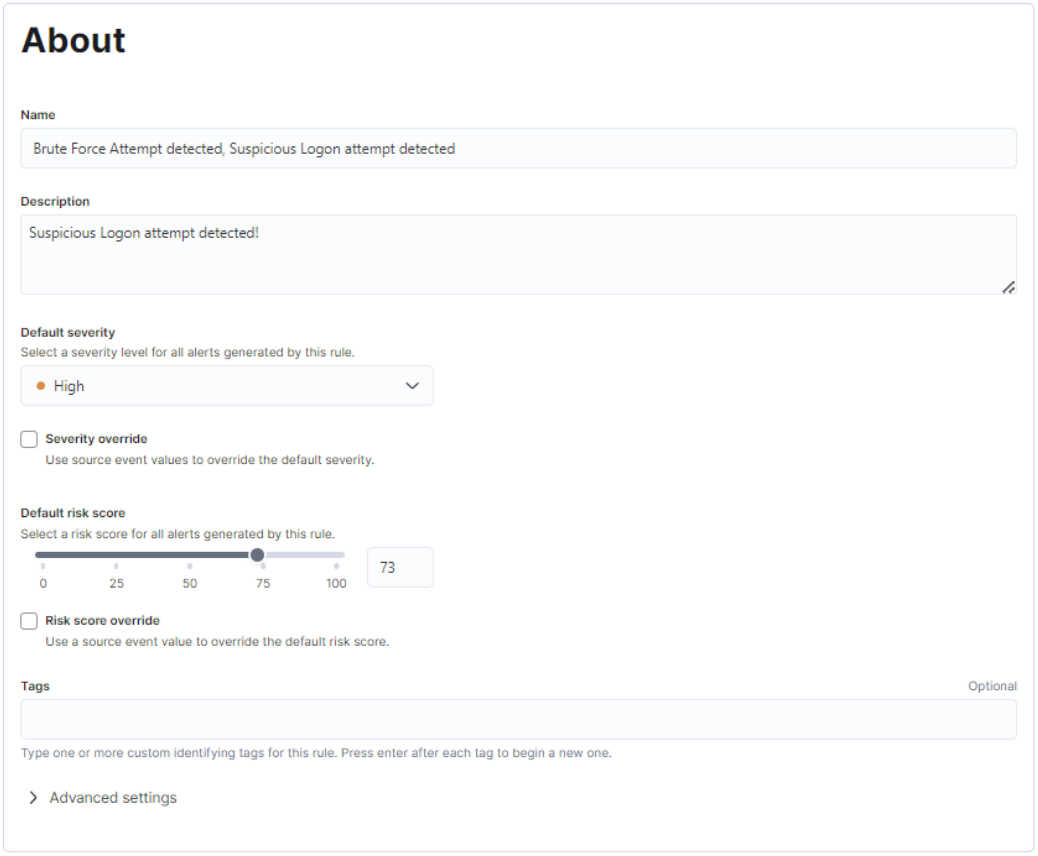

Fig. 6.9.2 Detection rule about section

6.9.3 SCHEDULING THE DETECTION RULE

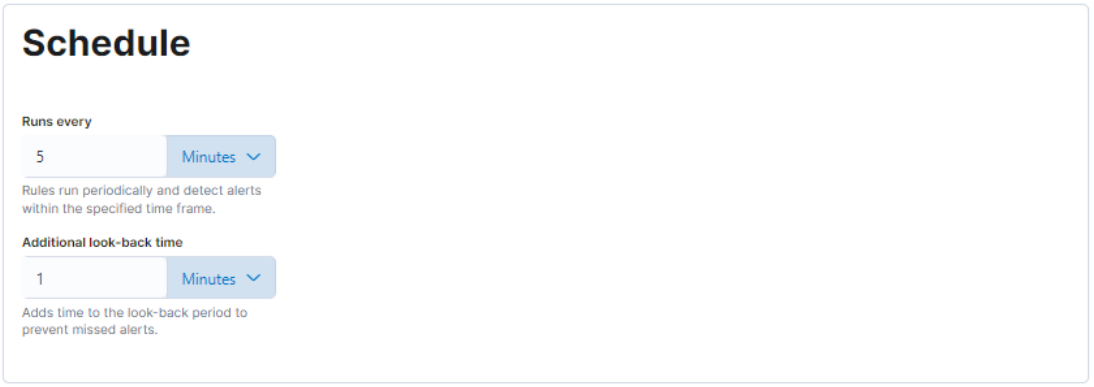

Fig. 6.9.3 Detection rule scheduling

6.9.4 SPECIFYING THE ACTIONS TO BE TRIGGERED UPON DETECTION

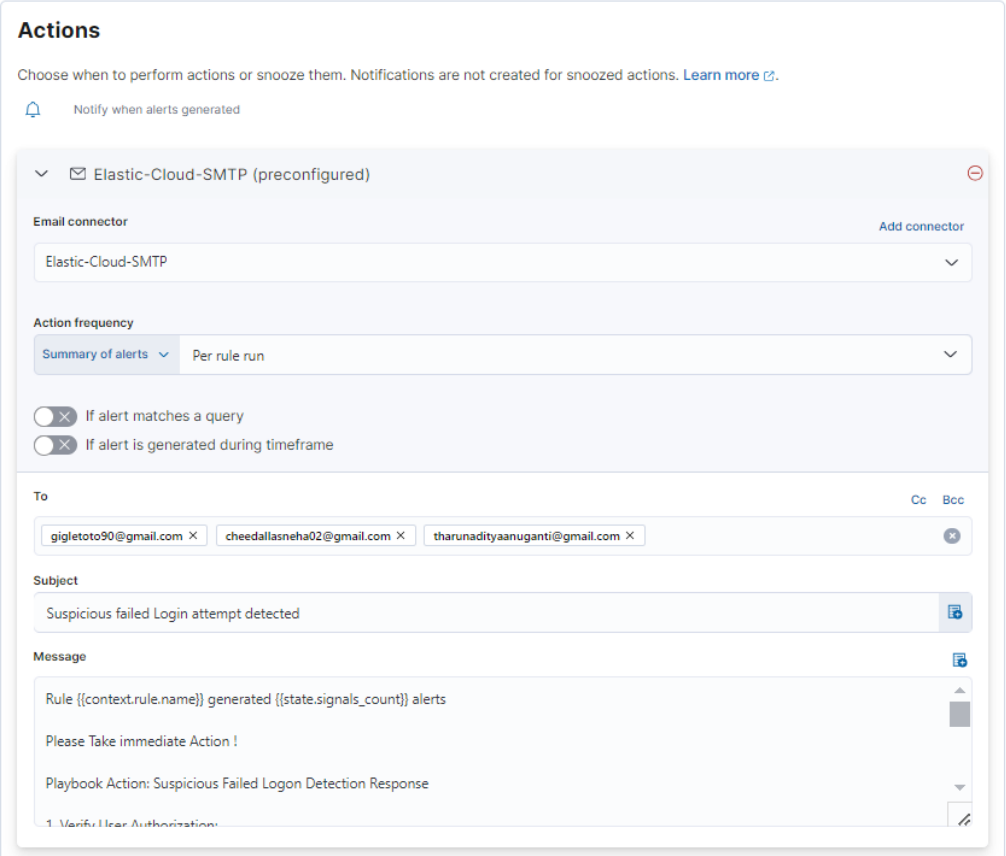

Fig. 6.9.4 Detection rule trigger actions

6.9.5 BRUTE FORCING PASSWORD AT SYSTEM LOGIN

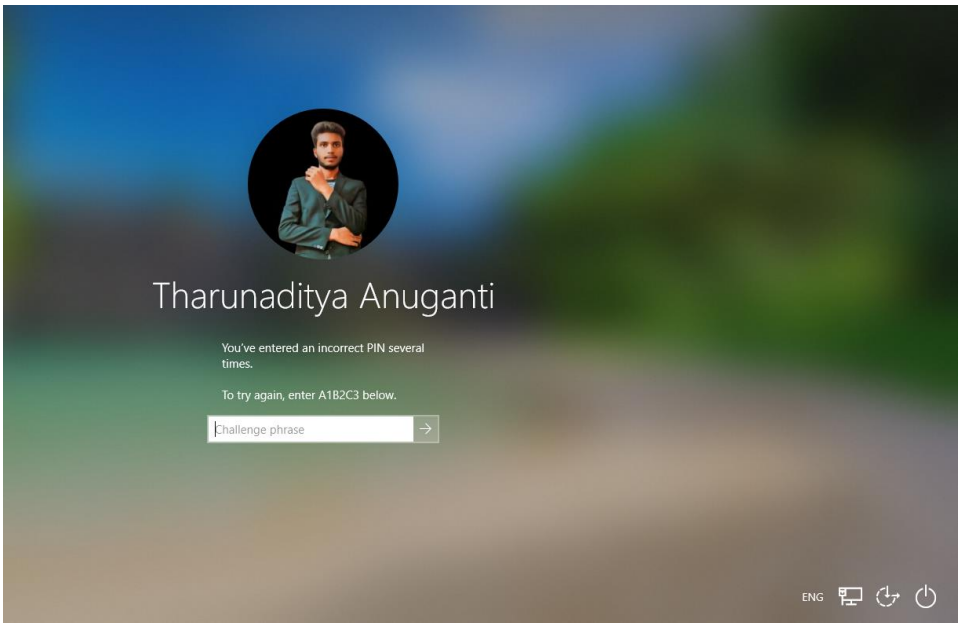

Fig. 6.9.6 Several failed login attempts made at system login

6.9.6 DETECTION AND RESPONSE DASHBOARD

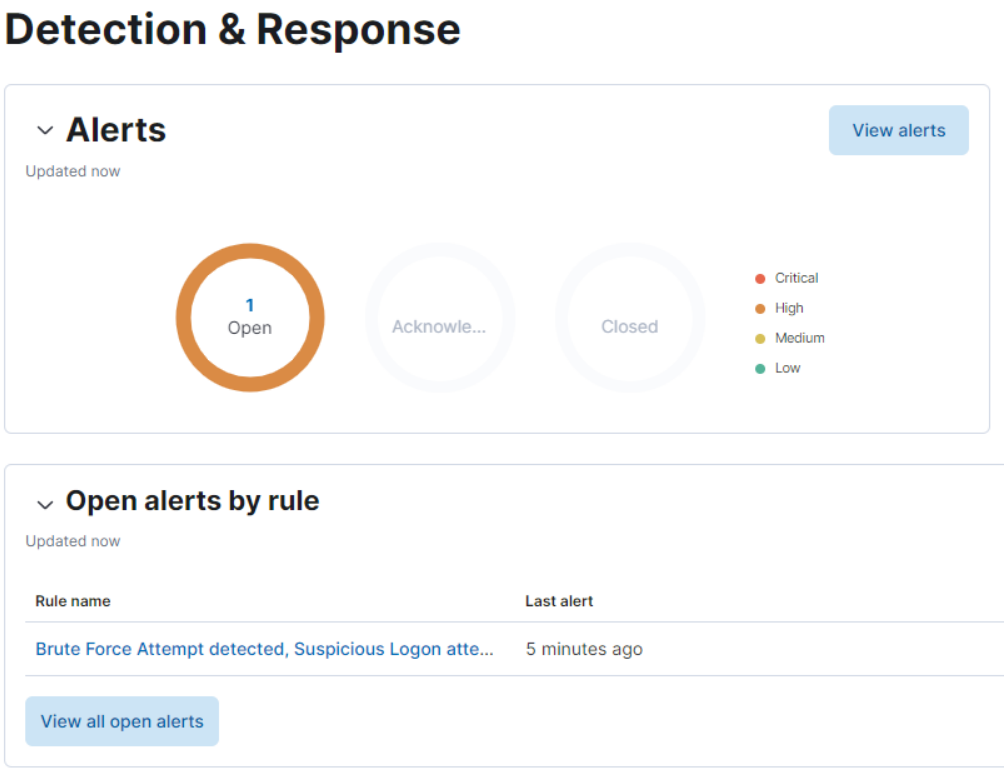

Fig. 6.9.7 Alert raised in detection & Response dashboard

6.9.7 MAILING ALERTS FOR EFFECTIVE INCIDENT RESPONSE

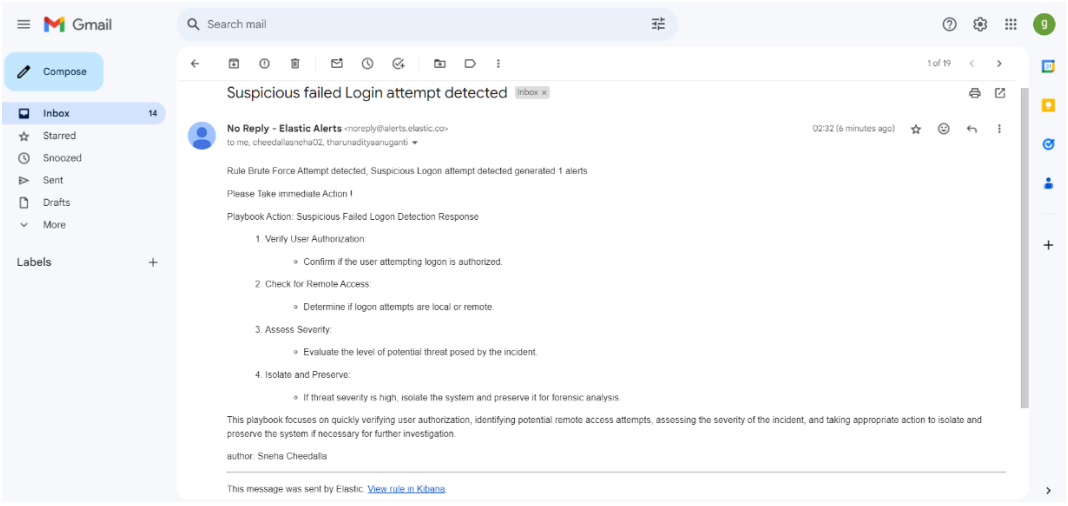

Fig. 6.9.8 Mail received on detecting a suspicious login attempt

CHAPTER 07

TESTING AND DEBUGGING

7. TESTING AND DEBUGGING

7.1 TESTING PROCESS

Firstly, we will start testing with individual module and will perform unit testing on that.So we successfully verify and validate all modules by successively integrating each module. Moreover, checking the work done was very important to reduce risk factor. Checking was being ultimately handled by testing but interim checking was required. So we planned work done by one member was tested by other for some time and again revolved for other level check. This technique proved to be very much helpful as it came out with innovative ideas to reduce error very low level. The objective of this testing phase is to prove that the developed system satisfies the requirements defined earlier. Several types of tests will be conducted in this phase. Testing is an important phase of system development because it can ensure the system matches the specifications. Besidesthat, testing also ensures that the system functions in the correct and proper manner with the minimum amount of errors. Applications are susceptible to bugs and malfunctions, such as scripts that not run properly. Besides that, there might be compatibility problems because a project may run perfectly on one device but may not display properly on another. Bottom-up testing strategy is used in this system to avoid unnecessary duplication of effort. Individual objects will be tested in isolation using unit testing and gradually integrated for the higher level integration testing and system testing. Failed components will be migrated back to the development phase for rework, and componentsthat work properly will migrate ahead for implementation.

7.1.1 UNIT TESTING

Unit testing reveals syntax and semantic errors from the smallest programming unit. In this thesis, unit testing is used to test each individual page. Errors that are found in a particular page of the application are thoroughly debugged and removed before startingto develop another. Due to the dynamic nature of testing, there is no proper testing documentation created.

7.1.2 LINK /INTEGRATION TESTING

Testing when each application of a particular Section in the System passed the unit testing, integration test was carried out to ensure that pages are linked in the correct flow an integrate properly into the entire website. All the buttons, text boxes and navigation bars were tested.

7.1.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available asspecified by the business and technical requirements, system documentation, and user manuals. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identifiedand the effective value of current tests is determined.

7.1.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It testsa configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptionsand flows, emphasizing pre-driven process links and integration points.

7.1.5 WHITE BOX TESTING

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached.

7.1.6 BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

7.1.6.1 TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

7.1.6.2 TEST OBJECTIVES

All field entries must work properly.

Pages must be activated from the identified link.

The entry screen, messages and responses must not be delayed.

7.1.6.3 FEATURES TO BE TESTED

Verify that the entries are of the correct format.

No duplicate entries should be allowed.

All links should take the user to the correct page.

7.1.7 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or one step up software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.1.8 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.2 TEST CASES

TEST ID	TEST DESCRIPTION	TEST DESIGN	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
1.	Suspicious login attempt detection	Attempting number of passwords	Raising alert Regarding Brute force attempt Working Action trigger	Alert raised Received email With incident response guide	Pass

Table :7.1 Test cases for system

CHAPTER 08

CONCLUSION

8. CONCLUSION

1.**Enhanced SOC Capabilities:**

The "NexGen SIEM" project significantly enhances the capabilities of Security Operations Centers (SOCs) by integrating open-source technologies. Through centralized data collection, automated incident management, and proactive threat detection mechanisms, the project empowers SOC professionals to respond rapidly and effectively to cyber threats. This results in improved agility, efficiency, and effectiveness in safeguarding organizational assets and mitigating security risks.

2.Promotion of Collaboration and Knowledge Sharing:

A key focus of the project is to promote collaborative workflows and knowledge sharing among SOC teams. By fostering a culture of collective intelligence and teamwork, the project enhances SOC effectiveness and resilience. Collaborative efforts enable SOC professionals to leverage each other's expertise and insights, leading to more informed decision-making and better responses to security incidents.

3.Continuous Improvement and Innovation:

The "NexGen SIEM" project underscores a commitment to continuous improvement and innovation within the cybersecurity landscape. By leveraging open-source technologies and integrating with threat intelligence platforms, the project ensures adaptability to evolving cyber threats. This proactive approach to security allows organizations to stay ahead of emerging challenges and effectively mitigate risks, thereby contributing to a more secure operational environment.

CHAPTER 09

FUTURE ENHANCEMENTS

9. FUTURE ENCHANCEMENTS

Machine Learning Integration: Implementing machine learning algorithms to enhance threat detection and automate response processes further. This would involve developing models for anomaly detection, behaviour analysis, and predictive analytics.

Enhanced Threat Intelligence Feeds: Integrate additional threat intelligence feeds to enrich the data available to the SOC, allowing for more comprehensive analysis and quicker identification of emerging threats.

Cloud-Native Architecture: Transitioning to a cloud-native architecture to leverage scalability, elasticity, and agility offered by cloud platforms. This could involve containerization of SOC components and deployment on Kubernetes for easier management and resource optimization.

Real-Time Visualization and Reporting: Developing interactive dashboards and real-time reporting tools to provide stakeholders with instant insights into security posture, threat landscape, and SOC performance metrics.

Incident Response Playbooks: Creating and refining incident response playbooks tailored to specific types of threats and attack scenarios. These playbooks would streamline response procedures and ensure consistency in handling incidents.

Integration with DevOps Pipelines: Integrating security processes seamlessly into DevOps pipelines to facilitate continuous security monitoring and automated remediation of vulnerabilities in development and production environments.

Threat Hunting Capabilities: Empowering SOC analysts with advanced threat hunting tools and techniques to proactively search for and identify hidden threats that may evade automated detection mechanisms.

User Behaviour Analytics (UBA): Implementing UBA solutions to monitor and analyze user activities across the network, endpoints, and applications, enabling early detection of insider threats and unauthorized access.

Regulatory Compliance Automation: Developing automated workflows and tools to streamline compliance with industry regulations and standards, such as GDPR, HIPAA, or PCI DSS, by ensuring timely reporting and adherence to security policies.

Enhanced Collaboration Tools: Introducing collaboration platforms and communication tools tailored for SOC teams to facilitate knowledge sharing, incident coordination, and effective communication during security incidents.

CHAPTER 10

REFERENCES

10. REFERENCES

[1] A SURVEY ON: “LOG ANALYSIS WITH ELK STACK TOOL” Pranita P. Bavaskar, PG Student, Department of Computer Science Engineering, Sandip University, Nasik, India (2019) IJRAR 965 – 968

[2] Toward a Secure ELK Stack

Fatimetou Abdou VADHIL 1, Mohamed Lemine SALIHI

2, Mohamedade Farouk NANNE 3 Université de Nouakchott Alassriya. Nouakchott, Mauritanie (2019) IJCSIS 139 – 143

[3] Repository related to Open SOC home lab setup using the similar tools

https://github.com/archanchoudhury/SOC-OpenSource