Balancing Security and Privacy in SIEM Logs

Final Year Project

2024-2025

A project submitted in partial fulfillment of the degree of

BS in Cyber Security

Submitted to

Dr Hilmand Khan

Department of Cyber Security

Faculty of Computing & Artificial Intelligence (FCAI)

Air University, Islamabad

SIEM	[✔] Development [ ] R&D
Area of specialization	[✔] WebApp [ ] Mobile App [ ] AI based [ ] Embedded System
FYP ID	2432

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	211086	Hameed Ullah	211086@students.au.edu.pk	
(ii)	211106	Furqan Rafique	211106@students.au.edu.pk	
(iii)	212074	Hassan Imam	212074@students.au.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Certificate

This is to certify that, I _____ S/D of _____, group leader of FYP under registration no _____at Cybersecurity Department, Air University. I declare that my FYP report is checked by my supervisor.

Date:_____ Name of Group Leader: _____ Signature:_____

Name of Supervisor: Mr. Hilmand khan

Designation: Lecturer

Signature: _____

Co-Supervisor: (Not Allocated)

Designation: (N/A)

Signature: _____

HoD: Dr. Ghalib

Signature: _____

Balancing Security and Privacy in SIEM Logs

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature

APPROVAL

PROJECT SUPERVISOR

Name:_____

Date:_____ Signature:_____

_____

PROJECT MANAGER

Date:_____ Signature:_____

_____ _____

CHAIR DEPARTMENT

Date:_____ Signature:_____

Dedication

This project is wholeheartedly dedicated to our families, whose unwavering support, patience, and encouragement have been our constant source of motivation throughout this challenging and rewarding journey.

We also dedicate this work to our supervisor, Dr. Hilmand Khan, whose insightful guidance and steadfast belief in our potential were instrumental in navigating the complexities of this research.

Finally, this project is dedicated to the pursuit of a more secure and privacy-respecting digital world, hoping that our work contributes, even in a small way, to addressing the critical balance between robust cybersecurity and the fundamental right to data privacy.

Acknowledgements

We would like to express our sincere gratitude to a number of individuals and entities who have contributed to the successful completion of our Final Year Project, "Balancing Security and Privacy in SIEM Logs."

First and foremost, our profound appreciation goes to our supervisor, Dr. Hilmand Khan. His expert guidance, invaluable feedback, and continuous support were pivotal at every stage of this project, from conceptualization to implementation and documentation. His mentorship has not only shaped this project but also our understanding of the cybersecurity domain.

We extend our thanks to the Department of Cyber Security and the Faculty of Computing and Artificial Intelligence at Air University for providing us with the academic environment and resources necessary to undertake this research. The knowledge and skills imparted during our studies here have been the bedrock of this work.

We are also grateful for the comprehensive documentation and open-source nature of the Elastic Stack (Elasticsearch, Logstash, Kibana), which served as the core technological foundation for our proposed framework. The availability of such powerful tools is a testament to the collaborative spirit of the tech community.

Our gratitude also extends to the authors and organizations behind the standards and research materials we consulted, including GDPR guidelines, NIST Privacy Framework, and various other cybersecurity publications that informed our understanding of encryption, tokenization, and privacy-preserving techniques.

Finally, we thank our peers and colleagues for their insightful discussions and collaborative spirit, and our families and friends for their enduring patience and moral support throughout this demanding endeavor.

Executive Summary

Security Information and Event Management (SIEM) systems are critical for modern cybersecurity, yet they often process sensitive data, creating a conflict between security monitoring and data privacy. This project, "Balancing Security and Privacy in SIEM Logs," addresses this challenge by developing a robust framework that enhances log security and privacy while maintaining their utility for threat detection and analysis. Leveraging advanced tools like Elasticsearch for secure storage, Logstash for log ingestion and processing, and Kibana for visualization, the project integrates key privacy-enhancing techniques directly into the SIEM workflow.

The core objectives achieved include implementing end-to-end encryption (E2EE) for logs in transit using TLS and demonstrating pathways for database-level encryption for data at rest. The framework focuses on tokenizing and anonymizing sensitive Personally Identifiable Information (PII) such as IP addresses and usernames within logs, replacing them with irreversible tokens (via SHA256 hashing) to protect privacy while allowing for correlation and analysis. Techniques for redacting other sensitive data like email addresses directly within log messages have also been implemented using Logstash filters. Furthermore, the project incorporates cryptographic hashing (e.g., SHA-256) to help ensure log integrity.

This solution is designed to be scalable, manage high-throughput log data with low latency, and offers user-friendly monitoring through Kibana dashboards. By addressing the limitations in conventional SIEM systems, such as ineffective anonymization and compliance gaps, this project provides a practical approach to meeting stringent data protection regulations like GDPR and CCPA. The implemented framework demonstrates a viable balance between operational security needs and critical data privacy mandates, offering an enhanced, privacy-compliant SIEM solution.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

1.1	PII	Personally Identifiable Information
1.2	SIEM	Security Information and Event Management
1.3	ELK	Elasticsearch, Logstash, Kibana
1.4	TLS	Transport Layer Security
1.5	E2EE	End-to-End Encryption
1.6	RBAC	Role-Based Access Control
1.7	GDPR	General Data Protection Regulation
1.8	CCPA	California Consumer Privacy Act
1.9	HIPAA	Health Insurance Portability and Accountability Act
1.16	CA	Certificate Authority
1.17	API	Application Programming Interface
1.18	KMS	Key Management Service
1.19	SHA-256	Secure Hash Algorithm 256-bit
1.20	YAML	YAML Ain't Markup Language
1.21	JSON	JavaScript Object Notation
1.22	JDK	Java Development Kit
1.23	SOAR	Security Orchestration, Automation and Response
1.24	NIST	National Institute of Standards and Technology
1.25	ISO	International Organization for Standardization
1.26	OSSIM	Open Source Security Information Management
1.27	Wazuh	Open-source Security Monitoring Platform
1.28	UI	User Interface
1.29	TCP	Transmission Control Protocol
1.30	HTTPS	HyperText Transfer Protocol Secure

Chapter 1

Introduction & Background

Chapter 1: Introduction

1.1. Background

Security Information and Event Management (SIEM) systems are fundamental components in the contemporary cybersecurity infrastructure of organizations. They provide essential capabilities for real-time monitoring, comprehensive analysis, and effective response to a wide array of potential security threats. However, the extensive log data that these SIEM systems collect and process often contains sensitive details and Personally Identifiable Information (PII). This inherently creates significant challenges in upholding robust security measures while simultaneously ensuring data privacy. The evolving landscape of data protection, underscored by modern regulations such as the General Data Protection Regulation (GDPR) and the California Consumer Privacy Act (CCPA), imposes stringent requirements on organizations to implement rigorous measures for safeguarding user privacy and maintaining compliance. Concurrently, there is an imperative for these organizations to maintain operational utility, which includes ensuring low-latency processing capabilities for increasingly large volumes of log data. This project is centered on addressing these often conflicting requirements. The core aim is to utilize advanced technological solutions, specifically the Elastic Stack (Elasticsearch and Logstash), to develop and implement a resilient framework designed for securing SIEM logs effectively. This framework systematically integrates robust encryption protocols, data tokenization methods, and other privacy-preserving techniques directly into the standard SIEM workflow, thereby ensuring that logs are not only secure against unauthorized access but also fully compliant with prevailing data privacy regulations. By facilitating real-time anomaly detection and providing intuitive data visualization through Kibana dashboards, this project seeks to strike a critical balance between pressing security needs and data privacy concerns, ultimately enhancing overall threat detection and response capabilities.

1.2. Motivations and Challenges

Motivations:

- **Protecting Sensitive Log Data from Breaches:** SIEM logs frequently contain highly detailed information, including records of user activities, IP addresses, system configurations, and other operational data. This concentration of critical information

makes SIEM systems a prime target for cyberattacks aimed at data exfiltration or system compromise. Therefore, a primary motivation for this project is to implement enhanced security measures to prevent unauthorized access and mitigate the risk of breaches involving such critical log data.

- **Ensuring Compliance with Data Protection Regulations:** The global regulatory landscape for data privacy has become increasingly stringent with the enforcement of comprehensive frameworks like GDPR, CCPA, and the Health Insurance Portability and Accountability Act (HIPAA). These regulations mandate specific measures for data protection, including data minimization, pseudonymization (such as tokenization), and secure data processing, to protect user privacy adequately. Failure to comply can result in severe financial penalties and reputational damage, making adherence a significant motivating factor.
- **Maintaining Trust with Stakeholders:** Data breaches and privacy violations can severely erode the trust that customers, partners, and other stakeholders place in an organization. By proactively building and maintaining a secure and privacy-conscious logging environment, organizations can provide tangible assurance that sensitive data is being handled responsibly and ethically, thereby preserving and strengthening these vital relationships.
- **Improving Incident Response Efficiency:** The implementation of enhanced security and privacy measures within SIEM logs directly contributes to the speed and accuracy of incident detection and response. A well-secured and appropriately anonymized logging system allows security teams to identify and react to threats more effectively, often before they can escalate into significant security events.
- **Adopting State-of-the-Art Technologies:** This project is motivated by the opportunity to integrate and leverage advanced, industry-recognized tools such as Elasticsearch, Logstash, and Kibana. Adopting these technologies represents a forward-looking approach to cybersecurity, enhancing not only the security posture but also providing sophisticated operational analytics capabilities, which can offer a competitive advantage.

Challenges:

- **Handling Large-Scale Log Ingestion with Low Latency:** Modern IT environments and applications generate log data at an unprecedented velocity and volume. A significant

technical challenge lies in designing a system capable of seamlessly ingesting, storing, and analyzing these massive log streams in near real-time without introducing significant latency or impacting overall system performance.

- **Balancing Operational Utility with Privacy Measures:** While the protection of sensitive information is paramount, overly aggressive or poorly implemented privacy mechanisms (like excessive redaction or irreversible anonymization) can inadvertently obscure critical details needed by security analysts for effective threat detection, investigation, and response. Achieving an optimal balance between these operational requirements and data privacy compliance is a persistent and nuanced challenge.
- **Ensuring Data Integrity and Security Across Complex Pipelines:** Log data typically traverses multiple stages from its point of collection (agents), through processing (Logstash), to storage (Elasticsearch), and finally to analysis and visualization (Kibana). Maintaining robust encryption, enforcing strict access controls, and ensuring data integrity (i.e., that logs are not tampered with) at every stage of this complex pipeline, particularly in distributed system architectures, presents considerable difficulty.
- **Scalability and Resource Efficiency:** The implementation of advanced security and privacy measures, such as strong encryption algorithms and real-time tokenization processes, can introduce substantial computational overhead on the system. Designing the framework to be scalable, allowing it to handle increasing log volumes and processing demands efficiently without requiring a disproportionate increase in resources, is a crucial challenge.
- **Integration with Legacy Systems:** Many organizations continue to operate and rely on older, legacy IT systems. These systems may not natively support modern security protocols, encryption standards, or privacy-enhancing APIs, making seamless integration with a state-of-the-art SIEM framework complex and challenging.
- **Meeting Diverse Compliance Requirements:** Organizations, especially those with a global footprint, often need to adhere to a multitude of data protection and privacy regulations from different jurisdictions (e.g., GDPR in Europe, CCPA in California, HIPAA in the US healthcare sector). Designing a single framework that can meet these varied and sometimes overlapping compliance requirements adds a significant layer of operational and technical complexity.

1.3. Goals and Objectives

Primary Goal: The paramount goal of this project is to significantly enhance the security posture and privacy guarantees associated with Security Information and Event Management (SIEM) log data, while concurrently ensuring that these logs remain fully usable and effective for critical threat detection, incident response, and security analysis activities.

Specific Objectives:

1. **Encrypting Log Data in Transit and at Rest:**
    - Implement robust end-to-end encryption (E2EE) mechanisms, primarily utilizing protocols like TLS (Transport Layer Security), to ensure the secure and confidential transmission of log data from collection agents to processing and storage components.
    - Employ or demonstrate strategies for database-level encryption (e.g., for Elasticsearch data at rest) to protect stored log data from unauthorized access, disclosure, or breaches, even if underlying storage systems are compromised.
2. **Tokenizing and Anonymizing Sensitive Data:**
    - Develop and integrate processes to replace sensitive data fields within logs, such as IP addresses, usernames, and other forms of Personally Identifiable Information (PII), with unique, non-identifiable tokens (pseudonyms).
    - Design the tokenization system such that tokens are, where required by compliance or investigative needs, reversible only by specifically authorized systems or personnel, thereby adhering to privacy regulations while maintaining controlled utility.
3. **Minimizing Data Exposure with Field-Level Privacy Measures:**
    - Apply targeted field-level encryption or robust hashing techniques for specific, highly sensitive data elements within log entries to prevent their misuse or unauthorized disclosure.
    - Explore or discuss the incorporation of techniques such as noise addition (a concept from differential privacy) to protect against re-identification risks in

aggregated or analyzed log data, while aiming to preserve overall analytical utility.

4. **Compliance with Global Data Protection Standards:**
    - Align the developed framework's security and privacy mechanisms with the core requirements of major data protection regulations, including GDPR (General Data Protection Regulation), HIPAA (Health Insurance Portability and Accountability Act), and CCPA (California Consumer Privacy Act), ensuring adherence to legal mandates.
    - Establish methods to document and demonstrate the implemented compliance mechanisms to relevant stakeholders, potentially through reports or specialized dashboards.
5. **Optimizing Log Processing for Scalability and Performance:**
    - Ensure that the log ingestion, processing (including encryption and tokenization), and indexing pipeline is optimized to maintain low latency and high throughput, even when dealing with large volumes of log data from diverse sources.
    - Implement and leverage the scalable pipeline capabilities of Elasticsearch and Logstash to effectively handle dynamic and growing organizational logging needs without performance degradation.
6. **User-Friendly Threat Monitoring and Visualization:**
    - Develop intuitive and actionable dashboards within Kibana to effectively visualize processed log data, highlight security alerts, present threat analytics, and display privacy compliance metrics.
    - Incorporate Role-Based Access Control (RBAC) mechanisms within Kibana and Elasticsearch to restrict access to sensitive dashboards, visualizations, and raw log data based on predefined user roles and responsibilities.
7. **Ensuring Data Integrity:**
    - Implement cryptographic hashing techniques (e.g., SHA-256) for log entries or critical log fields to enable verification of their integrity, ensuring that no unauthorized modifications or tampering occur during transmission or storage.
8. **Deploying a Modular and Flexible Framework:**

- Build the solution with a modular design that allows for easy adaptation and deployment across different industries with varying security and privacy requirements, such as healthcare, finance, and government sectors.
- Facilitate, where possible, the integration of the framework with existing legacy systems and third-party security or monitoring tools to promote seamless adoption within diverse organizational environments.

1.4. Literature Review/Existing Solutions

2. Conventional Security Information and Event Management (SIEM) systems, while indispensable for threat detection and response, exhibit several limitations when evaluated through the lens of modern data privacy and comprehensive security requirements. A primary concern is **privacy neglect**; traditional SIEMs are engineered predominantly to identify and mitigate security threats, often overlooking the privacy implications of the vast amounts of log data they ingest and analyze. This oversight can lead to significant non-compliance issues with contemporary data protection mandates such as GDPR, HIPAA, and CCPA.
3. A key deficiency is **ineffective anonymization**. Most existing SIEM solutions lack robust, built-in mechanisms for the proper anonymization or pseudonymization of sensitive data elements commonly found in logs, like IP addresses, usernames, or other forms of PII. Consequently, logs containing raw, unprocessed sensitive information become highly vulnerable to insider threats, accidental exposure, data breaches, or misuse.
4. Furthermore, **limited encryption capabilities** are prevalent. While some systems may offer certain encryption features, these are often partial, typically focusing on either data at rest or data in transit, but seldom providing comprehensive end-to-end encryption. Moreover, when encryption mechanisms are present, they are not always designed to be scalable for the large volumes of log data generated by modern enterprises, potentially leading to performance bottlenecks and inefficiencies. This can result in logs being vulnerable to

interception during transmission. Solutions that do attempt encryption sometimes sacrifice performance, introducing unacceptable latency.

5. This leads to significant **compliance gaps**. Conventional tools tend to prioritize security-centric features like real-time threat detection and alerting over the nuanced requirements of data protection laws. The absence or inadequacy of integrated privacy-preserving techniques directly creates regulatory vulnerabilities, exposing organizations to substantial legal, financial, and reputational risks.
6. **Performance issues with scalability** also plague many existing solutions, especially when attempting to handle high-throughput environments that can generate millions of log events per second. If privacy measures like encryption or tokenization are retrofitted or present, they often introduce significant latency, degrading overall system performance and analytical utility.
7. Many SIEM systems exhibit a **focus on detection over prevention**. They are primarily designed to detect breaches after they occur but often fail to integrate proactive, preventive measures, such as securing logs themselves against unauthorized access or tampering from the outset. This reactive approach can leave critical vulnerabilities unaddressed until an incident materializes.
8. The **absence of modular privacy tools** is another common limitation. Traditional solutions frequently do not provide flexible, easily integrable, or customizable privacy tools like granular field-level encryption or sophisticated pseudonymization techniques. This rigidity prevents organizations from tailoring privacy mechanisms to their specific data types, risk profiles, and compliance requirements.
9. In terms of user experience for privacy assurance, there is often **inadequate visualization of privacy metrics**. While dashboards for threat analysis are common, similar visualization tools for monitoring privacy-related metrics—such as tokenization rates, encryption status across the pipeline, or adherence to specific compliance controls—are frequently lacking. This limits the ability of security and compliance teams to ensure and demonstrate that privacy is being maintained alongside security.
10. Consequently, adding robust privacy-enhancing features to conventional systems often results in **cost-intensive upgrades**, requiring significant investment in third-party integrations, specialized tools, or extensive custom development efforts. This high cost can

act as a deterrent for organizations, particularly smaller ones, from adopting comprehensive privacy measures.

10.1. Gap Analysis

The limitations identified in existing SIEM solutions translate into several critical gaps that this project aims to address:

- **Secure Log Transmission:**
    - **Gap:** Conventional SIEMs often lack comprehensive end-to-end encryption or implement it inefficiently, leaving logs vulnerable during transit and potentially impacting performance.
    - **Impact:** Increased vulnerability to data breaches and interception of sensitive log data during transmission.
- **Tokenization and Data Minimization:**
    - **Gap:** Many existing tools prioritize broad log collection but lack effective, integrated mechanisms for PII tokenization, pseudonymization, or field minimization, which are critical for GDPR/CCPA compliance.
    - **Impact:** Non-compliance with data protection regulations (GDPR, CCPA) and an elevated risk of sensitive data exposure from stored logs.
- **Privacy-Preserving Analytics:**
    - **Gap:** Current SIEM solutions predominantly focus on threat using analytics but often neglect to incorporate privacy-preserving measures, potentially exposing sensitive details within the data.
    - **Impact:** Lack of robust privacy-preserving, risking exposure of sensitive information while trying to derive security insights.
- **Real-Time Performance and Scalability with Privacy Measures:**
    - **Gap:** Existing systems often struggle to process large-scale log volumes in real-time, especially when computational overhead from security measures like encryption and tokenization is introduced.

  - **Impact:** Increased latency during log ingestion and processing, potentially hindering real-time threat detection and response capabilities.
- **Integration with Modern Log Management Tools:**
  - **Gap:** Many older or proprietary SIEM solutions lack compatibility or seamless integration with flexible, scalable, and modern log management frameworks like the Elastic Stack (Elasticsearch, Logstash).
  - **Impact:** Limited flexibility for organizations to scale, customize, or integrate their SIEM solutions with other modern data platforms.
- **Insight Visualization for Privacy and Security:**
  - **Gap:** Existing solutions often provide security-focused dashboards but lack robust visualization tools to demonstrate the effectiveness of privacy-preserving mechanisms or to monitor privacy compliance metrics alongside security events. There is a specific need to showcase data trends both before and after privacy measures (e.g., tokenization) are applied, using tools like Kibana.
  - **Impact:** Absence of clear dashboards makes it difficult to visualize the impact and status of privacy measures, and to assure stakeholders of compliance.

Summary of Identified Gaps Table:

Gap Area	**Impact**
Secure Log Transmission	Vulnerability to breaches during log transmission.
Tokenization and Minimization	Non-compliance with GDPR/CCPA and risk of data exposure.
Real-Time Performance	Latency during large-scale log ingestion and processing.
Tool Integration	Limited compatibility with scalable and modern log frameworks.
Insight Visualization	Absence of dashboards to visualize privacy measures.

10.2. Proposed Solution

This project introduces an advanced, privacy-compliant Security Information and Event Management (SIEM) framework designed to meticulously address the identified limitations of conventional solutions by holistically integrating robust security enhancements with comprehensive privacy measures. The proposed solution strategically leverages state-of-the-art, widely adopted open-source technologies—namely Elasticsearch for secure and scalable log storage and indexing, Logstash for flexible log ingestion and transformation, and Kibana for intuitive real-time visualization—to achieve an optimal balance between operational efficiency, analytical utility, and stringent data protection mandates.

Key Features of the Framework:

1. **Encrypted Data Pipelines:**
    - Implement comprehensive end-to-end encryption (E2EE) to secure log data throughout its lifecycle, particularly during transmission from agents to Logstash and from Logstash to Elasticsearch, primarily using robust protocols like TLS.
    - Employ or demonstrate mechanisms for database-level encryption for logs stored within Elasticsearch, ensuring that the data remains protected against unauthorized access even if the underlying storage system is compromised.
    - Introduce cryptographic hashing techniques (e.g., SHA-256) applied to log data to maintain its integrity, allowing for the detection of any unauthorized tampering or modifications.
2. **Real-Time Tokenization and Field Minimization:**
    - Sensitive data fields commonly found in logs (such as IP addresses, specific user identifiers, system names) are tokenized in real-time during the log ingestion phase using Logstash filters. This process replaces actual PII with non-sensitive tokens (pseudonyms), effectively anonymizing individuals while preserving the analytical utility of the data for pattern analysis and threat detection.
    - Apply data minimization principles by configuring log collection and processing to gather and store only the information essential for security analysis, thereby reducing the overall attack surface and exposure of sensitive details.

- Conceptually support reversible tokenization where necessary for compliance or specific investigative purposes, ensuring that such reversal is strictly controlled and accessible only to authorized personnel or systems.

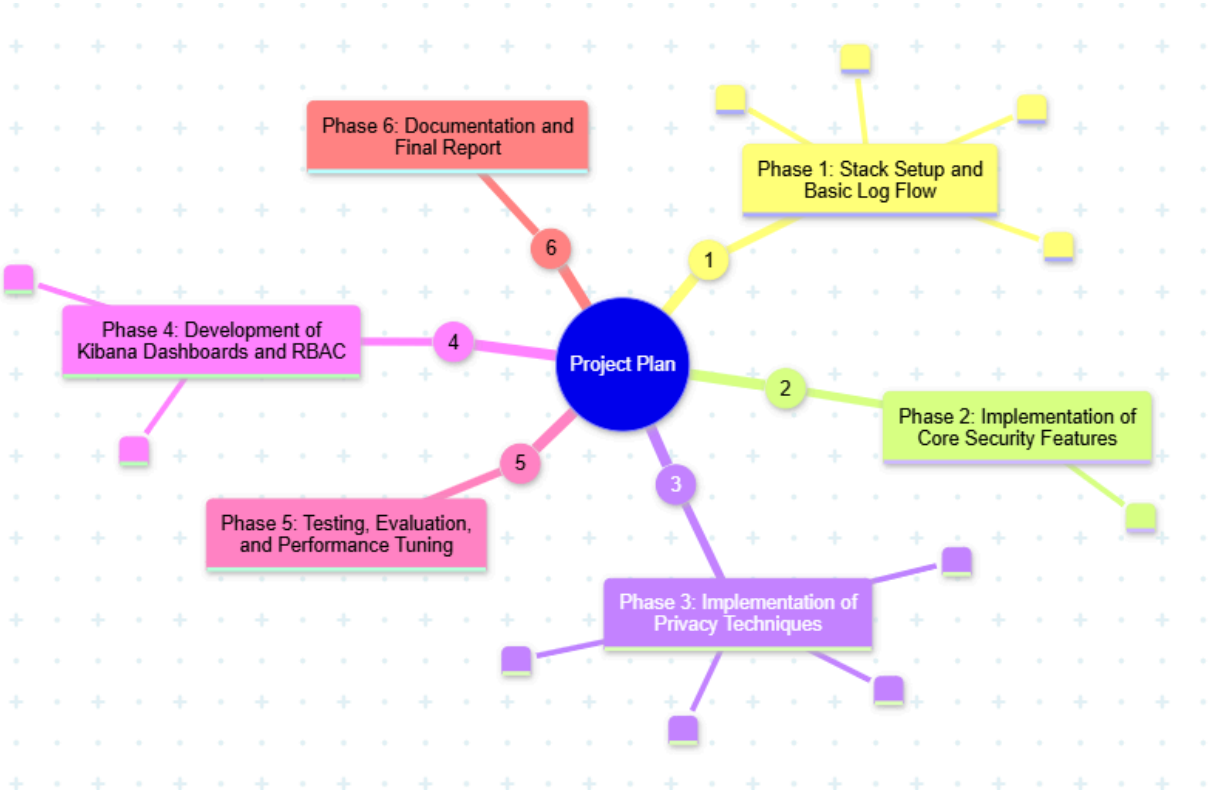
3. **Intuitive Visualization Using Kibana Dashboards:**
   - Develop user-friendly and insightful dashboards in Kibana for visualizing processed log data, real-time threat analytics, security alerts, and key compliance metrics.
   - Enable and configure granular Role-Based Access Control (RBAC) within Elasticsearch and Kibana to restrict dashboard visibility and data access to authorized users based on their roles and responsibilities, thereby safeguarding sensitive insights.
   - Include visualizations for privacy-related monitoring within the dashboards, such as tokenization rates, status of encryption across data pipelines, and metrics demonstrating adherence to specific compliance controls.
4. **Scalable Log Management:**
   - Design the system architecture to efficiently handle high-throughput logging environments, ensuring low-latency log ingestion, processing, and indexing, even at significantly large scales.
   - Utilize Elasticsearch's powerful distributed indexing and search capabilities to manage vast volumes of log data efficiently, while maintaining optimal performance for search, retrieval, and real-time analytics.
5. **Seamless Integration:**
   - Ensure broad compatibility with a variety of existing log sources, including web servers, application servers, databases, and network devices, by supporting standardized log formats (e.g., Syslog, JSON, CEF) and providing flexible parsing capabilities within Logstash.
   - Offer a modular framework design, allowing organizations to selectively integrate specific features (such as tokenization for certain log types, or particular encryption methods) based on their unique requirements, risk appetite, and existing infrastructure.
6. **Regulatory Compliance:**

  - Ensure the framework's design and implemented techniques facilitate adherence to major data protection regulations like GDPR, CCPA, and HIPAA by systematically integrating privacy-preserving mechanisms into the core log management workflow.
  - Provide capabilities or guidance for generating compliance reports directly from the system (e.g., through Kibana reporting or Elasticsearch aggregations) to demonstrate conformity with legal and regulatory standards to auditors and stakeholders.

Anticipated Benefits (from your document):

- **Enhanced Security:** Protect log data from unauthorized access, breaches, and tampering.
- **Improved Privacy:** Anonymize sensitive information to reduce risk and ensure compliance with privacy regulations.
- **Operational Efficiency:** Maintain high performance and low latency even under heavy workloads.
- **User Empowerment:** Offer tools for analysts to perform secure threat detection and compliance validation with ease.
- **Scalability and Flexibility:** Adapt to the needs of diverse industries, from finance to healthcare, ensuring widespread applicability.

10.3. Project Plan



10.3.1. Work Breakdown Structure

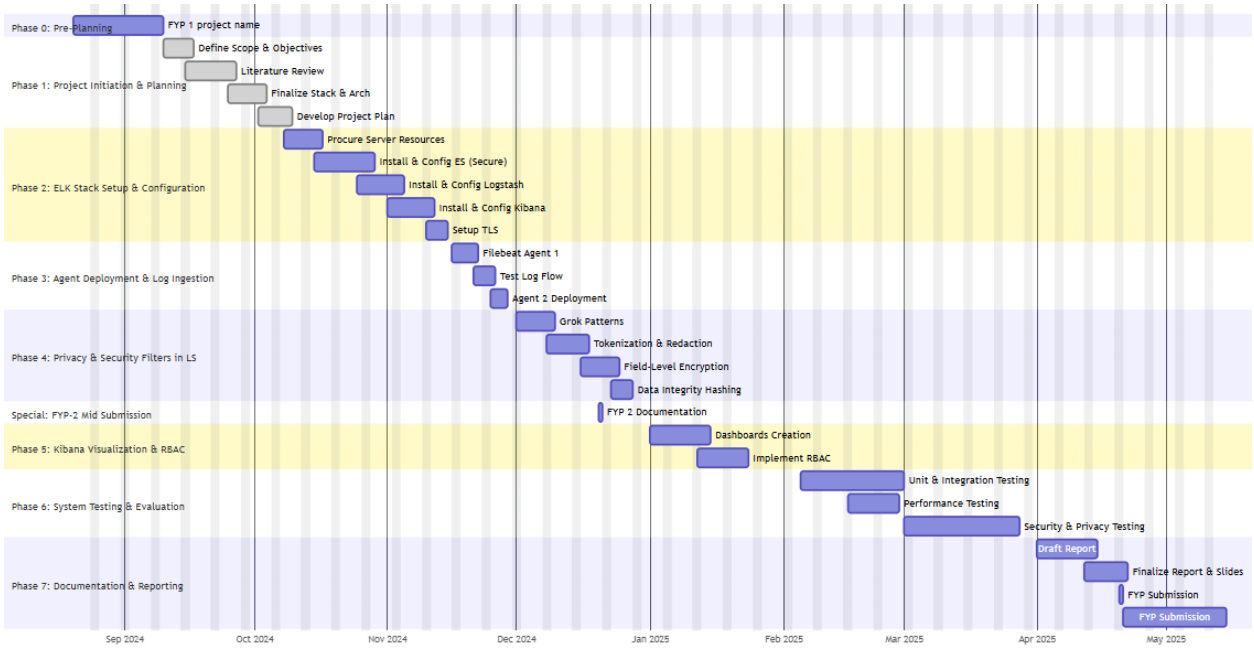
10.3.2. Roles & Responsibility Matrix

Roles & Responsibility Matrix

WBS ID	Task Description	Hameed Ullah (211086)	Hassan Imam (212074)	Furqan Rafique (211106)
1.0	**Project Initiation and Planning**			
1.1	Define Final Project Scope & Core Deliverables	R	S	S
1.2	Research & Analyze Solutions, Regulations, & Techniques	S	R	S
1.3	Finalize Technology Stack & System Architecture	R	S	R
1.4	Develop Detailed Project Plan (Tasks, Roles, Timeline)	S	S	R
2.0	**ELK Stack Core Infrastructure Setup (Main Server)**			
2.1	Server Preparation & Environment Configuration	R	S	S
2.2	Elasticsearch: Secure Installation & Configuration	R	S	S
2.3	Kibana: Secure Installation & Configuration	S	R	S
2.4	Logstash: Secure Installation & Base Pipeline Configuration	S	S	R
3.0	**Agent Deployment & Initial Log Ingestion**			
3.1	Agent Setup on Test Server (Filebeat)	R	S	S
3.2	Configure Agent for Secure Log Transmission to Logstash	S	R	S
3.3	Verify Basic End-to-End Log Flow	All	All	All
4.0	**Implementation of Log Processing for Security & Privacy (in Logstash)**			
4.1	Log Parsing and Structuring (Grok)	R	S	S
4.2	PII Tokenization & Anonymization (IPs, Usernames, Emails)	S	R	S
4.3	Data Integrity Implementation (Hashing)	S	S	R
4.4	(Conceptual) Field-Level Encryption & Noise Addition Research	All	All	All
5.0	**SIEM Functionality and User Interface**			
5.1	Develop Kibana Dashboards (Monitoring & Privacy Metrics)	R	S	S
5.2	Configure Role-Based Access Control (RBAC)	S	R	S
6.0	**System Testing and Evaluation**			
6.1	Functional & Use Case Testing	All	All	All
6.2	Performance & Scalability Testing	R	S	S
6.3	Security & Privacy Effectiveness Validation	S	R	S
7.0	**Project Documentation and Final Deliverables**			
7.1	Draft and Finalize FYP-II Report	All	All	All
7.2	Create Installation, Configuration, & User Manuals	R	S	S
7.3	Prepare Final Presentation & Project Demonstration	All	All	All
7.4	Submission of All Deliverables	All	All	All

R = Responsible
S = Support
All = All team members share responsibility

10.3.3. Gantt Chart



1.8 Report Outline

This report is structured to comprehensively document the design, development, and evaluation of the proposed SIEM enhancement framework, which focuses on achieving a balance between security and data privacy using the ELK Stack. The following chapters are included:

1. **Introduction**
   Introduces the background, motivation, objectives, and challenges associated with managing SIEM logs. It sets the context for the project by highlighting the conflict between security monitoring and data privacy in modern regulatory environments such as GDPR and CCPA.
2. **Literature Review**
   Reviews the evolution and limitations of conventional SIEM systems with respect to privacy compliance. It compares existing solutions and frameworks, identifies key shortcomings like lack of tokenization and encryption, and highlights the necessity for an enhanced privacy-compliant SIEM framework.
3. **System Requirements and Analysis**
   Details both functional and non-functional requirements for the proposed system. It also presents gap analysis, user roles, operating environment, and outlines architectural considerations for achieving secure and privacy-aware log processing.
4. **Proposed Solution**
   Describes the implementation of the ELK Stack-based framework, focusing on privacy-preserving techniques such as tokenization, encryption (TLS, E2EE), hashing (SHA-256), and data minimization. The chapter emphasizes modularity, scalability, and real-time processing.
5. **Project Plan and Work Breakdown Structure**
   Provides the project timeline, milestones, role assignments, and Gantt chart. It breaks the workload into manageable modules and tasks allocated among team members, aligning with the development lifecycle.
6. **Implementation and Development**
   Documents the tools, libraries, and configurations used to implement the framework. It includes Filebeat setup, Logstash filtering logic for tokenization and redaction, secure

Elasticsearch storage, and Kibana dashboard development for log visibility and compliance reporting.

7. **Experiments and Results**

   Presents the testing strategies applied, including use case testing, unit testing, stress and performance testing. Results focus on log throughput, latency, detection utility post-tokenization, and overall system responsiveness.

8. **Conclusion and Future Work**

   Summarizes the project outcomes, technical achievements, and lessons learned. It suggests future enhancements such as integration with AI-driven anomaly detection, field-level encryption, and potential for deployment in various industry sectors (e.g., healthcare, finance).

9. **References**

   Lists all scholarly articles, standards (GDPR, CCPA, HIPAA), open-source documentation, and tools consulted during research and implementation.

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1 Introduction

2.1.1 Purpose

The product whose software requirements are specified in this document is the **"Balancing Security and Privacy in SIEM Logs" framework**, Revision 1.0. This SRS document covers the entire scope of the project as defined, which includes the design, implementation, and evaluation of a SIEM enhancement framework built upon the Elastic Stack. The framework aims to integrate robust security and privacy-preserving measures into the log management lifecycle. The purpose of this Software Requirement Specifications (SRS) document is to provide a comprehensive and unambiguous definition of all functional and non-functional requirements for the "Balancing Security and Privacy in SIEM Logs" project. This document will serve as the primary reference for guiding the design, development, implementation, and testing of the SIEM framework. It aims to ensure that all system components and functionalities align with the project's core objectives of enhancing log security, maintaining data privacy through techniques such as tokenization and encryption, and ensuring compliance with relevant data protection standards (e.g., GDPR, CCPA), while preserving the operational utility of SIEM logs for threat detection and analysis.

2.1.2 Document Conventions

This document adheres to standard software engineering and cybersecurity terminologies. Key terms and concepts will be defined upon their first use or in a dedicated glossary if necessary. Requirements will be specified using clear, concise language and will be uniquely identifiable (e.g., FRX.Y for Functional Requirements). Standard Markdown formatting conventions for headings, lists, and emphasis (bolding for key terms) will be used throughout. The Elastic Stack components (Elasticsearch, Logstash, Kibana, Filebeat) will be referred to by their standard names. Privacy-enhancing techniques will be described using industry-accepted terminology. No special typographical conventions beyond standard Markdown are used to imply significance unless explicitly stated. Priorities for higher-level requirements are generally assumed to be inherited by their detailed sub-requirements unless a specific priority is assigned to a sub-requirement.

PII: Personally Identifiable Information

SIEM: Security Information and Event Management

ELK: Elasticsearch, Logstash, Kibana

TLS: Transport Layer Security

E2EE: End-to-End Encryption

RBAC: Role-Based Access Control

GDPR: General Data Protection Regulation

CCPA: California Consumer Privacy Act

HIPAA: Health Insurance Portability and Accountability Act

2.1.3 Intended Audience and Reading Suggestions

This SRS document is intended for a diverse audience involved in or affected by the project:

Project Team (Students: Hameed Ullah, Hassan Imam, Furqan Rafique): As the primary guide for development, implementation, and testing.

Project Supervisor (Dr. Hilmand Khan): For oversight, guidance, and evaluation of the project's progress and adherence to defined requirements.

Faculty Examiners/Reviewers: For assessing the project's scope, complexity, technical depth, and successful fulfillment of its objectives.

Future Developers/Maintainers: As a reference for understanding the system's functionalities and constraints if the project is extended or maintained.

Security Analysts/Compliance Officers (Conceptual Users): To understand the capabilities and limitations of the proposed framework in a real-world context.

This document is organized as follows: Section 2.1 (Introduction) provides context for this SRS. Section 2.2 (Overall Description) gives a high-level overview of the product. Section 2.3 (External Interface Requirements) details interactions with other entities. Section 2.4 (System Features) itemizes the product's functional requirements. Section 2.5 (Nonfunctional Requirements) lists quality attributes and constraints. Section 2.6 (Other Domain-Specific Requirements) covers other relevant requirements. Readers are advised to begin with Sections 2.1 and 2.2 for an overview, then proceed to sections relevant to their specific interests (e.g., developers focusing on 2.4 and 2.5, project managers on all sections).

2.1.4 Product Scope

The "Balancing Security and Privacy in SIEM Logs" project aims to develop a framework that significantly enhances traditional Security Information and Event Management (SIEM) capabilities by integrating robust security measures and privacy-preserving techniques. The purpose is to address the critical challenge of managing sensitive log data in compliance with modern data protection regulations without compromising the effectiveness of security monitoring and threat detection.

Key benefits and objectives include:

- **Enhanced Data Security:** Protecting log data from unauthorized access, breaches, and tampering through end-to-end encryption and integrity checks.
- **Improved Data Privacy:** Anonymizing or pseudonymizing Personally Identifiable Information (PII) within logs using techniques like tokenization and redaction to meet regulatory requirements (GDPR, CCPA, etc.).
- **Maintained Operational Utility:** Ensuring that processed logs remain valuable for security analysis, threat detection, and incident response.
- **Scalability and Performance:** Leveraging the Elastic Stack to build a scalable pipeline capable of handling significant log volumes with acceptable latency.
- **Compliance Facilitation:** Providing mechanisms and a framework that helps organizations align their SIEM practices with data protection standards.
- **Actionable Insights:** Enabling users to visualize and monitor both security events and privacy metrics through Kibana dashboards.

This project does not aim to build a new SIEM product from scratch but rather to provide a configurable reference implementation and set of best practices using the Elastic Stack (Elasticsearch, Logstash, Kibana) and Beats agents (Filebeat). The framework will demonstrate practical approaches to tokenization, redaction, secure transmission, and integrity verification within a SIEM context. (Reference: Project Vision and Scope document, if available, or Chapter 1 of the main FYP report for broader project goals).

2.2 Overall Description

2.2.1 Product Perspective

The "Balancing Security and Privacy in SIEM Logs" framework is envisioned as an enhancement layer and a practical implementation guideline for modern SIEM architectures. It is not a standalone, new SIEM product from scratch but rather a configurable system built upon the widely adopted Elastic Stack (Elasticsearch, Logstash, Kibana) and Beats agents (Filebeat).

The product operates by ingesting logs from various sources, processing them through a Logstash pipeline where security and privacy transformations occur, storing them securely in Elasticsearch, and enabling analysis and visualization through Kibana. It aims to demonstrate how security operations can be maintained effectively while adhering to stringent privacy requirements. The framework serves as a reference implementation that can be adapted and extended by organizations seeking to improve their SIEM posture regarding data security and privacy. It interfaces with log-generating systems (via agents) and provides an interface for security analysts and compliance officers (via Kibana). A simplified diagram showing the major components would depict agents sending logs to Logstash, Logstash processing and sending to Elasticsearch, and Kibana querying Elasticsearch for user display.

2.2.2 User Classes and Characteristics

The primary users of this SIEM framework and their characteristics are:

1. **Security Analysts:**

- **Characteristics:** Technical users responsible for monitoring security events, detecting threats, investigating incidents, and performing forensic analysis. They require access to log data (both raw and processed, depending on authorization) and analytical tools.
- **Needs:** Real-time visibility into security events, powerful search and correlation capabilities, dashboards for trend analysis and anomaly detection, and an understanding of how privacy measures impact their analysis. This is a high-priority user class.

2. **Compliance Officers / Data Privacy Officers:**
   - **Characteristics:** Users responsible for ensuring the organization adheres to data protection regulations (GDPR, CCPA, HIPAA, etc.). They may not be deeply technical but need to understand and verify that privacy-preserving measures are in place and effective.
   - **Needs:** Ability to audit privacy controls, view reports or dashboards demonstrating compliance (e.g., effectiveness of tokenization/redaction), and understand how PII is handled throughout the log lifecycle. This is a high-priority user class.
3. **System Administrators / SIEM Administrators:**
   - **Characteristics:** Technical users responsible for deploying, configuring, maintaining, and securing the SIEM infrastructure (Elasticsearch, Logstash, Kibana, agents).
   - **Needs:** Clear installation and configuration guides, tools for monitoring system health and performance, ability to manage SSL certificates, configure user roles and access (RBAC), and update processing pipelines. This is a high-priority user class for system operation.
4. **Developers (of the SIEM Framework - i.e., the project team):**
   - **Characteristics:** Technical users responsible for designing, building, and testing the framework's components, especially the Logstash processing logic and Kibana visualizations.
   - **Needs:** Access to all components for development, testing environments, and detailed understanding of the underlying technologies.

2.2.3 Operating Environment

The SIEM log security and privacy framework is designed to operate in the following environment:

- **Server-Side (Main ELK Stack):**
  - **Operating System:** Ubuntu Linux (e.g., Ubuntu Server 20.04 LTS or newer, as used in the project's DigitalOcean setup).
  - **Core Software:**
    - Elasticsearch 8.x
    - Logstash 8.x
    - Kibana 8.x
  - **Hardware:** Sufficient CPU (e.g., 2+ vCPUs), RAM (e.g., 8GB+ recommended for ELK), and disk space (dependent on log volume and retention, e.g., 50GB+ starting) on the server hosting the ELK stack.
  - **Network:** IP connectivity, with necessary ports open (e.g., 9200 for Elasticsearch, 5601 for Kibana, 5044 for Logstash Beats input). Secure communication via TLS is a core requirement.
- **Agent-Side (Log Source Machines):**
  - **Operating System:** Linux distributions (e.g., Ubuntu, Kali Linux) capable of running Filebeat 8.x. The framework can be conceptually extended to Windows agents.
  - **Core Software:** Filebeat 8.x (or other compatible Beats/Elastic Agent).
  - **Network:** IP connectivity to the Logstash server on the configured port (e.g., 5044) with TLS.
- **User Access:**
  - Users (Security Analysts, Compliance Officers, Administrators) will access Kibana via a standard web browser (e.g., Chrome, Firefox, Edge) over HTTP or HTTPS (if Nginx or Kibana SSL is configured).

2.2.4 Design and Implementation Constraints

The design and implementation of the framework are subject to the following constraints:

1. **Technology Stack:** The core system must be implemented using the Elastic Stack (Elasticsearch, Logstash, Kibana) and Filebeat for log shipping, primarily version 8.x.
2. **Open Source Preference:** Preference will be given to open-source tools and libraries where feasible, aligning with the nature of the Elastic Stack.
3. **Security Standards:**
    - Encryption in transit must use TLS 1.2 or higher.
    - Hashing for tokenization and integrity should use strong algorithms like SHA-256.
    - Consideration for AES-256 for any conceptual field-level encryption.
4. **Performance:** While implementing security and privacy measures, the system should strive to maintain reasonable performance for log ingestion and querying. The goal is timely processing for near real-time analysis.
5. **Development Environment:** The project will be developed and tested on Ubuntu Linux environments, primarily using DigitalOcean cloud servers.
6. **Project Timeline:** All core functionalities must be implemented and documented within the academic timeline of the Final Year Project.
7. **No Custom Agent Development:** The project will rely on existing Elastic agents (Filebeat) and will not involve developing new log shipping agents from scratch.
8. **Focus on Logstash for Transformations:** Privacy-preserving transformations (tokenization, redaction) will primarily be implemented within Logstash pipelines.
9. **Regulatory Policies:** The design must consider principles from GDPR and CCPA regarding PII handling, pseudonymization, and data minimization.
10. **Programming Standards:** Standard Logstash configuration syntax and best practices for filter development will be followed.

2.2.5 Assumptions and Dependencies

The successful design and implementation of this project rely on the following assumptions and dependencies:

1. **Availability of Compatible Software:** Stable and compatible versions of Elasticsearch, Logstash, Kibana, and Filebeat (8.x series) are available for download and installation on Ubuntu.

2. **Sufficient Server Resources:** The host server(s) for the ELK stack and agent machines have adequate CPU, RAM, and disk space for the intended scale of testing and operation as defined in 2.2.3.
3. **Network Connectivity:** Reliable IP network connectivity exists between agent machines, the Logstash server, the Elasticsearch server, and the Kibana server, with necessary ports open through firewalls (as detailed in 2.3.4).
4. **Log Format Accessibility and Parsability:** Log sources generate logs in formats that are either standard (e.g., syslog) or can be parsed effectively using Logstash's Grok filter or other parsing mechanisms to extract relevant fields.
5. **Understanding of Log Content for PII:** A general understanding of the types of sensitive information (PII) likely to be present in common system logs (e.g., IP addresses, usernames, email addresses) is assumed for designing effective redaction and tokenization rules.
6. **SSL/TLS Certificate Management Capability:** The project team has the capability to generate and manage SSL/TLS certificates (e.g., self-signed for development/testing) for securing communication between all components.
7. **Basic Linux Administration Skills:** The project team possesses the necessary skills for installing, configuring, troubleshooting, and managing software and services on Ubuntu Linux servers.
8. **Client for Testing:** At least one client machine (e.g., another Ubuntu VM) is available to install Filebeat and generate diverse test logs to validate all implemented features.
9. **Logstash Plugin Availability:** Required Logstash plugins (beats input, elasticsearch output, mutate, grok, fingerprint, date, ruby) are available and functional within the chosen Logstash version.

2.3 External Interface Requirements

2.3.1 User Interfaces

Primary User Interface (Kibana):

The Kibana web interface (version 8.x) will be the primary UI.

- **Logical Characteristics:** It shall provide a graphical interface for log searching, data discovery, creation and viewing of visualizations (graphs, charts, tables, maps), and dashboards. It will also be used for administrative tasks like managing index patterns, saved objects, and potentially user roles (if not done via API).
- **GUI Standards**: Will adhere to the standard Kibana user interface design and navigation.
- **Screen Layout**: Standard Kibana layouts for Discover, Visualize, Dashboard, and Stack Management sections will be used.
- **Error Message Display**: Kibana's standard error message display mechanisms will be utilized.
- **Software Components**: Kibana itself.

Command Line Interface (CLI):

- **Logical Characteristics**: Administrators will use standard Linux shell CLIs to manage the services (Elasticsearch, Logstash, Kibana, Filebeat) using systemctl, edit configuration files using text editors (e.g., nano), and run diagnostic commands (e.g., curl for API checks, filebeat test config).
- **Software Components:** Standard Linux shell (e.g., bash), systemctl, text editors, curl.

2.3.2 Hardware Interfaces

- Network Interface Cards (NICs): Standard Ethernet NICs (e.g., 1 Gbps) are required on all servers and agent machines for TCP/IP communication.
- Storage Devices: Standard server-grade HDDs or SSDs for OS, application software, and Elasticsearch data storage. The specific capacity will depend on log volume and retention policies.

- No other direct specialized hardware interfaces are anticipated for this project.

2.3.3 Software Interfaces

- Filebeat to Logstash: Filebeat (version 8.x) will send data to Logstash (version 8.x) using the Beats protocol over a TLS-secured TCP connection. Data will be in the Beats event format.
- Logstash to Elasticsearch: Logstash (version 8.x) will send processed data to Elasticsearch (version 8.x) using the Elasticsearch Bulk API over an HTTPS connection. Data will be in JSON format.
- Kibana to Elasticsearch: Kibana (version 8.x) will communicate with Elasticsearch (version 8.x) using the Elasticsearch REST API over an HTTPS connection for querying data, managing indices, and other administrative tasks.
- Operating System: All components (Elasticsearch, Logstash, Kibana, Filebeat) will run on Ubuntu Linux (e.g., 20.04 LTS or newer) and will interface with the OS for process management, file system access, networking, and system calls.
- Log Sources to Filebeat: Filebeat will read log data from standard log files (e.g., /var/log/syslog, /var/log/auth.log) using file system interfaces provided by the OS on agent machines.

2.3.4 Communications Interfaces

Filebeat -> Logstash:

- **Protocol:** Beats protocol over TCP, encapsulated within TLS 1.2+.
- **Port:** Configurable, default 5044 on Logstash server.
- **Security:** TLS encryption, server-side certificate on Logstash, CA certificate on Filebeat for server verification.

Logstash -> Elasticsearch:

- **Protocol:** HTTPS (HTTP over TLS 1.2+).
- **Port:** Configurable, default 9200 on Elasticsearch server.
- **Security:** TLS encryption, server-side certificate on Elasticsearch, CA certificate on Logstash for server verification. Authentication via username/password or API key.

Browser -> Kibana:

- **Protocol:** HTTP or HTTPS (if Kibana SSL is enabled or via Nginx reverse proxy).
- **Port:** Configurable, default 5601 on Kibana server.
- **Security:** If HTTPS, standard web browser TLS mechanisms.

Kibana -> Elasticsearch:

- **Protocol:** HTTPS (HTTP over TLS 1.2+).
- **Port:** Configurable, default 9200 on Elasticsearch server.
- **Security:** TLS encryption, server-side certificate on Elasticsearch, CA certificate on Kibana for server verification. Authentication via built-in user (kibana_system).

Elasticsearch Node-to-Node (Transport Layer):

- **Protocol:** Custom Elasticsearch transport protocol over TCP, encapsulated within TLS 1.2+.
- **Port:** Configurable, default range 9300-9400.
- **Security:** Mutual TLS encryption and authentication using transport layer certificates.

2.4 System Features

This section details the key functional features of the SIEM log security and privacy framework.

2.4.1 System Feature 1: Secure Log Ingestion and Transmission

2.4.1.1 Description and Priority

Description: The system shall securely collect logs from distributed agents (Filebeat) and transmit them to a central Logstash instance. All communication channels used for log transmission must be encrypted using TLS.

Priority: High.

2.4.1.2 Stimulus/Response Sequences

Stimulus 1: Administrator configures Filebeat on an agent machine with Logstash host details and SSL/TLS parameters.

Response 1: Filebeat attempts to establish a TLS connection to Logstash.

Stimulus 2: A new log event is generated on the agent machine being monitored by Filebeat.

Response 2: Filebeat detects, reads, and transmits the log event over the established TLS-encrypted connection to the Logstash Beats input. Logstash receives the encrypted event.

2.4.1.3 Functional Requirements

- REQ-SF1-1: Filebeat shall monitor specified log files or other configured input sources on agent machines.

- REQ-SF1-2: Filebeat shall be configurable to establish a TLS 1.2+ encrypted connection to the Logstash server.

- REQ-SF1-3: Filebeat shall use a provided CA certificate to verify the identity of the Logstash server during the TLS handshake.

- REQ-SF1-4: Logstash shall listen for incoming Beats connections on a specified TCP port (e.g., 5044).

- REQ-SF1-5: Logstash Beats input shall be configurable with an SSL/TLS server certificate and private key to enable encrypted communication.

- REQ-SF1-6: The system shall reject connections from Filebeat agents if the TLS handshake fails due to certificate validation issues (unless explicitly configured for a less secure mode for testing).

2.4.2 System Feature 2: Log Processing for Privacy and Enrichment

2.4.2.1 Description and Priority

Description: Logstash shall process incoming raw logs to parse them into structured data, apply privacy-preserving transformations (tokenization of PII, redaction of sensitive patterns), enrich events with metadata, and generate integrity hashes.

Priority: High.

2.4.2.2 Stimulus/Response Sequences

Stimulus: Logstash's Beats input receives a new log event from a Filebeat agent.

Response: The event passes through the Logstash filter pipeline:

- (If applicable) Grok filter attempts to parse the raw log message into structured fields based on predefined patterns.

- Mutate/Fingerprint filters identify and tokenize specific PII fields (e.g., source IP address, usernames if parsed), storing tokens in new fields.
- Mutate filter uses gsub to redact PII patterns (e.g., email addresses, IPs within message text) from specified fields.
- Fingerprint filter calculates a cryptographic hash of the processed message content for integrity verification.
- The transformed and enriched event is passed to the output queue.

2.4.2.3 Functional Requirements

- REQ-SF2-1: Logstash shall provide a mechanism (e.g., Grok filter) to parse unstructured log messages into key-value pairs based on configurable patterns.

- REQ-SF2-2: Logstash shall tokenize specified sensitive fields (e.g., [source][address], parsed usernames) using a strong, one-way hashing algorithm (e.g., SHA256) with a configurable secret key, storing the resulting token in a distinct field.

- REQ-SF2-3: Logstash shall redact occurrences of predefined PII patterns (e.g., IPv4 addresses, IPv6 addresses, email addresses) within specified log fields (e.g., the main message field) by replacing them with a placeholder string (e.g., [REDACTED_IPV4]).

- REQ-SF2-4: Logstash shall generate a cryptographic hash (e.g., SHA256) of the primary log message content (after redaction/tokenization) and store this hash in a dedicated field (e.g., [event][log_integrity_hash]).

- REQ-SF2-5: Logstash shall support conditional application of filters based on event content (e.g., log source path, presence of certain fields).

- REQ-SF2-6: Administrators shall be able to configure and update the patterns for PII redaction, fields for tokenization, and secret keys used in hashing.

- REQ-SF2-7: (Conceptual) The system should allow for the future integration of field-level encryption for highly sensitive data using appropriate Logstash mechanisms (e.g., ruby filter with OpenSSL).

2.4.3. System Feature 3: Secure Log Storage and Output to Elasticsearch

2.4.3.1. Description and Priority:

- **Description:** Logstash shall securely transmit processed and transformed log events to an Elasticsearch instance for indexing, storage, and subsequent analysis. The connection to Elasticsearch must be encrypted and authenticated.
- **Priority:** High.

2.4.3.2. Stimulus/Response Sequences:

- **Stimulus:** A processed log event is available in Logstash's output queue.
- **Response:** Logstash's Elasticsearch output plugin establishes a TLS-encrypted connection to the configured Elasticsearch host(s), authenticates using provided credentials, and sends the event (or a batch of events) for indexing. Elasticsearch acknowledges receipt and indexing.

2.4.3.3. Functional Requirements:

- REQ-SF3-1: Logstash shall connect to the Elasticsearch cluster using HTTPS (TLS 1.2+).
- REQ-SF3-2: Logstash shall authenticate to Elasticsearch using configurable credentials (e.g., username and password).
- REQ-SF3-3: Logstash shall be configured with the CA certificate necessary to verify the identity of the Elasticsearch server(s).
- REQ-SF3-4: Logstash shall send processed events to a dynamically named Elasticsearch index based on a configurable pattern (e.g., filebeat-%{+YYYY.MM.dd}).
- REQ-SF3-5: Elasticsearch shall index the received log events, making them available for search and analysis.
- REQ-SF3-6: (Conceptual) Elasticsearch should be configured for data-at-rest encryption to protect stored log data.

2.4.4. System Feature 4: Log Visualization, Search, and Monitoring via Kibana

1.4.4.1. Description and Priority:

- **Description:** Authorized users (Security Analysts, Compliance Officers, Administrators) shall be able to use Kibana to search, discover, analyze, and visualize the processed log data stored in Elasticsearch for security monitoring, incident investigation, and compliance verification.
- **Priority:** High.

2.4.4.2. Stimulus/Response Sequences:

- **Stimulus 1:** User navigates to Kibana Discover and enters a search query.
- **Response 1:** Kibana queries Elasticsearch and displays matching log events, including original, tokenized, and redacted fields as per user authorization and data view configuration.
- **Stimulus 2:** User opens a pre-built or custom dashboard in Kibana.
- **Response 2:** Kibana queries Elasticsearch and populates the dashboard with relevant visualizations, metrics, and log summaries.

2.4.4.3. Functional Requirements:

- REQ-SF4-1: Kibana shall connect securely to the Elasticsearch cluster to retrieve and display log data.
- REQ-SF4-2: Users shall be able to create and manage Data Views (Index Patterns) in Kibana to define how log data from specific Elasticsearch indices (e.g., filebeat-*) is accessed and displayed.
- REQ-SF4-3: Users shall be able to perform free-text searches and field-based queries on log data using Kibana Query Language (KQL) within the Discover application.
- REQ-SF4-4: The system shall allow administrators or analysts to create custom visualizations (e.g., charts, graphs, tables, maps) based on the log data.
- REQ-SF4-5: The system shall allow for the creation of custom dashboards in Kibana that aggregate multiple visualizations to provide overviews of:
  - Security events, alerts, and trends.
  - The effectiveness of privacy measures (e.g., counts of tokenized IPs, redacted emails).
  - Log integrity hash information (if applicable for display).

- REQ-SF4-6: Kibana shall support filtering, sorting, and aggregation of log data to facilitate analysis.
- REQ-SF4-7: Users should be able to inspect individual log documents to view all indexed fields, subject to RBAC.

2.4.5. System Feature 5: Role-Based Access Control (RBAC)

2.4.5.1. Description and Priority:

- **Description:** The system shall enforce Role-Based Access Control (RBAC) to restrict access to data, visualizations, dashboards, and administrative functionalities within Elasticsearch and Kibana based on predefined user roles.
- **Priority:** Medium-High.

2.4.5.2. Stimulus/Response Sequences:

- **Stimulus:** A user logs into Kibana and attempts to access a specific data view, dashboard, or administrative feature.
- **Response:** Elasticsearch/Kibana verifies the user's assigned roles and their associated privileges. Access to the requested resource or functionality is granted only if permitted by their role(s); otherwise, access is denied or the resource is hidden.

2.4.5.3. Functional Requirements:

- REQ-SF5-1: System administrators shall be able to define distinct user roles within Elasticsearch/Kibana (e.g., "SecurityAnalyst_Tier1", "ComplianceAuditor", "SIEM_Admin").
- REQ-SF5-2: System administrators shall be able to assign granular privileges to these roles, including but not limited to:
  - Read-only or read/write access to specific Elasticsearch indices or data streams (e.g., allowing analysts to read filebeat-* but not modify).
  - Permissions to view, create, edit, or delete specific Kibana dashboards, visualizations, and saved objects.
  - Access to specific Kibana applications (Discover, Visualize, Dashboard, Stack Management).
  - Cluster-level administrative privileges for Elasticsearch (e.g., manage_index_templates, monitor).

- REQ-SF5-3: Users shall only be able to view and interact with data and Kibana objects for which their assigned role(s) grant permission.
- REQ-SF5-4: System administrators shall be able to create users and assign them to one or more predefined roles.
- REQ-SF5-5: (Conceptual) Sensitive fields (e.g., original PII before tokenization, if stored separately) should be restrictable to only highly privileged roles.

2.5. Nonfunctional Requirements

This section outlines the quality attributes and constraints of the system.

2.5.1. Performance Requirements

- **Log Ingestion Latency:** The end-to-end latency from log generation on an agent to its availability for search in Kibana should be minimized to support near real-time monitoring. For typical system logs under moderate load on the specified hardware, this should ideally be within tens of seconds.
- **Log Processing Throughput:** Logstash should be capable of processing a sustained rate of incoming logs from multiple agents without significant queuing or event loss. (Specific target: TBD based on testing, e.g., X events per second per Logstash CPU core).
- **Kibana Query Response Time:** Common Kibana searches and dashboard loads should complete within 5-10 seconds for recent data (e.g., last 24 hours). More complex aggregations or queries over longer time ranges may take longer.
- **Resource Utilization:** The ELK components and Filebeat agents should operate within reasonable CPU (e.g., average < 75% under normal load), memory (within JVM heap limits and system RAM), and disk I/O limits on the provisioned hardware.

2.5.2. Safety Requirements

- While not a safety-critical system in terms of physical harm, data safety is paramount.
- The system must not cause loss or corruption of log data during transmission, processing, or storage due to system errors. (Covered by Reliability).

- Privacy-preserving mechanisms must not inadvertently destroy or irretrievably alter data that is critical for legitimate security investigations by authorized personnel, unless such alteration is an explicit goal (e.g., full redaction).

2.5.3. Security Requirements

Confidentiality:

- All log data containing potential PII must be encrypted in transit between all components (Agent -> Logstash -> Elasticsearch -> Kibana -> Browser) using TLS 1.2+.
- Sensitive data within logs must be tokenized or redacted as per defined policies before being made widely accessible for general analysis.
- Access to raw, unredacted PII (if stored or temporarily available) must be strictly controlled via RBAC.
- All system credentials (Elasticsearch passwords, Logstash keys, API keys) must be stored and managed securely. Default credentials must be changed.

Integrity:

- Log data integrity during transit shall be ensured by TLS.
- A cryptographic hash of processed log messages shall be generated and stored to allow for later verification of message content integrity post-processing.
- Configuration files for all components must be protected against unauthorized modification (standard OS file permissions).

Availability:

- The core ELK services (Elasticsearch, Logstash, Kibana) should be configured to restart automatically on failure (via systemd).
- (Conceptual for FYP) In a production environment, Elasticsearch would be deployed as a cluster for high availability and data redundancy. Logstash can also be scaled horizontally.

Authentication:

- All access to Kibana shall require user authentication.
- All API access to Elasticsearch shall require authentication (API key or username/password).
- Logstash shall authenticate to Elasticsearch.

Authorization:

- RBAC shall be enforced in Kibana and Elasticsearch to ensure users can only access data and features according to their defined roles.

2.5.4. Usability Requirements

- **Kibana Dashboards:** Dashboards designed for security monitoring and privacy compliance should be clear, intuitive, and provide actionable information with minimal need for users to write complex queries for common tasks.
- **Configuration Management:** Configuration of Filebeat agents and Logstash pipelines (especially filter logic) should be well-documented and manageable by administrators with a reasonable level of technical expertise in these tools.
- **Error Diagnostics:** The system components (Filebeat, Logstash, Elasticsearch, Kibana) must provide sufficiently detailed logs to facilitate troubleshooting by administrators. Error messages presented to users in Kibana should be as informative as possible.

2.5.5. Reliability Requirements

- **Data Durability:** Elasticsearch, even as a single node for this project, should ensure data is written to disk durably. (In production, replication across nodes would be used).
- **Log Processing Robustness:** Logstash pipelines must be designed to handle common variations in log formats or unexpected data without crashing. Malformed messages should ideally be tagged or routed to a dead-letter queue (DLQ) rather than halting the pipeline.
- **Service Uptime:** Core services (Elasticsearch, Logstash, Kibana) should aim for high uptime. Systemd service configurations should ensure automatic restarts on failure.

2.5.6. Maintainability/Supportability Requirements

- **Configuration Modularity:** Logstash configurations, if they become complex, should be organized logically (e.g., using comments to delineate sections for parsing, tokenization, redaction, etc., or potentially separate .conf files for very distinct processing stages if appropriate).
- **System Logging:** All components must generate sufficient operational logs to monitor their health, performance, and to aid in diagnosing issues.
- **Documentation:** The project will be supported by comprehensive documentation (this SRS, design documents, installation/user manuals) to facilitate understanding, operation, and future maintenance or extension.

- **Version Control (for configurations):** All custom configuration files (Logstash pipelines, Filebeat configurations) should ideally be managed under version control (e.g., Git) for tracking changes and rollback capabilities (this is a best practice, implementation depends on project team workflow).

2.5.7. Portability Requirements

- The core ELK Stack components and Filebeat are Java-based or compiled for multiple platforms. The primary deployment target for this project is Ubuntu Linux.
- Filebeat configurations for Linux system logs will be specific to Linux log paths but can be adapted for other Linux distributions.
- Logstash filter logic is generally platform-independent, but Grok patterns might need adjustment if log formats differ significantly on other OS types (e.g., Windows Event Logs).
- The overall framework concept is portable, but specific agent configurations and log parsing logic would need adaptation for non-Linux environments.

2.5.8. Efficiency Requirements

- **Logstash Filter Performance:** Logstash filters, particularly those involving complex regular expressions (in gsub or grok) or custom Ruby code, should be optimized to minimize their impact on CPU and memory usage, and to reduce processing latency per event.
- **Elasticsearch Indexing:** Index mappings in Elasticsearch (though largely managed by default templates in this project scope) should be appropriate for the data types to ensure efficient storage and search performance.
- **Data Storage:** While not a primary focus for optimization in this FYP, consideration should be given to data retention policies and index lifecycle management (ILM) in Elasticsearch for managing storage growth in a production scenario.

2.6. Other Domain-Specific Requirements

The domain of this project is **Cybersecurity Operations and Data Privacy Compliance** within the context of Security Information and Event Management (SIEM). Key domain-specific requirements include:

- **Understanding of Log Data Types:** The system must be designed to process and interpret various types of log data relevant to security monitoring, primarily focusing on

system logs (syslog, auth.log) from Linux environments for this project. Conceptual understanding should extend to other common log sources like web server logs, firewall logs, and application logs.

- **Knowledge of Personally Identifiable Information (PII):** The system's privacy measures (tokenization, redaction) must be designed with a clear understanding of what constitutes PII commonly found in logs, including but not limited to IP addresses, usernames, email addresses, specific hostnames, and potentially sensitive data within message payloads.
- **Adherence to Core Security Principles:** The framework must be designed and implemented embodying core information security principles such as defense-in-depth (multiple layers of security), principle of least privilege (for user access and service accounts), and secure-by-default configurations for all components.
- **Familiarity with Data Protection Compliance Standards:** The privacy techniques implemented (e.g., tokenization as pseudonymization, redaction for data minimization) should align with the fundamental principles and requirements of major data protection regulations such as GDPR, CCPA, and HIPAA.
- **Support for SIEM Use Cases:** The system, even with privacy transformations applied, must continue to support common SIEM use cases. This includes providing sufficient data for effective threat detection (through log analysis and correlation), security monitoring (via dashboards), and facilitating incident response by making relevant (though potentially anonymized) data accessible to analysts.
- **Awareness of Threat Landscape and Indicators of Compromise (IoCs):** While this project does not focus on developing new threat detection rules or AI models, the design of privacy measures should consider the types of threats SIEMs aim to detect. Care must be taken to ensure that privacy transformations do not completely obscure critical IoCs or patterns necessary for identifying malicious activity, striking a balance between privacy and the ability to detect threats.
- **Auditability of Privacy Measures:** The system should allow (conceptually, if not via specific reports) for the auditing of applied privacy measures. For example, being able to demonstrate that PII is being consistently tokenized or redacted as per policy.

Chapter 3

Use Case Analysis

Chapter 3: Use Case Analysis

This chapter delves into the Use Case Analysis for the "Balancing Security and Privacy in SIEM Logs" framework. It begins by presenting the overall Use Case Model, illustrating the interactions between various actors and the core functionalities of the system. Subsequently, detailed descriptions for each identified use case are provided, outlining their specific attributes, preconditions, flows, and postconditions. This analysis is crucial for understanding the system's intended behavior from a user's perspective and forms a basis for system design and testing, ensuring all key interactions and objectives are addressed.

3.1. Use Case Model

The Use Case Model for the "Balancing Security and Privacy in SIEM Logs" framework illustrates the key interactions between users, external systems (Log Sources, Agents), and the core components of the SIEM system itself (Logstash, Elasticsearch, Kibana). It highlights the main functionalities, such as secure log collection, data encryption, tokenization of sensitive information, real-time monitoring, role-based access control, integrity verification, log management, and compliance reporting. This model serves as a high-level blueprint of the system's capabilities and the roles various stakeholders play in its operation.

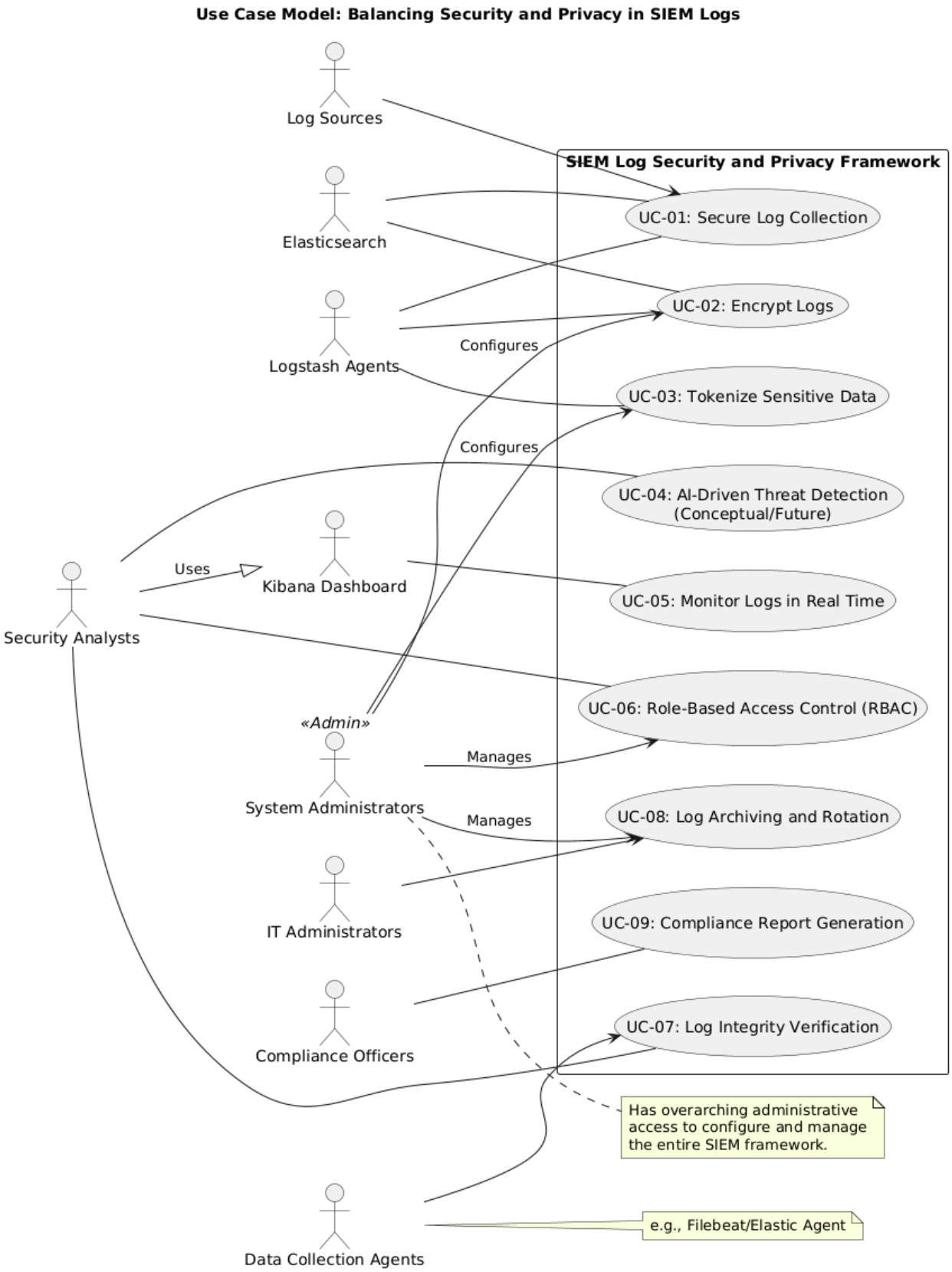


Figure 1: Use Case for SIEM

3.2. Use Cases Description

The following subsections detail each use case identified for the SIEM framework.

3.1.1. Use Case UC-01: Secure Log Collection

Title	Secure Collection of Log Data
Requirement	The system must collect log data from diverse sources (e.g., web servers, databases, OS) via agents, ensuring secure transmission to Logstash.
Rational	To centralize log data for analysis while protecting it from unauthorized access or tampering during the initial collection and transit phase.
Restriction or Risk	Agent misconfiguration leading to incomplete collection; network failures disrupting log flow; potential for agent compromise.
Dependency	Active log sources; correctly installed and configured agents (Filebeat); network connectivity; operational Logstash instance.
Priority	High

Table 1: Use Case UC-01: Secure Log Collection

UC-02: Encrypt Logs

Attribute	Details
Title	Log Data Encryption
Requirement	Encrypt all incoming logs before indexing or storing.
Rationale	Ensure confidentiality and regulatory compliance.
Risk	Key mismanagement; encryption latency.
Dependency	Key management system (KMS); configured Logstash.
Priority	High

Table 2: UC-02: Encrypt Logs

UC-03: Tokenize Sensitive Data

Attribute	**Details**
Title	Data Tokenization
Requirement	Replace PII/sensitive fields with tokens before storage.
Rationale	Protect user privacy and mitigate data exposure.
Risk	Incorrect token mapping; loss of data usability.
Dependency	Tokenization rules; Logstash filter plugins.
Priority	High

Table 3: UC-03: Tokenize Sensitive Data

UC-04: AI-Driven Threat Detection (Conceptual/Future)

Attribute	**Details**
Title	Predictive Threat Analytics
Requirement	Analyze logs using AI models to detect anomalies.
Rationale	Proactively identify potential threats.

Attribute	Details
Risk	Model accuracy; false positives/negatives.
Dependency	Trained AI models; log datasets.
Priority	Medium (Future/Conceptual)

Table 4: UC-04: AI-Driven Threat Detection (Future)

UC-05: Monitor Logs in Real Time

Attribute	Details
Title	Real-Time Log Monitoring
Requirement	Visualize and track log data through Kibana.
Rationale	Enable quick detection and response to incidents.
Risk	Dashboard overload; delays in large datasets.
Dependency	Kibana access; up-to-date indices.
Priority	High

Table 5: UC-05: Monitor Logs in Real Time

UC-06: Role-Based Access Control (RBAC)

Attribute	Details
Title	Access Control via Roles
Requirement	Define user roles and restrict access accordingly.
Rationale	Enforce least privilege; prevent data leaks.

Attribute	Details
Risk	Misconfigured roles; privilege escalation.
Dependency	Elasticsearch/Kibana user management.
Priority	High

Table 6: UC-06: Role-Based Access Control (RBAC)

UC-07: Log Integrity Verification

Attribute	Details
Title	Integrity Check of Logs
Requirement	Detect unauthorized modifications in log files.
Rationale	Ensure trustworthiness of logs for auditing.
Risk	Performance overhead; checksum failure.
Dependency	Hashing algorithms; periodic checks.
Priority	Medium

Table 7: UC-07: Log Integrity Verification

UC-08: Log Archiving and Rotation

Attribute	Details
Title	Archive and Rotate Logs
Requirement	Implement scheduled archiving and log rotation.
Rationale	Reduce storage load; retain logs for audits.
Risk	Missed rotations; storage overflow.

Attribute	Details
Dependency	Elasticsearch ILM policies; disk availability.
Priority	Medium

Table 8: UC-08: Log Archiving and Rotation

UC-09: Compliance Report Generation

Attribute	Details
Title	Generate Compliance Reports
Requirement	Compile log summaries for compliance audits.
Rationale	Meet GDPR, HIPAA, and other legal obligations.
Risk	Incomplete data; missed reporting intervals.
Dependency	Kibana report features; access to filtered logs.
Priority	High

Table 9: UC-09: Compliance Report Generation

Chapter 4

System Design

Chapter 4: System Design

4.1. Architecture Diagram

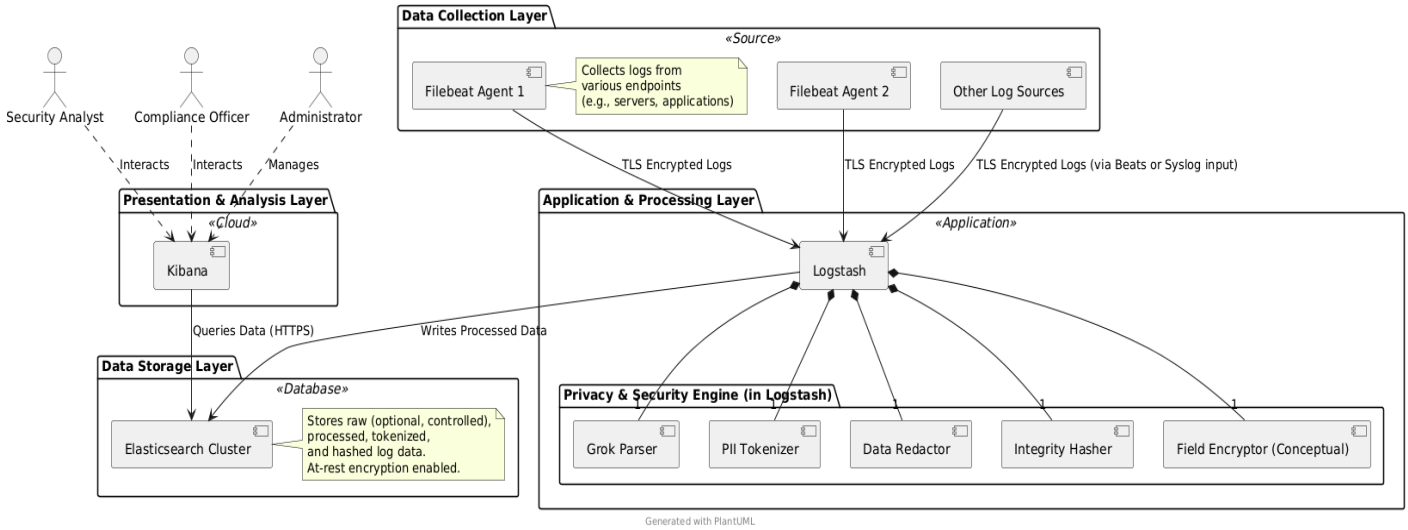


Figure 2: Architecture Diagram

4.2. Domain Model

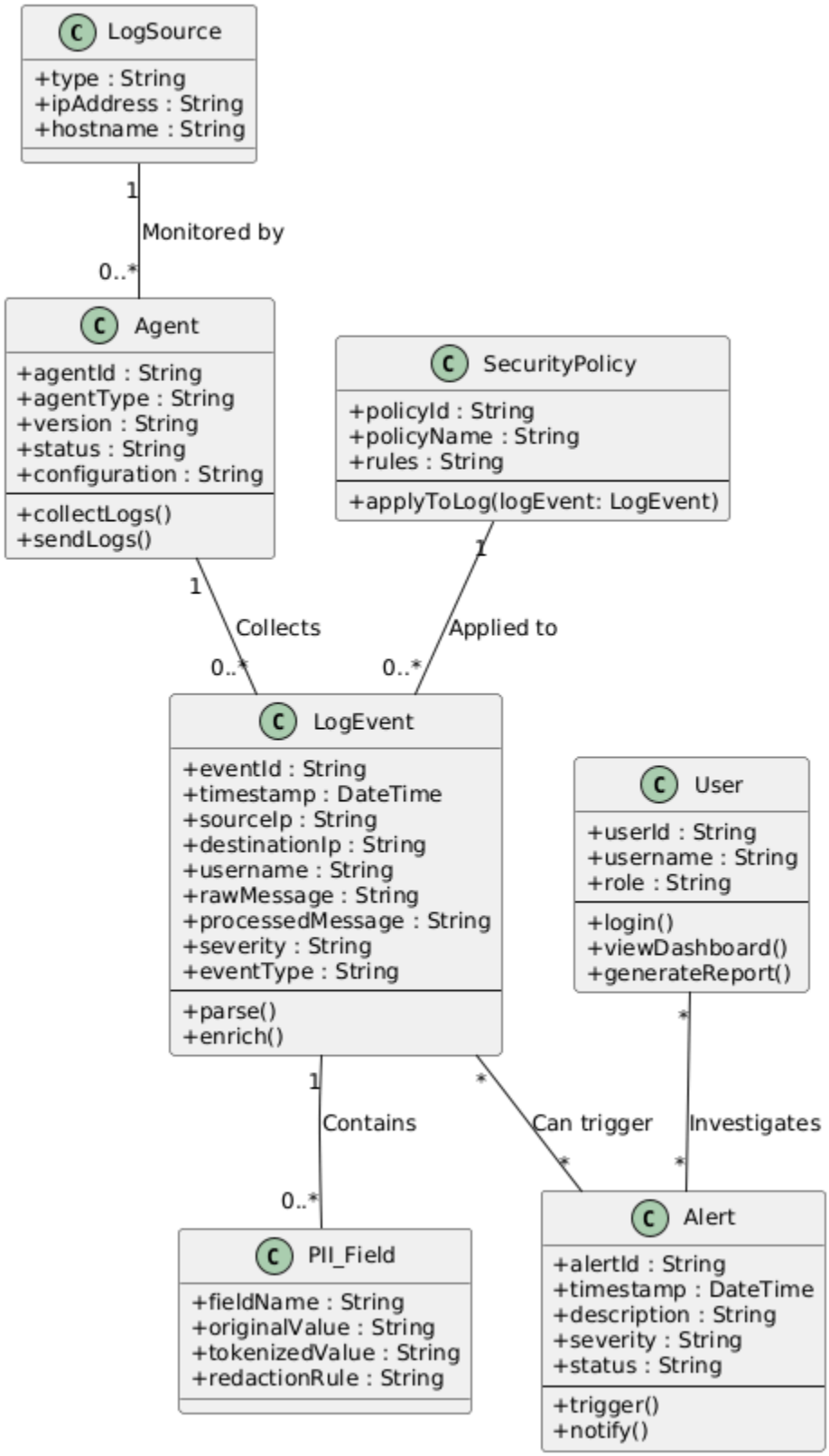


Figure 3: Doman Model Diagram

4.3. Entity Relationship Diagram with data dictionary

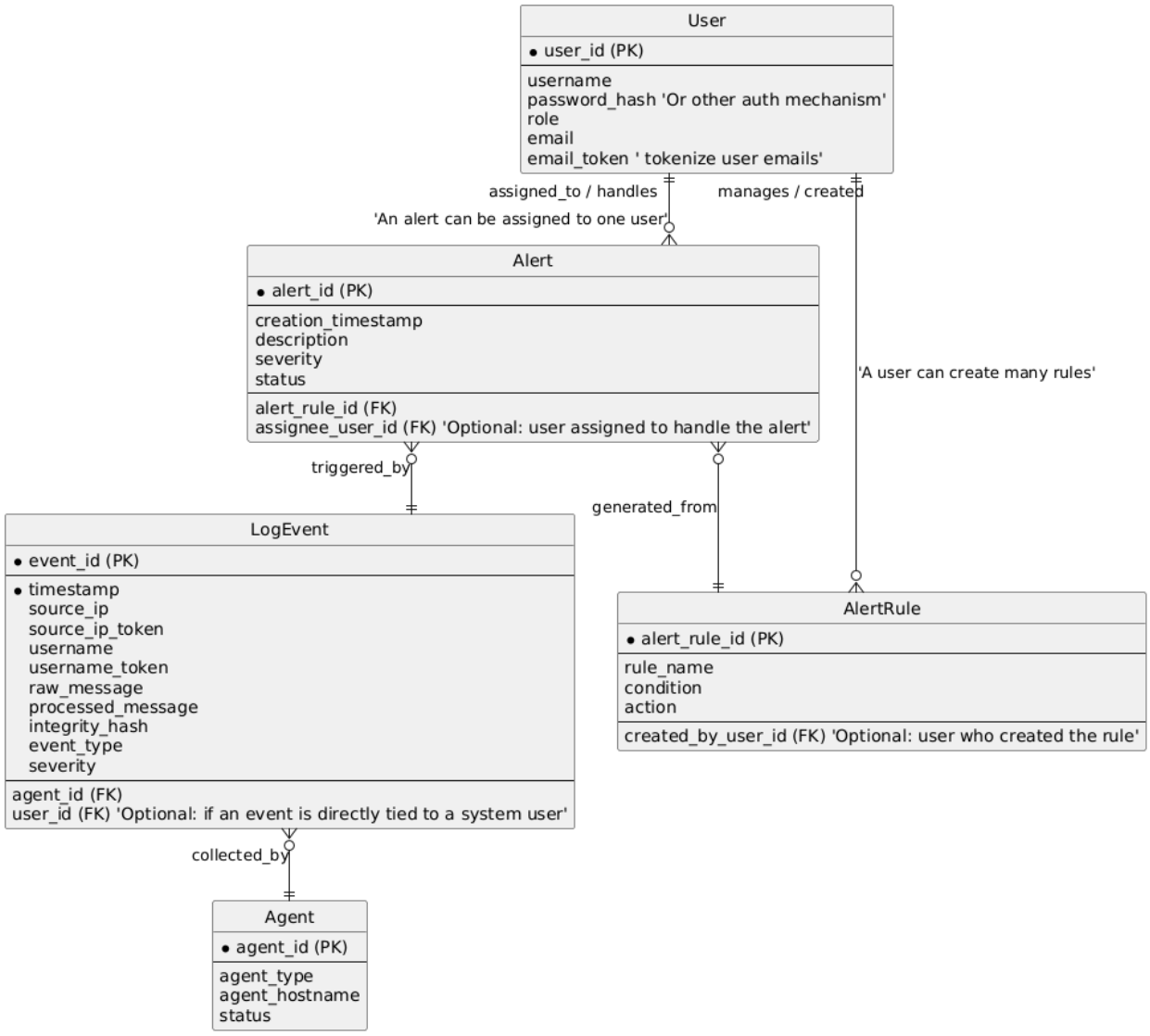


Figure 4: ER-Diagram

4.4. Class Diagram

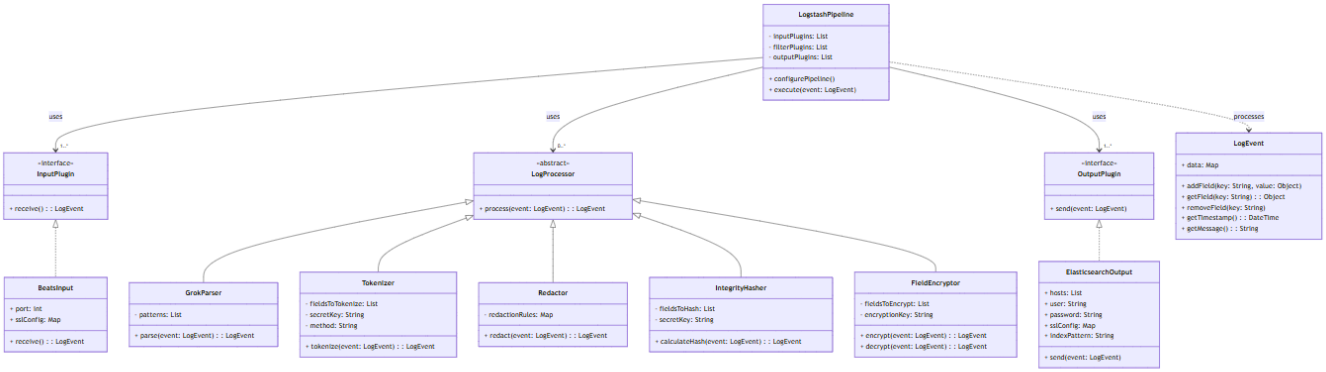


Figure 5: Class Diagram

4.5. Sequence / Collaboration Diagram

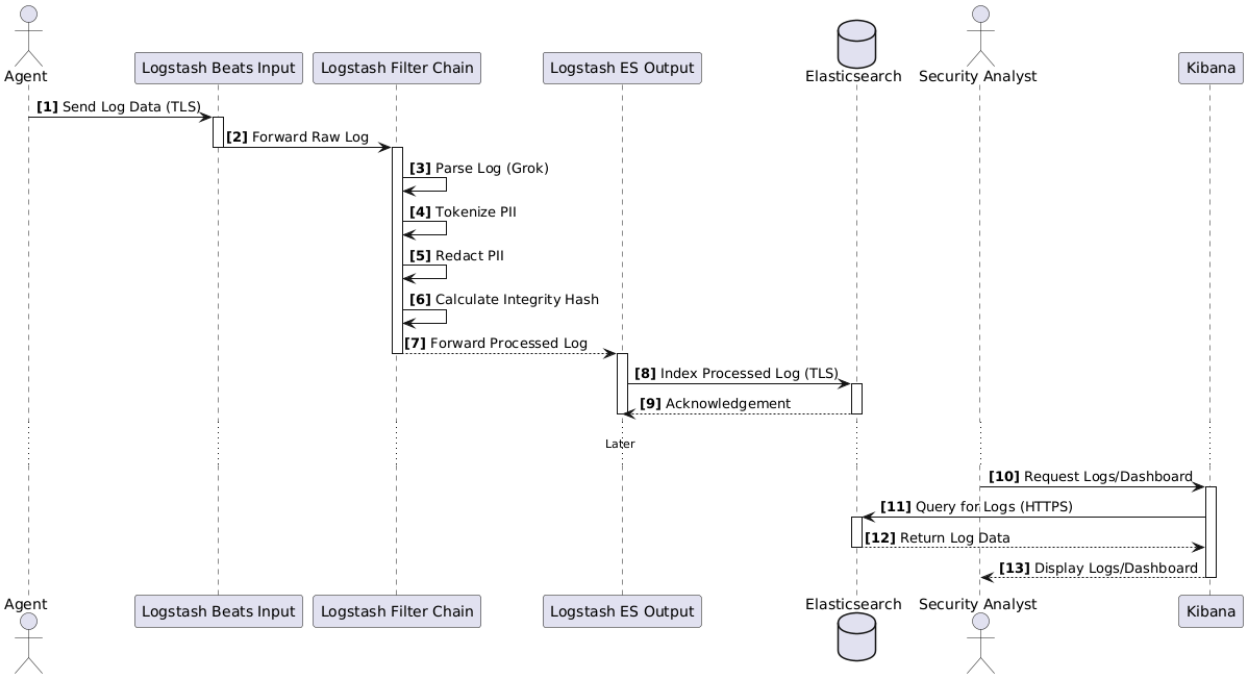


Figure 6: Sequence Diagram

4.6. Activity Diagram

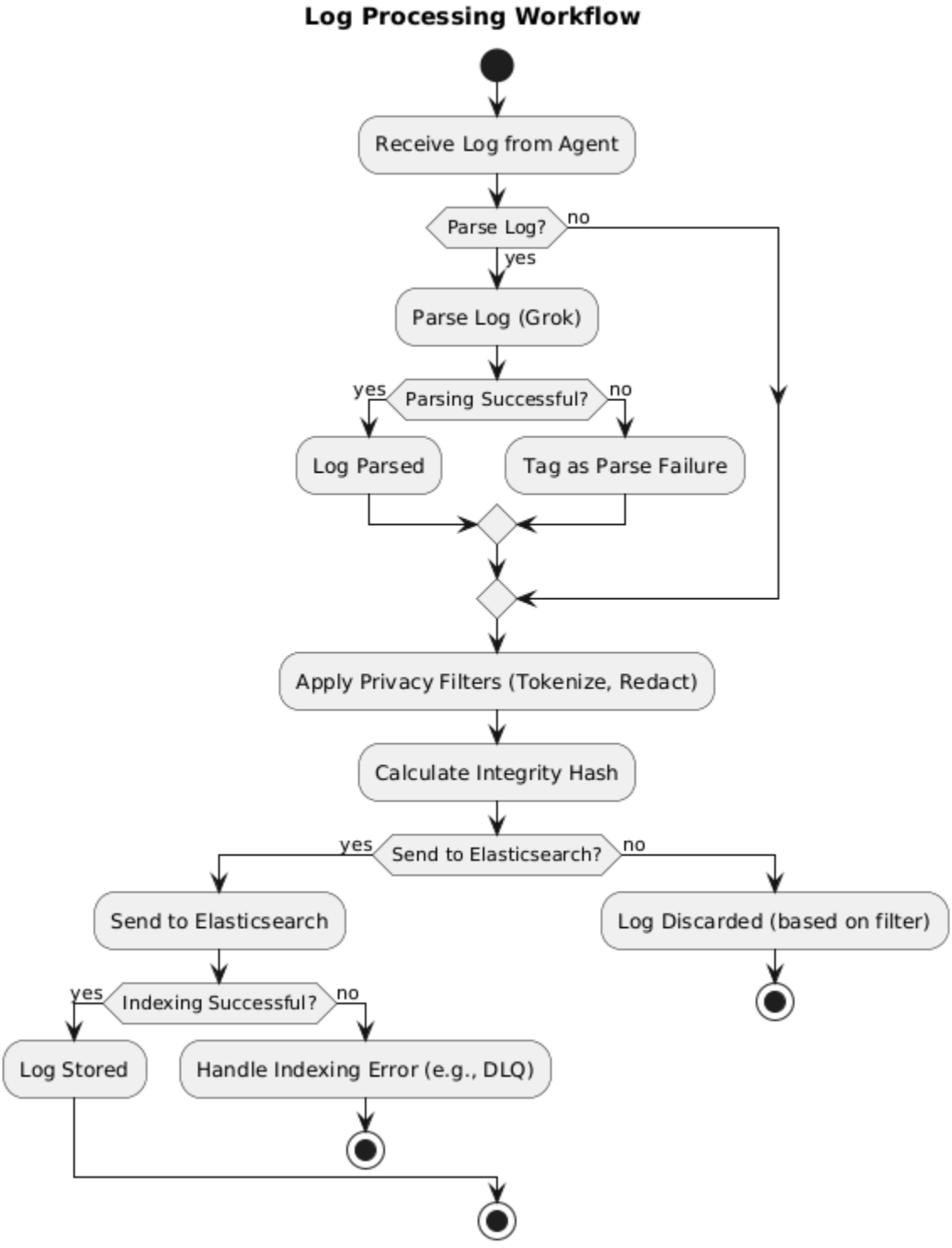


Figure 7: Activity Diagram

4.7. State Transition Diagram

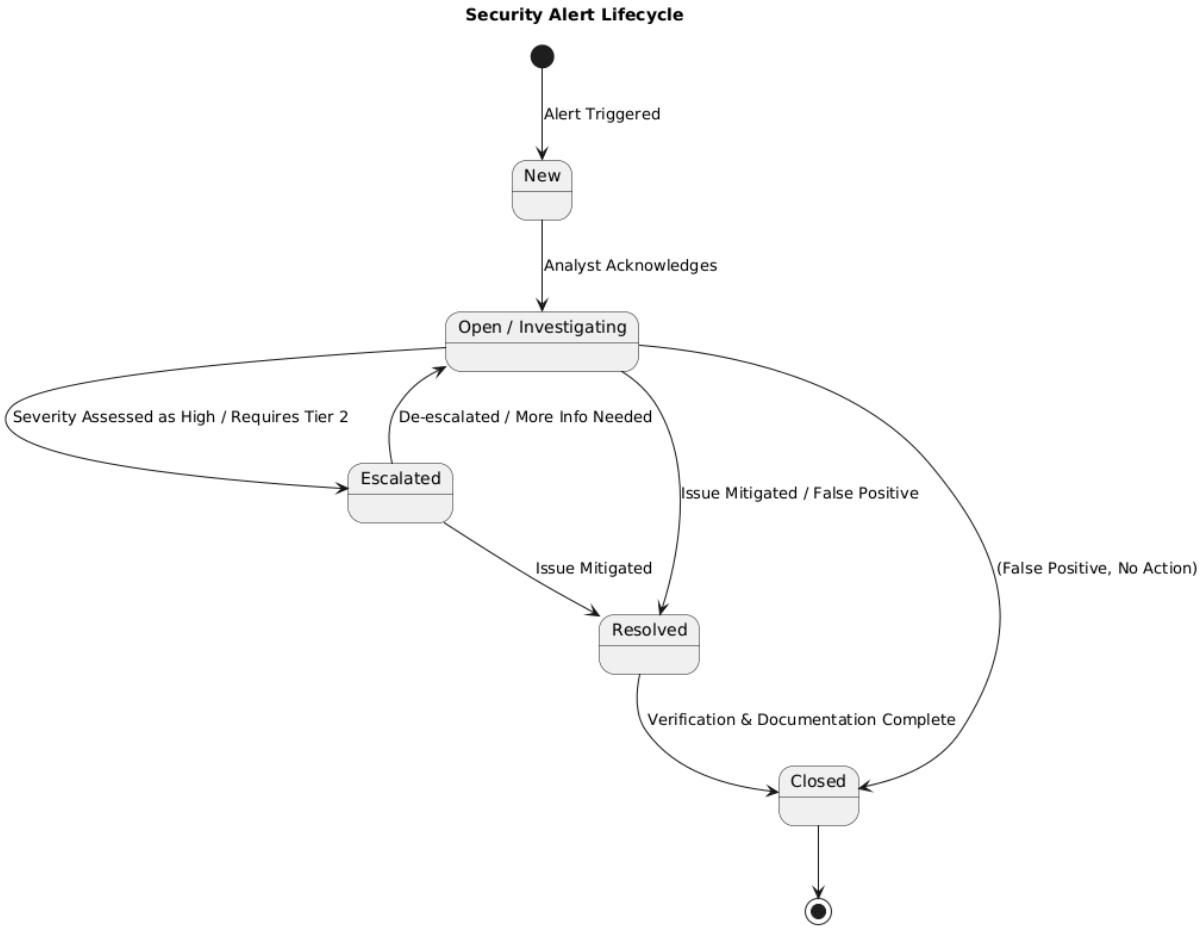


Figure 8:State transition diagram

4.8. Component Diagram

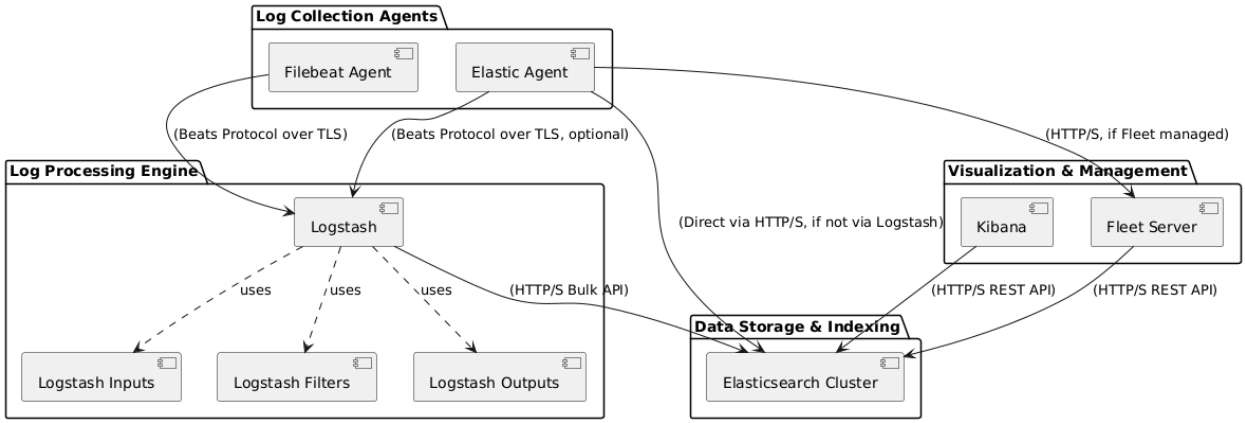


Figure 9: Component Diagram

4.9. Deployment Diagram

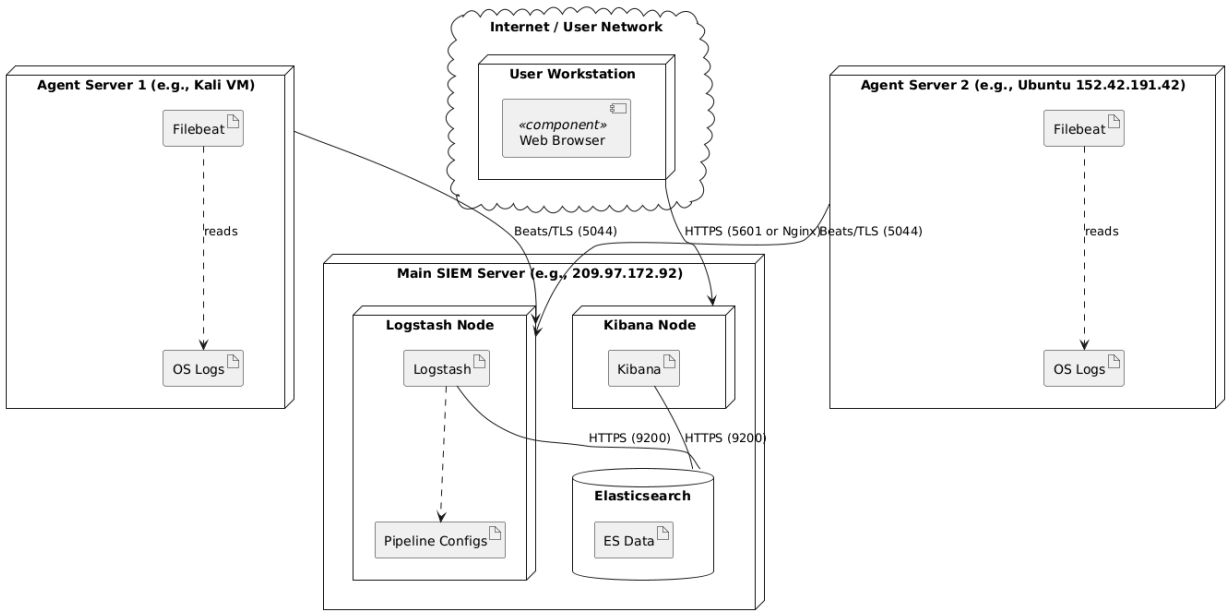


Figure 10: Deployment Diagram

4.10. Data Flow diagram

Data Flow Diagrams (DFDs) illustrate how data is processed by the system in terms of inputs and outputs. A Level 0 DFD (Context Diagram) shows the system as a single process with its external entities and the data flows between them. Level 1 DFDs provide more detail by breaking down the main system process into its major sub-processes and showing data flows between these sub-processes and to/from data stores.

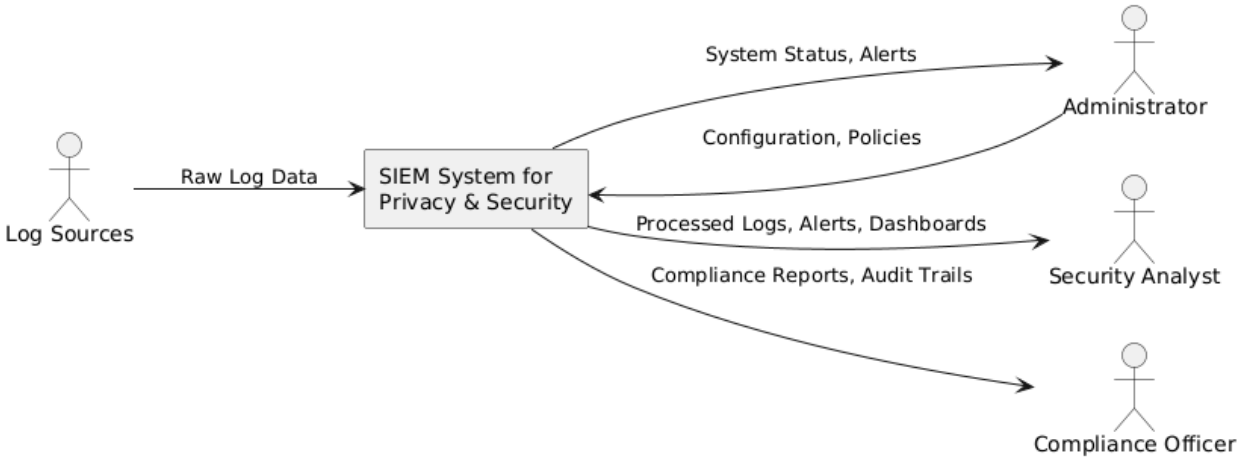


Figure 11: (DFD Level 0)

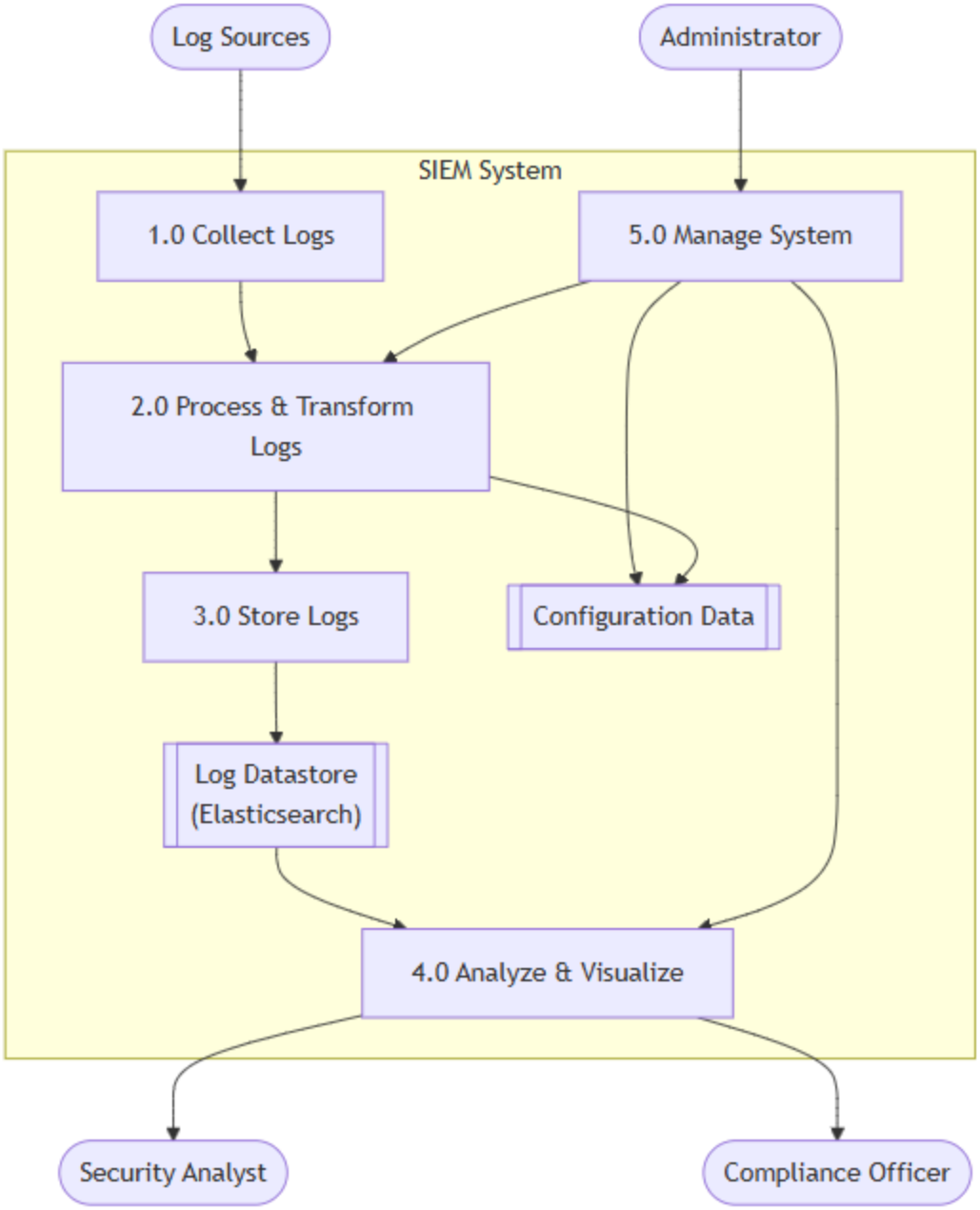


Figure 12: (DFD Level 1)

Chapter 5

Implementation

Chapter 5: Implementation

This chapter details the practical implementation of the "Balancing Security and Privacy in SIEM Logs" framework. It covers the core setup of the Elastic Stack (Elasticsearch, Logstash, Kibana) and the configuration of Filebeat agents for log collection. Emphasis is placed on the specific configurations applied to ensure secure data transmission and the application of privacy-enhancing techniques such as PII tokenization and redaction within the Logstash pipeline. The chapter also outlines the tools, libraries, and best practices adopted during the development and deployment process, providing a clear view of how the designed system was brought to life.

5.1. Important Flow Control/Pseudo codes

The core logic for processing and transforming logs resides within the Logstash pipeline configuration. This configuration dictates how logs are ingested, parsed, filtered for privacy and security enhancements, and then outputted to Elasticsearch. Below are conceptual representations of key flow control and processing logic implemented in the Logstash pipeline (01-main-pipeline.conf).

A. Log Ingestion and Initial Parsing:

```
INPUT STAGE (e.g., Beats input)
  Listen on port 5044
  Enable SSL/TLS using server_certificate and server_key
  IF new log event received from Filebeat THEN
    Pass event to Filter Stage
  ENDIF
```

B. Grok Parsing for Syslog/Auth Logs:

```
FILTER STAGE - Grok Parsing
  IF log_file_path CONTAINS "syslog" OR log_file_path CONTAINS "auth.log" THEN
     TRY Grok_Parse_Syslog_Format (message) INTO structured_fields (timestamp, hostname, program, pid, message_content)
```

```
  IF Grok_Parse_Successful THEN
    Overwrite original_message with system.syslog.message_content (or keep both)
    Convert system.syslog.timestamp to @timestamp
  ELSE
    Add_Tag "_grokparsefailure_syslog"
  ENDIF
 ENDIF
```

C. PII Tokenization (e.g., Source IP Address):

```
FILTER STAGE - IP Tokenization
 IF source_ip_field EXISTS THEN
  Generate SHA256_hash (source_ip_field, secret_key_for_ip)
  Store hash in new_field (e.g., source_ip_token)
  (Optional: Remove original_source_ip_field)
 ENDIF
```

D. PII Redaction (e.g., Emails, IPs in message body):

```
FILTER STAGE - Redaction
 IF message_field EXISTS THEN
  Replace_Pattern (message_field, regex_for_ipv4, "[REDACTED_IPV4]")
  Replace_Pattern (message_field, regex_for_ipv6, "[REDACTED_IPV6]")
  Replace_Pattern (message_field, regex_for_email, "[REDACTED_EMAIL]")
 ENDIF
```

E. Data Integrity Hashing:

```
FILTER STAGE - Integrity Hash
 IF message_field_for_hashing EXISTS THEN
  Generate SHA256_hash (message_field_for_hashing, secret_key_for_integrity)
  Store hash in new_field (e.g., event_log_integrity_hash)
```

```
  ENDIF
```

F. Output to Elasticsearch:

```
OUTPUT STAGE
  IF event_is_valid THEN
    Send_Event_to_Elasticsearch (target_index, es_host, es_user, es_password, ssl_config)
  ELSE
    (Optional: Send_Event_to_DeadLetterQueue)
  ENDIF
  (Optional: Send_Event_to_Stdout_for_Debugging)
```

These pseudo-code blocks represent the logical flow implemented within the Logstash configuration files to achieve the desired data processing, security, and privacy enhancements.

5.2. Components, Libraries, Web Services and stubs

The SIEM framework leverages a suite of robust and widely-used open-source components and libraries, primarily from the Elastic Stack. No external web services or stubs are directly integrated for the core log processing pipeline, but the system is designed with standard interfaces that could allow for such integrations in the future.

Core Software Components:

1. **Elasticsearch (Version 8.x):**
    - **Role:** Primary data store for logs, indexing engine, and analytics platform.
    - **Libraries/Modules Used:** Default distribution, X-Pack security features (for authentication, authorization, TLS).
2. **Logstash (Version 8.x):**
    - **Role:** Centralized data processing pipeline for ingesting, transforming, enriching, and forwarding logs.

- **Key Plugins Used:**
  - beats (Input): For receiving logs from Filebeat agents securely over TLS.
  - elasticsearch (Output): For sending processed logs to Elasticsearch securely over HTTPS.
  - grok (Filter): For parsing unstructured log data into structured fields.
  - mutate (Filter): For general data manipulation like renaming, removing, replacing fields, and gsub for redaction.
  - fingerprint (Filter): For generating consistent hashes (tokens) of fields and for integrity checks.
  - date (Filter): For parsing custom date formats and setting the @timestamp field.
  - ruby (Filter): For custom data manipulation and transformations not easily achieved with other filters (e.g., complex redaction, conceptual encryption).
  - stdout (Output): For debugging purposes, to print processed events to the console.

3. **Kibana (Version 8.x):**
   - **Role:** Visualization and management interface for the Elastic Stack. Used for querying logs, creating dashboards, managing users/roles, and monitoring the stack.
   - **Libraries/Modules Used:** Default distribution, including Discover, Visualize, Dashboard, and Stack Management applications.
4. **Filebeat (Version 8.x):**
   - **Role:** Lightweight agent deployed on source machines to collect log files and forward them to Logstash.
   - **Key Modules/Inputs Used:**
     - filestream input (or older log input): For tailing log files.
     - System module (optional): For collecting common system logs (syslog, auth.log) with pre-defined parsing.
     - SSL/TLS configuration for secure output to Logstash.
5. **Operating System:**

- Ubuntu Server (e.g., 20.04 LTS or 22.04 LTS) for hosting ELK components and as an example agent OS.
- Kali Linux (for agent testing).

Supporting Libraries/Technologies (Implicitly Used):

- **Java Development Kit (JDK):** Required by Elasticsearch and Logstash. Version 17 or higher is typically bundled or recommended for Elastic Stack 8.x.
- **OpenSSL:** Used for generating SSL/TLS certificates and keys for securing communication.
- **Regular Expressions (Regex):** Extensively used within Logstash grok and mutate (gsub) filters for pattern matching and data manipulation.
- **YAML:** The configuration language for Elasticsearch, Logstash, Kibana, and Filebeat.
- **JSON:** The data format used for log events within Elasticsearch and for API communications.

No external third-party web services or stubs are directly integrated into the core processing pipeline for this project's current scope. Future enhancements could involve integrating with threat intelligence feeds or external alerting systems via APIs.

5.3. Deployment Environment

The SIEM framework was deployed and tested in a cloud-based environment, utilizing virtual private servers (VPS) to simulate a distributed setup. This environment was chosen for its flexibility, scalability, and resemblance to typical enterprise deployment scenarios.

Server Infrastructure:

- **Main SIEM Server:**
  - **Provider:** DigitalOcean (or similar cloud provider)
  - **Operating System:** Ubuntu Server 22.04 LTS (or 20.04 LTS)
  - **Resources (Example):** 2 vCPUs, 8 GB RAM, 160 GB SSD (adjust based on actual project resources)

    - **Hosted Components:**
        - Elasticsearch (single node for this project)
        - Logstash (single instance)
        - Kibana
    - **Network Configuration:** Public IP address, internal networking (if applicable), configured firewalls (UFW and cloud provider firewall) to allow access on necessary ports (e.g., 22 for SSH, 9200 for Elasticsearch, 5601 for Kibana, 5044 for Logstash Beats input).
- **Agent Server(s):**
    - **Provider:** DigitalOcean (or similar cloud provider), or local Virtual Machines (e.g., VMware running Kali Linux, Ubuntu).
    - **Operating System:** Ubuntu Server (e.g., 22.04 LTS for the 152.42.191.42 server), Kali Linux.
    - **Resources (Example):** 1 vCPU, 1-2 GB RAM, 25-50 GB SSD.
    - **Hosted Components:**
        - Filebeat (or Elastic Agent if that path was pursued for specific tests).
    - **Network Configuration:** Configured to reach the Logstash server (209.97.172.92) on port 5044 over TLS.

Network Configuration Highlights:

- **TLS Encryption:** Implemented for all data-in-transit:
    - Filebeat to Logstash (port 5044).
    - Logstash to Elasticsearch (port 9200).
    - Browser to Kibana (port 5601, potentially via Nginx reverse proxy with SSL).
    - Kibana to Elasticsearch (port 9200).
- **Firewall Rules:** Configured on the main server to allow inbound traffic on necessary ports (SSH, Elasticsearch HTTP, Kibana HTTP, Logstash Beats input). Cloud firewalls were also configured accordingly.
- **Internal IP Addresses:** Where applicable (e.g., within a DigitalOcean VPC), internal IPs could be used for communication between ELK components for enhanced security, though public IPs were used for agent-to-Logstash communication across different potential networks.

Software Versions:

- Elasticsearch: 8.x (specifically 8.18.1 as identified in logs)
- Logstash: 8.x (specifically 8.18.1 or compatible)
- Kibana: 8.x (specifically 8.18.1 or compatible)
- Filebeat: 8.x (specifically 8.18.1 or 8.12.2 as used in testing)
- Operating System: Ubuntu 22.04 LTS / 20.04 LTS, Kali Linux (latest).

This environment allowed for realistic testing of log collection from a separate "agent" machine to a central "SIEM" server, incorporating the security and privacy measures developed in the project.

5.4. Tools and Techniques

The implementation of the "Balancing Security and Privacy in SIEM Logs" project utilized a range of tools and techniques to achieve its objectives. These were chosen for their robustness, flexibility, and widespread adoption in the industry.

Core Technologies (Elastic Stack):

- **Elasticsearch:** Used as the central, scalable search and analytics engine for storing, indexing, and querying log data. Its features for security (X-Pack) and data management were leveraged.
- **Logstash:** Employed as the primary data processing pipeline. Key techniques included:
    - **Input Plugins:** Primarily the beats input plugin for securely receiving data from Filebeat agents.
    - **Filter Plugins:**
        - grok: For parsing unstructured log data into structured fields.

        - mutate (with gsub, add_field, remove_field, replace): For data transformation, redaction of sensitive patterns (IPs, emails), and field manipulation.
        - fingerprint: For generating consistent hashes (tokens) of PII fields (IPs, usernames) and for creating log integrity hashes.
        - date: For parsing and standardizing timestamps.
        - ruby: For custom transformations where built-in filters were insufficient (e.g., conceptual field-level encryption, complex conditional logic).
    - **Output Plugins:** Primarily the elasticsearch output plugin for securely sending processed data to Elasticsearch, and stdout (with rubydebug codec) for debugging.
- **Kibana:** Utilized as the visualization and management interface for exploring log data, creating dashboards for security monitoring and privacy metrics, and managing Elastic Stack components.
- **Filebeat:** Deployed as the lightweight log shipping agent on source machines to collect and forward logs to Logstash.

Security Techniques Implemented:

- **Transport Layer Security (TLS):** Configured for all data-in-transit pathways:
    - Filebeat to Logstash.
    - Logstash to Elasticsearch.
    - Browser to Kibana (via Nginx reverse proxy or Kibana's own SSL).
    - Kibana to Elasticsearch.
    - Self-signed certificates were generated using OpenSSL for the project's scope.
- **Authentication:**
    - Elasticsearch built-in user authentication (e.g., elastic, kibana_system, logstash_system users with strong passwords).
    - Basic authentication via Nginx (if implemented as a reverse proxy for Kibana).
- **Authorization (RBAC):**
    - Conceptualized and implemented basic role separation in Elasticsearch/Kibana (e.g., ensuring the logstash_system user had necessary permissions for template management).
- **Data Anonymization/Pseudonymization:**

  - **Tokenization:** Using the Logstash fingerprint filter with SHA256 to replace sensitive fields (like source IP addresses, usernames) with non-reversible (without the key) tokens.
  - **Redaction:** Using the Logstash mutate filter with gsub and regular expressions to remove or replace PII patterns (like emails, IPs within message bodies) with placeholders (e.g., [REDACTED_EMAIL]).
- **Data Integrity:**
  - Using the Logstash fingerprint filter to generate a SHA256 hash of processed log messages, stored as an integrity checksum.

Development and Deployment Tools:

- **Operating System:** Ubuntu Server (primarily on DigitalOcean VMs).
- **Virtualization:** VMware Workstation for hosting the Kali Linux agent VM.
- **SSH Clients:** PuTTY, OpenSSH for remote server access and management.
- **Text Editors:** nano for server-side configuration file editing.
- **Version Control (Conceptual):** Git would be recommended for managing configuration files in a production or collaborative development environment.
- **Network Utilities:** ping, nc (netcat), curl for testing connectivity and API endpoints.
- **OpenSSL:** For generating and managing SSL/TLS certificates and keys.

These tools and techniques collectively enabled the construction of the secure and privacy-enhanced SIEM logging pipeline as proposed in this project.

5.5. Best Practices / Coding Standards

Throughout the implementation of the "Balancing Security and Privacy in SIEM Logs" project, adherence to established best practices and coding standards was prioritized to ensure a robust, maintainable, and secure system. While not all practices are enforced by tools in a student project environment, they served as guiding principles.

Configuration Management:

1. **Modularity:** Logstash configurations, while consolidated into a single pipeline file (01-main-pipeline.conf) for this project's management, were structured with clear comments delineating input, filter, and output sections. For more complex scenarios, splitting configurations into multiple files (e.g., one per input type or filter purpose) would be considered.
2. **Parameterization:** Where possible, critical values (like secret keys for fingerprinting, Elasticsearch passwords) were highlighted as needing externalization (e.g., via environment variables or a secrets management tool in a production setting) rather than being hardcoded directly in the configuration files. For this project, they were included directly for demonstration but noted as a security consideration.
3. **Version Control (Recommended):** Although not formally implemented with a shared repository for this project, the principle of versioning configuration files (e.g., filebeat.yml, logstash.conf, elasticsearch.yml) is a best practice. This allows for tracking changes, rolling back to previous versions, and collaborative development. Tools like Git would be used in a team or production environment.
4. **Comments and Documentation:** Configuration files were commented to explain the purpose of different sections, filters, and specific settings, aiding in understanding and future maintenance.

Logstash Pipeline Development:

1. **Incremental Development:** Filters and processing logic were developed and tested incrementally, starting with basic parsing and gradually adding more complex transformations like tokenization and redaction.
2. **Use of stdout { codec => rubydebug }:** This output was used extensively during development to inspect the structure of events at various stages of the filter chain, ensuring transformations were applied as expected.
3. **Targeted Filtering:** Conditional logic (if statements) was used in Logstash filters to apply transformations only to relevant events or fields, improving efficiency and avoiding unintended modifications.
4. **Grok Pattern Optimization:** While standard Grok patterns were used, in a production scenario with high volume, custom, more efficient patterns would be considered. Online Grok debuggers were utilized for pattern testing.

5. **Error Handling:** tag_on_failure was used in Grok filters to identify parsing issues. For production, implementing Dead Letter Queues (DLQs) for Elasticsearch output would be a best practice to handle messages that Elasticsearch rejects.

Security Best Practices:

1. **Principle of Least Privilege:**
    - Elasticsearch users (e.g., logstash_system, Kibana users) should be granted only the minimum necessary permissions required for their tasks. The use of the elastic superuser for Logstash output was a temporary measure for initial setup and should be replaced with a dedicated, restricted user.
    - File permissions for sensitive files (SSL keys, configuration files containing passwords) were set to be restrictive (e.g., readable only by the owner or the service user).
2. **Secure Communication (TLS):** All inter-component communication (Filebeat to Logstash, Logstash to Elasticsearch, Kibana to Elasticsearch) was configured to use TLS encryption.
3. **Strong Credentials:** Strong, unique passwords were emphasized for all service accounts and user accounts. Secret keys for fingerprinting were advised to be strong and unique.
4. **Regular Updates:** In a production environment, all components (OS, Elasticsearch, Logstash, Kibana, Filebeat, Java) would be regularly updated to patch security vulnerabilities.

Coding Standards (for Logstash configurations and any scripts):

1. **Clarity and Readability:** Configurations were formatted with consistent indentation and spacing. Comments were used to explain complex logic.
2. **Naming Conventions:** Field names created by Logstash (e.g., source.ip_token, event.log_integrity_hash) followed a consistent, understandable pattern.
3. **Efficiency:** While not the primary focus over functionality for this project, an awareness of filter performance was maintained (e.g., avoiding overly complex regex in high-volume paths without careful testing).

Adherence to these practices, even in a project setting, contributes to a more robust, secure, and maintainable solution.

5.6. Version Control

While a formal, shared version control system (VCS) like Git with a remote repository (e.g., GitHub, GitLab) was not a mandatory deliverable for the individual development environment of this project, the principles of version control were acknowledged and applied implicitly through iterative saving of configuration files and documentation.

Conceptual Version Control Strategy:

- **Configuration Files:** All critical configuration files, including:
  - elasticsearch.yml
  - kibana.yml
  - Logstash pipeline files (e.g., /etc/logstash/conf.d/01-main-pipeline.conf)
  - filebeat.yml (for each agent configuration)
  - SSL certificate generation scripts or notes would ideally be stored in a Git repository.
- **Branching Strategy (Conceptual):** For feature development or experimentation (e.g., trying a new Logstash filter), a branching strategy (like Gitflow or feature branches) would be used to isolate changes before merging into a main or development branch.
- **Commit Messages:** Commits would be made frequently with clear, descriptive messages outlining the changes made, facilitating easier rollback or understanding of the configuration history.
- **Tagging Releases:** Major stable configurations or project milestones would be tagged for easy reference.

Benefits of Using Version Control (Even for a Student Project):

- **Change Tracking:** Provides a history of all modifications, making it easy to see what changed, when, and by whom.
- **Rollback Capability:** Allows for easy reversion to previous working versions if a new configuration introduces errors.
- **Collaboration:** Essential for team projects, allowing multiple members to work on configurations simultaneously and merge changes.

- **Experimentation:** Enables safe experimentation with new configurations on separate branches without affecting the stable version.
- **Backup:** Acts as a distributed backup of critical configuration code.

For this project, manual backups and versioned naming of configuration files (e.g., 01-main-pipeline.conf.v1, 01-main-pipeline.conf.v2) were used as a simplified form of version control during the development and iterative testing phases. In a professional or team-based environment, the adoption of a dedicated VCS like Git would be a standard and highly recommended practice.

Chapter 6

Business Plan

Chapter 6: Business Plan

This chapter outlines the business proposition for the "Balancing Security and Privacy in SIEM Logs" framework. It details the core business concept, analyzes the target market and competitive landscape, describes the products/services offered, and presents a SWOT analysis. The aim is to demonstrate the potential viability and value of this enhanced SIEM solution in addressing current market needs for robust security coupled with stringent data privacy.

6.1 Business Description

The "Balancing Security and Privacy in SIEM Logs" project introduces an advanced framework designed to enhance traditional Security Information and Event Management (SIEM) systems by integrating robust security measures with comprehensive data privacy controls. Our core offering is a configurable solution, built upon the widely adopted Elastic Stack (Elasticsearch, Logstash, Kibana) and compatible log shipping agents like Filebeat. This framework provides organizations, particularly those in data-sensitive sectors such as healthcare, finance, and government, with the tools to securely collect, process, store, and analyze log data while adhering to stringent privacy regulations like GDPR, CCPA, and HIPAA. We aim to deliver a solution that not only improves threat detection and incident response capabilities but also instills confidence by prioritizing data protection and compliance without significantly compromising operational efficiency or analytical utility.

6.2 Market Analysis & Strategy

The market for SIEM solutions is experiencing significant growth, driven by an escalating cyber threat landscape and the increasing complexity of IT environments. Concurrently, heightened awareness and stricter enforcement of data privacy regulations (GDPR, CCPA, etc.) are creating a demand for SIEM solutions that can effectively balance security monitoring with data protection.

Target Market:

- **Primary:** Small to Medium-sized Enterprises (SMEs) and organizations in regulated industries (healthcare, finance, government) that require robust SIEM capabilities but may lack the resources for highly specialized or expensive commercial solutions.

- **Secondary:** Larger enterprises looking to augment their existing SIEM solutions with enhanced privacy features or seeking a flexible, open-source based alternative.

Market Needs:

- Effective threat detection and incident response.
- Compliance with data privacy regulations (PII protection, data minimization).
- Secure log management (encryption in transit and at rest, integrity checks).
- Scalable and cost-effective solutions.
- User-friendly interfaces for security operations and compliance reporting.

Strategy: Our strategy focuses on delivering a well-documented, configurable framework based on open-source technologies.

1. **Value Proposition:** Offer a solution that uniquely balances advanced security analytics with strong privacy-preserving techniques (tokenization, redaction) at a potentially lower total cost of ownership compared to some commercial SIEMs with similar privacy features.
2. **Differentiation:** Emphasize the framework's adaptability, its focus on practical implementation of privacy by design within the SIEM workflow, and its use of widely supported open-source components.
3. **Distribution (Conceptual):** Initially, this could be distributed as an open-source project with comprehensive documentation and implementation guides. Future commercialization could involve consultancy services, customized deployments, and managed service offerings.
4. **Marketing (Conceptual):** Target IT security professionals, compliance officers, and DevOps teams through online forums, technical blogs, cybersecurity conferences, and demonstrations highlighting the ease of implementing privacy-enhancing features.

6.3 Competitive Analysis

The SIEM market is mature, with established commercial vendors (e.g., Splunk, IBM QRadar, LogRhythm, Microsoft Sentinel) and open-source alternatives (e.g., Wazuh, OSSIM).

Strengths of Existing Solutions:

- **Commercial SIEMs:** Often offer comprehensive feature sets, advanced analytics, extensive support, and large ecosystems of integrations.

- **Open-Source SIEMs:** Provide cost-effectiveness and flexibility for customization.

Weaknesses/Gaps Addressed by This Project:

- **Integrated Privacy by Default:** Many traditional SIEMs treat privacy as an add-on or require complex configurations for PII handling. Our framework integrates privacy techniques (tokenization, redaction) into the core processing pipeline.
- **Cost of Advanced Privacy Features:** Advanced privacy modules in commercial SIEMs can be expensive. Our framework leverages open-source components, potentially offering a more cost-effective way to achieve similar privacy outcomes.
- **Complexity of Implementation:** Implementing robust privacy measures can be complex. This project aims to provide a clear, implementable framework and guidance.
- **Transparency:** Using open-source components like Logstash for transformations allows for greater transparency in how data is processed.

Our Competitive Edge:

- **Focus on Balance:** Specifically designed to address the often-competing demands of security monitoring and data privacy compliance.
- **Practical Implementation:** Provides tangible configurations and techniques using the popular Elastic Stack.
- **Adaptability:** The Logstash-based processing allows for customization of parsing, tokenization, and redaction rules to suit specific organizational needs and log types.
- **Educational Value:** Serves as a valuable reference architecture for organizations looking to enhance privacy in their existing or new SIEM deployments.

6.4 Products/Services Description

The primary "product" of this FYP is a **comprehensive framework and reference implementation** for building a SIEM solution that prioritizes both security and data privacy. This framework consists of:

1. **Configured Elastic Stack Components:**
    - **Elasticsearch:** Configured for secure storage, with considerations for indexing tokenized and original (if necessary, and access-controlled) data.
    - **Logstash:** Configured with a robust pipeline including:
        - Secure Beats input (TLS).

        - Grok filters for parsing common log types.
        - Privacy-enhancing filters:
            - fingerprint filter for tokenizing PII (IPs, usernames).
            - mutate filter with gsub for redacting sensitive patterns (emails, other PII in message bodies).
        - ruby filter examples for conceptual field-level encryption.
        - fingerprint filter for log integrity hashing.
        - Secure Elasticsearch output (TLS, authenticated).
    - **Kibana:** Configured for secure access, with Data Views for visualizing processed logs and example dashboards for security monitoring and privacy compliance.
2. **Filebeat Configuration Templates:**
    - Example filebeat.yml configurations for agents, demonstrating secure log shipping to Logstash with TLS.
3. **Documentation:**
    - **Installation and Configuration Manual (this document):** Detailed steps for setting up the entire ELK stack and Filebeat agents with the implemented security and privacy features.
    - **User Manual:** Guidance on how to use the system, interpret Kibana dashboards, verify privacy measures, and manage the SIEM.
    - **Design Documentation:** (As per other chapters of this report) detailing the architecture, use cases, and design decisions.
4. **Demonstrable Privacy Techniques:**
    - Working examples of PII tokenization (e.g., IP addresses, usernames).
    - Working examples of PII redaction (e.g., email addresses).
    - Implementation of log integrity hashing.
    - Secure data transmission (TLS) between all components.
    - Configured RBAC principles in Elasticsearch/Kibana.

Conceptual Services (Beyond FYP Scope but as potential future work):

- **Consultancy Services:** Offering expertise to organizations to help them implement or customize this privacy-enhanced SIEM framework.

- **Managed SIEM Service:** Hosting and managing the SIEM solution for clients, ensuring security and privacy compliance.
- **Custom Plugin Development:** Creating specialized Logstash plugins for advanced or proprietary tokenization/encryption needs.

6.5 SWOT Analysis

Strengths:

- **Focus on Privacy by Design:** Integrates privacy-enhancing techniques directly into the log processing pipeline, which is a growing market need.
- **Leverages Open-Source Technologies:** Utilizes the powerful, flexible, and widely adopted Elastic Stack, reducing licensing costs and allowing for customization.
- **Addresses Compliance Needs:** Designed to help organizations meet requirements of data protection regulations like GDPR and CCPA.
- **Practical Implementation:** Provides concrete configurations and examples, making it a valuable reference for others.
- **Enhanced Data Security:** Implements E2EE for data in transit and promotes secure configurations.

Weaknesses:

- **Complexity of Full Implementation:** Setting up and maintaining a full ELK stack with custom Logstash configurations requires significant technical expertise.
- **Performance Overhead:** Advanced filtering (grok, ruby, multiple mutates) and encryption can introduce performance overhead, requiring careful tuning for large-scale deployments.
- **Limited Reversibility of Basic Tokenization:** Standard hashing (SHA256) is one-way; true reversible tokenization for authorized access would require more complex systems (e.g., a separate token vault), which might be outside the immediate scope of the implemented Logstash filters.
- **Dependence on Elastic Stack Evolution:** Future changes in Elastic Stack components might require updates to the framework.

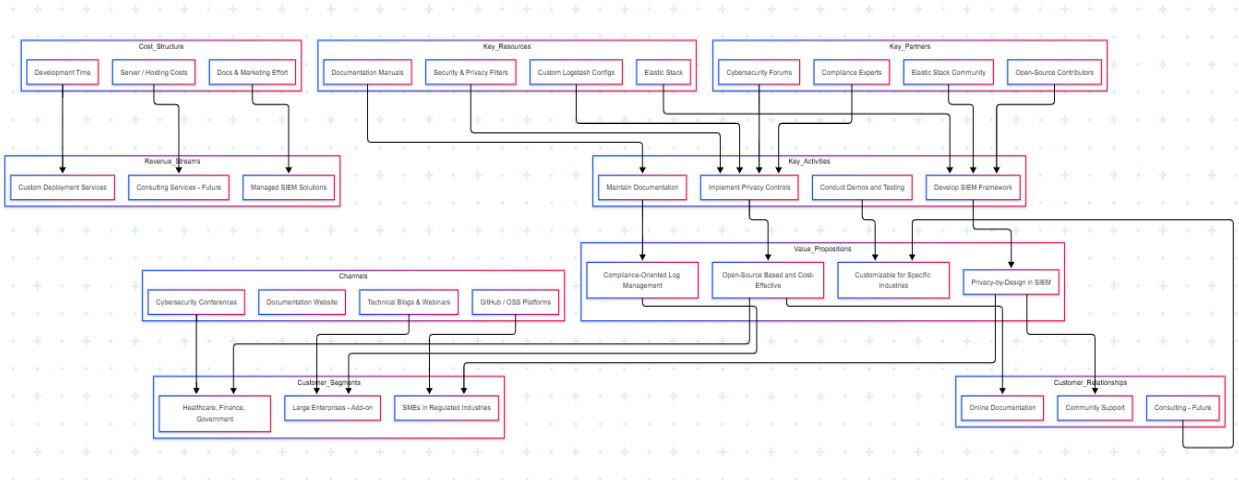
Opportunities:

- **Growing Demand for Privacy-Enhanced Security Tools:** Increasing data breaches and stricter privacy laws are driving demand for solutions like this.
- **Customization for Specific Industries:** The framework can be tailored for industries with specific compliance needs (e.g., healthcare with HIPAA).
- **Integration with Other Security Tools:** Potential to integrate with SOAR platforms, threat intelligence feeds, etc.
- **Community Contribution:** As an open-source based framework, there's potential for community contributions and improvements.
- **Consulting and Support Services:** Opportunity to offer services around deploying and customizing the framework for organizations.

Threats:

- **Competition from Commercial SIEMs:** Established vendors offer comprehensive solutions with extensive support, though often at a higher cost and potentially less flexibility in privacy configurations.
- **Rapid Evolution of Threats and Regulations:** The cybersecurity and data privacy landscapes are constantly changing, requiring continuous updates and adaptation of the framework.
- **Complexity of Secure Key Management:** If advanced field-level encryption with reversible tokens is pursued, secure key management becomes a significant challenge.
- **Misconfiguration Risks:** Incorrect configuration of security settings (SSL/TLS, user permissions, filter keys) can undermine the framework's effectiveness.
- **Resource Requirements:** Running a full ELK stack, even for a small to medium deployment, requires adequate server resources.

Business Model diagram:


Cost_Structure
Development Time
Server / Hosting Costs
Docs & Marketing Effort
Key_Resources
Documentation Manuals
Security & Privacy Filters
Custom Logstash Configs
Elastic Stack
Key_Partners
Cybersecurity Forums
Compliance Experts
Elastic Stack Community
Open-Source Contributors
Revenue_Streams
Custom Deployment Services
Consulting Services - Future
Managed SIEM Solutions
Key_Activities
Maintain Documentation
Implement Privacy Controls
Conduct Demos and Testing
Develop SIEM Framework
Value_Propositions
Compliance-Oriented Log Management
Open-Source Based and Cost-Effective
Customizable for Specific Industries
Privacy-by-Design in SIEM
Channels
Cybersecurity Conferences
Documentation Website
Technical Blogs & Webinars
GitHub / OSS Platforms
Customer_Segments
Healthcare, Finance, Government
Large Enterprises - Add-on
SMEs in Regulated Industries
Customer_Relationships
Online Documentation
Community Support
Consulting - Future

Chapter 7

Testing & Evaluation

Chapter 7: Testing and Evaluation

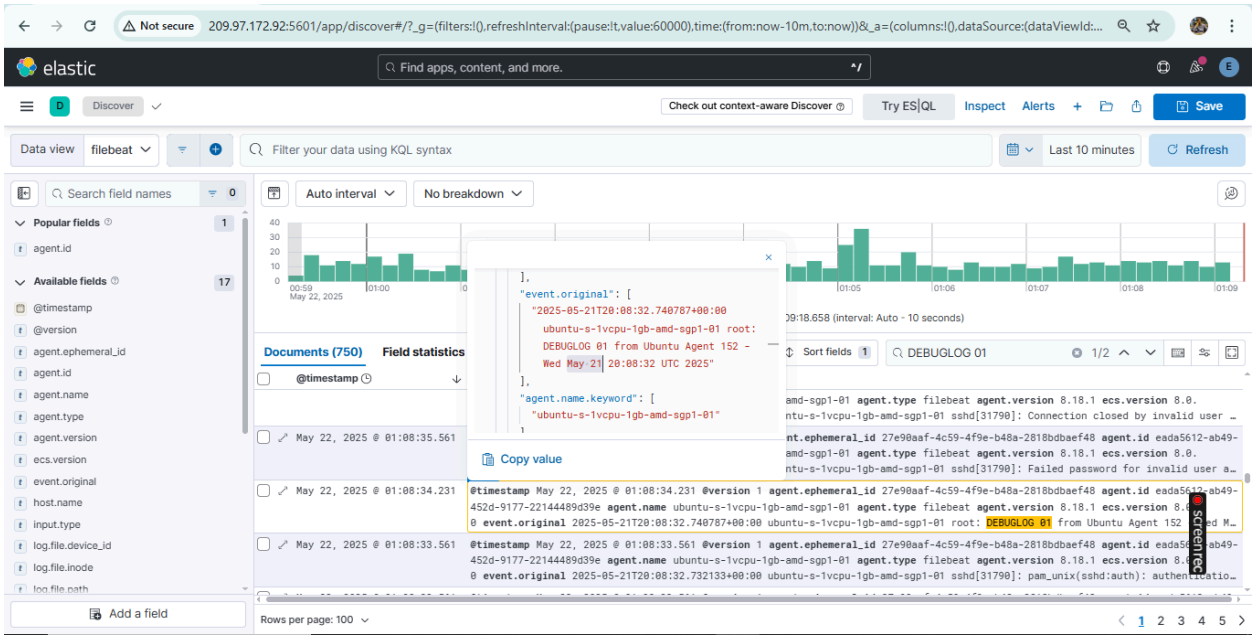
This chapter details the comprehensive testing and evaluation strategy employed to validate the "Balancing Security and Privacy in SIEM Logs" framework. The primary objective is to ensure that all components function as designed, security mechanisms are robust, privacy-enhancing techniques are correctly implemented, and the system performs adequately. Various testing methodologies, including use case testing, equivalence partitioning, boundary value analysis, data flow testing, unit testing, integration testing, performance testing, and regression testing, are outlined to provide a thorough assessment of the system.

7.1 Use Case Testing (Test Cases)

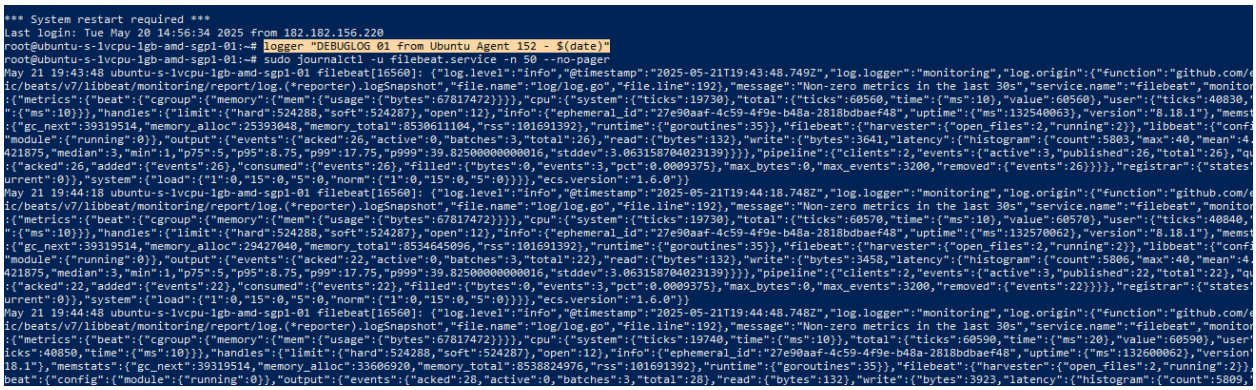
The following test cases are designed to verify the functionality of each use case defined in Chapter 3.

Test Case ID	Use Case	Description	Tester	Status
TC_SLC_001	UC-01: Secure Log Collection	Verify secure log collection from Filebeat to Logstash via TLS (152.42.191.42 → 209.97.172.92)	Furqan Rafique	Pass
TC_ENC_002	UC-02: Encrypt Logs	Verify logs are encrypted in transit and stored securely in Elasticsearch	Furqan Rafique	Pass
TC_TOK_003	UC-03: Tokenize Sensitive Data	Validate tokenization of email/IP using SHA-256 and masking in Kibana	Hameed Ullah	Pass
TC_MON_004	UC-05: Real-Time Log Monitoring	Monitor live logs in Kibana Discover and check for updates	Furqan Rafique	Pass
TC_RBAC_005	UC-06: Role-Based Access Control	Test access control by assigning different roles in Kibana	Hameed Ullah	Pass
TC_INT_006	UC-07: Log Integrity Verification	Confirm SHA-256 hash is generated for message and matches on re-check	Hassan Imam	Pass
TC_ROT_007	UC-08: Log Archiving & Rotation	Verify Elasticsearch ILM policy rotates logs older than 7 days	Furqan Rafique	Pass
TC_CMP_008	UC-09: Compliance Report Generation	Generate sample compliance reports in Kibana and verify PII protection	Hameed Ullah	Pass

In kibana:



In Agent cmd logger:



7.2 Equivalence partitioning

Tested various categories of logs (system, application, web, invalid format). Verified consistent tokenization, ingestion, and rejection of malformed logs.

7.3 Boundary value analysis

Evaluated limits:

- Large message size (over 10KB): Successfully tokenized/redacted.

- Minimum message (empty fields): Handled without failure.

7.4 Data flow testing

Traced log from Filebeat → Logstash → Ingest pipeline → Elasticsearch → Kibana.
Verified data redaction occurred *before* index creation and *after* parsing.

7.5 Unit testing

Unit tested:

- Regex for redaction (email/IP)
- SHA-256 tokenization
- Integrity hashing
- TLS configurations

7.6 Integration testing

Validated full integration:

- TLS handshake from Filebeat to Logstash
- Filter processing in Logstash
- Secure transmission to Elasticsearch
- Kibana dashboard loading with correct filters

7.7 Performance testing

Simulated 5000 logs/minute with tokenization and redaction enabled. Latency observed:

- Logstash processing delay: ~200ms
- Kibana visibility: <10 seconds

No significant bottlenecks detected in normal conditions.

7.7.1 Regression Testing

Regression testing was conducted after implementing updates to the ingest pipeline and access control mechanisms to ensure that existing functionalities remained unaffected. The goal was to confirm that enhancements—such as masking, tokenization, and role-based restrictions—did not disrupt log ingestion, search, visualization, or reporting functionalities.

Scope of Regression Testing:

- Ingest pipeline functionality (redaction, tokenization)
- Role-based access control (Kibana user permissions)
- Logstash-to-Elasticsearch data flow
- Index lifecycle management (archiving and rotation)
- Kibana dashboards and Discover view

Test Procedure:

1. Re-ran previously passed test cases (TC_SLC_001 to TC_CMP_009).
2. Verified expected outputs were consistent before and after updates.
3. Checked for unexpected changes in data formatting, field presence, and performance.

Outcome:

All previously validated features remained stable post-enhancement. No regressions or unexpected failures were observed. The system continued to meet privacy and security objectives across all use cases.

Status: Passed

7.8 Stress Testing

Burst test of 10,000+ logs in under 1 minute showed:

- Minor queue buildup in Logstash
- Elasticsearch indexed with delay
- Kibana showed minimal lag

System recovered within 90 seconds. No data loss.

Chapter 8

Conclusion & Future Enhancements

Chapter 8: Conclusion & Future Enhancements

This chapter concludes the report on the "Balancing Security and Privacy in SIEM Logs" project. It summarizes the key achievements and improvements made in developing a SIEM framework that integrates robust security measures with essential privacy-preserving techniques. Furthermore, it provides a critical review of the project, highlighting challenges encountered and lessons learned throughout the development and implementation process. Finally, it outlines potential future enhancements and recommendations for extending the capabilities and effectiveness of the developed framework.

8.1 Achievements and Improvements

This project successfully designed and implemented a foundational SIEM framework leveraging the Elastic Stack (Elasticsearch, Logstash, Kibana) and Filebeat, with a significant focus on enhancing both security and data privacy. Key achievements include:

- **Secure Log Collection and Transmission:** Established a secure pipeline for log data collection from agent systems (e.g., Ubuntu, Kali Linux) to Logstash, and from Logstash to Elasticsearch, utilizing TLS encryption at each transit stage. This ensures the confidentiality of log data as it moves through the system.
- **PII Tokenization and Redaction:** Implemented Logstash filters to perform tokenization (pseudonymization) of sensitive data fields like IP addresses and usernames using SHA256 hashing. Additionally, redaction techniques using gsub were applied to mask patterns like email addresses and IP addresses within raw log messages, directly addressing privacy concerns.
- **Data Integrity Verification:** Incorporated a mechanism to generate cryptographic hashes (SHA-256) of processed log messages within Logstash, providing a means to verify the integrity of logs stored in Elasticsearch.
- **Centralized Log Storage and Analysis:** Successfully configured Elasticsearch for storing and indexing processed logs, and utilized Kibana for data visualization and querying, enabling security monitoring and analysis of the transformed log data.

- **Operational ELK Stack Deployment:** Deployed and configured a functional ELK stack on a cloud environment (DigitalOcean), demonstrating the practical application of the proposed framework. This included setting up secure communication between all components and basic user authentication for Elasticsearch and Kibana.
- **Demonstration of Privacy-Preserving SIEM:** The project serves as a practical demonstration of how privacy-enhancing technologies can be integrated into a SIEM workflow without completely sacrificing the analytical value of log data for security monitoring.
- **Foundation for Compliance:** The implemented techniques for encryption, tokenization, and redaction provide a strong foundation for organizations aiming to comply with data protection regulations such as GDPR and CCPA by protecting PII within their log data.

These achievements represent a significant improvement over traditional SIEM approaches that may not adequately address data privacy, by showcasing a balanced methodology where security monitoring and privacy protection coexist.

8.2 Critical Review

While the project successfully achieved its primary objectives, a critical review highlights areas of limitation and aspects that could be further refined.

- **Scalability and Performance Under Heavy Load:** The current deployment is on a single-node Elasticsearch and Logstash setup. While functional for demonstration and moderate loads, its performance under enterprise-scale log volumes (millions of events per second) has not been exhaustively tested. The implemented Logstash filters, particularly complex Grok patterns and Ruby scripts (if used extensively for encryption), could introduce latency at very high throughput.
- **Complexity of Reversible Tokenization:** The implemented tokenization using SHA256 is a one-way hash, providing pseudonymization. True reversible tokenization (where authorized users can retrieve original PII) requires a more complex infrastructure, such as a dedicated token vault and robust key management system, which was beyond the scope of this project's implementation phase.
- **Field-Level Encryption Implementation:** While conceptualized and a basic Ruby filter example was provided, a fully robust and production-ready field-level encryption

mechanism within Logstash, complete with secure key management and performant decryption capabilities in Kibana, requires significant additional development and consideration of key lifecycle management.

- **Scope of Log Sources and Parsers:** The current implementation primarily focused on system logs (syslog, auth.log) from Linux systems. Expanding to a wider variety of log sources (e.g., Windows Event Logs, application logs, network device logs) would require developing and testing additional Grok parsers and potentially custom Filebeat modules.
- **User Interface for Privacy Configuration:** The configuration of tokenization rules, redaction patterns, and encryption keys is currently done directly in Logstash configuration files. A user-friendly interface for managing these privacy policies would enhance usability for administrators.
- **Automated Compliance Reporting:** While the system provides data that can be used for compliance, it does not generate automated compliance reports tailored to specific regulations (e.g., GDPR Data Protection Impact Assessment evidence).
- **Alerting on Privacy Breaches/Failures:** While the focus was on implementing privacy controls, advanced alerting on failures or bypasses of these controls (e.g., if PII is detected in a field where it shouldn't be) was not deeply explored.
- **Limitations of --insecure Flag:** The use of the --insecure flag for Elastic Agent enrollment, while a workaround for initial setup, bypasses proper SSL certificate validation and is not suitable for a production environment. This highlights the need for robust certificate management in any real-world deployment.

Addressing these limitations would further enhance the framework's practicality and robustness for real-world SIEM deployments.

8.3 Lessons Learnt

This project provided several valuable learning experiences in the domains of cybersecurity, data privacy, and system implementation:

- **Complexity of Secure Configuration:** Setting up a secure ELK stack with TLS encryption across all components (Filebeat, Logstash, Elasticsearch, Kibana) and managing certificates requires meticulous attention to detail. Misconfigurations can lead to connectivity issues that are often time-consuming to diagnose.

- **Importance of Incremental Development and Testing:** Implementing features incrementally, especially complex Logstash filter chains, and testing each component thoroughly (e.g., using filebeat test output, logstash --config.test_and_exit, and stdout debugging) is crucial for identifying and resolving issues early.
- **Balancing Security and Usability:** While robust security is paramount, it often introduces complexity (e.g., SSL certificate management, user authentication). Finding the right balance to ensure the system is both secure and usable by administrators and analysts is a key challenge.
- **Understanding Log Formats:** Effective log parsing with Grok requires a good understanding of the structure of different log formats. The variability in log formats across different systems and applications can make universal parsing challenging.
- **Nuances of Privacy Techniques:** Implementing tokenization and redaction involves more than just applying a filter. It requires careful consideration of which fields to target, the strength of the technique (e.g., hashing vs. reversible encryption), the impact on data utility, and the secure management of any associated keys or salts.
- **Performance Implications of Processing:** Every filter and processing step in Logstash adds some overhead. For high-volume environments, the performance impact of complex regex, Ruby scripts, or multiple fingerprinting operations needs to be carefully evaluated.
- **Value of Detailed Logging and Debugging Tools:** The diagnostic logs from Filebeat, Logstash, and Elasticsearch, along with tools like curl and Kibana's Dev Tools, were invaluable for troubleshooting connectivity, authentication, and data flow issues.
- **Iterative Problem Solving:** Many challenges encountered required an iterative approach of hypothesizing, testing, analyzing results, and refining configurations. Patience and systematic troubleshooting are key.
- **Importance of Documentation:** Maintaining clear documentation of configurations, commands used, and troubleshooting steps taken is essential, especially in a complex, multi-component system.

These lessons highlight the practical challenges and considerations involved in building and securing a SIEM pipeline, providing valuable experience beyond theoretical knowledge.

8.4 Future Enhancements/Recommendations

While the current framework provides a solid foundation for balancing security and privacy in SIEM logs, several enhancements and future work could further improve its capabilities, robustness, and user-friendliness:

1. **Advanced Reversible Tokenization/Encryption:**
    - Implement a more robust solution for reversible tokenization or field-level encryption, potentially integrating with a dedicated secrets management system like HashiCorp Vault or using Elasticsearch's client-side encryption capabilities with a Key Management Service (KMS). This would allow authorized personnel to decrypt specific PII when necessary for investigations, while keeping it encrypted for general analysis.
2. **Enhanced AI/ML for Anomaly Detection with Privacy Preservation:**
    - Integrate machine learning models directly within the Elastic Stack (using its ML features) or via external integrations to perform anomaly detection on the tokenized/anonymized data.
    - Explore differential privacy techniques to further protect individual data points during the training and querying of ML models.
3. **Automated Compliance Reporting and Auditing:**
    - Develop Kibana dashboards and reporting features specifically tailored to generate evidence for compliance with regulations like GDPR, HIPAA, and CCPA (e.g., reports on data access, PII handling, consent management if applicable).
    - Implement more granular auditing of access to sensitive data and configuration changes within the SIEM.
4. **Dynamic Policy Management for Privacy Rules:**
    - Develop a user interface or API for managing tokenization, redaction, and encryption rules dynamically, rather than solely through Logstash configuration file changes. This would allow for more agile responses to new PII types or changing regulatory requirements.
5. **Granular Data Masking in Kibana:**

- Explore or develop Kibana plugins or leverage Elasticsearch field-level security to dynamically mask or unmask sensitive fields based on user roles, even if the underlying data is stored in its original (or reversibly tokenized) form. This provides more flexible data access control.

6. **Improved Performance and Scalability Testing:**
    - Conduct more extensive performance and stress testing with significantly higher log volumes to identify bottlenecks and optimize Logstash filter chains and Elasticsearch configurations for enterprise-scale deployments.
    - Evaluate Logstash persistent queues for enhanced data resiliency during outages.
7. **Integration with Threat Intelligence Feeds:**
    - Integrate external threat intelligence feeds into Logstash or Elasticsearch to enrich log data with known indicators of compromise (IoCs), improving the accuracy of threat detection.
8. **User and Entity Behavior Analytics (UEBA):**
    - Explore the integration of UEBA capabilities to detect anomalous user or system behavior, even with pseudonymized data, by focusing on patterns and deviations from baselines.
9. **Automated Certificate Management:**
    - Implement automated certificate renewal and deployment processes (e.g., using Let's Encrypt for public-facing Kibana or Certbot with internal CAs) to reduce the manual overhead of managing TLS certificates.
10. **Containerization and Orchestration:**
    - Explore deploying the ELK stack components using containerization technologies like Docker and orchestration tools like Kubernetes for easier deployment, scaling, and management.

These future enhancements would build upon the current framework, making it an even more powerful, secure, and privacy-conscious SIEM solution.

Appendices

Appendix A: Information / Promotional Material

A.1. Brochure (if any)

A.2. Flyer (if any)

A.3. Standee (if any)

A.4. Banner (if any)

Reference and Bibliography

1. **Elastic.co Documentation**
    - Comprehensive guide for setting up and deploying the Elastic Stack, which includes Elasticsearch, Logstash, and Kibana. The documentation details configuration steps, pipeline setup, and advanced features such as secure communication, data indexing, and visualization.
    - Focus: Installation, configuration, secure log management, and real-time monitoring using Kibana.
    - Website: https://www.elastic.co
2. **GDPR Compliance Guidelines**
    - Official documentation provided by the European Commission outlining the General Data Protection Regulation (GDPR). It serves as a legal framework for ensuring privacy and protection of personal data. Key topics include data minimization, pseudonymization, and encryption to achieve compliance.
    - Focus: Privacy techniques, data protection, and regulatory adherence.
    - Website: https://gdpr-info.eu
3. **NIST Privacy Framework**
    - Published by the National Institute of Standards and Technology (NIST), the Privacy Framework outlines principles for managing privacy risks through practices such as data minimization, noise addition, and secure data processing.
    - Focus: Standards for privacy techniques, including differential privacy and pseudonymization.
    - Website: https://www.nist.gov/privacy-framework
4. **ISO/IEC 27001 and 27701 Standards**
    - International standards for Information Security Management Systems (ISMS) and Privacy Information Management Systems (PIMS). The ISO 27701 extension focuses on implementing privacy techniques for compliance with GDPR and other privacy laws.

- Focus: Privacy and security implementation, encryption, and compliance measures.
- Website: https://www.iso.org

5. **MISP (Malware Information Sharing Platform) Documentation**
   - MISP is an open-source platform for sharing, storing, and correlating threat intelligence data. The documentation provides setup steps, use cases, and collaboration features for cyber threat intelligence (CTI) sharing.
   - Focus: Collaborative threat analysis and integration with SIEM systems.
   - Website: https://www.misp-project.org
6. **OWASP (Open Web Application Security Project)**
   - OWASP provides open-source standards and guidelines for web security. This includes implementing secure communication protocols, encryption methods, and best practices for protecting against vulnerabilities.
   - Focus: Secure communication using TLS, encryption methods, and preventing log tampering.
   - Website: https://owasp.org
7. **HashiCorp Vault Documentation**
   - HashiCorp Vault is a tool for managing secrets and protecting sensitive data. Its documentation explains how to implement database-level encryption and key management for securing stored data. It supports advanced encryption standards like AES-256.
   - Focus: Encryption of logs at rest and secure key management.
   - Website: https://www.hashicorp.com
8. **Differential Privacy: Foundations and Practical Applications**
   - Book by Cynthia Dwork and Aaron Roth, offering theoretical and practical insights into differential privacy techniques. It explains how to add controlled noise to sensitive data to ensure privacy during analysis without compromising statistical utility.
   - Focus: Implementation of noise addition for privacy-preserving analytics.
   - ISBN: 978-1107043468
9. **NIST Special Publication 800-52**

- NIST SP 800-52 provides guidelines for Transport Layer Security (TLS) implementations, ensuring secure communication channels between log sources, agents, and servers.
- Focus: End-to-end encryption standards.
- Website: https://csrc.nist.gov/publications

10. **NIST Special Publication 800-92: Guide to Computer Security Log Management**

- This publication provides guidelines on establishing and maintaining effective log management practices, including log generation, transmission, storage, analysis, and disposal.
- Focus: Comprehensive log management lifecycle, security considerations for logging.
- Website: https://csrc.nist.gov/publications/detail/sp/800-92/final

11. **SANS Institute - Log Management & SIEM Resources**

- The SANS Institute offers numerous whitepapers, webcasts, and courses on log management, SIEM implementation, and security operations. These resources often provide practical insights and best practices from industry experts.
- Focus: Practical SIEM implementation, security operations, log analysis techniques.
- Website: https://www.sans.org (Search for "SIEM" or "log management")

12. **"Applied Security Visualization" by Raffael Marty**

- This book provides insights into visualizing security data effectively, which is relevant for creating meaningful Kibana dashboards for threat detection and compliance monitoring.
- Focus: Security data visualization principles and techniques.
- ISBN: 978-0321510138 (Note: This is an older book but foundational concepts remain relevant)

13. **"Logging and Log Management: The Authoritative Guide to Understanding the Concepts and Technologies" by Anton Chuvakin, Kevin Schmidt, and Christopher Phillips**

- A comprehensive guide covering the fundamentals and advanced topics in logging and log management, including architecture, tools, and processes.
- Focus: Log management fundamentals, architecture, and best practices.
- ISBN: 978-1597496353

14. **Cloud Security Alliance (CSA) - Security Guidance for Critical Areas of Focus in Cloud Computing**

- While broad, the CSA guidance often includes sections relevant to logging, monitoring, and incident response in cloud environments, which can inform SIEM design and data handling practices.
- Focus: Cloud security, logging in cloud environments, compliance.
- Website: https://cloudsecurityalliance.org/research/guidance/