People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

Mohamed El Bachir El Ibrahimi University of Bordj Bou Arréridj

Faculty of Mathematics and Computer Science

Department of Computer Science



DISSERTATION

Presented in fulfillment of the requirements of obtaining the degree

Master in Computer Science

Specialty: Networking & Multimedia

THEME

Design and Implementation of a SIEM Platform for Enhanced Cybersecurity Monitoring and Incident Management

Presented by:

SANA Ishak Elmouatassem Billah, AOURAGH Omar Mehdi, OULD AMER Nassim, OUCHENE Hicham, ROUABAH Adem

Publicly defended on June 24, 2026, before a jury consisting of:

President: Dr. Mouhoub Belazougue

Examiner 1: Dr. Nadjib Benaouda

Examiner 2: Dr. Adel Bounegab

Supervisor: Dr. Boubakeur Moussaoui

2025/2026

Dedication

We dedicate this work to our supervisor, Dr. MOUSSAOUI Boubakeur, whose guidance, patience, and support made this project possible. Your belief in us pushed us further than we thought we could go.

To Dr. BENAOUDA Nadjib — a teacher who goes far beyond what is expected. Your knowledge, your generosity, and the way you give your time and energy to students without hesitation make you one of those rare people who leave a lasting mark. We are genuinely grateful to have crossed paths with you.

To our teammates — working with you never felt like pressure. Every challenge became lighter, every late night became a memory, and every obstacle became something we laughed about together. This project is as much yours as it is ours.

To our families — thank you for your patience, your encouragement, and for never doubting us, even when we doubted ourselves.

And to our colleagues, classmates, and friends who supported us along the way — your advice and moral support meant more than you know.

Acknowledgment

We would like to express our sincere gratitude to all those who supported us during the completion of this dissertation.

First, we warmly thank our supervisor, Dr. Boubakeur MOUSSAOUI, for his valuable guidance, constructive feedback, patience, and continuous encouragement throughout this project.

Our appreciation also goes to the jury members for the time they devoted to evaluating this work and for their useful remarks.

We also thank the faculty members of the Department of Computer Science at Mohamed El Bachir El Ibrahimi University for their academic support and for providing an environment favorable to learning and innovation.

Abstract

This project presents the design and implementation of a Security Information and Event Management platform aimed at improving cybersecurity monitoring and incident management within organizations. The proposed solution addresses the limitations of existing SIEM systems, particularly in terms of cost, complexity, and lack of adaptation to local operational environments.

The platform is designed using a modular architecture that enables the collection, normalization, and analysis of security events from multiple sources. It incorporates a rule-based detection engine to identify suspicious activities and generate alerts in near real-time. The system also provides an intuitive interface for Security Operations Center analysts to monitor alerts, investigate incidents, and manage detection rules efficiently.

The solution targets different SOC roles, including Level 1, Level 2, and Level 3 analysts, detection engineers, and system administrators, ensuring a structured and role-based approach to security operations.

Keywords: SIEM, cybersecurity, SOC, log management, event correlation, incident response, detection rules, threat intelligence.

Résumé

Ce projet présente la conception et la mise en œuvre d'une plateforme SIEM (Security Information and Event Management) visant à améliorer la surveillance de la sécurité et la gestion des incidents au sein des organisations. La solution proposée répond aux limites des systèmes SIEM existants, notamment en termes de coût, de complexité et d'adaptation aux environnements locaux.

La plateforme repose sur une architecture modulaire permettant la collecte, la normalisation et l'analyse des journaux de sécurité provenant de différentes sources. Elle intègre un moteur de détection basé sur des règles afin d'identifier les activités suspectes et de générer des alertes en quasi temps réel.

Le système propose également une interface intuitive permettant aux analystes du SOC de surveiller les alertes, d'investiguer les incidents et de gérer les règles de détection.

La solution cible différents profils d'utilisateurs, notamment les analystes SOC de niveau 1, 2 et 3, les ingénieurs de détection ainsi que les administrateurs système, garantissant ainsi une gestion structurée des opérations de sécurité.

En mettant l'accent sur la scalabilité, la facilité d'utilisation et l'adaptabilité, cette plateforme SIEM vise à fournir une solution efficace et accessible, particulièrement adaptée aux organisations disposant de ressources limitées en cybersécurité.

Mots-clés : SIEM, cybersécurité, SOC, gestion des journaux, détection d'intrusion, corrélation d'événements.

ملخص

يهدف هذا المشروع إلى تصميم وتطوير منصة لإدارة معلومات وأحداث الأمن (SIEM) من أجل تحسين مراقبة الأمن السيبراني وإدارة الحوادث داخل المؤسسات. تأتي هذه المنصة لمعالجة محدوديات الأنظمة الحالية، خاصة من حيث التكلفة، التعقيد، وصعوبة التكيف مع البيئات المحلية.

تعتمد المنصة على بنية معيارية تتيح جمع وتحليل وتوحيد سجلات الأحداث الأمنية من مصادر متعددة. كما تتضمن محرك كشف يعتمد على قواعد محددة من أجل اكتشاف الأنشطة المشبوهة وتوليد تنبيهات بشكل شبه فوري.

توفر المنصة كذلك واجهة استخدام سهلة تمكّن محللي مركز العمليات الأمنية (SOC) من مراقبة التنبيهات، تحليل الحوادث، وإدارة قواعد الكشف بكفاءة.

تستهدف هذه المنصة عدة فئات من المستخدمين، بما في ذلك محللي SOC من المستوى الأول والثاني والثالث، مهندسي الكشف، ومديري النظام، مما يضمن تنظيمًا فعالًا لعمليات الأمن.

من خلال التركيز على القابلية للتوسع وسهولة الاستخدام والتكيف، تهدف هذه المنصة إلى تقديم حل فعال ومنخفض التكلفة، خاصة للمؤسسات التي تعاني من نقص في الموارد في مجال الأمن السيبراني.

الكلمات المفتاحية: SIEM، الأمن السيبراني، مركز العمليات الأمنية، إدارة السجلات، كشف التسلل، تحليل الأحداث.

List of Abbreviations

- **SIEM :** Security Information and Event Management
- **SIM :** Security Information Management
- **SEM :** Security Event Management
- **SOC :** Security Operations Center
- **CIA :** Confidentiality, Integrity, and Availability
- **DoS :** Denial of Service
- **DDoS :** Distributed Denial of Service
- **SQL :** Structured Query Language
- **APT :** Advanced Persistent Threat
- **IS :** Information System
- **IDS :** Intrusion Detection System
- **EDR :** Endpoint Detection and Response
- **SOAR :** Security Orchestration, Automation, and Response
- **UEBA :** User and Entity Behavior Analytics
- **UML :** Unified Modeling Language
- **API :** Application Programming Interface
- **REST :** Representational State Transfer
- **NAT :** Network Address Translation
- **VM :** Virtual Machine
- **DC :** Domain Controller

- **CDM :** Conceptual Data Model
- **L1 :** Level 1 (SOC Analyst)
- **L2 :** Level 2 (SOC Analyst)
- **L3 :** Level 3 (SOC Analyst)
- **SME :** Small and Medium-sized Enterprise
- **RBAC :** Role-Based Access Control

Contents

Dedication **i**

Acknowledgment **ii**

Abstract **iii**

Résumé **iv**

v **ملخص**

List of Abbreviations **vi**

1 General Introduction **1**
1.1 Background 1
1.2 Problem Statement 2
1.3 Objective and Methodology 3
1.4 Report Structure 4

2 Literature Review and Requirements Analysis **6**
2.1 Introduction 6
2.2 Cybersecurity Definitions and Common Attacks 7
2.2.1 Cybersecurity 7
2.2.2 Information System 7
2.2.3 The CIA Triad 7
2.2.4 Common Cyberattacks 7
2.2.4.1 Brute Force Attack 8
2.2.4.2 Malware 8
2.2.4.3 Denial of Service and Distributed Denial of Service (DoS / DDoS) 8
2.2.4.4 Privilege Escalation 8
2.2.5 Cybersecurity Actors: White, Gray, and Black Hats 9
2.3 Logs and SIEM Concept 9
2.3.1 Logs 9
2.3.1.1 Types of Logs 10

2.3.1.2 Role of Logs in Security Monitoring . . . 12
2.3.2 Security Information and Event Management . . . 12
2.3.2.1 Definition . . . 12
2.3.2.2 Brief History and Evolution . . . 12
2.3.2.3 Core SIEM Functions . . . 13
2.3.2.4 SIEM Architecture Overview . . . 14
2.3.3 The Relationship Between Logs and SIEM . . . 15
2.4 Overview of Existing SIEM Solutions and Requirements Analysis . . . 15
2.4.1 Insights into Existing SIEM Solutions . . . 15
2.4.2 Cybersecurity Landscape in Algeria . . . 16
2.4.3 Project Objectives . . . 16
2.4.3.1 Short-Term Goals . . . 16
2.4.3.2 Long-Term Goals . . . 17
2.4.4 Target Users of the System . . . 17
2.4.5 System Requirements . . . 18
2.4.5.1 Functional Requirements . . . 18
2.4.5.2 Non-Functional Requirements . . . 18
2.4.6 Conclusion . . . 19

3 Methodology and Design **20**
3.1 Introduction . . . 20
3.2 Use Case Diagram . . . 20
3.2.1 Definition . . . 20
3.2.2 Key Components of a Use Case Diagram for the SIEM Platform . . . 20
3.2.3 Benefits of Use Case Diagrams . . . 21
3.2.4 Presentation of Our Platform's Use Case Diagrams . . . 22
3.2.5 Use Case Diagrams of Our Platform . . . 22
3.2.5.1 Use Case Diagram for SOC Analyst Level 1, 2, and 3 . . . 23
3.2.5.2 Use Case Diagram for Connected SOC Manager . . . 24
3.2.5.3 Use Case Diagram for Connected Detection Engineer . . . 25
3.2.6 Sequence Diagrams for System-Level Interactions . . . 26
3.2.6.1 Sequence diagram for Sign In . . . 27
3.2.6.2 Sequence diagram for Alert Investigation . . . 28
3.2.6.3 Sequence diagram for Alert Handling . . . 29
3.2.6.4 Sequence diagram for Request Analysis . . . 30
3.2.6.5 Sequence diagram for Managing Users . . . 31
3.3 Database Design . . . 31
3.3.1 Introduction to SQL Databases . . . 31
3.3.2 Database Design Process . . . 32

3.3.3 Conceptual Data Model . . . 33
3.3.4 Our Logical Data Design . . . 34
3.4 Conclusion . . . 35

4 Implementation **36**
4.1 Introduction . . . 36
4.2 Technical Stack and Architecture . . . 36
4.2.1 Frontend (React) . . . 37
4.2.2 Backend (Node.js) . . . 37
4.2.3 Database (PostgreSQL) . . . 38
4.2.4 Security Engine (Wazuh) . . . 38
4.2.5 Monitoring (Prometheus) . . . 38
4.2.6 Virtualization (VMware) . . . 38
4.3 Features of Our SIEM Platform . . . 38
4.3.1 Login . . . 39
4.3.2 Dashboard . . . 40
4.3.3 Agent Management . . . 42
4.3.4 Log Management . . . 44
4.3.5 Alert Management . . . 46
4.3.6 Incident Management . . . 48
4.3.7 Detection Rule Management . . . 50
4.3.8 Reporting . . . 52
4.3.9 Role-Based Access Control . . . 54
4.4 Conclusion . . . 57

5 Tests and Evaluation **58**
5.1 Introduction . . . 58
5.2 Functional Testing . . . 58
5.3 Attack Simulation and Detection Evaluation . . . 58
5.3.1 Scenario 1: SSH Brute-Force Attack (Hydra) . . . 59
5.3.1.1 Description . . . 59
5.3.1.2 Expected Behavior . . . 59
5.3.1.3 Steps . . . 59
5.3.1.4 Result . . . 59
5.3.2 Scenario 2: Network Port Scan (Nmap) . . . 62
5.3.2.1 Description . . . 62
5.3.2.2 Expected Behavior . . . 62
5.3.2.3 Steps . . . 62
5.3.2.4 Result . . . 62
5.3.3 Scenario 3: Remote Connection using Evil-WinRM (Pass the Hash) 65

5.3.3.1 Description . . . 65
5.3.3.2 Expected Behavior . . . 65
5.3.3.3 Steps . . . 65
5.3.3.4 Result . . . 65
5.3.4 Scenario 4: Privilege Escalation . . . 68
5.3.4.1 Description . . . 68
5.3.4.2 Expected Behavior . . . 68
5.3.4.3 Steps . . . 68
5.3.4.4 Result . . . 68

6 General Conclusion **71**
6.1 Contributions . . . 71
6.2 Limitations . . . 72
6.3 Future Work . . . 72

List of Figures

2.1 Showing Logs using the cmdlet Get-EventLog in PowerShell 10
2.2 Log Files from Windows Event Viewer 11
2.3 Security and Application Logs in Windows Event Viewer 11
2.4 The core operational functions of a SIEM system [14] 14

3.1 Use Case Diagram for SOC Analyst Level 1, 2, and 3 23
3.2 Use Case Diagram for SOC Manager 24
3.3 Use Case Diagram for Detection Engineer 25
3.4 Sequence diagram for Sign In 27
3.5 Sequence diagram for Alert Investigation 28
3.6 Sequence diagram for Incident Handling 29
3.7 Sequence diagram for Request Analysis 30
3.8 Sequence diagram for Managing Users 31
3.9 Conceptual Data Model (CDM) 33

4.1 SIEM system architecture. 37
4.2 Login page of the SIEM platform. 39
4.3 Dashboard view of the SIEM platform. 41
4.4 Agent monitoring view of the SIEM platform. 43
4.5 Logs view of the SIEM platform. 45
4.6 Alerts view of the SIEM platform. 47
4.7 Incidents view of the SIEM platform. 49
4.8 Rule management view of the SIEM platform. 51
4.9 Final reports view of the SIEM platform. 53
4.10 User management view of the SIEM platform. 55
4.11 Groups management view of the SIEM platform. 56

5.1 Hydra brute-force attack running on Kali Linux 60
5.2 Brute-force alert displayed on the SIEM dashboard 61
5.3 Nmap port scan executed from the Kali Linux machine 63
5.4 Nmap port scan generating a large volume of connection attempts 64
5.5 Port scan alert displayed on the SIEM dashboard 64
5.6 Evil-WinRM remote connection executed from the Kali Linux machine . . 66

5.7 Alerts raised on the SIEM dashboard following the Evil-WinRM connection 66
5.8 Alert details for the pass-the-hash detection 67
5.9 PowerShell commands executed through the Evil-WinRM session 69
5.10 Privilege escalation alerts displayed on the SIEM dashboard 69
5.11 Critical alert details for the Domain Admins group membership change . . 70

Chapter 1

General Introduction

1.1 Background

In the digital era, information systems have become the backbone of nearly every sector, including government institutions, banking systems, telecommunications, energy infrastructures, healthcare organizations, and private enterprises. The rapid digitization of services, cloud adoption, remote work environments, and the interconnection of networks have significantly increased the attack surface of modern infrastructures. While digital transformation enhances efficiency and accessibility, it simultaneously introduces complex cybersecurity risks that threaten data confidentiality, system integrity, and service availability (CIA triad).

Globally, cyberattacks are increasing in frequency and sophistication. Organizations are constantly targeted by ransomware campaigns, phishing attacks, insider threats, advanced persistent threats (APT), distributed denial-of-service (DDoS) attacks, and data exfiltration attempts. To address these challenges, enterprises rely on cybersecurity monitoring solutions capable of detecting suspicious activities in real time and enabling rapid incident response. Among these solutions, Security Information and Event Management platforms play a pivotal role.

A SIEM system centralizes logs from various sources such as servers, firewalls, endpoints, applications, and network devices. It aggregates, normalizes, correlates, and analyzes this data to detect security incidents. By applying detection rules and behavioral analytics, SIEM solutions allow Security Operations Centers teams to monitor infrastructure, investigate alerts, and respond to incidents effectively.

In Algeria, digital transformation is accelerating across both public and private sectors. Financial institutions, telecom operators, energy companies, and governmental entities increasingly rely on interconnected information systems. However, cybersecurity moni-

toring solutions within the Algerian market remain largely dependent on foreign vendors. Most existing SIEM platforms deployed locally are international proprietary solutions that are expensive, complex to customize, and not fully adapted to the Algerian regulatory, operational, and linguistic context. Moreover, when organizations use foreign platforms, they lose full control over their own data. Sensitive information such as logs, alerts, and incident records is stored and processed on external systems managed by foreign providers, which raises serious concerns about data privacy and national security.

Moreover, there is currently no fully Algerian-developed SIEM platform specifically designed to address local infrastructure realities, compliance needs, and organizational constraints. This absence represents both a gap and an opportunity. Developing a locally engineered SIEM solution tailored to Algerian organizations can reduce dependency on external technologies, lower implementation costs, foster cybersecurity expertise within our country, and contribute to the national digital sovereignty strategy.

In this context, this project aims to design and implement a SIEM Security Solution adapted to the Algerian context. The platform focuses on log collection, correlation, detection rule management, alert generation, incident tracking, and SOC workflow support, while remaining scalable and modular for future enhancements. This initiative aligns with the broader objective of strengthening cybersecurity posture in Algeria and promoting the emergence of national cybersecurity startups.

1.2 Problem Statement

While Algerian institutions are rapidly digitalizing, cybersecurity monitoring practices remain fragmented and reactive. Many organizations rely primarily on traditional security mechanisms such as firewalls, antivirus software, and perimeter defenses. These tools provide essential protection, they do not offer centralized visibility or advanced correlation capabilities across heterogeneous systems.

Several operational challenges can be identified within the current Algerian cybersecurity landscape:

- Logs generated by servers, endpoints, network devices, and applications are often stored locally without centralized aggregation.
- Security events are rarely correlated across multiple sources, making it difficult to detect complex multi-stage attacks.
- Incident detection is frequently manual and reactive rather than proactive and real-time.

- Organizations lack structured SOC workflows to classify alerts, escalate incidents, and track remediation processes.
- Existing international SIEM solutions are costly, require significant technical expertise, and may not be fully aligned with local regulatory requirements.
- Small and medium-sized enterprises (SMEs) in Algeria often cannot afford enterprise-grade SIEM solutions.

These limitations create critical security blind spots. Without centralized monitoring, suspicious activities such as lateral movement within networks, privilege escalation, brute-force attacks, or data exfiltration attempts may go undetected until significant damage has occurred.

Furthermore, reliance on foreign SIEM platforms introduces additional constraints. Licensing models are often expensive, customization is limited, and technical support may not always be locally available. This dependency reduces technological autonomy and slows incident response processes.

Another major issue is the lack of a structured Security Operations Center workflow adapted to Algerian organizational structures. In many institutions, responsibilities between Level 1 (L1), Level 2 (L2), and Level 3 (L3) analysts are not clearly defined within a centralized monitoring platform. Detection rule engineering, alert triage, and incident management processes are often dispersed and unstandardized.

Consequently, the fundamental question arises:

How can we design and implement a scalable, cost-effective, and locally adapted SIEM solution that enhances real-time threat detection, supports structured SOC operations, and contributes to strengthening cybersecurity resilience in Algeria?

Addressing this problem requires not only technical development but also architectural design that integrates log collection agents, detection engines, rule management modules, alerting systems, and incident management workflows into a unified platform tailored for Algerian organizations.

1.3 Objective and Methodology

The primary objective of this project is to design and develop a comprehensive SIEM Security Solution tailored to the Algerian market. The platform aims to centralize log collection, implement real-time event correlation, enable customizable detection rules, generate actionable alerts, and support structured SOC workflows for incident investigation and response.

More specifically, the solution seeks to:

- Collect and normalize logs from multiple sources including endpoints, servers, and network devices.
- Apply correlation rules to detect suspicious or malicious activities.
- Provide role-based access control for SOC Analysts (L1, L2, L3), Detection Engineers, and SOC Managers.
- Enable alert classification, filtering, and status management.
- Support incident creation and lifecycle tracking.
- Offer a modular architecture to allow future integration of threat intelligence and automation features.

The development approach follows a structured software engineering methodology, beginning with requirement analysis and system modeling using UML diagrams (use case diagrams, sequence diagrams, and the class diagram). The system architecture is designed using a layered and modular approach, separating log ingestion, processing, storage, and presentation layers.

The backend is implemented using a modern web framework with a relational database system for structured data management. The solution also integrates role-based authentication and secure communication mechanisms between agents and the central engine. Development is conducted iteratively, allowing incremental validation of core SIEM functionalities such as log ingestion, rule evaluation, alert generation, and incident management.

1.4 Report Structure

This report is organized to provide a comprehensive presentation of our SIEM Security Solution. While Chapter 1 introduces the project context, the subsequent chapters are structured as follows:

- **Chapter 2** provides the theoretical foundation of our project. It covers key cybersecurity concepts, definitions of logs and SIEM systems, and an analysis of existing solutions. It also examines the cybersecurity landscape in Algeria, defines the target users, and specifies the functional and non-functional requirements of the platform.

- **Chapter 3** describes the design phase of the system, including use case diagrams, sequence diagrams, a class diagram, and the database design based on PostgreSQL.
- **Chapter 4** details the implementation phase, presenting the technical stack, the system architecture, the main features of the platform, and a visual walkthrough of the application interfaces.
- **Chapter 5** presents the testing and evaluation phase, including functional test cases and attack simulation scenarios such as brute-force attacks, port scanning, remote access via Evil-WinRM, and privilege escalation, verifying the detection capabilities of the platform.
- **General Conclusion** summarizes the contributions of the project, discusses its current limitations, and outlines future enhancements to further strengthen the platform.

This structure ensures a progressive presentation from contextual background and problem identification to system design, implementation, and future perspectives.

Chapter 2

Literature Review and Requirements Analysis

2.1 Introduction

Security Information and Event Management systems represent one of the most significant milestones in the evolution of modern cybersecurity infrastructure, born from a simple but powerful idea: to provide organizations with a single, unified platform capable of monitoring their entire IT environment. At its core, a SIEM collects and processes log files generated continuously by devices, applications, and services across a network, normalizes them into a common format, and presents actionable insights to analysts through dashboards, alerts, and reports.

Over the decades, SIEM technology has evolved from basic log aggregation tools to sophisticated platforms incorporating threat intelligence, behavioral analytics, and automated incident response. Despite this progress, many existing solutions remain complex to deploy, costly to maintain, and ill-suited to the realities faced by organizations in developing regions.

This chapter analyzes the current state of SIEM solutions and the cybersecurity landscape in Algeria, identifies the limitations of existing systems, defines the target users of the proposed platform, and specifies the functional and non-functional requirements that guide its design.

2.2 Cybersecurity Definitions and Common Attacks

2.2.1 Cybersecurity

Cybersecurity refers to the set of practices, technologies, and processes designed to protect digital systems, networks, and data from unauthorized access, damage, or attacks. It encompasses all measures taken to ensure the confidentiality, integrity, and availability of information in an increasingly interconnected digital environment [18].

2.2.2 Information System

An information system (IS) is an organized set of resources — including hardware, software, data, personnel, and communication networks — that collects, processes, stores, and distributes information within an organization [17].

2.2.3 The CIA Triad

The CIA triad is the foundational model of information security, built upon three core principles that any secure system must uphold.

- **Confidentiality:** Ensures that information is accessible only to those authorized to access it. Breaches of confidentiality include unauthorized data disclosure, credential theft, and espionage.
- **Integrity:** Guarantees that information remains accurate and unaltered except through authorized processes. Integrity violations include data tampering, unauthorized modification of logs, and man-in-the-middle attacks.
- **Availability:** Ensures that systems and data are accessible to authorized users whenever needed. Availability is threatened by denial-of-service attacks, hardware failures, and ransomware.

A SIEM system directly supports the CIA triad by continuously monitoring for events that indicate a violation of any of these three principles.

2.2.4 Common Cyberattacks

The following attack types represent the most frequently encountered threats in modern IT environments and are among the primary categories of events that a SIEM system is designed to detect.

2.2.4.1 Brute Force Attack

A brute force attack consists of systematically attempting all possible combinations of credentials until the correct one is found [20]. This type of attack directly targets user authentication mechanisms and generates a high volume of failed login attempts, which are captured in authentication logs and serve as a key detection signal for SIEM systems.

2.2.4.2 Malware

Malware (malicious software) is any program designed to disrupt, damage, or gain unauthorized access to a system. It encompasses several subtypes, including:

- **Viruses:** Self-replicating programs that attach to legitimate files and spread across systems.
- **Ransomware:** Malware that encrypts a victim's data and demands payment for the decryption key.
- **Trojans:** Programs disguised as legitimate software that provide attackers with backdoor access.

Malware activity generates distinctive patterns in endpoint and system logs, making it an important detection target for SIEM platforms.

2.2.4.3 Denial of Service and Distributed Denial of Service (DoS / DDoS)

A Denial of Service (DoS) attack aims to make a system or network resource unavailable by overwhelming it with illegitimate requests [21]. When carried out from multiple distributed sources simultaneously, it becomes a Distributed Denial of Service (DDoS) attack. These attacks directly threaten the availability pillar of the CIA triad and produce anomalous traffic patterns detectable through network logs.

2.2.4.4 Privilege Escalation

Privilege escalation occurs when an attacker gains higher-level access rights than originally granted, either by exploiting vulnerabilities in the system or by abusing misconfigured permissions. This type of attack allows adversaries to move from a low-privilege account to administrative control, and leaves traces in access control and system logs that correlation rules can identify.

2.2.5 Cybersecurity Actors: White, Gray, and Black Hats

In the cybersecurity field, practitioners and adversaries are commonly categorized by their intent and the legality of their actions. This classification uses the metaphor of hat colors, borrowed from old Western films where heroes wore white hats and villains wore black ones.

- **White Hat (Ethical Hacker):** White hat hackers are security professionals who use their skills legally and with explicit authorization. They perform penetration testing, vulnerability assessments, and security audits to help organizations identify and fix weaknesses before malicious actors can exploit them. Their work is governed by contracts and operates strictly within legal boundaries.
- **Gray Hat:** Gray hat hackers occupy an ambiguous position between ethical and malicious hacking. They may probe systems without explicit authorization, but typically without malicious intent — often disclosing discovered vulnerabilities to the affected organization, sometimes in exchange for a fee. Their activities exist in a legal gray area, as unauthorized access is generally illegal regardless of intent.
- **Black Hat (Malicious Hacker):** Black hat hackers operate with malicious intent and without authorization. They exploit vulnerabilities for personal gain, financial profit, espionage, or sabotage. The attacks described in Section 2.2.4 are primarily carried out by black hat actors, and the events they generate within IT infrastructure are precisely what a SIEM system is designed to detect and respond to.

2.3 Logs and SIEM Concept

2.3.1 Logs

A log is a timestamped record automatically generated by an operating system, application, network device, or security tool to document events and activities occurring within a digital system [19]. Each log entry typically contains a set of structured fields including the date and time of the event, its source, the type of action performed, and its outcome. Logs serve as the primary source of truth for understanding what has happened within an IT environment, making them an indispensable resource for security monitoring, troubleshooting, and forensic investigation.

2.3.1.1 Types of Logs

Different components of an IT infrastructure generate different categories of logs, each capturing a distinct aspect of system activity.

- **System Logs:** Document operating system-level events such as service starts and stops, hardware errors, kernel messages, and configuration changes. They provide visibility into the health and integrity of individual machines.
- **Security Logs:** Produced by dedicated security tools such as firewalls, intrusion detection systems (IDS), antivirus software, and endpoint detection and response (EDR) solutions. They directly report on detected threats, policy violations, and blocked activities.
- **Application Logs:** Generated by software applications to record user interactions, errors, transactions, and internal state changes. Web servers, databases, and enterprise applications all produce application logs that can reveal injection attacks, data exfiltration, or abnormal usage patterns.
- **Authentication Logs:** Record all login attempts, logouts, password changes, and privilege assignments. They are essential for detecting unauthorized access, brute force attacks, and account compromise.
- **Network Logs:** Capture traffic flows, connection requests, firewall decisions, and protocol activity across the network. They are critical for identifying anomalous communication patterns, port scanning, and denial-of-service attempts.

```
PS C:\Users\ALEM> Get-EventLog

cmdlet Get-EventLog at command pipeline position 1
Supply values for the following parameters:
LogName: *

  Max(K) Retain OverflowAction       Entries Log
  ------ ------ --------------       ------- ---
  20 480      0 OverwriteAsNeeded      25 420 Application
  20 480      0 OverwriteAsNeeded           0 HardwareEvents
     512      7 OverwriteOlder              0 IntelAudioServiceLog
     512      7 OverwriteOlder              0 Internet Explorer
  20 480      0 OverwriteAsNeeded           0 Key Management Service
     128      0 OverwriteAsNeeded           0 OAlerts
     512      7 OverwriteOlder          3 177 OneApp_IGCC
                                              Security
  20 480      0 OverwriteAsNeeded       5 904 System
  15 360      0 OverwriteAsNeeded         589 Windows PowerShell
```

Figure 2.1: Showing Logs using the cmdlet Get-EventLog in PowerShell

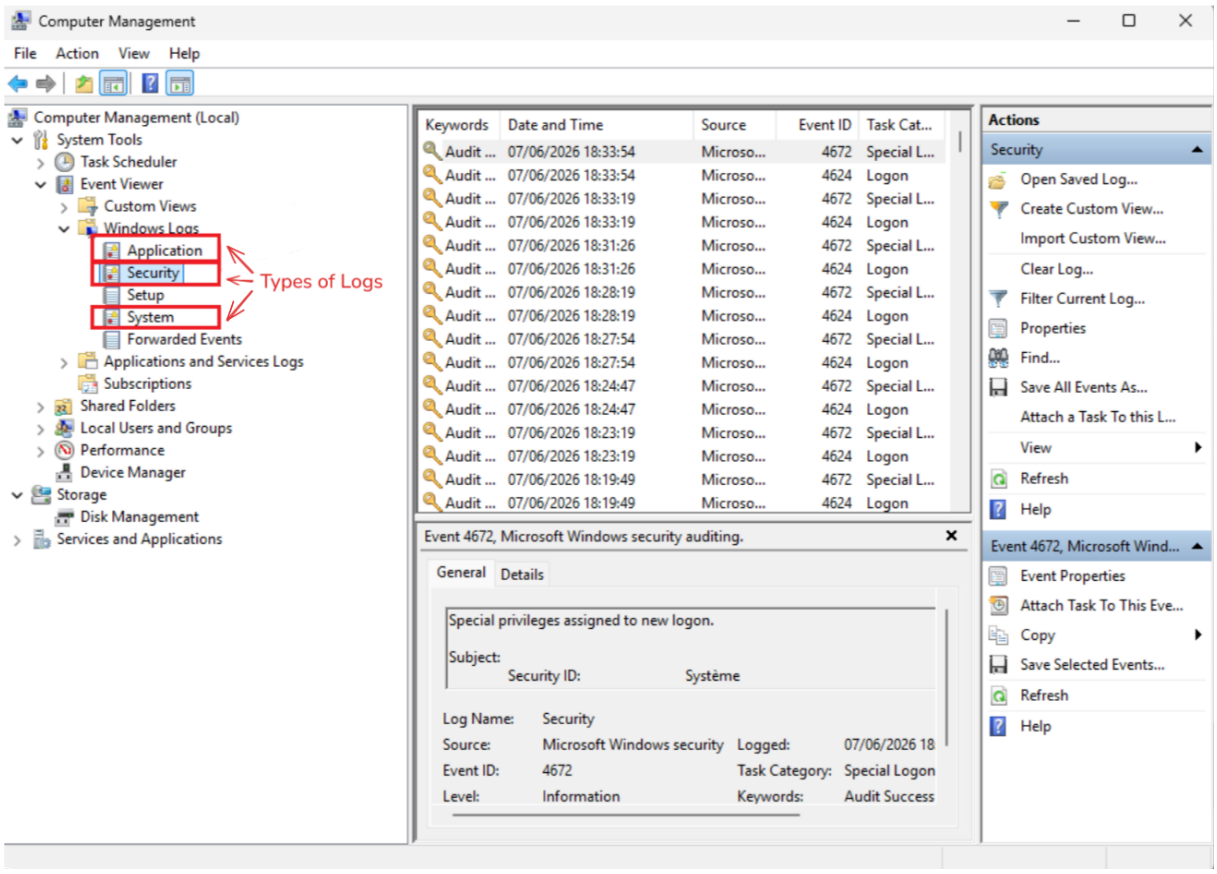


Figure 2.2: Log Files from Windows Event Viewer

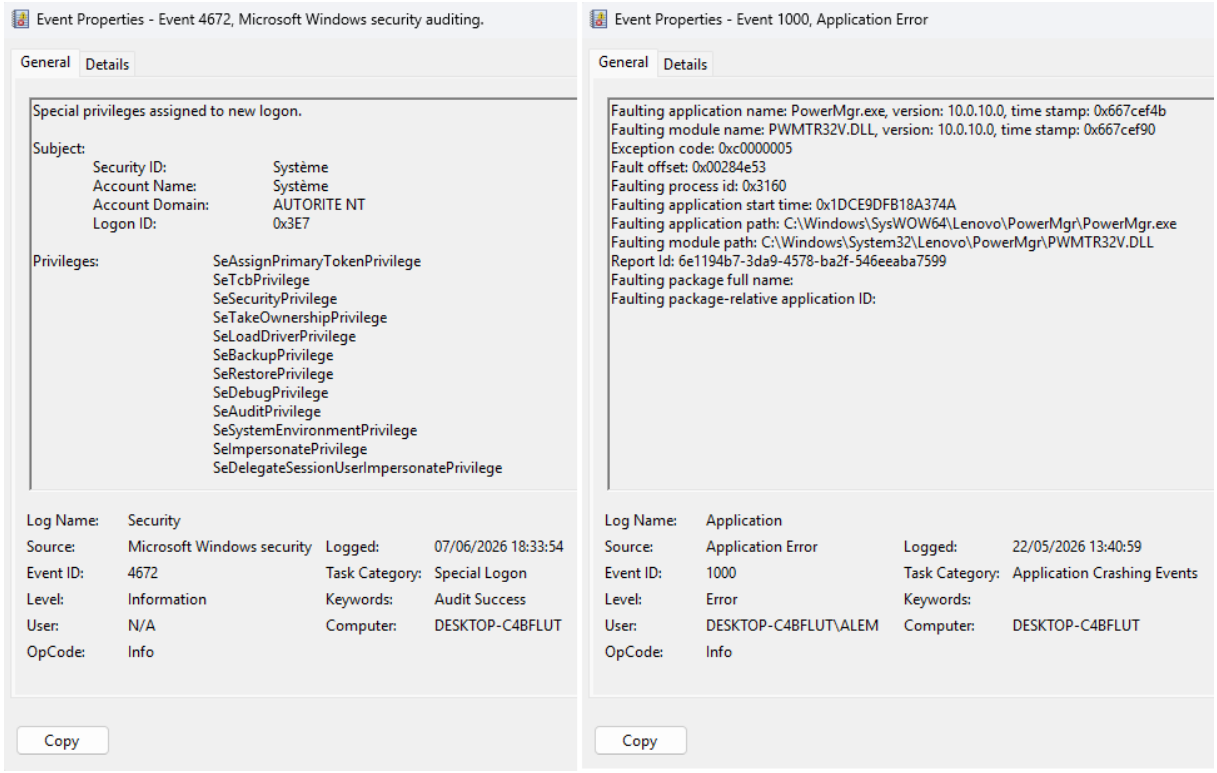


Figure 2.3: Security and Application Logs in Windows Event Viewer

2.3.1.2 Role of Logs in Security Monitoring

Logs play a central role in every phase of the security monitoring lifecycle. During normal operations, they provide continuous visibility into system behavior, enabling analysts to establish baselines of expected activity. When an incident occurs, logs become the primary evidence source for detection, investigation, and reconstruction of the attack timeline. In the aftermath of an incident, they support forensic analysis, compliance reporting, and the refinement of detection rules.

However, the sheer volume and heterogeneity of logs generated across a modern IT environment make manual analysis impractical. Authentication logs, network traffic records, and application events are produced at a rate of thousands of entries per second in large organizations, each in its own format and structure. This reality necessitates automated tools capable of aggregating, normalizing, and analyzing logs at scale — a need that SIEM systems are specifically designed to address.

2.3.2 Security Information and Event Management

2.3.2.1 Definition

A Security Information and Event Management system is a centralized platform that combines two complementary disciplines: Security Information Management (SIM), which focuses on the long-term storage and analysis of log data, and Security Event Management (SEM), which focuses on real-time monitoring and correlation of security events. Together, they form a unified solution capable of providing organizations with comprehensive visibility into their security posture, enabling rapid detection of threats, and supporting structured incident response.

2.3.2.2 Brief History and Evolution

The concept of SIEM emerged in the early 2000s, coined by analysts Mark Nicolett and Amrit Williams of Gartner, who recognized the need to unify log management and real-time event monitoring into a single platform. Early SIEM solutions were primarily focused on log aggregation and compliance reporting, helping organizations meet regulatory requirements such as PCI-DSS and SOX.

Over time, SIEM platforms evolved significantly. The second generation introduced real-time correlation engines capable of linking related events across multiple sources to identify complex attack patterns. More recent generations have incorporated User and Entity Behavior Analytics (UEBA), machine learning-based anomaly detection, and integration

with Security Orchestration, Automation, and Response (SOAR) platforms, enabling not only detection but also automated response to security incidents.

2.3.2.3 Core SIEM Functions

A modern SIEM platform performs the following core functions:

- **Log Collection:** The SIEM ingests log data from a wide variety of sources across the IT environment, including endpoints, servers, network devices, cloud services, and security tools. Collection is typically performed through lightweight agents installed on monitored systems or via agentless protocols such as Syslog and SNMP.
- **Normalization:** Logs collected from heterogeneous sources arrive in different formats and structures. The SIEM normalizes this data into a common schema, allowing events from different sources to be compared, correlated, and analyzed uniformly.
- **Correlation:** The correlation engine applies predefined rules and logic to identify relationships between events that, in isolation, may appear benign but together indicate a security threat. For example, a failed login followed by a successful one from an unusual location may trigger a compromised account alert.
- **Alerting:** When a correlation rule is triggered or a threshold is exceeded, the SIEM generates an alert and notifies the appropriate security personnel. Alerts are prioritized by severity to help analysts focus on the most critical threats first.
- **Reporting:** The SIEM produces structured reports summarizing security events, alert trends, and incident statistics over a given period. These reports support both operational decision-making and compliance obligations.



Figure 2.4: The core operational functions of a SIEM system [14]

2.3.2.4 SIEM Architecture Overview

The SIEM architecture uses a layered design that separates log ingestion, processing, storage, correlation, and presentation. The main architectural components are:

- **Log collectors and agents:** Components deployed on monitored assets to gather logs and forward them securely to the platform.
- **Ingestion and normalization layer:** Receives raw events, normalizes diverse log formats, and prepares them for analysis.
- **Correlation and detection engine:** Applies detection rules and correlates events across multiple sources to identify suspicious patterns.
- **Alert manager:** Stores generated alerts, supports triage workflows, and tracks alert lifecycle states.
- **Incident management module:** Converts alerts into incidents, supports escalation, and manages response tracking.
- **Presentation layer:** Provides dashboards, alert views, investigation screens, and administrative interfaces.

This architecture supports scalability by keeping the ingestion pipeline separate from the analysis engines and from the user interface. It also allows future enhancement with threat intelligence feeds, machine learning-based anomaly detection, and automation workflows.

2.3.3 The Relationship Between Logs and SIEM

Logs and SIEM systems are inseparable by design. A SIEM without logs has no data to process, no events to correlate, and no threats to detect. Conversely, logs without a SIEM remain scattered, unanalyzed, and largely invisible to the security team — a vast reservoir of information with no mechanism to extract its value.

The SIEM transforms raw log data into actionable security intelligence through a structured pipeline: logs are collected from across the IT environment, normalized into a unified format, enriched with contextual information, and evaluated against correlation rules that encode knowledge of known attack patterns. The result is a continuous stream of prioritized alerts that guide analysts toward real threats, dramatically reducing the time between initial compromise and detection.

This interdependency means that the quality of a SIEM's detection capability is directly tied to the completeness and quality of its log sources. An organization that logs authentication events but neglects network traffic, for instance, will have blind spots that attackers can exploit undetected. A well-designed SIEM platform must therefore not only process logs efficiently but also guide operators in ensuring that the right sources are monitored — a principle that directly informs the architecture of the system presented in the following sections.

2.4 Overview of Existing SIEM Solutions and Requirements Analysis

2.4.1 Insights into Existing SIEM Solutions

Despite the availability of both commercial and open-source SIEM solutions, most of these platforms are developed for global markets and are not specifically tailored to the needs of local environments. In the context of Algeria, no fully localized SIEM solution designed to address the specific operational, economic, and organizational constraints of Algerian institutions has been widely adopted.

As a result, organizations often rely on foreign solutions that may be costly, complex to deploy, and difficult to adapt to local requirements. This situation highlights the need for a SIEM platform that is both accessible and aligned with the specific needs of the Algerian cybersecurity landscape.

2.4.2 Cybersecurity Landscape in Algeria

The rapid digital transformation in Algeria has led to increased reliance on information systems across various sectors, including government, finance, telecommunications, and energy. While this transformation enhances operational efficiency, it also exposes organizations to a growing number of cybersecurity threats.

Currently, many Algerian organizations rely on traditional security mechanisms such as firewalls and antivirus software. These tools, although essential, do not provide centralized visibility or advanced correlation capabilities across multiple systems. As a result, security monitoring remains fragmented and reactive rather than proactive.

Another major challenge is the absence of structured Security Operations Center workflows. In many organizations, responsibilities between different levels of analysts (L1, L2, and L3) are not clearly defined, leading to inefficiencies in alert handling and incident response.

Furthermore, the adoption of foreign SIEM solutions introduces additional constraints, including high costs, limited customization, and dependency on external vendors. Small and medium-sized enterprises, in particular, often lack the financial and technical resources required to deploy and maintain such systems.

These challenges highlight the need for a locally adapted SIEM solution that addresses the specific requirements of Algerian organizations while remaining cost-effective and scalable.

2.4.3 Project Objectives

The proposed SIEM platform aims to provide a structured and scalable solution for security monitoring and incident management. The objectives of the project are divided into short-term and long-term goals.

2.4.3.1 Short-Term Goals

The short-term goals focus on the development of a functional and reliable prototype that addresses the essential needs of a Security Operations Center.

- Design and implement a basic SIEM architecture capable of collecting and processing logs from multiple sources.
- Develop a rule-based detection engine to identify suspicious activities and generate alerts.
- Provide a user-friendly interface for monitoring alerts and managing incidents.

- Implement role-based access control to support different types of SOC users.
- Ensure real-time or near real-time alert generation and visualization.

2.4.3.2 Long-Term Goals

The long-term goals aim to enhance the system's capabilities and extend its applicability to more advanced cybersecurity scenarios.

- Integrate advanced correlation techniques and machine learning for anomaly detection.
- Support large-scale deployments with improved scalability and distributed architecture.
- Incorporate threat intelligence feeds to enrich alerts and improve detection accuracy.
- Develop automated response mechanisms to reduce incident response time.
- Adapt the platform to comply with international cybersecurity standards and best practices.

2.4.4 Target Users of the System

The SIEM platform is designed to support different categories of users within a Security Operations Center. Each user interacts with the system according to specific responsibilities and access levels.

- **SOC Analyst Level 1 (L1):** Responsible for continuous monitoring of incoming alerts, performing initial triage to determine their relevance and priority, and escalating suspicious cases to higher-level analysts.
- **SOC Analyst Level 2 (L2):** Conducts in-depth investigation of alerts, analyzes suspicious activities, and validates true positive incidents before escalation.
- **SOC Analyst Level 3 (L3):** Handles advanced incident response, manages the incident lifecycle, and ensures proper remediation and recovery.
- **Detection Engineer:** Designs, tests, and maintains detection rules, ensuring accurate and efficient threat detection.
- **SOC Manager:** Supervises the overall SOC operations, manages users and access rights, and ensures coordination between different teams.

These users interact with the SIEM platform through role-based access, ensuring efficient collaboration and structured security operations.

2.4.5 System Requirements

This section defines the requirements that the SIEM platform must satisfy. These requirements are divided into functional and non-functional categories.

2.4.5.1 Functional Requirements

The system must provide the following functionalities:

- Collect logs from multiple sources, including endpoints, servers, and network devices.
- Normalize and store log data for efficient processing and analysis.
- Apply detection rules to identify suspicious or malicious activities.
- Generate alerts based on predefined rules and event correlation.
- Allow users to view, filter, and manage alerts.
- Support incident creation and lifecycle management.
- Provide role-based access control for different types of users.
- Enable creation, modification, and testing of detection rules.

2.4.5.2 Non-Functional Requirements

In addition to functional capabilities, the system must satisfy the following quality attributes:

- **Performance:** The system must process large volumes of log data efficiently and provide near real-time alerting.
- **Scalability:** The architecture must support increasing data volumes and additional agents without significant performance degradation.
- **Security:** Communication between components must be secure, and access to the system must be controlled through authentication and authorization mechanisms.

- **Usability:** The user interface must be intuitive and accessible to different types of users.
- **Reliability:** The system must ensure high availability and data integrity.

2.4.6 Conclusion

This chapter provided an overview of existing SIEM solutions and highlighted their limitations in the context of Algerian organizations. It also examined the current cybersecurity landscape in Algeria, emphasizing the need for a centralized and structured monitoring solution.

Furthermore, the chapter identified the target users of the system and defined the functional and non-functional requirements that guide the design of the proposed SIEM platform. These elements form the foundation for the system conception and architectural design presented in the next chapter.

Chapter 3

Methodology and Design

3.1 Introduction

This chapter describes a conceptual model of how the system operates. We begin by presenting the use case diagrams, followed by the sequence diagrams, and then provide a detailed explanation of each use case. We then present the class diagram, which models the system's core entities and their relationships, and conclude with the database design that supports the platform's data management requirements.

3.2 Use Case Diagram

3.2.1 Definition

A use case diagram is a visual representation of a system's functionality from the user's perspective. It helps to identify the interactions between users (actors) and system functions (use cases). These diagrams are widely used in software engineering and system analysis to model system behavior [15].

3.2.2 Key Components of a Use Case Diagram for the SIEM Platform

The main components of a use case diagram, applied to our SIEM platform, are described below:

- **Actors:** Represent the different users or systems that interact with the SIEM platform.

  - **Endpoint Agent:** Collects and sends logs from monitored systems to the SIEM.
  - **SOC Analysts (L1, L2, L3):** Monitor alerts, investigate incidents, and perform triage and advanced analysis.
  - **Detection Engineer:** Creates, updates, tests, and manages detection rules.
  - **SOC Manager:** Supervises system operations, manages user accounts, and ensures compliance.

- **Use Cases:** Represent the services or actions that the SIEM platform provides to its actors.
  - Examples include *collect logs, view alerts, filter events, create incidents, manage rules, manage users*, etc.

- **System Boundary:** Defines the scope of the SIEM platform and encloses all the system's use cases.

- **Relationships:**
  - **Association:** Links actors to their use cases (solid line).
  - **Include:** Represents mandatory functionality that is always executed within another use case (e.g., *collect logs* included in *generate alert*).
  - **Extend:** Represents optional behavior that extends a base use case under certain conditions (e.g., *notify manager* may extend *create incident*).
  - **Generalization:** Represents inheritance among actors or use cases (e.g., L2 and L3 analysts inherit base L1 monitoring capabilities).

3.2.3 Benefits of Use Case Diagrams

Use case diagrams offer numerous advantages, including:

- Help in requirement gathering and defining system scope.
- Provide clear communication between stakeholders.
- Support identification of system functionalities and user interactions.
- Act as a foundation for system design and documentation.

3.2.4 Presentation of Our Platform's Use Case Diagrams

This section presents the use case diagrams for each primary actor in the SIEM platform:

- **SOC Analyst Level 1 (Figure 3.1):** Monitoring and initial triage tasks.
- **SOC Analyst Level 2 (Figure 3.1):** Investigation and incident creation.
- **SOC Analyst Level 3 (Figure 3.1):** Incident management and advanced response.
- **SOC Manager (Figure 3.2):** User and system management.
- **Detection Engineer (Figure 3.3):** Rule creation and maintenance.

3.2.5 Use Case Diagrams of Our Platform

This section presents the various use case diagrams relevant to the primary actors in our SIEM platform:

3.2.5.1 Use Case Diagram for SOC Analyst Level 1, 2, and 3

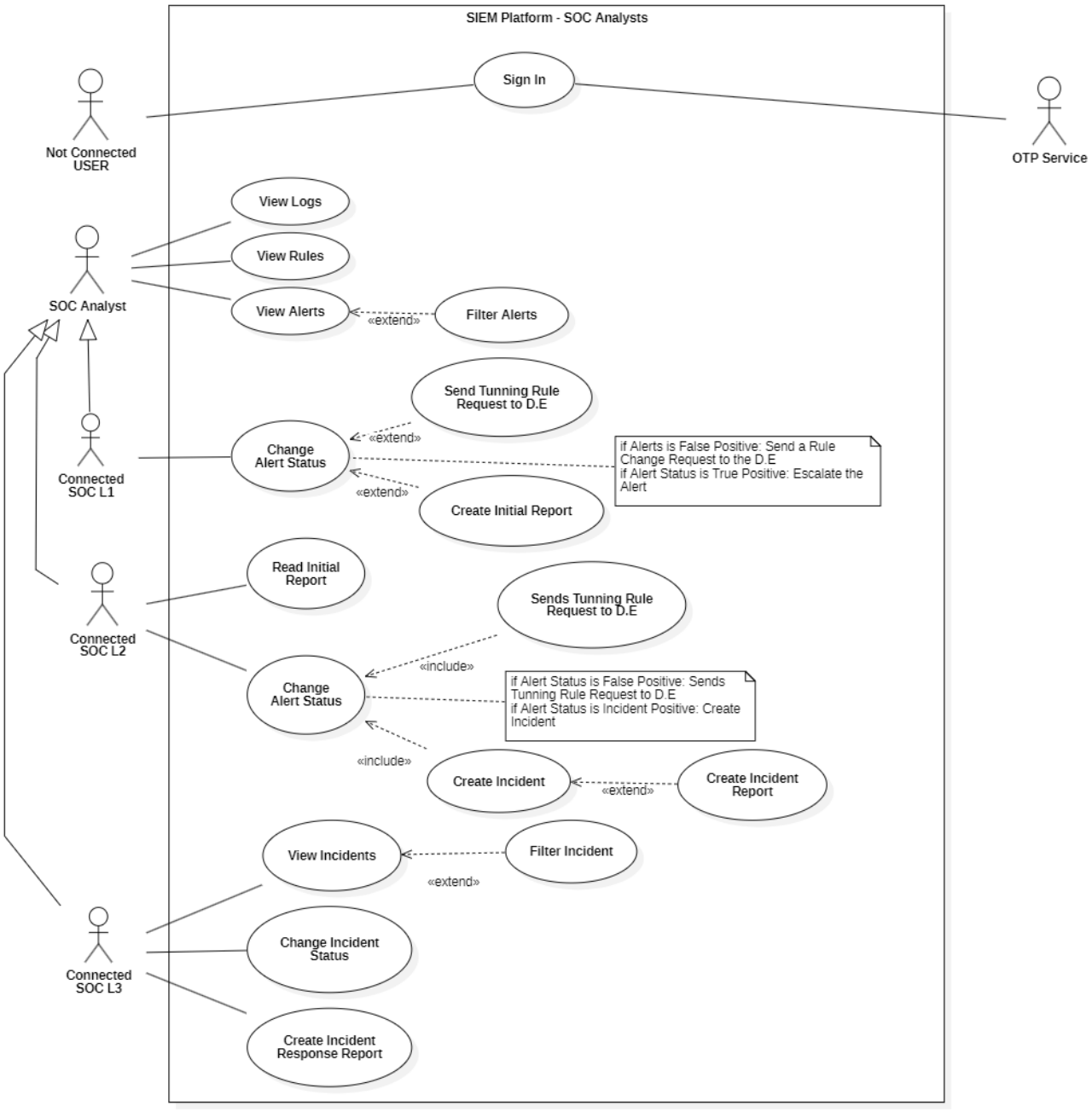


Figure 3.1: Use Case Diagram for SOC Analyst Level 1, 2, and 3

3.2.5.2 Use Case Diagram for Connected SOC Manager

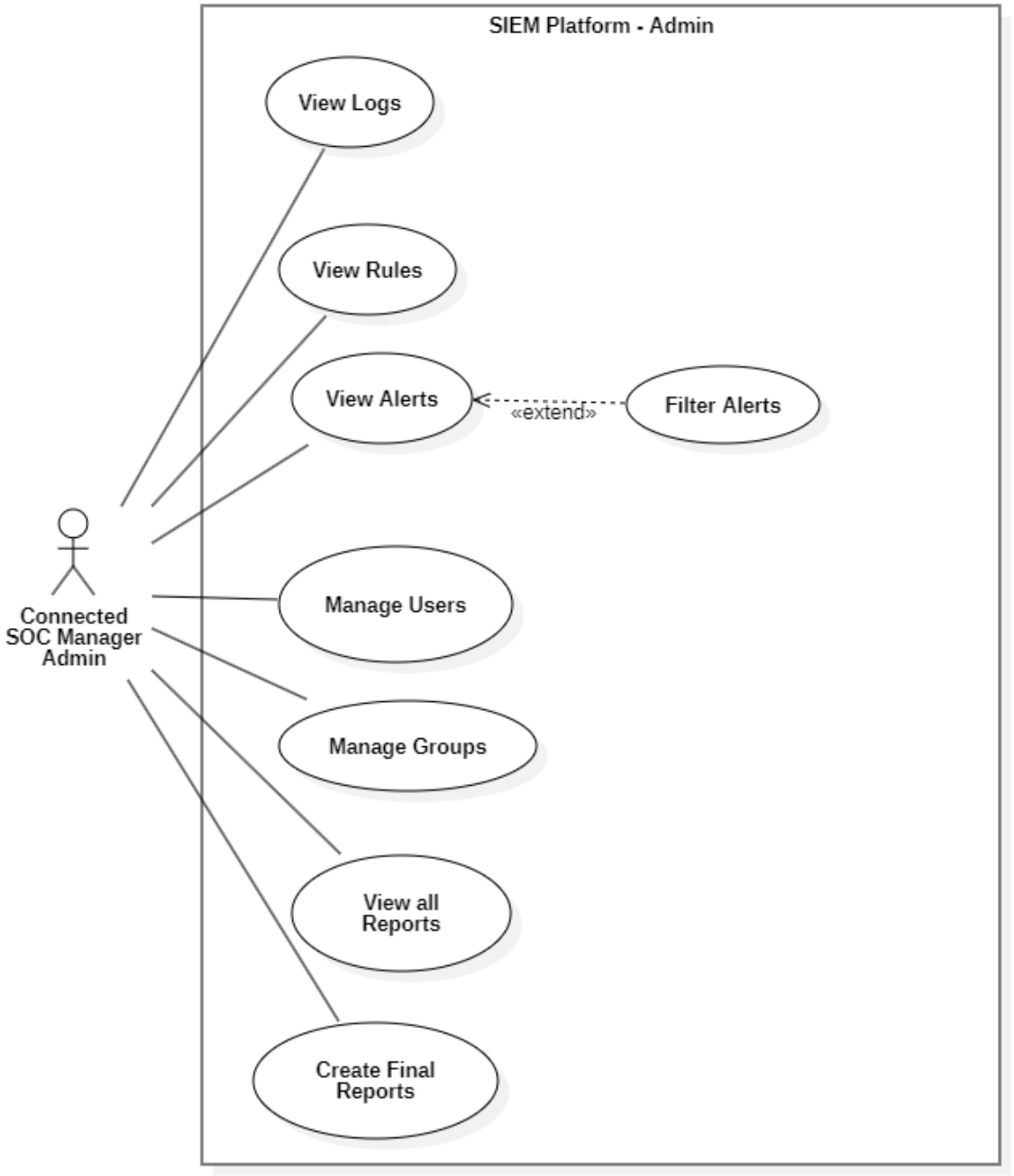


Figure 3.2: Use Case Diagram for SOC Manager

3.2.5.3 Use Case Diagram for Connected Detection Engineer

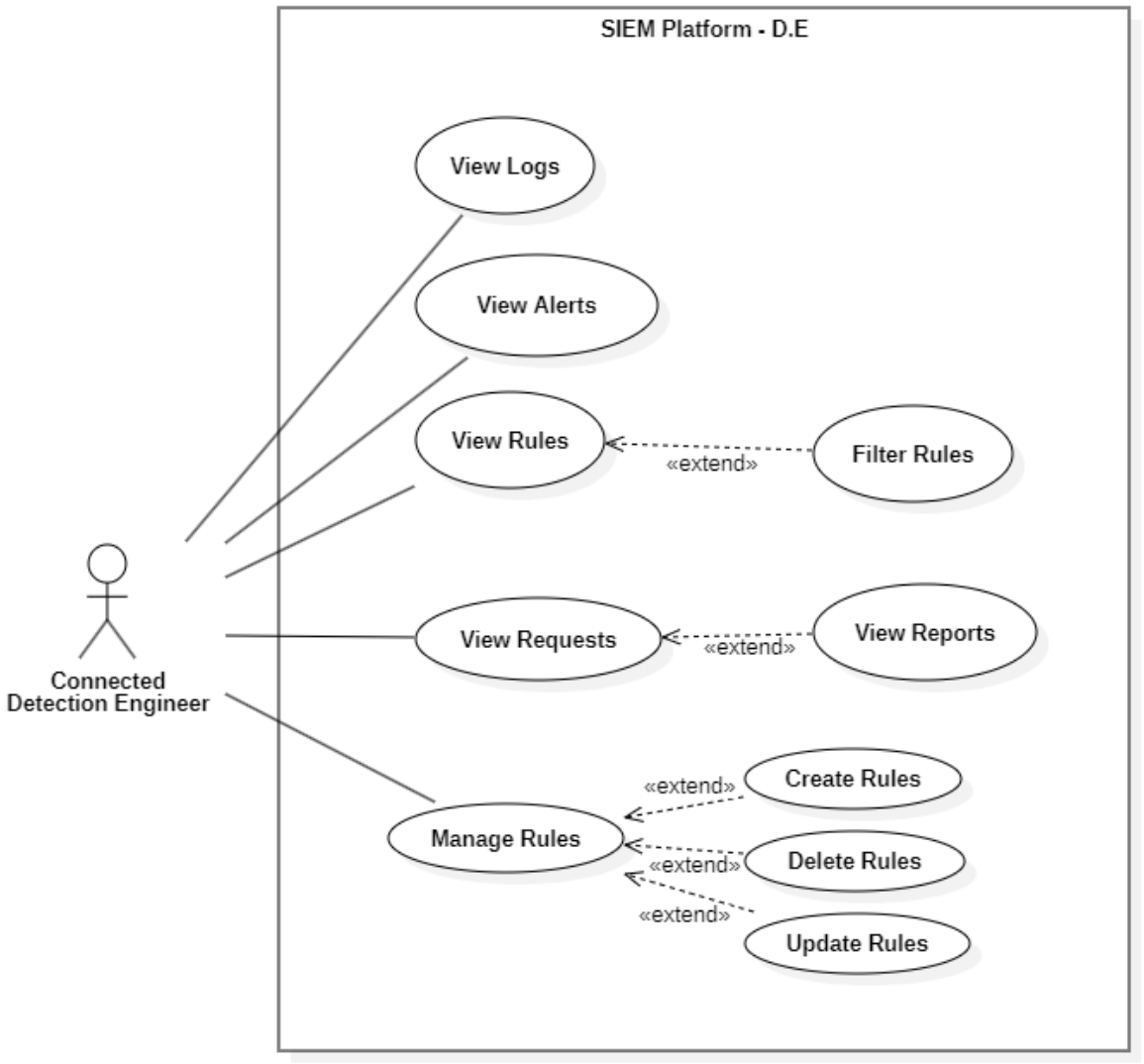


Figure 3.3: Use Case Diagram for Detection Engineer

These diagrams illustrate the interactions between each actor and the system, providing a clear understanding of system functionalities and user roles.

3.2.6 Sequence Diagrams for System-Level Interactions

This section presents several sequence diagrams illustrating system-level interactions for selected use cases within the SIEM platform. These diagrams, commonly referred to as System Sequence Diagrams (SSDs), depict the interactions between the system and the involved actors in order to achieve the intended functionality of each use case.

In what follows, we present the SSD corresponding to the use case *Sign In*, along with the SSDs of the related system interactions. The objective is to illustrate the overall workflow of the SIEM platform, from user authentication to alert analysis, incident handling, and system administration.

Specifically, we present the SSDs corresponding to the following use cases:

- **Sign In** (Figure 3.4).
- **Alert Investigation** (Figure 3.5).
- **Incident Handling** (Figure 3.6).
- **Request Analysis** (Figure 3.7).
- **Managing Users** (Figure 3.8).

3.2.6.1 Sequence diagram for Sign In

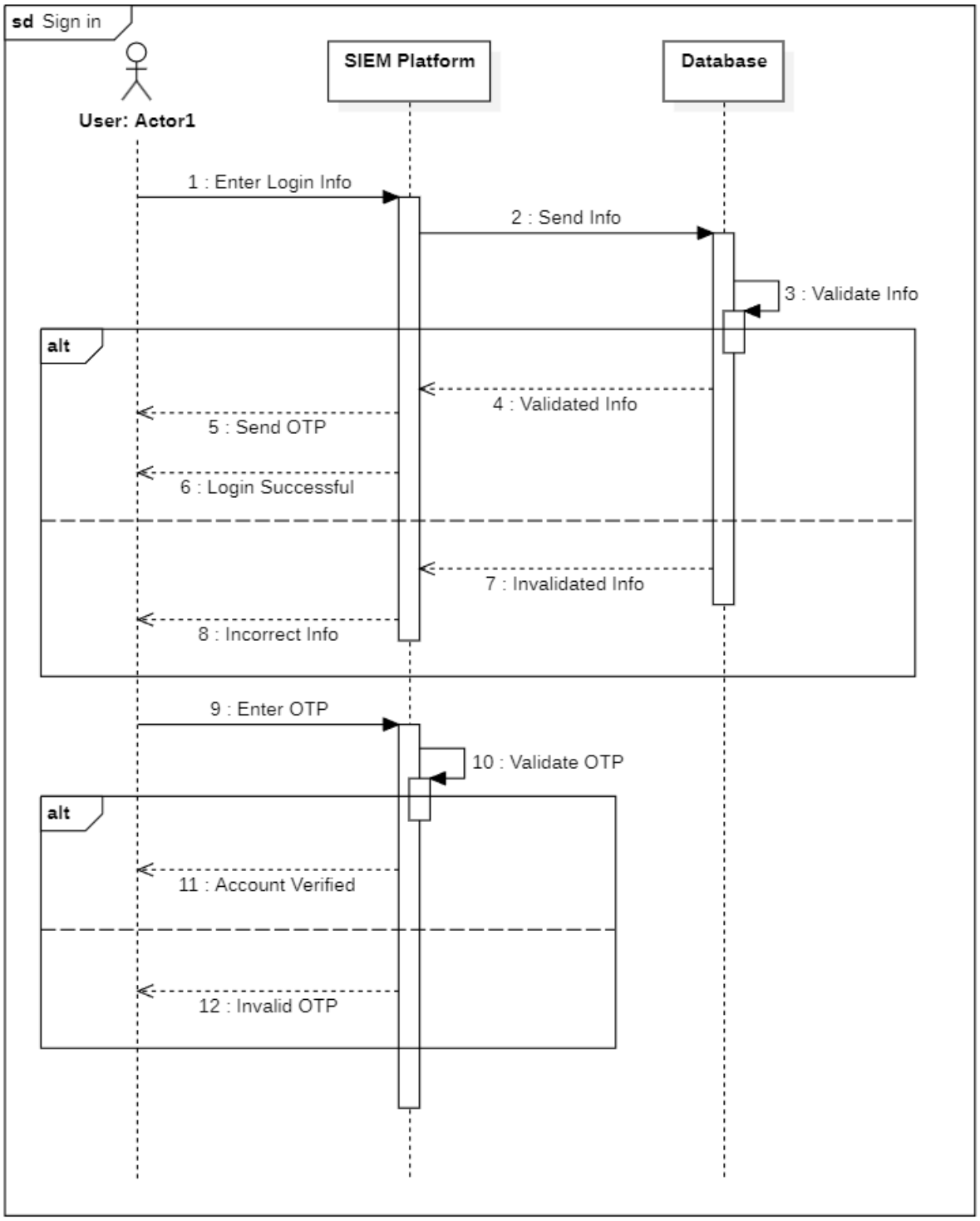


Figure 3.4: Sequence diagram for Sign In

3.2.6.2 Sequence diagram for Alert Investigation

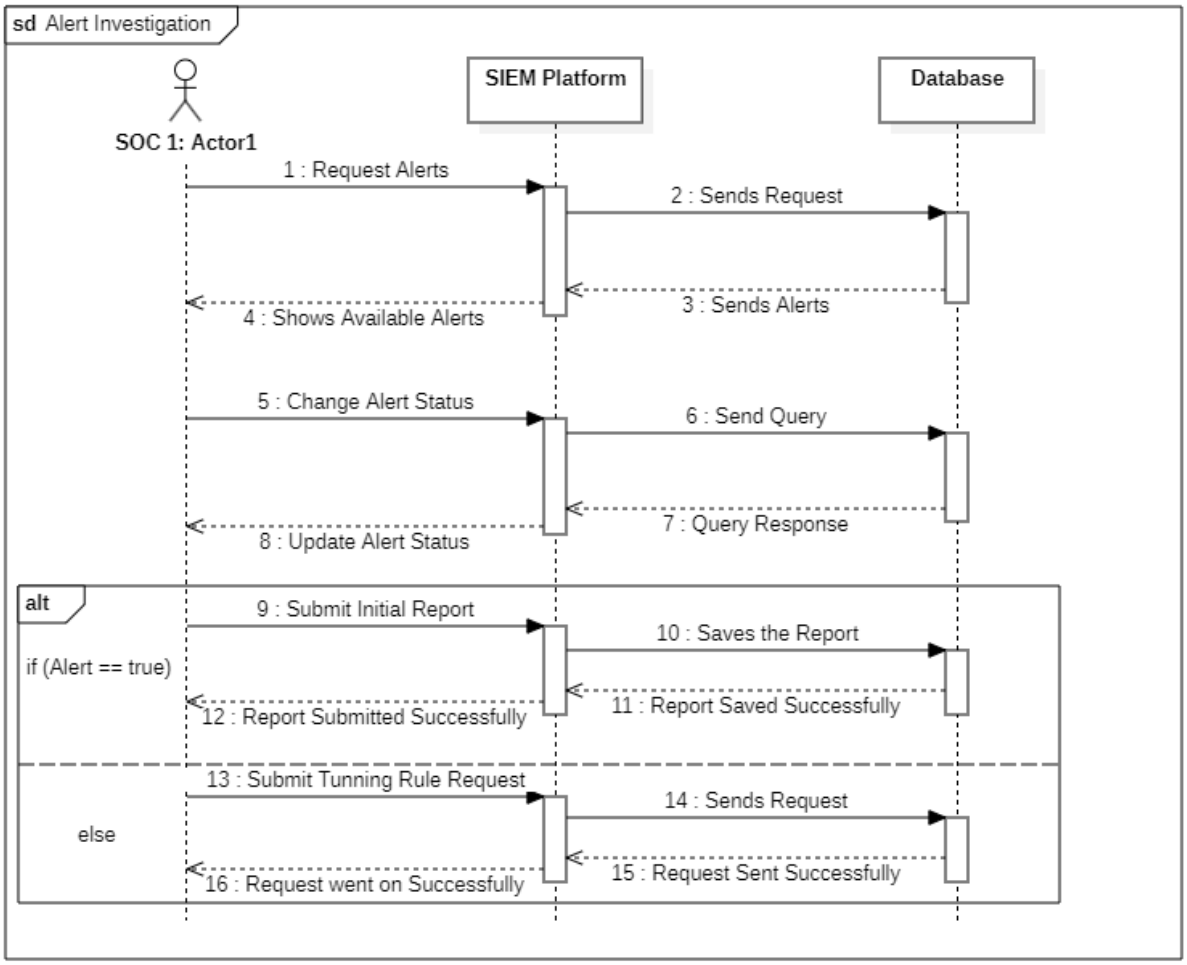


Figure 3.5: Sequence diagram for Alert Investigation

3.2.6.3 Sequence diagram for Alert Handling

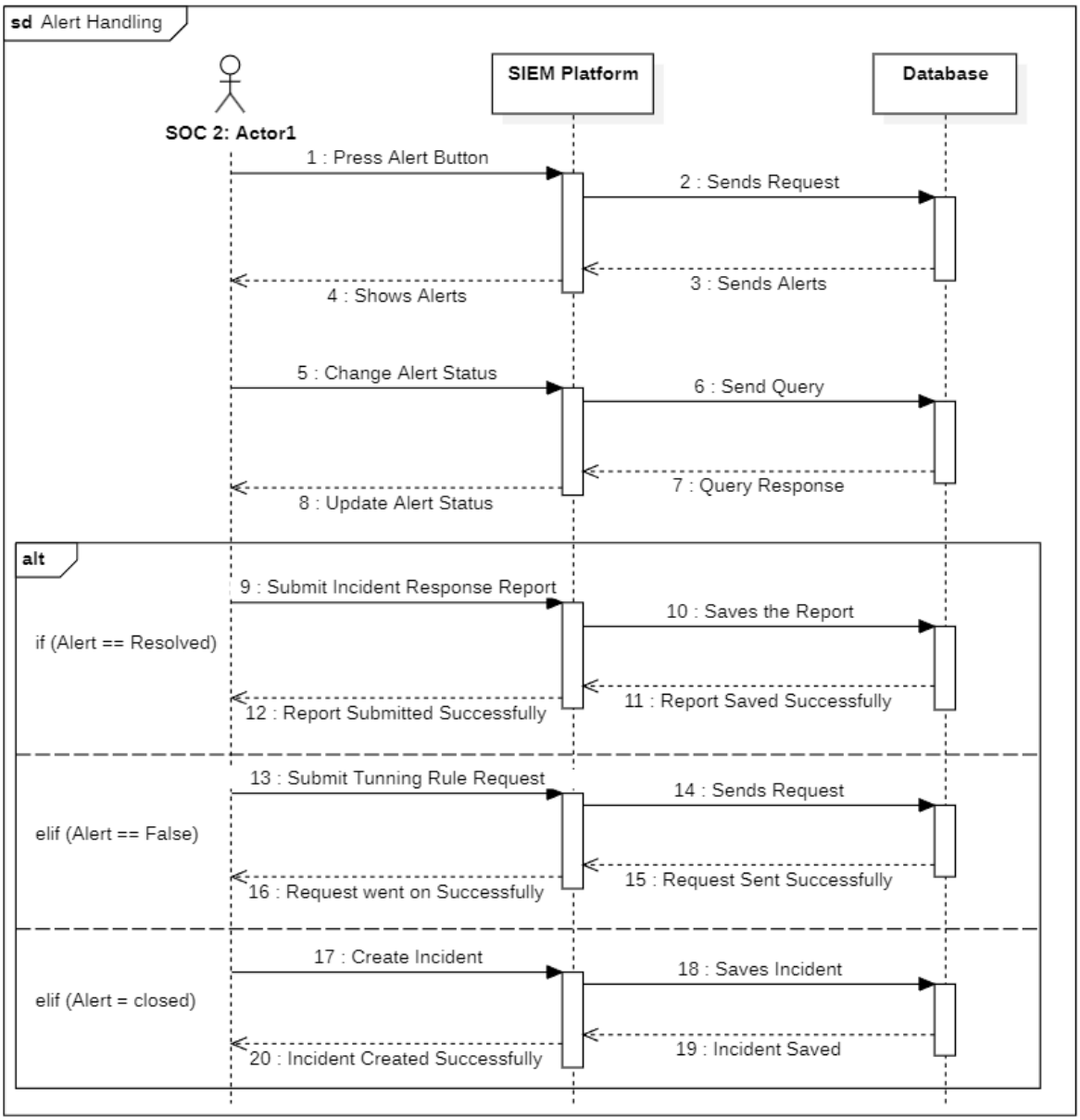


Figure 3.6: Sequence diagram for Incident Handling

3.2.6.4 Sequence diagram for Request Analysis

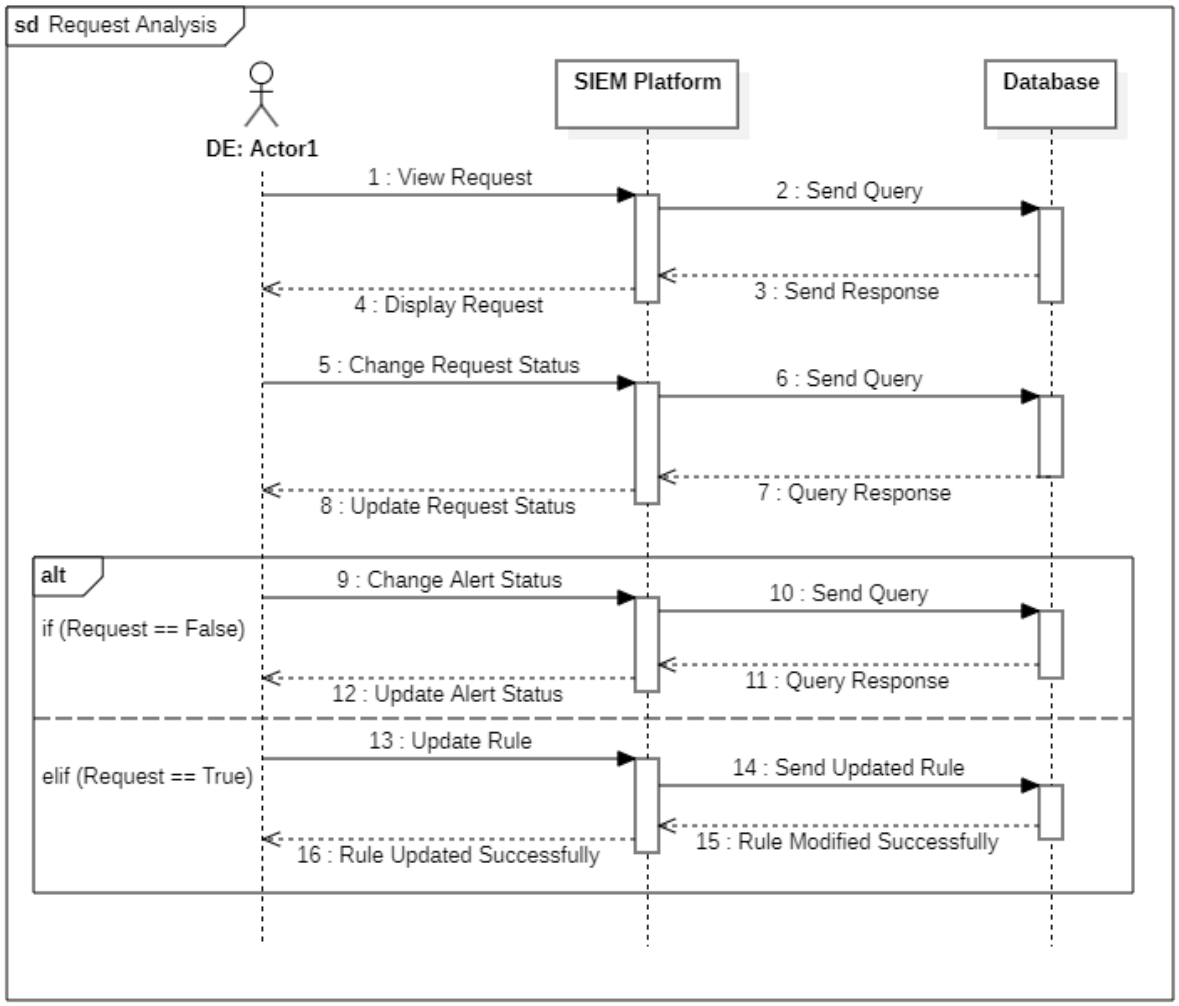


Figure 3.7: Sequence diagram for Request Analysis

3.2.6.5 Sequence diagram for Managing Users

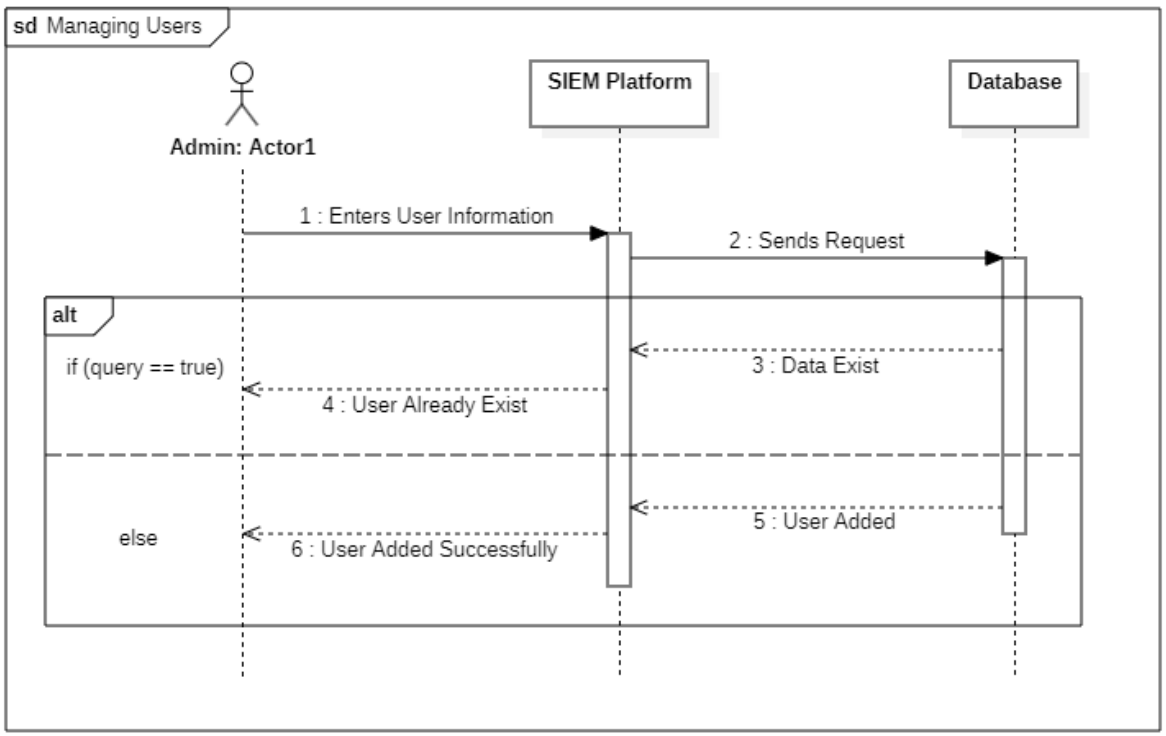


Figure 3.8: Sequence diagram for Managing Users

3.3 Database Design

3.3.1 Introduction to SQL Databases

Relational databases represent a traditional and widely adopted category of database management systems that organize data into structured tables with predefined schemas. These systems ensure data integrity, consistency, and reliability through the use of constraints, relationships, and standardized query languages such as SQL (Structured Query Language) [6].

Unlike NoSQL databases, relational databases enforce a strict schema, making them particularly suitable for systems that require strong consistency, complex relationships, and transactional integrity. This makes them an appropriate choice for our SIEM platform, where data accuracy, traceability, and structured relationships between entities (such as alerts, incidents, users, and rules) are critical.

In our project, we utilize PostgreSQL, an advanced open-source relational database management system known for its robustness, extensibility, and compliance with ACID (Atomicity, Consistency, Isolation, Durability) properties. PostgreSQL provides powerful fea-

tures such as indexing, complex querying, and support for large-scale data processing, making it well-suited for handling security events, logs metadata, and incident records within the SIEM platform.

3.3.2 Database Design Process

The database design process for our SIEM platform followed a structured methodology to ensure alignment with both functional and non-functional requirements. The process consisted of the following steps:

- **Requirement Analysis:** This phase involved identifying the data requirements based on system actors such as SOC analysts (L1, L2, L3), detection engineers, and SOC managers. Key data elements included alerts, logs metadata, incidents, users, roles, and detection rules. The system requires efficient querying and correlation of events, which guided the choice of a relational database.
- **Conceptual Design:** A high-level model of the system's data entities and their relationships was developed using UML class diagrams. Core entities such as *User*, *Alert*, *Incident*, *Rule*, and *Log* were identified, along with their interactions. This phase ensured that all system functionalities, such as alert investigation and incident management, were properly represented.
- **Logical Design:** The conceptual model was translated into a relational schema compatible with PostgreSQL. Tables were defined with appropriate attributes, primary keys, and foreign key relationships to maintain referential integrity. Normalization techniques were applied to reduce redundancy and improve data consistency.
- **Physical Design:** The database was implemented in PostgreSQL with optimizations for performance and scalability. Indexes were created on frequently queried fields such as alert timestamps, severity levels, and user identifiers. Additional configurations were applied to ensure efficient storage and fast query execution.
- **Testing and Validation:** The database schema was validated through testing scenarios to ensure it supports all system functionalities, including alert monitoring, investigation workflows, and incident lifecycle management. Performance testing was conducted to verify the system's ability to handle large volumes of security events.

This structured approach ensured that the database design is reliable, scalable, and aligned with the operational requirements of the SIEM platform.

3.3.3 Conceptual Data Model

Figure 3.9 illustrates the class diagram representing the system's data requirements. This diagram serves as a conceptual data model (CDM), highlighting the key entities and relationships necessary for the proper functioning of the SIEM platform.

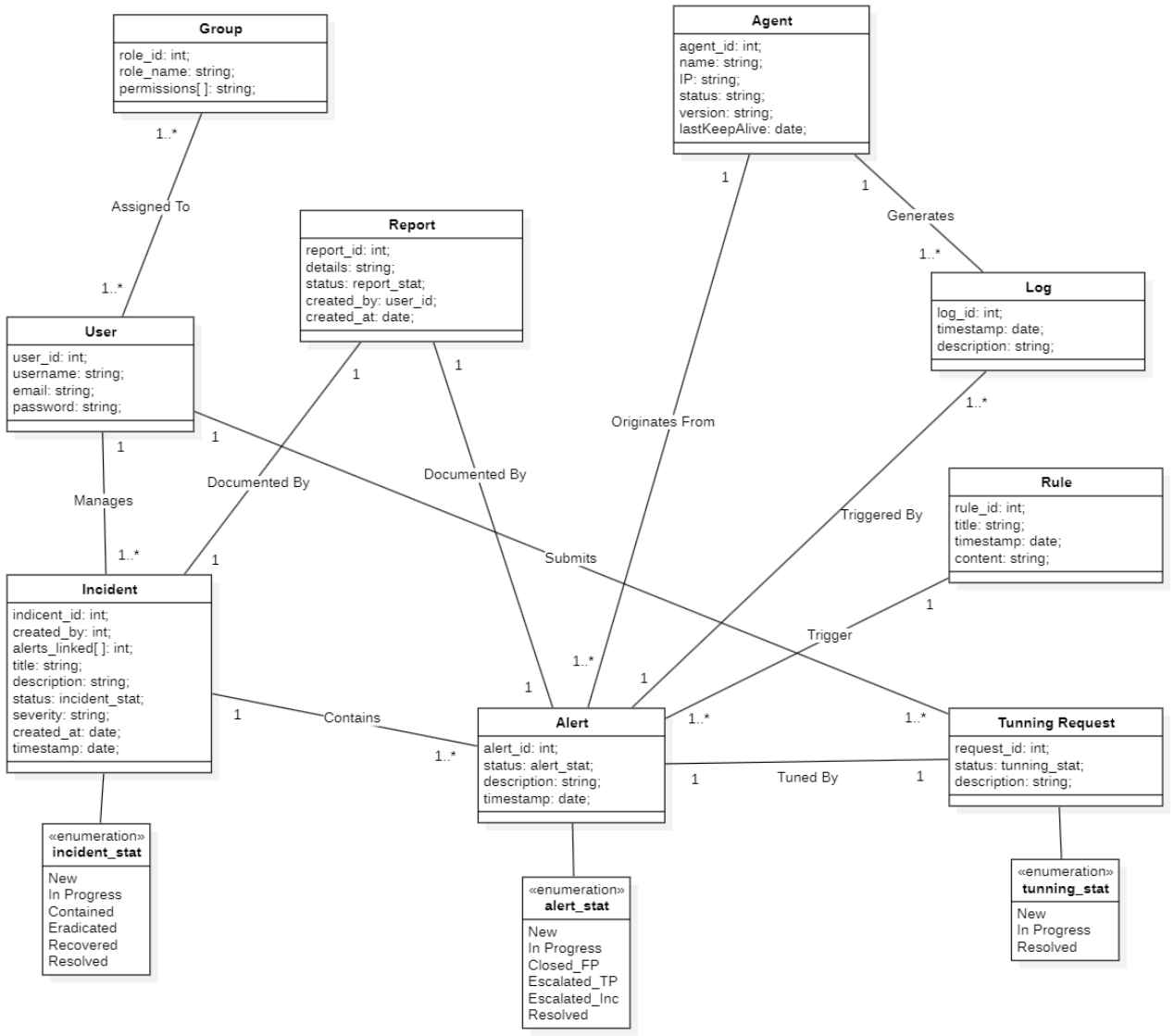


Figure 3.9: Conceptual Data Model (CDM)

The conceptual model provides a clear representation of how data flows between different components of the system and supports core functionalities such as alert processing, incident management, and rule-based detection.

3.3.4 Our Logical Data Design

The database of the SIEM platform is structured into several relational tables, each representing a core entity of the system. The main tables are described as follows:

User: Stores information about system users, including SOC analysts, detection engineers, and administrators. Each user is identified by a unique *userId* and has attributes such as username, password, and email. This table supports authentication and access to the platform.

Agent: Represents monitored endpoints that generate logs. Each agent is identified by *agentId* and includes attributes such as name, IP address, status, version, and last keep-alive timestamp. This table is essential for tracking system health and log sources.

Role: Defines user roles within the system (e.g., SOC L1, L2, L3, Admin). Each role is associated with a user through *userId* and determines the level of access and responsibilities within the platform.

Permission: Stores the set of permissions associated with each role. This enables fine-grained access control, ensuring that users can only perform actions authorized by their assigned roles.

Log: Contains log entries collected from agents. Each log is linked to an agent via *agentId* and includes attributes such as timestamp, location, and description. This table represents the raw data ingested by the SIEM system.

Rule: Represents detection rules used to analyze logs and generate alerts. Each rule is identified by *ruleId* and may be associated with a user (e.g., detection engineer). It includes a description, timestamp, and rule-specific details.

Alert: Stores alerts generated when logs match specific detection rules. Each alert is linked to a rule and a log through *ruleId* and *logId*. It includes details, timestamp, and an optional reference to a report.

Incident: Represents security incidents created from one or more alerts. Each incident is associated with a user (typically SOC L3) and includes attributes such as title, description, status, severity, creation date, and linked alerts. Additional details may include affected hosts and response actions.

Report: Stores reports generated during incident analysis or response. Each report includes a unique identifier, detailed content, and a creation timestamp.

Request: Represents analysis or action requests triggered by alerts. Each request is linked to an alert and is used to initiate further investigation or processing.

The relational design enforces data integrity through primary and foreign key constraints,

ensuring consistent relationships between entities such as logs, alerts, incidents, and users. Indexes can be applied to frequently queried attributes such as timestamps, severity levels, and agent identifiers to optimize performance.

This schema is designed to be scalable and extensible, allowing the integration of additional features such as advanced analytics, correlation rules, and automated response mechanisms without requiring major structural changes.

3.4 Conclusion

This chapter presented the design of our SIEM platform for SOC operations, using UML diagrams to model how analysts interact with the system during log analysis, alert investigation, and incident handling. The database design was also defined to support key SIEM functions such as event storage, correlation, and auditability. The next chapter will focus on implementation through Wazuh, PostgreSQL, and the web interface.

Chapter 4

Implementation

4.1 Introduction

This chapter presents the implementation of the SIEM platform. It describes the technical architecture, the implementation stack, the API design, the key user workflows, and the main application components. The chapter emphasizes how the architectural design was materialized into a practical security monitoring solution.

4.2 Technical Stack and Architecture

The implementation of our SIEM platform relies on a combination of modern development technologies, security tools, and infrastructure components, chosen to meet the functional and non-functional requirements of SOC operations.

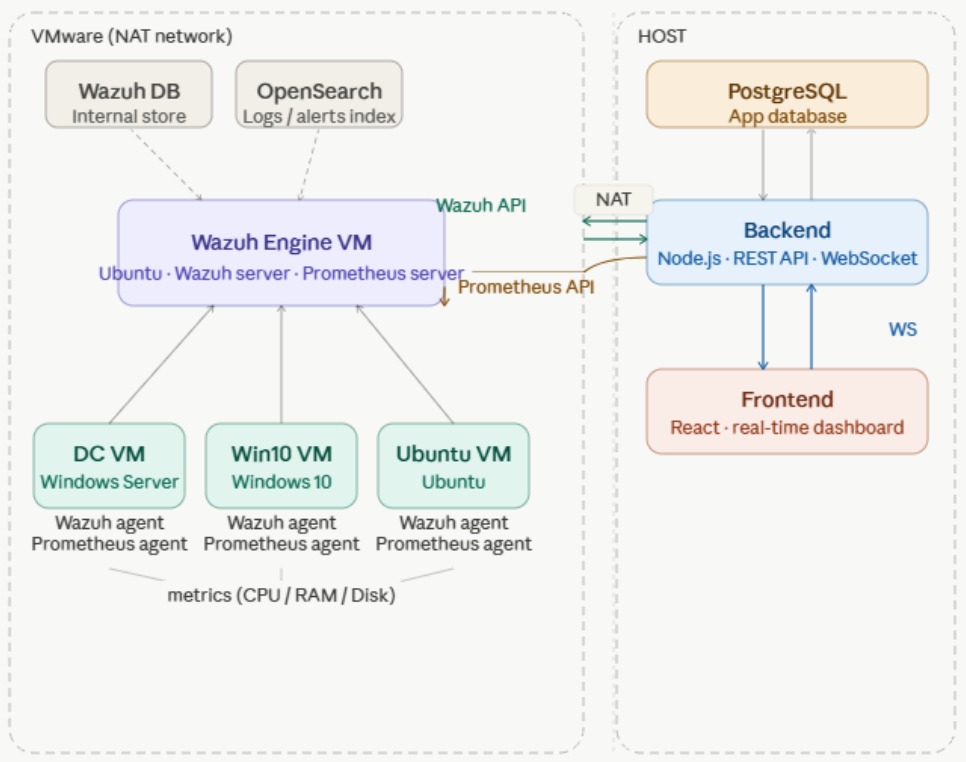


Figure 4.1: SIEM system architecture.

4.2.1 Frontend (React)

- **React:** A JavaScript library for building dynamic, component-based user interfaces, chosen for its real-time rendering capabilities and ecosystem of supporting libraries.
- **Tailwind CSS:** A utility-first CSS framework used for building responsive and consistent layouts without writing custom stylesheets.

4.2.2 Backend (Node.js)

- **Node.js:** A JavaScript runtime built on the V8 engine, chosen for its non-blocking I/O model which makes it well-suited for handling concurrent API requests and real-time WebSocket connections simultaneously.
- **Express.js:** A minimal web framework for Node.js used to define and expose the REST API endpoints consumed by the frontend.
- **WebSocket:** A full-duplex communication protocol used to push real-time alerts and system metric updates from the backend to the frontend without requiring repeated polling.

4.2.3 Database (PostgreSQL)

- **PostgreSQL:** An open-source relational database management system used to persist application-level data including user accounts, roles, incidents, and platform configuration. Its support for complex queries and transactional integrity makes it well-suited for investigation workflows and audit trail management.

4.2.4 Security Engine (Wazuh)

- **Wazuh:** An open-source security platform that provides threat detection, log analysis, integrity monitoring, and incident response capabilities. It serves as the core security engine of the platform, processing events forwarded by agents and generating structured alerts.
- **OpenSearch:** A distributed search and analytics engine used internally by Wazuh to index logs and alerts, enabling fast querying and historical analysis of large volumes of security data.

4.2.5 Monitoring (Prometheus)

- **Prometheus:** An open-source systems monitoring toolkit that collects and stores time-series metrics. It is used to monitor the performance of the virtual machines in the lab environment, exposing CPU, RAM, and disk usage data to the backend via its API.

4.2.6 Virtualization (VMware)

- **VMware:** A virtualization platform used to host the lab environment. It provides the isolated NAT network within which the Wazuh Engine VM and the three monitored machines operate, simulating a realistic multi-endpoint organizational infrastructure.

4.3 Features of Our SIEM Platform

The platform is organized around nine core functional areas, each addressing a specific aspect of SOC operations.

4.3.1 Login

The application begins with a secure authentication screen. Users log in with their credentials and are directed to the appropriate interface based on their assigned role. Authentication is enforced through secure password policies and role-based access control.

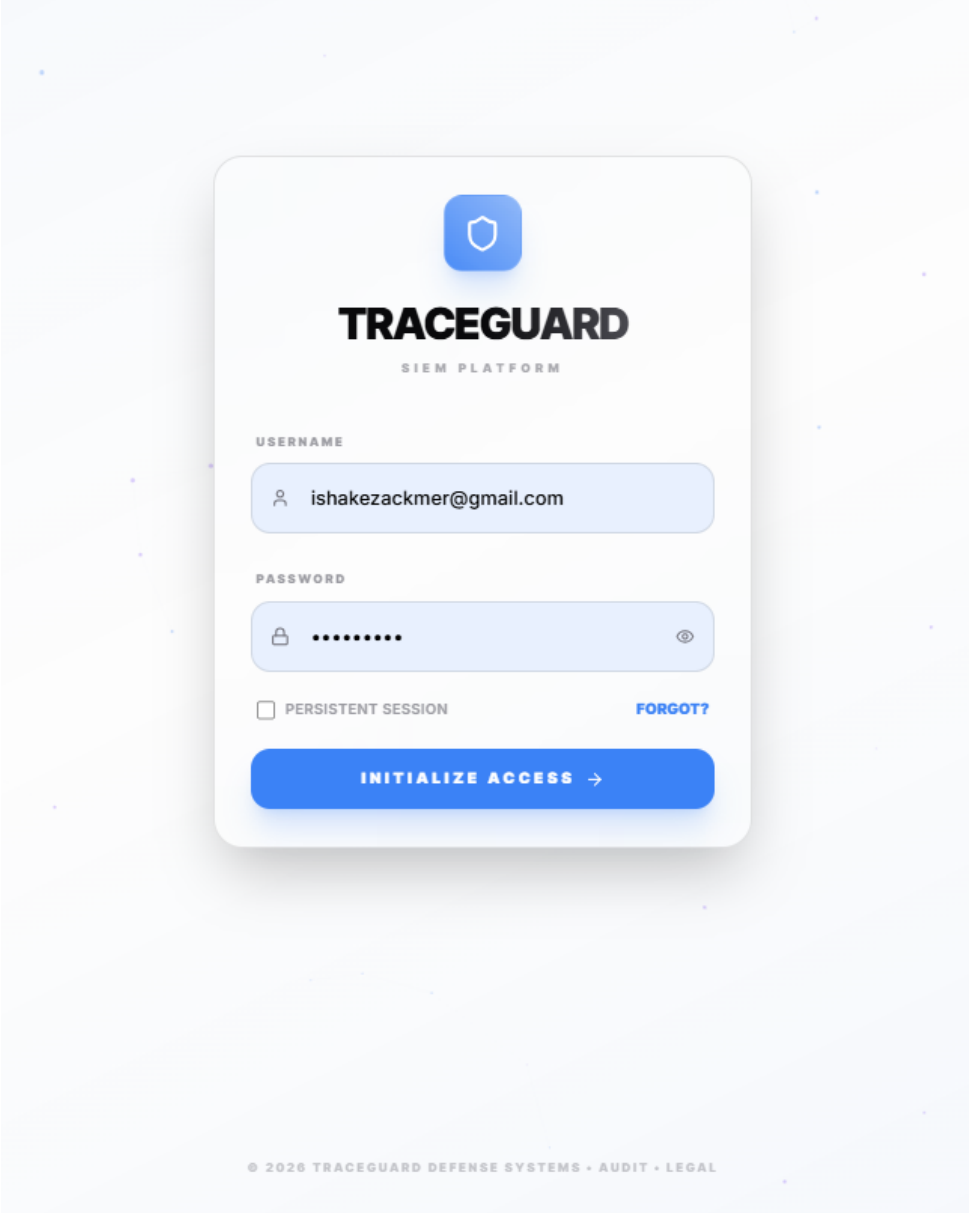


Figure 4.2: Login page of the SIEM platform.

4.3.2 Dashboard

The dashboard provides a real-time overview of security activity across the monitored infrastructure. It aggregates and visualizes statistics about ongoing network events, endpoint states, alert trends, and agent health, giving analysts an immediate situational awareness of the environment.

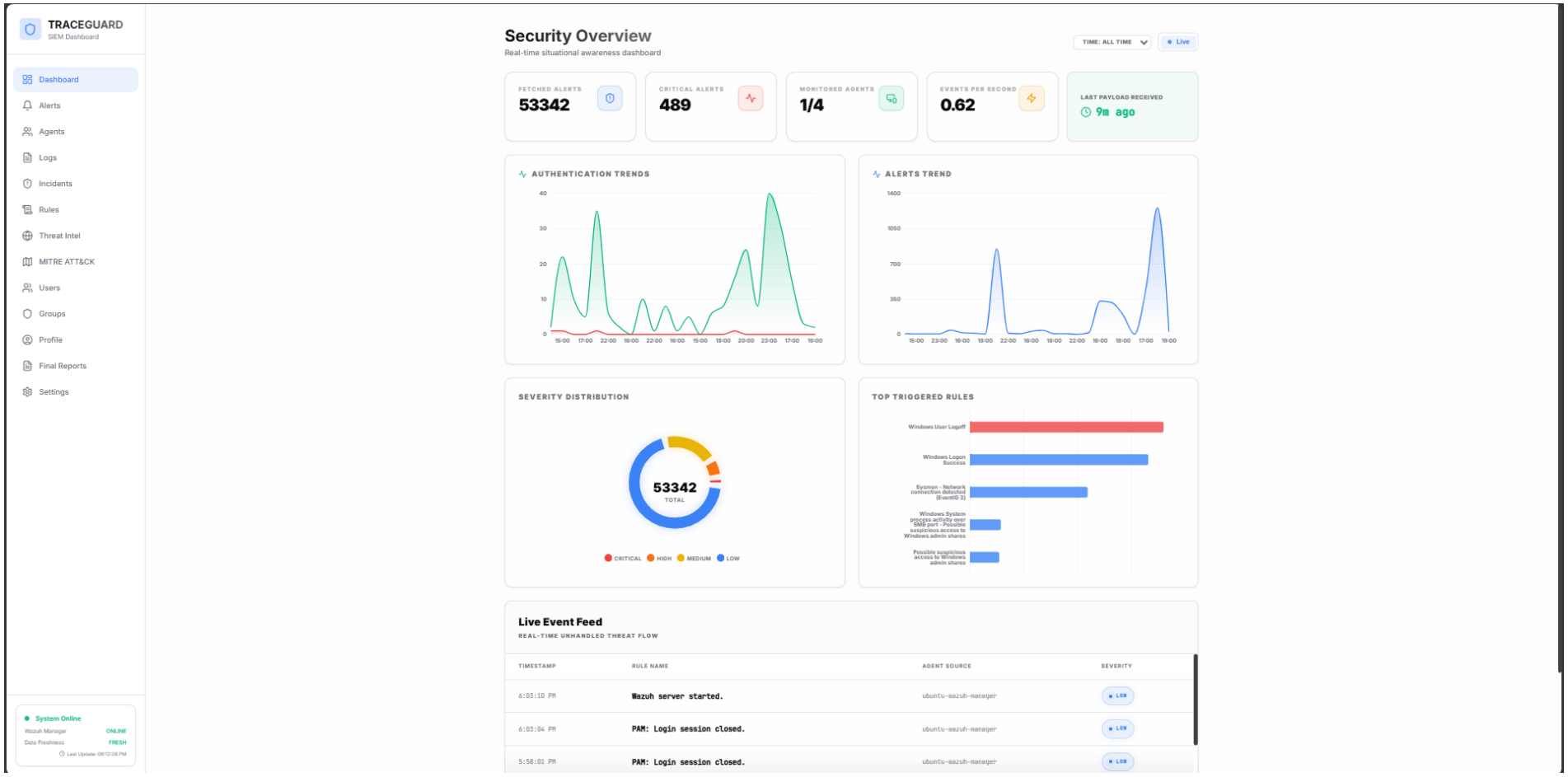


Figure 4.3: Dashboard view of the SIEM platform.

4.3.3 Agent Management

Agents are retrieved directly from Wazuh and listed within the platform. For each agent, the platform displays system performance metrics collected by Prometheus, including CPU usage, RAM consumption, and disk utilization, allowing analysts to monitor both the security and operational health of monitored endpoints from a single interface.

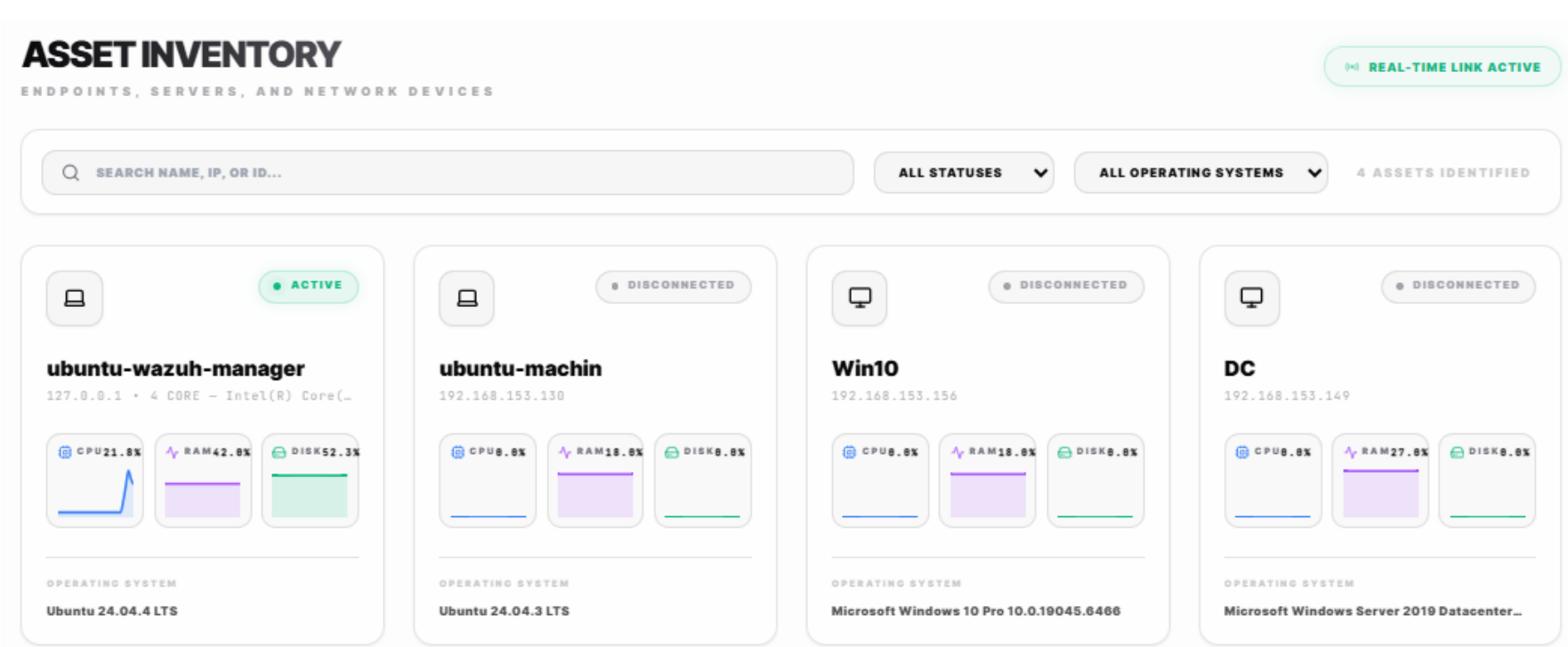


Figure 4.4: Agent monitoring view of the SIEM platform.

4.3.4 Log Management

Logs are ingested from Wazuh and presented in a searchable, filterable view. Analysts can narrow down log entries by source, time range, severity, or keyword to investigate specific events or reconstruct activity timelines.

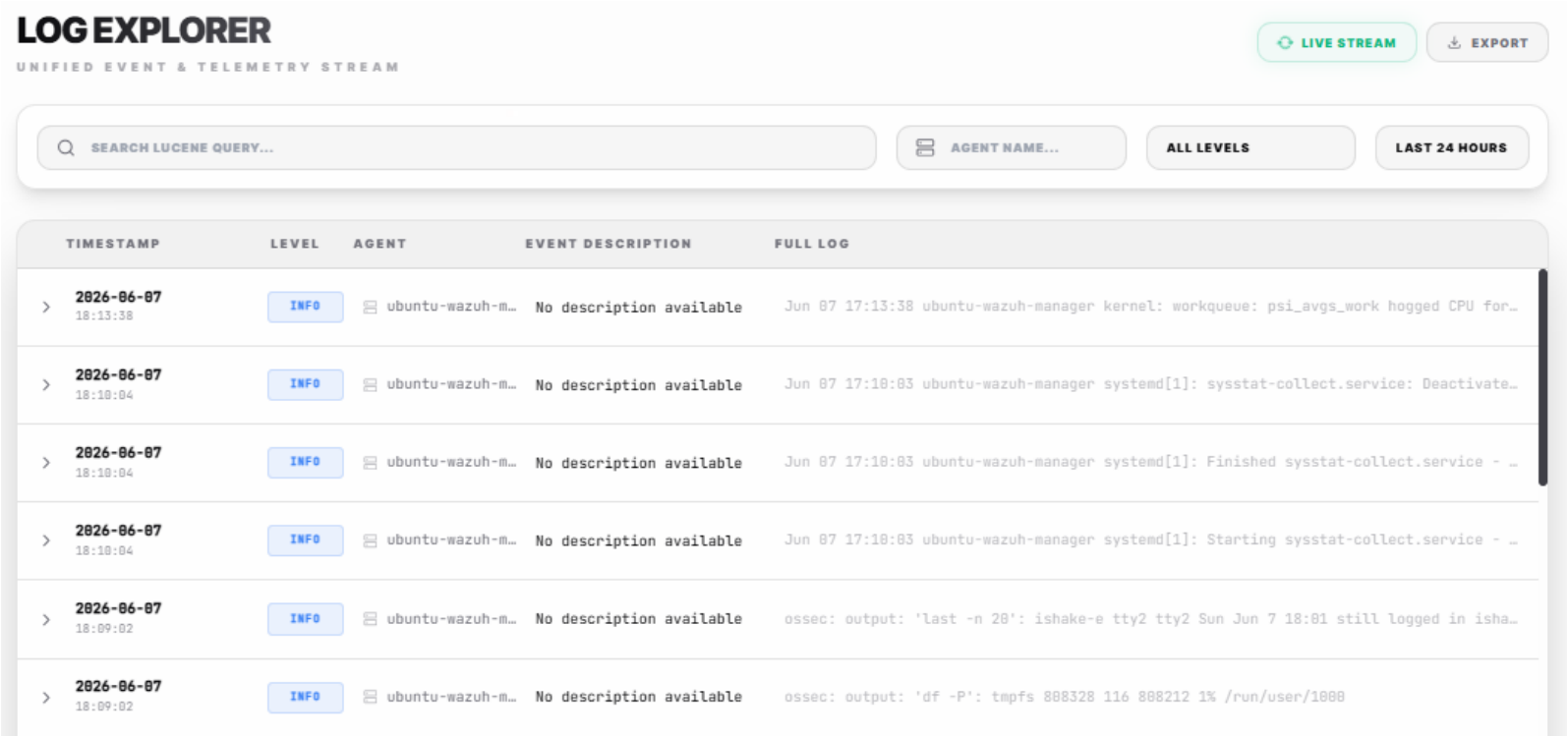


Figure 4.5: Logs view of the SIEM platform.

4.3.5 Alert Management

Alerts generated by the Wazuh correlation engine are retrieved and displayed within the platform. Analysts can filter alerts by various criteria and update their status throughout the triage lifecycle, supporting structured L1, L2, and L3 workflows.

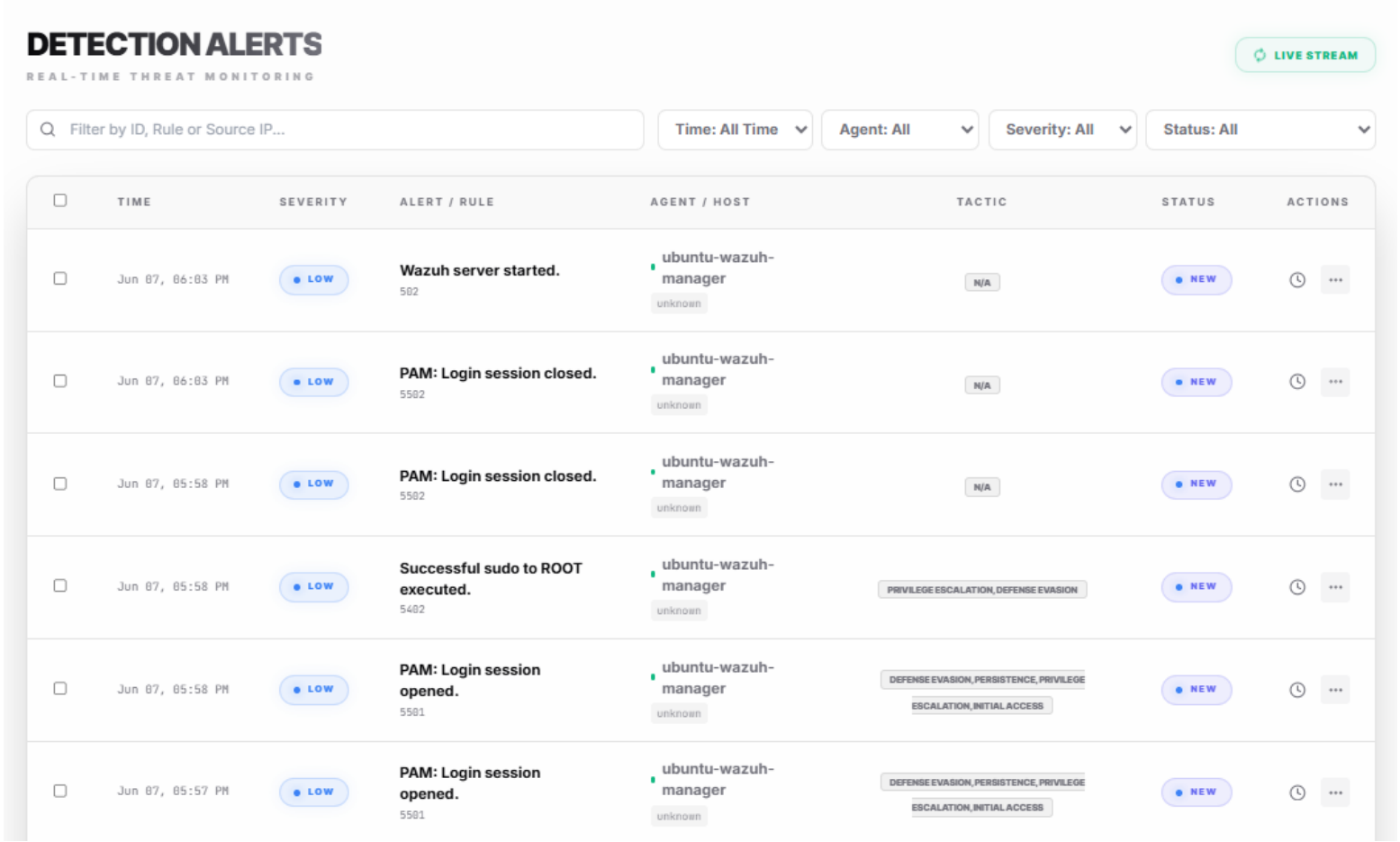


Figure 4.6: Alerts view of the SIEM platform.

4.3.6 Incident Management

The platform provides full incident lifecycle management. Analysts can create, update, and delete incidents, link related alerts to a single incident, and track each incident through its investigation and remediation stages. All active and resolved incidents are listed and accessible for review.

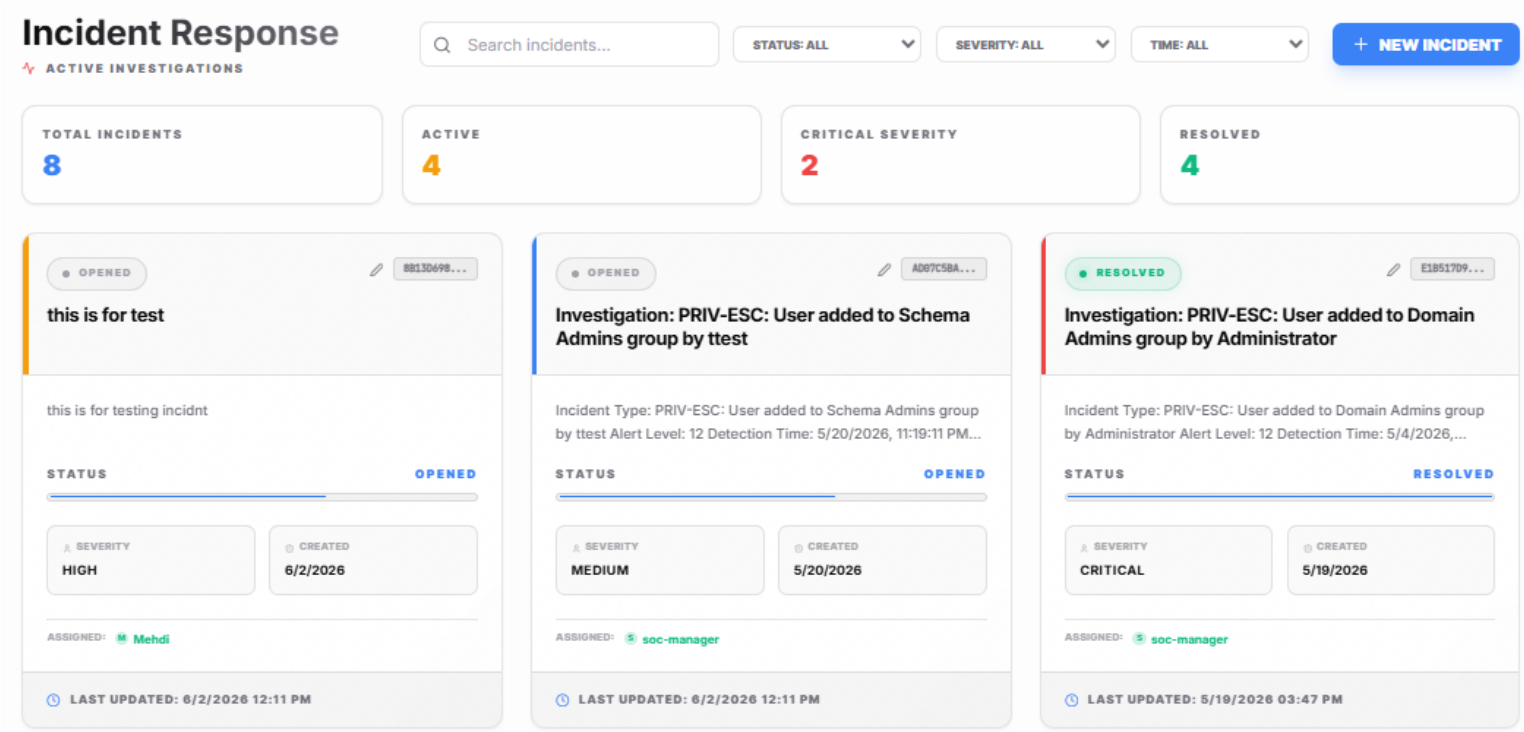


Figure 4.7: Incidents view of the SIEM platform.

4.3.7 Detection Rule Management

Detection engineers can create, update, delete, and list detection rules directly within the platform. Rules define the conditions under which alerts are generated, and managing them through the interface ensures that detection logic remains current and maintainable without requiring direct access to the underlying Wazuh configuration.

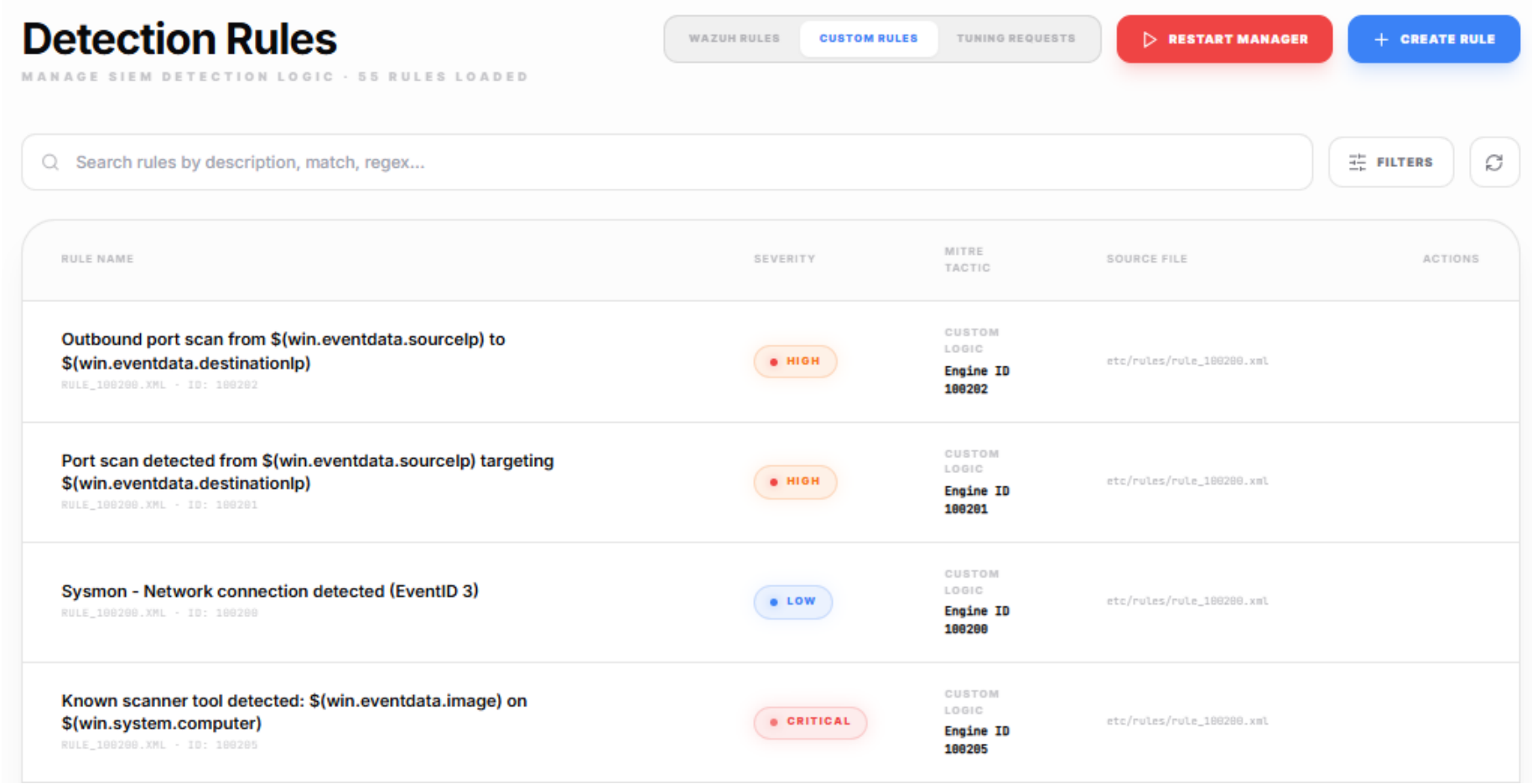


Figure 4.8: Rule management view of the SIEM platform.

4.3.8 Reporting

The platform supports the generation of several report types:

- **Final Reports:** Comprehensive summaries of security activity over a defined period.
- **Incident Reports:** Detailed documentation of individual incidents, including timeline, affected assets, and response actions taken.
- **Alert Reports:** Focused reports on alert activity and trends.
- **Tuning Request Reports:** Reports used to document and communicate requests for detection rule adjustments based on observed false positives or missed detections.

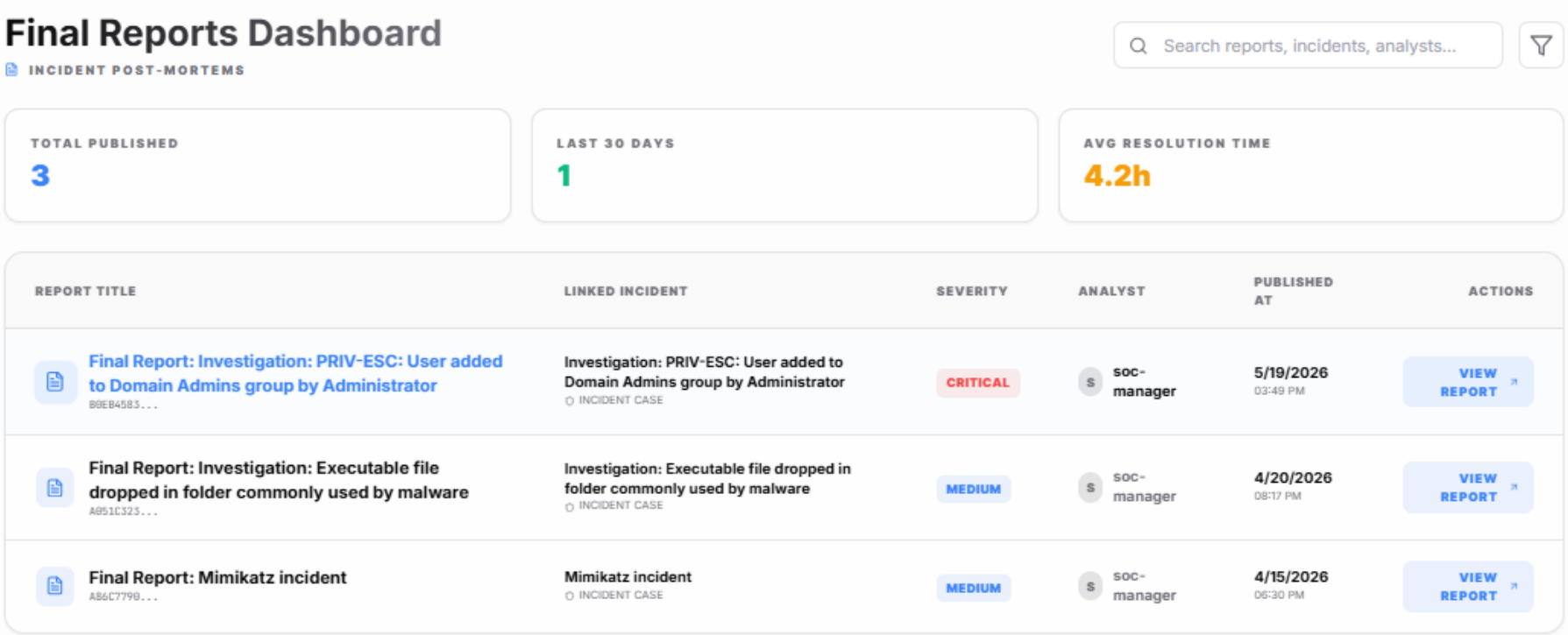


Figure 4.9: Final reports view of the SIEM platform.

4.3.9 Role-Based Access Control

The platform implements a full RBAC system. Administrators can create and manage users and groups, assign users to groups, and define permissions at the group level. Users and groups can be updated or deleted as organizational needs evolve, ensuring that access to platform capabilities is always aligned with each user's role and responsibilities within the SOC.

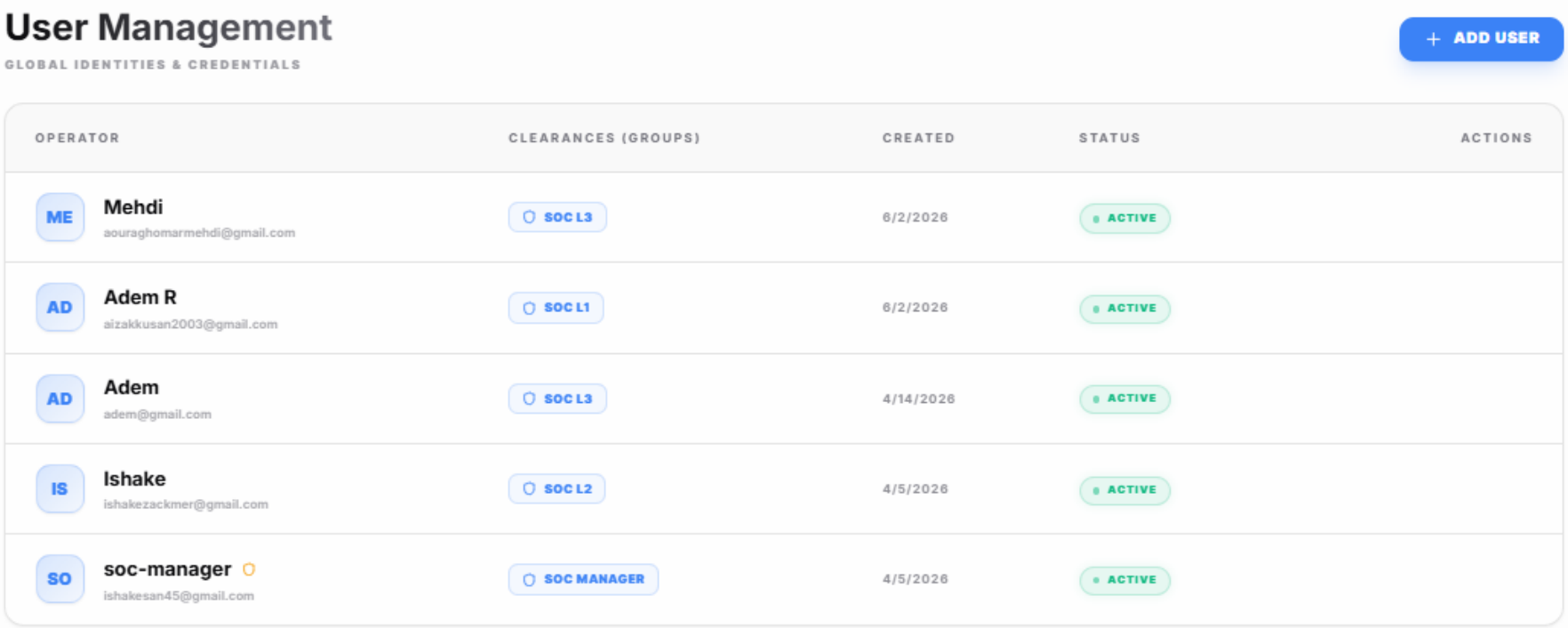


Figure 4.10: User management view of the SIEM platform.

Figure 4.11: Groups management view of the SIEM platform.

4.4 Conclusion

The implementation chapter demonstrates how the SIEM design was realized as a working security platform. The system architecture supports modular log ingestion, correlation, alert management, and incident handling. The API design supports secure data flows, user workflows, and administrative control.

By implementing the SIEM platform with a layered architecture and structured SOC workflows, the solution provides centralized visibility, real-time detection, and coordinated incident response. Future implementation work can extend the platform with threat intelligence integration, automated response actions, and advanced analytics to further enhance detection effectiveness.

Chapter 5

Tests and Evaluation

5.1 Introduction

After implementing our solution, we move on to presenting several test cases in this chapter to ensure that our system is effective and works as intended. The final step then consists of examining the various data and graphs produced by our monitoring tools and evaluating the effectiveness of the solution through different scenarios and tests.

5.2 Functional Testing

Before conducting attack simulations, we first verify that all components of the SIEM platform are functioning correctly under normal conditions. The following test cases cover the core functionalities of the system.

5.3 Attack Simulation and Detection Evaluation

In this section, we simulate real-world attack scenarios to evaluate the detection capabilities of our SIEM platform. The attacker is represented by a Kali Linux machine that has already gained access to the internal NAT network and has knowledge of the IP addresses of the monitored endpoints.

5.3.1 Scenario 1: SSH Brute-Force Attack (Hydra)

5.3.1.1 Description

In this scenario, the attacker attempts to gain unauthorized access to the Ubuntu VM by performing an SSH brute-force attack. Using tools available on Kali Linux such as `Hydra` or `Medusa`, the attacker sends a large number of successive login attempts with different username and password combinations.

5.3.1.2 Expected Behavior

Since the Ubuntu VM runs a Wazuh agent, all authentication-related events are collected and forwarded to the Wazuh Engine in real time. The Wazuh server applies its built-in rules for brute-force detection, specifically rules that trigger when a threshold of failed authentication attempts is exceeded within a short time window. An alert is then generated and indexed in OpenSearch, and subsequently displayed on the real-time dashboard of our application.

5.3.1.3 Steps

1. Launch the brute-force attack from the Kali Linux machine targeting the SSH service of the Ubuntu VM.
2. Monitor the Wazuh Engine for incoming authentication failure logs.
3. Verify that the brute-force detection rule is triggered and an alert is raised.
4. Confirm that the alert appears in real time on the frontend dashboard.

5.3.1.4 Result

The brute-force attack was successfully detected by the Wazuh Engine. As shown in the figures below, multiple failed SSH authentication attempts triggered the brute-force detection rule, and the corresponding alert was displayed in real time on the dashboard.

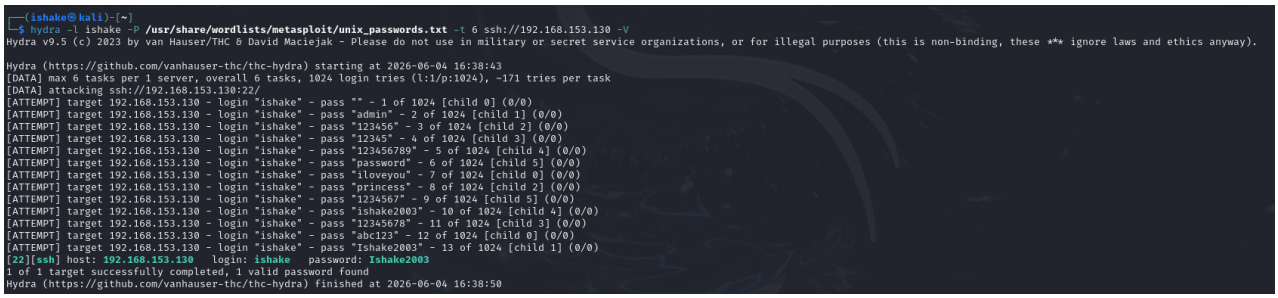


Figure 5.1: Hydra brute-force attack running on Kali Linux

The command executed from the Kali Linux machine was:

```
hydra -l ishake -P /usr/share/wordlists/metasploit/unix_passwords.txt
      -t 6 ssh://192.168.153.130 -V
```

Where `-l ishake` specifies the target username, `-P` points to the password wordlist located in the Metasploit directory, `-t 6` sets the number of parallel tasks to 6, and `-V` enables verbose mode to display each attempt. Hydra ran a total of 1024 login tries against the Ubuntu VM at `192.168.153.130`, and successfully found the valid credentials (`ishake:Ishake2003`) after 13 attempts.

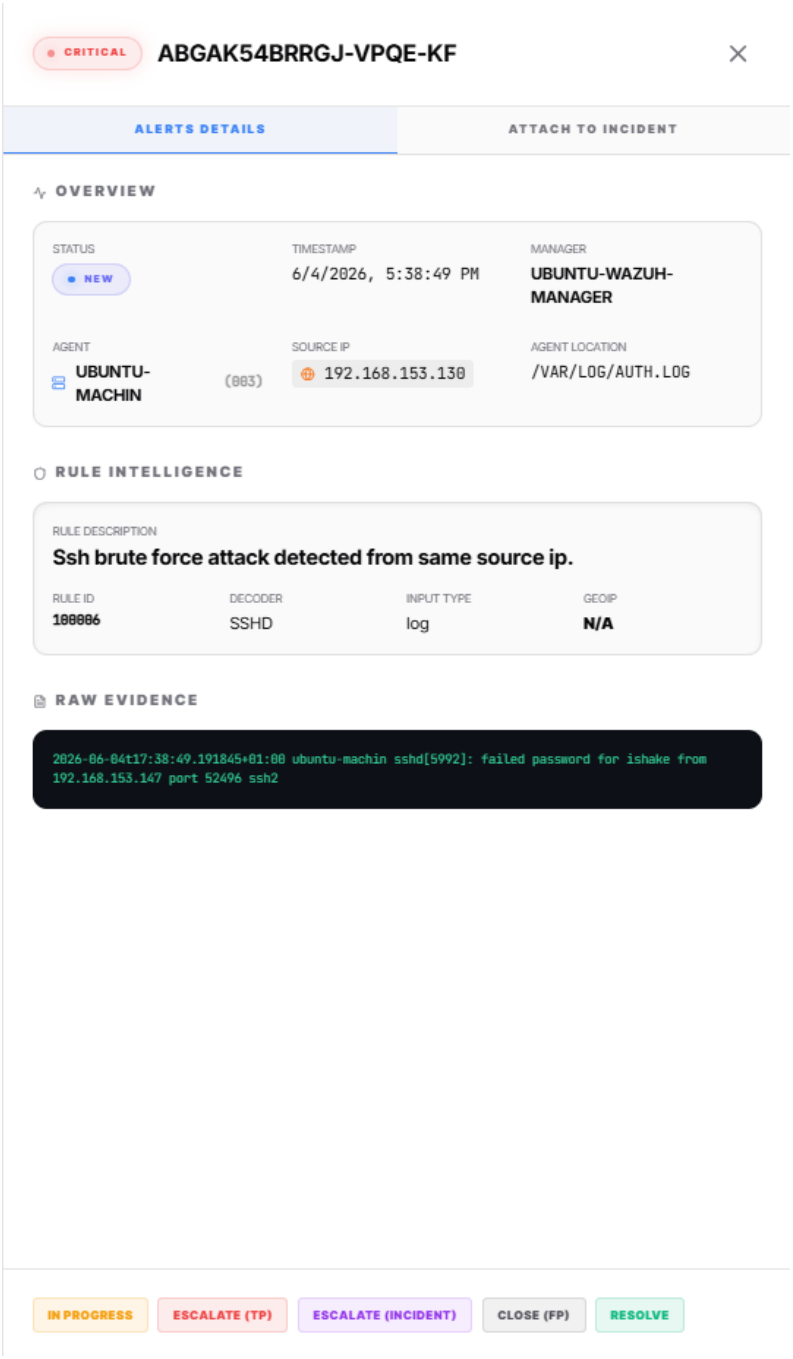


Figure 5.2: Brute-force alert displayed on the SIEM dashboard

As illustrated in Figure 5.2, the repeated failed authentication attempts were detected by the Wazuh Engine, which triggered its built-in brute-force detection rule. The corresponding alert was immediately raised and displayed in real time on our SIEM dashboard, confirming that the platform successfully identified the attack.

5.3.2 Scenario 2: Network Port Scan (Nmap)

5.3.2.1 Description

In this scenario, the attacker performs an aggressive network reconnaissance using `Nmap` from the Kali Linux machine. A large volume of probe packets is sent to one or more machines in the network in order to discover open ports and running services. This type of scan generates an abnormally high amount of network traffic and connection attempts, which are characteristic of a reconnaissance phase.

5.3.2.2 Expected Behavior

The Wazuh agents on the targeted machines capture the abnormal network activity and forward the corresponding logs to the Wazuh Engine. The engine applies its detection rules to identify the port scanning behavior and raises an alert. The alert is indexed in OpenSearch and displayed on the dashboard in real time.

5.3.2.3 Steps

1. Execute an Nmap scan from the Kali Linux machine targeting one or more machines in the NAT network (e.g., `nmap -sS -A <target_ip>`).
2. Monitor the Wazuh Engine for logs related to abnormal connection attempts.
3. Verify that the port scan detection rule is triggered and an alert is raised.
4. Confirm that the alert appears in real time on the frontend dashboard.

5.3.2.4 Result

The port scan was successfully detected by the Wazuh Engine. The following command was executed from the Kali Linux machine targeting the Domain Controller VM at `192.168.153.149`:

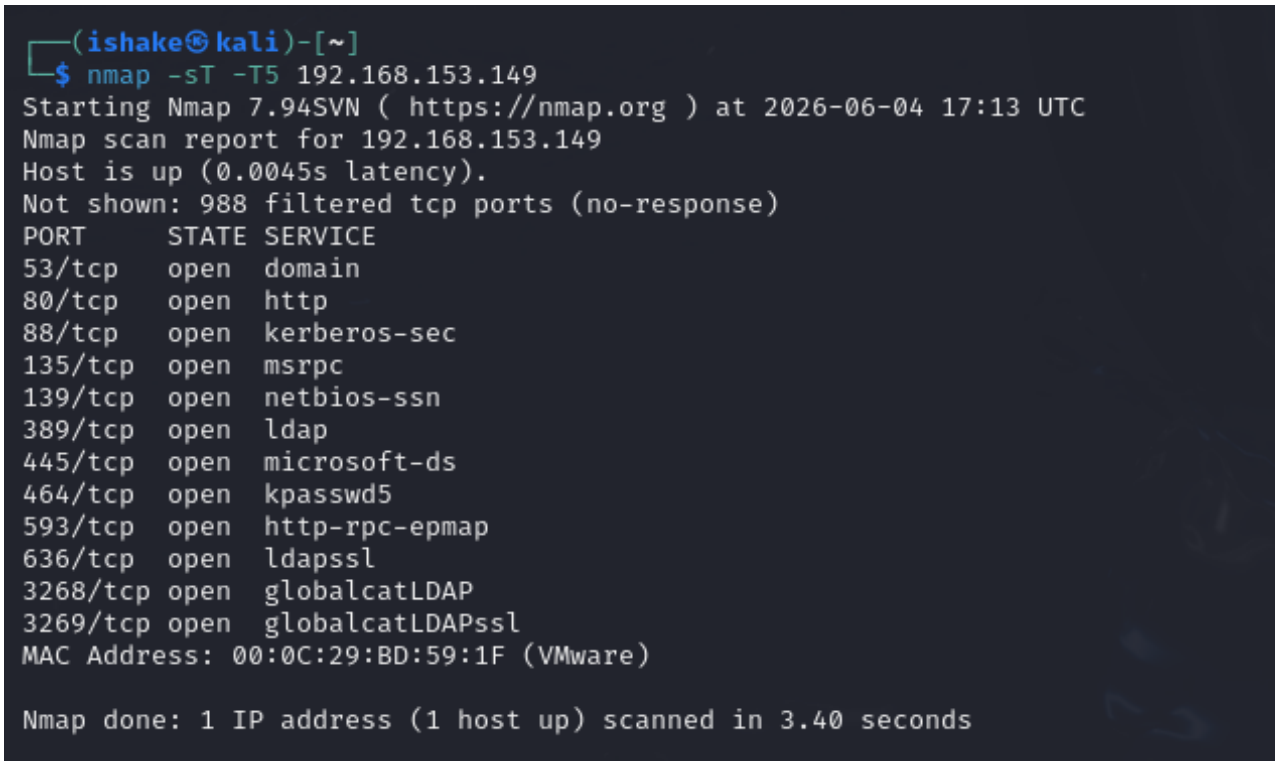


Figure 5.3: Nmap port scan executed from the Kali Linux machine

The command executed from the Kali Linux machine was:

```
nmap -sT -T5 192.168.153.149
```

Where `-sT` performs a full TCP connect scan, and `-T5` sets the timing template to the most aggressive level, sending packets as fast as possible. The scan completed in 3.40 seconds and revealed 12 open ports on the target machine, including `53` (domain), `80` (http), `88` (kerberos-sec), `389` (ldap), and `445` (microsoft-ds), among others.

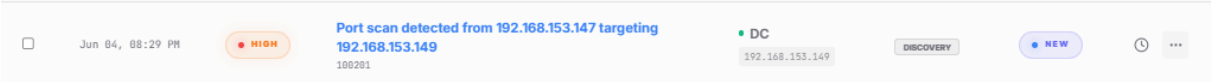


Figure 5.4: Nmap port scan generating a large volume of connection attempts

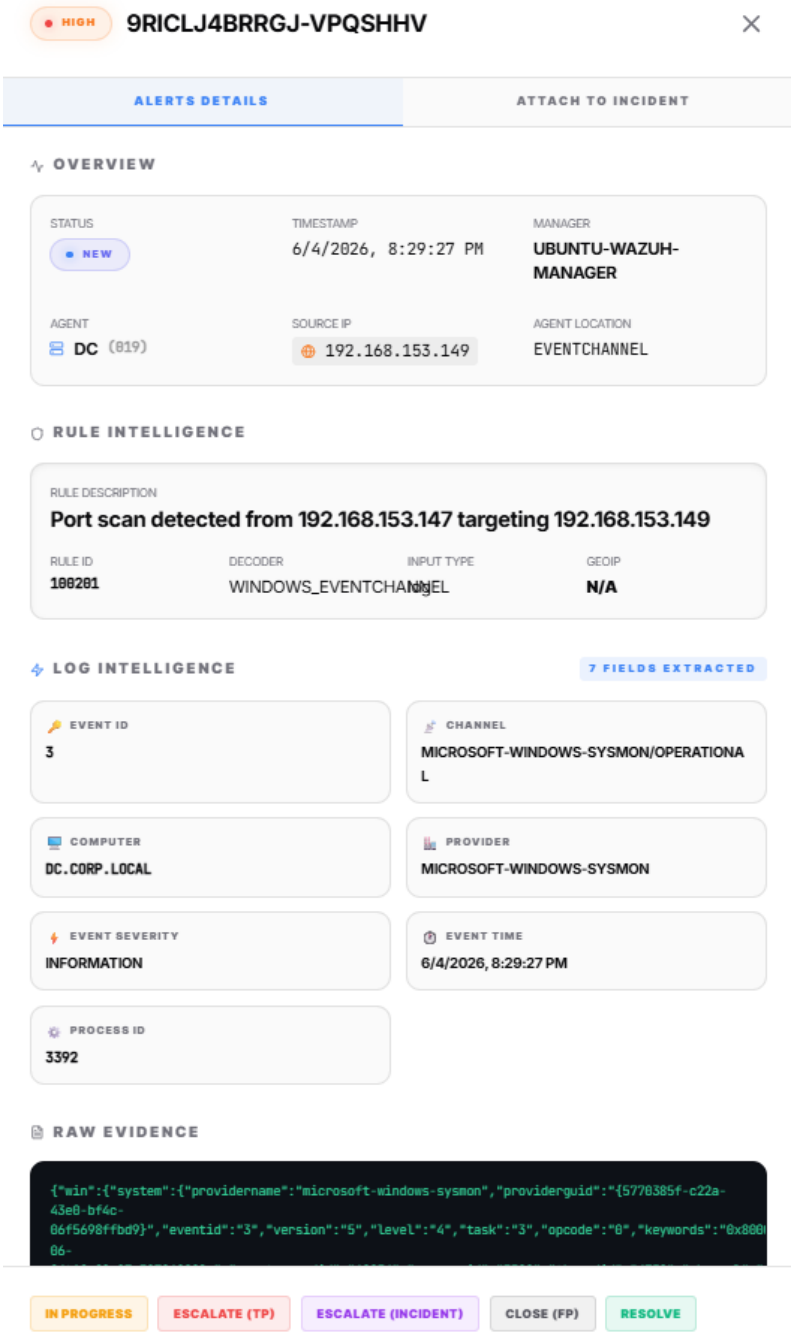


Figure 5.5: Port scan alert displayed on the SIEM dashboard

As illustrated in Figure 5.3, the aggressive scan generated a large volume of connection attempts that were captured by the Wazuh agent running on the DC VM. The Wazuh Engine successfully identified the behavior and raised a high severity alert with the descrip-

tion *"Port scan detected from 192.168.153.147 targeting 192.168.153.149"*, corresponding to rule ID `100281`. The alert was timestamped at 6:07:44 PM on June 4, 2026, and appeared in real time on our SIEM dashboard, confirming that the platform correctly detected the reconnaissance activity.

5.3.3 Scenario 3: Remote Connection using Evil-WinRM (Pass the Hash)

5.3.3.1 Description

In this scenario, the attacker uses Evil-WinRM to establish a remote connection to the Domain Controller by leveraging valid credentials obtained from a previous reconnaissance phase. This technique, known as Pass the Hash, allows an attacker to authenticate to a remote machine using a known username and password without needing to crack any hashes, simulating a lateral movement or privilege escalation attempt within the network.

5.3.3.2 Expected Behavior

The Wazuh agent running on the DC VM captures the remote authentication event and forwards it to the Wazuh Engine. The engine applies its detection rules and raises alerts related to suspicious remote logon activity, user account modifications, and potential pass-the-hash behavior. The alerts are then displayed in real time on our SIEM dashboard.

5.3.3.3 Steps

1. Execute the Evil-WinRM command from the Kali Linux machine targeting the Domain Controller at `192.168.153.149`.
2. Monitor the Wazuh Engine for remote authentication and account activity logs.
3. Verify that the detection rules are triggered and alerts are raised.
4. Confirm that the alerts appear in real time on the frontend dashboard.

5.3.3.4 Result

The attack was successfully detected by the Wazuh Engine. The following command was executed from the Kali Linux machine:

```
evil-winrm -u "CORP\Administrator" -p "Ishake2003" -i "192.168.153.149"
```

Where `-u` specifies the target username, `-p` provides the password, and `-i` specifies the target IP address of the Domain Controller. Evil-WinRM successfully established a remote connection to the DC VM using the Administrator credentials.



Figure 5.6: Evil-WinRM remote connection executed from the Kali Linux machine

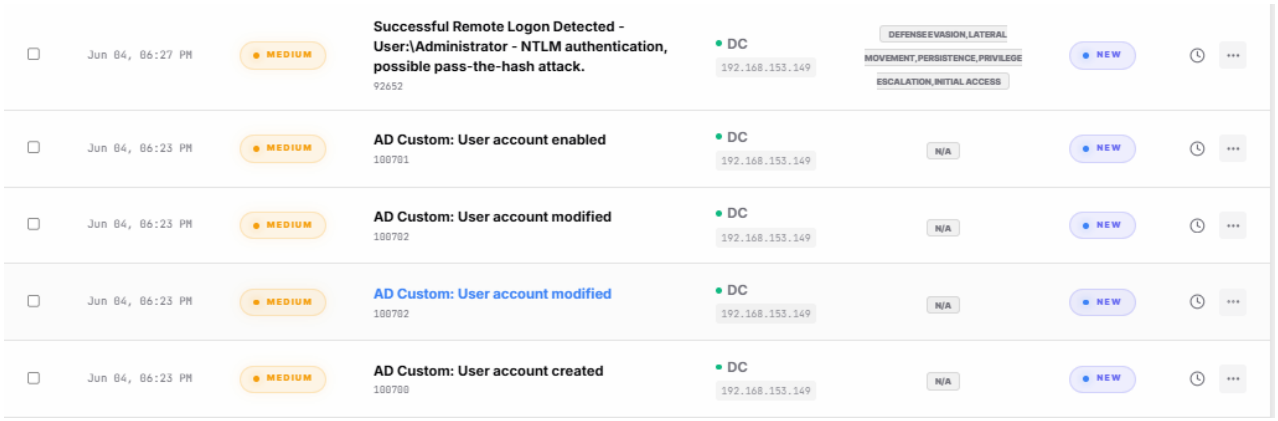


Figure 5.7: Alerts raised on the SIEM dashboard following the Evil-WinRM connection

As illustrated in Figures 5.7 and 5.8, the Wazuh Engine successfully identified the malicious activity and raised multiple medium severity alerts, including a *"Successful Remote Logon Detected - User:\Administrator - NTLM authentication, possible pass-the-hash attack"* (rule ID `92652`), as well as several Active Directory related alerts such as *"AD Custom: User account created"*, *"AD Custom: User account modified"*, and *"AD Custom: User account enabled"*, all triggered at 6:23 PM on June 4, 2026 on the DC agent at `192.168.153.149`. The alert details confirm that the event was captured via the Windows Security Auditing channel with an `AUDIT_SUCCESS` severity, proving that the platform successfully detected the lateral movement attempt in real time.

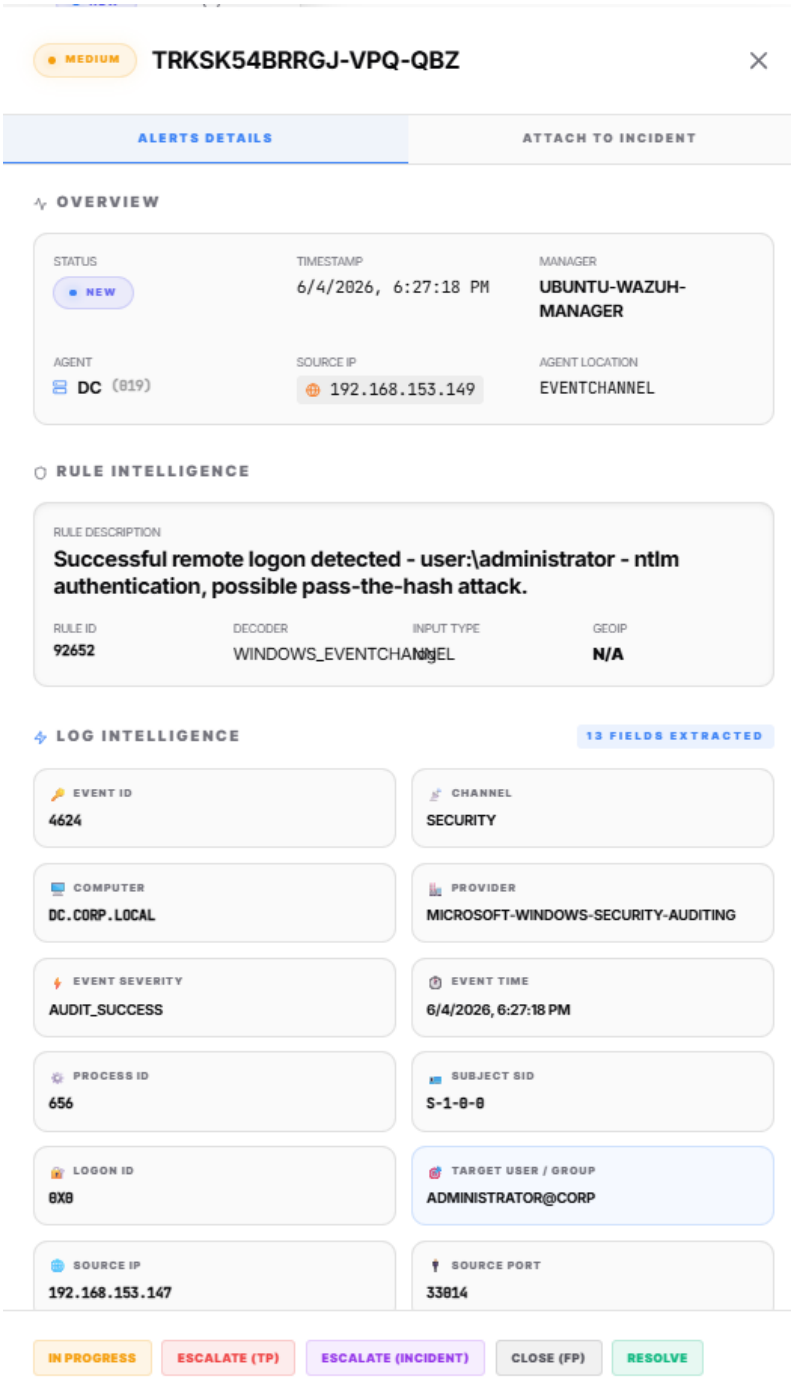


Figure 5.8: Alert details for the pass-the-hash detection

5.3.4 Scenario 4: Privilege Escalation

5.3.4.1 Description

In this scenario, the attacker leverages the previously established Evil-WinRM session as `Administrator` to escalate privileges within the Active Directory environment. Using PowerShell commands directly through the remote shell, the attacker creates a new user account and adds it to the `Domain Admins` group, effectively granting it the highest level of privileges in the domain.

5.3.4.2 Expected Behavior

The Wazuh agent on the DC VM captures the Active Directory changes and forwards them to the Wazuh Engine. The engine applies its privilege escalation detection rules and raises critical and medium severity alerts related to the new user creation, group membership changes, and suspicious remote logon activity. The alerts are displayed in real time on our SIEM dashboard.

5.3.4.3 Steps

1. From the active Evil-WinRM session, create a new Active Directory user using PowerShell.
2. Add the newly created user to the `Domain Admins` group.
3. Monitor the Wazuh Engine for privilege escalation related alerts.
4. Confirm that the alerts appear in real time on the frontend dashboard.

5.3.4.4 Result

Once inside the Evil-WinRM remote shell, the following PowerShell commands were executed:

```
$password = ConvertTo-SecureString "Ishake2003" -AsPlainText -Force

New-ADUser -Name "John Smith" -GivenName "John" -Surname "Smith"
           -SamAccountName "jsmith" -UserPrincipalName "jsmith@corp.local"
           -AccountPassword $password -Enabled $true

Add-ADGroupMember -Identity "Domain Admins" -Members "jsmith"
```

The first command converts a plain text password into a secure string. The second command creates a new Active Directory user named `John Smith` with the account name `jsmith`. The third command adds the newly created user to the `Domain Admins` group, granting them full administrative privileges over the domain.

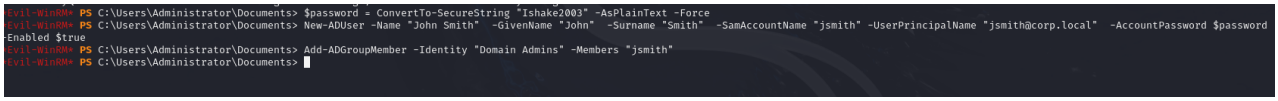


Figure 5.9: PowerShell commands executed through the Evil-WinRM session

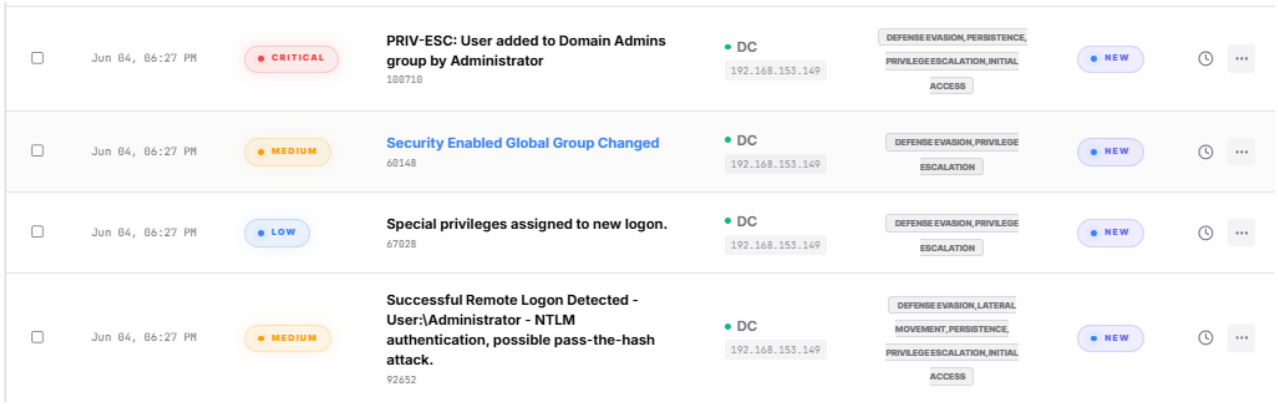


Figure 5.10: Privilege escalation alerts displayed on the SIEM dashboard

As illustrated in Figures 5.10 and 5.11, the Wazuh Engine successfully detected the privilege escalation activity and raised multiple alerts at 6:27 PM on June 4, 2026. A **critical** severity alert was triggered with the description *"PRIV-ESC: User added to Domain Admins group by Administrator"* (rule ID `100710`), along with a medium severity alert for *"Security Enabled Global Group Changed"* (rule ID `60148`) and another medium severity alert for a suspicious remote logon via NTLM authentication (rule ID `92652`). The alert details confirm that the event was captured via the Windows Security Auditing channel, with `ADMINISTRATOR@CORP` as the subject user and `DOMAIN ADMINS@CORP` as the target group, proving that the platform successfully detected the privilege escalation in real time.

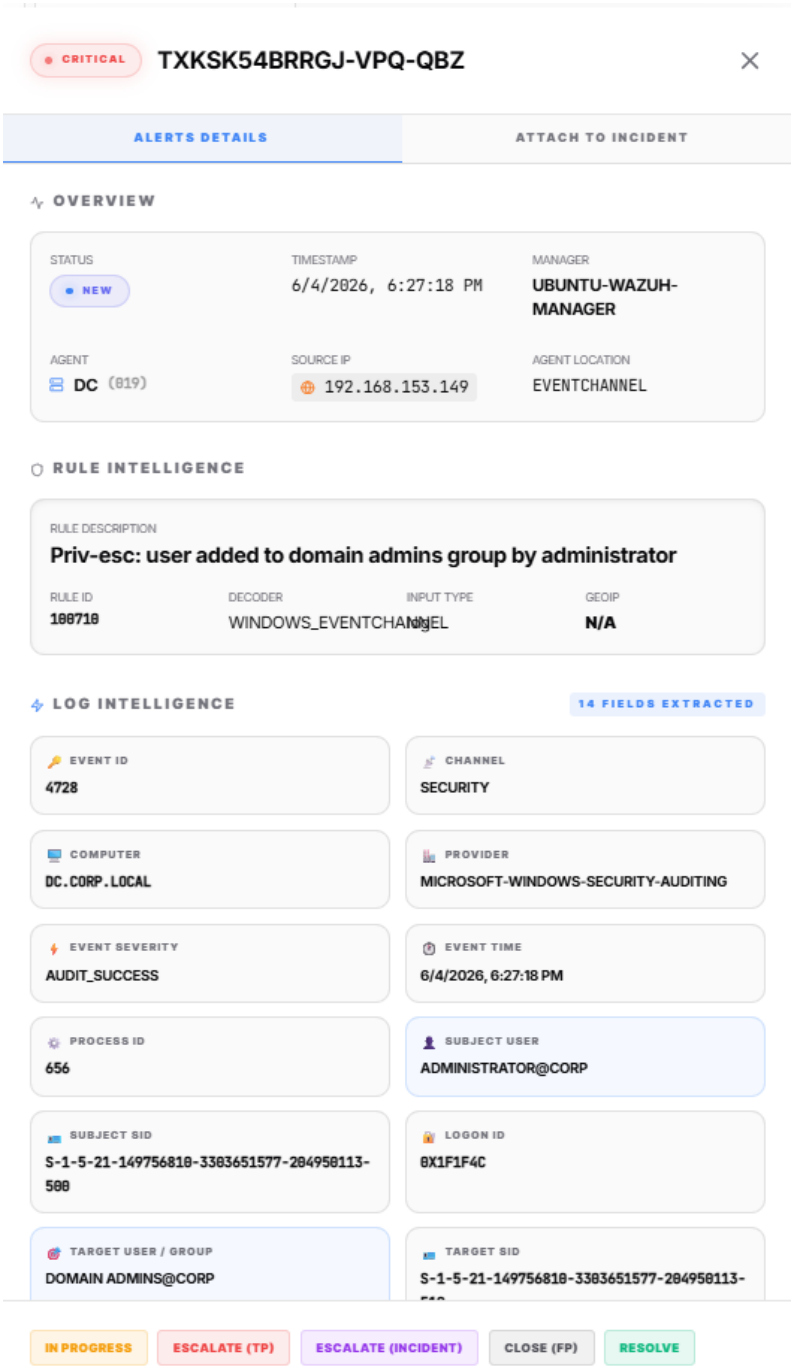


Figure 5.11: Critical alert details for the Domain Admins group membership change

Chapter 6

General Conclusion

6.1 Contributions

This project developed a robust SIEM platform tailored for real-time security monitoring and incident management. The platform integrates a web-based dashboard accessible via standard browsers, enabling security analysts, detection engineers, and administrators to monitor alerts, manage incidents, and track system health in real-time. Key achievements include:

- Implementing real-time dashboards for alerts, agent status, and system performance.
- Providing comprehensive incident management, including creation, assignment, escalation, and documentation.
- Developing flexible rule management for detection engineers to author, test, and deploy detection rules efficiently.
- Including user and group management, access control, and profile settings to enforce organizational security policies.
- Offering reporting and audit capabilities to support SOC performance analysis and regulatory compliance.

The platform delivers secure, reliable, and user-friendly tools for monitoring, investigation, and administration of security events.

6.2 Limitations

Despite its robust functionality, the platform has some limitations:

- Automated response systems is not yet implemented.
- Performance under extremely high alert volumes has not been extensively stress-tested, which may reveal bottlenecks.
- Certain advanced analytics and visualization features, such as AI-based anomaly detection, were not included due to time and resource constraints.

6.3 Future Work

Future enhancements will aim to address these limitations and strengthen the SIEM platform:

- Automated response capabilities for proactive security measures.
- Implementing AI-driven analytics and anomaly detection to improve threat detection accuracy.
- Conducting stress tests and performance optimization to ensure scalability under high alert volumes.
- Enhancing reporting features with customizable dashboards and advanced visualization tools for SOC managers.

These improvements will ensure that the platform becomes a comprehensive, scalable, and intelligent SIEM solution, capable of supporting modern security operations effectively.

Bibliography

[1] Node.js Contributors, *Node.js API Documentation*, https://nodejs.org/docs/latest/api/, accessed: 2026.

[2] Express.js Contributors, *Express.js Documentation*, https://expressjs.com/, accessed: 2026.

[3] Vite Contributors, *Vite Documentation*, https://vitejs.dev/, accessed: 2026.

[4] Microsoft, *TypeScript Documentation*, https://www.typescriptlang.org/docs/, accessed: 2026.

[5] Tailwind Labs, *Tailwind CSS Documentation*, https://tailwindcss.com/docs, accessed: 2026.

[6] PostgreSQL Global Development Group, *PostgreSQL Documentation*, https://www.postgresql.org/docs/, accessed: 2026.

[7] Wazuh Contributors, *Wazuh REST API Documentation*, https://documentation.wazuh.com/current/user-manual/api/index.html, accessed: 2026.

[8] MITRE, *MITRE ATT&CK Documentation*, https://attack.mitre.org/, accessed: 2026.

[9] Shadcn Contributors, *Shadcn UI Documentation*, https://ui.shadcn.com/docs/, accessed: 2026.

[10] Git Community, *Git Documentation*, https://git-scm.com/doc, accessed: 2026.

[11] Postman, Inc., *Postman Documentation*, https://learning.postman.com/docs/, accessed: 2026.

[12] Microsoft, *Visual Studio Code Documentation*, https://code.visualstudio.com/docs, accessed: 2026.

[13] Generic SIEM Literature, *Security Information and Event Management: Concepts and Best Practices*, various sources, accessed: 2026.

[14] I. Akhamal, *Into the SIEM Architecture and Data Flow*, Medium, https://medium.com/@imaneakhamal/into-the-siem-architecture-and-data-flow-0a35c528f537, accessed: 2026.

[15] M. Seidl et al., *UML @ Classroom: An Introduction to Object-Oriented Modeling*, Springer, 2015.

[16] TryHackMe, *Introduction to SIEM*, https://tryhackme.com, accessed: 2026.

[17] K. C. Laudon and J. P. Laudon, *Management Information Systems: Managing the Digital Firm*, 16th ed. Pearson, 2020.

[18] National Institute of Standards and Technology (NIST), *Cybersecurity Framework*, version 2.0, U.S. Department of Commerce, 2024. https://www.nist.gov/cyberframework

[19] K. Kent and M. Souppaya, *Guide to Computer Security Log Management*, NIST Special Publication 800-92, National Institute of Standards and Technology, 2006. https://csrc.nist.gov/publications/detail/sp/800-92/final

[20] MITRE, *ATT&CK Technique T1110: Brute Force*, https://attack.mitre.org/techniques/T1110/, accessed: 2026.

[21] MITRE, *ATT&CK Technique T1498: Network Denial of Service*, https://attack.mitre.org/techniques/T1498/, accessed: 2026.