DESIGN AND IMPLEMENTATION OF ALERT MANAGEMENT SYSTEM FOR SMALL AND MEDIUM ENTERPRISES.

BY

GABRIEL A. IGHIETSEMHE

PSC2102129

DEPARTMENT OF COMPUTER SCIENCE,

FACULTY OF COMPUTING,

UNIVERSITY OF BENIN,

BENIN CITY,

EDO STATE, NIGERIA.

NOVEMBER 2025

DESIGN AND IMPLEMENTATION OF ALERT MANAGEMENT SYSTEM FOR SMALL AND MEDIUM ENTERPRISES.

BY

GABRIEL A. IGHIETSEMHE

PSC2102129

A PROJECT REPORT SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE, FACULTY OF COMPUTING, UNIVERSITY OF BENIN, BENIN CITY, IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF A BACHELOR OF SCIENCE (B.Sc.) DEGREE IN COMPUTER SCIENCE

NOVEMBER 2025

CERTIFICATION

This is to certify that this project work was carried out by **IGHIETSEMHE GABRIEL AMIEMENOGHENA** with Matriculation Number **PSC21020129** under my supervision. It is adequate and satisfactory, both in scope and content, for the award of Bachelor of Science (B.Sc.) Degree in Computer Science of the University of Benin.

_____ _____

Dr. G.O. AZIKEN **DATE**

Project Supervisor

DECLARATION

I, **IGHIETSEMHE GABRIEL AMIEMENOGHENA**, with Matriculation number **PSC2102129**, do hereby declare that:

1. This project work is based on a study undertaken by me in the Department of Computer Science, University of Benin, Benin City, under the supervision **Dr. G.O. AZIKEN.**

2. This research work has not been previously submitted for the award of degree elsewhere.

3. All ideas and views are a product of my personal research; and where the views of others have been expressed; they were duly acknowledged.

4. All liabilities arising from the study are entirely mine and not of the supervisor.

_____ _____

GABRIEL A. IGHIETSEMHE **DATE**

APPROVAL

This project work is hereby approved in partial fulfillment of the requirements for the award of Bachelor of Science (B.Sc.) Degree in Computer Science from the University of Benin.

DR. (MRS.) A. R. USIOBAFO

Head of Department

DATE

DEDICATION

I want to dedicate this project to God Almighty for seeing me through from the beginning till now, to my wonderful family and well-wishers for their constant love, support and care, and finally to all the friends Joshua, Paschal, Purity, Nehita and the Zeqah community for contributing to my academic journey.

ACKNOWLEDGEMENT

My utmost acknowledgement goes to God Almighty for giving me the strength, wisdom and direction throughout my academic journey. I would like to express my gratitude to my project supervisor DR. (MRS.) G.O AZIKEN for her consistent guidance and support towards ensuring the successful completion of this project. I would also like to specially thank to the Head of Computer Science Department Dr. (Mrs.) A.R. Usiobaifo and other lecturers in the Department of Computer Science who I have been opportune to cross paths with, and have impacted me immensely these past few years: Prof. (Mrs.) S. Konyeha, Prof. G.O. Ekuobase, Prof. K.C. Ukaoha, Prof. A.A. Imiavan, Prof. (Mrs.) F. Egbokhare, Prof. (Mrs.) V.V.N. Akwukwuma, Prof. F.I. Amadin, Prof. (Mrs.) V.I. Osubor, Dr. (Mrs.) Aziken, Dr. F.O. Chete, Dr. (Mrs) R.O. Osaseri, Dr. F.O. Oliha, Dr. J.C. Obi, Mr. P. E.B. Imiefoh, Mr. I.E. Obasohan, Mr. K.O. Otokiti, Mr. I.E. Obayagbonna, Mrs. R.I. Izevbizua, Mr. E.C. Igodan, Miss L.O.Usiosefe , Mr J. Okhuoya, Prof. F.A.U. Imouokhome, Mrs. J.I. Adun, Dr. E. Nweli and Mr. D.N. Idehen.

ABSTRACT

Small and Medium Enterprises (SMEs) often face challenges in managing the overwhelming number of security alerts generated by their IT systems. Traditional alert systems lack contextual intelligence, leading to alert fatigue, delayed responses, and missed critical incidents. This study presents a **context-aware Alert Management System** that enhances prioritization accuracy by incorporating operational factors such as alert frequency, entity type, business hours, and historical severity.

The system was designed and implemented using a React-based simulation environment with 50 synthetic alerts representing realistic SME security events. Comparative evaluation between a baseline model ((Severity + Criticality)/2) and an enhanced model ((Severity + Criticality + Context Factor)/3) demonstrated a 42.42% reduction in alert fatigue and complete elimination of false-positive high-priority alerts while maintaining 100% detection of genuine threats.

The findings confirm that context-aware alert management significantly improves prioritization accuracy and analyst efficiency. The proposed framework provides SMEs with a cost-effective, transparent, and scalable solution for strengthening their cybersecurity posture and improving real-time incident response.

TABLE OF CONTENTS

CERTIFICATION i

DECLARATION ii

APPROVAL iii

DEDICATION iv

ACKNOWLEDGEMENT v

ABSTRACT vi

LIST OF FIGURES xii

LIST OF TABLES xiii

CHAPTER ONE 1

INTRODUCTION 1

1.1 Background of the Study 1

1.2 Statement of Problem 2

1.3 Aim and Objectives 3

1.4 Research Questions 4

1.5 Significance of the Study 5

1.6 Scope of the Study 6

1.7 Definition of Terms 6

CHAPTER TWO 8

LITERATURE REVIEW 8

2.0 Introduction to Literature Review 8

2.1 Structure of the Chapter 8

2.2 Evolution of IT Alert Management Systems 9

2.2.1 Historical Development of Security Alert Systems 9

2.2.2 Traditional Alert Management Approaches 9

2.2.3 Challenges with Early Alert Systems 9

2.2.4 Advancements in Modern Alert Management 10
2.3 Cybersecurity Challenges in SMEs 10
2.3.1 SME Vulnerabilities in Cybersecurity 10
2.3.2 Common Security Threats Faced by SMEs 11
2.3.3 Nigerian SME Context in Cybersecurity 11
2.3.4 Summary of Challenges 12
2.4 Alert Fatigue in Security Operations 12
2.4.1 Understanding Alert Fatigue 12
2.4.2 Causes of Alert Fatigue 12
2.4.3 Impact of Alert Fatigue on SMEs 13
2.4.4 Summary of Alert Management Problems 13
2.5 Alert Prioritization Techniques 14
2.5.1 Rule-Based Prioritization Methods 14
2.5.2 Severity and Entity-Based Approaches 14
2.5.3 Research on Reducing False Positives 15
2.5.4 Summary of Prioritization Techniques 15
2.6 Real-Time Notification Systems 15
2.6.1 Multi-Channel Notification Technologies 15
2.6.2 Applications in SMEs 16
2.6.3 Comparison with Traditional Notification Methods 16
2.7 Existing Studies on IT Alert Management 16
2.7.1 Summary of Key Related Works 16
2.7.2 Identified Research Gaps 17
2.7.3 How This Study Addresses the Gaps 18
2.8 Summary of Literature Review 19
2.9 Literature Summary Table 21

CHAPTER THREE 24

METHODOLOGY 24

3.0 Introduction 24

3.1 Research Design 24

3.2 Analysis of Existing System 25

3.2.1 Overview of Existing Alert Prioritization System 25

3.2.2 Existing System Workflow 25

3.2.3 Strengths of the Existing System 26

3.2.4 Limitations of the Existing System 27

3.3 Proposed System 27

3.3.1 Design Philosophy 27

3.3.2 Context-Aware Enhancement 28

3.3.3 Context-Aware Enhancement Components 28

3.3.4 Proposed System Workflow 30

3.3.5 System Architecture 31

3.4 Simulation Methodology 32

3.4.1 Simulation Approach 32

3.4.2 Understanding Severity and Criticality 32

3.4.3 Alert Dataset Configuration 36

3.4.4 Dataset Characteristics 37

3.4.5 Entity Classification 38

3.4.6 Simulation Process 38

3.5 Implementation Details 41

3.5.1 Code Structure 41

3.5.2 Key Functional Logic 42

3.6 Evaluation Framework 43

3.6.1 Comparison Methodology 43
3.7 Conclusion 44
CHAPTER 4 45
IMPLEMENTATION AND TESTING 45
4.0 Introduction 45
4.1 Simulation Execution Overview 45
4.2 Baseline Model Results 46
4.2.1 Baseline Priority Distribution 46
4.2.2 Baseline Ranking 48
4.3 Enhanced Model Results 50
4.3.1 Context Factor Analysis 50
4.3.2 Context Factor Distribution 51
4.3.3 Enhanced Priority Distribution 52
4.3.4 Prioritized Alerts (Enhanced Ranking) 55
4.4 Dashboard Representation 56
4.4.1 Alert Overview Dashboard 56
4.4.2 Prioritized Alerts Dashboard 58
4.4.3 Email Alert Notification 62
4.5 Comparative Analysis: Baseline Vs. Enhanced 63
4.5.1 Priority Score Changes 63
4.5.2 Ranking Adjustments and Context Influence 63
4.5.3 False Positive Reduction 64
4.5.4 Critical Alert Prioritization Accuracy 66
4.6 Discussion and Interpretation 66
4.6.1 Key Findings 66
4.6.2 Alignment with Research Questions 67

4.7 Conclusion 69

CHAPTER 5 70

SUMMARY, CONCLUSION, AND RECOMMENDATIONS 70

5.0 Introduction 70

5.1 Summary 70

5.2 Conclusion 70

5.3 Recommendations 71

5.4 Future Work 72

REFERENCES 74

LIST OF FIGURES

Figure 1: Existing System Flowchart (Bassey et al., 2024) 26

Figure 2: Proposed System Flowchart 30

Figure 3: System Architecture 31

Figure 4 Alert Dataset Configuration 37

Figure 5: Data Loading 39

Figure 6: Simulation Process 41

Figure 7: Code Structure 42

Figure 8: Baseline Priority Distribution 48

Figure 9: Context Factor Application 51

Figure 10: Context Factor Distribution 52

Figure 11:Enhanced Priority Distribution 53

Figure 12: Alert Overview Dashboard 56

Figure 13: Prioritized Active Alerts Table 61

Figure 14: Automated Security Email Notification 62

Figure 15: Category changes 64

LIST OF TABLES

Table 1: Literature Summary 23

Table 2: Context Factor Elements 29

Table 3: Severity Levels and Classifications 34

Table 4: Criticality levesls and Classification 36

Table 5: Dataset Characteristics 37

Table 6: Simulation Process 39

Table 7: Key functional logic 43

Table 8: Baseline Priority Distribution 47

Table 9: Baseline Priority Ranking 49

Table 10: Context Factor Analysis 50

Table 11: Context factor distribution] 51

Table 12: Enhanced priority distribution 53

Table 13: Enhanced priority ranking 55

Table 14: Ranking adjustments analysis 64

Table 15: Identified false positives 65

Table 16: Critical treats analysis 66

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Small and Medium Enterprises (SMEs) form a vital part of national economies, especially in developing countries such as Nigeria, where they contribute significantly to employment, innovation, and overall economic growth (Onuwabhagbe et al., 2023). However, as these organizations continue to adopt digital technologies to enhance efficiency and competitiveness, they are increasingly exposed to cybersecurity risks.

Unlike large corporations that maintain dedicated cybersecurity teams and advanced monitoring infrastructure, most SMEs lack the financial resources, technical expertise, and strategic capacity to manage their IT security operations effectively (Chidukwani et al., 2022). This limitation makes them more susceptible to cyberattacks that can disrupt business operations, erode customer confidence, and cause severe financial losses.

One major cybersecurity challenge confronting SMEs is the management of security alerts. Security tools such as firewalls, antivirus programs, intrusion detection systems (IDS), and network monitors generate hundreds or even thousands of alerts each day. Many of these alerts are repetitive, low-priority, or false positives, which often results in "alert fatigue." Alert fatigue occurs when IT personnel become desensitized to alerts because of their excessive volume (Tariq et al., 2025) . In such conditions, critical alerts that signal genuine threats may be ignored or delayed, increasing the risk of a successful cyberattack.

This problem is especially pronounced among Nigerian SMEs, which are often targeted by cybercriminals who exploit weak defenses and limited monitoring capabilities (Ali et al., 2025) . Without intelligent systems to filter and prioritize alerts, these businesses find it difficult to distinguish between normal system activities and actual security incidents.

Effective alert management involves more than simply generating alerts. It requires collecting, categorizing, prioritizing, and routing alerts to the appropriate personnel for timely response. Bassey et al. (2024) emphasize that prioritization techniques which consider both the severity level of an alert and the criticality of the affected entity can significantly improve response efficiency and decision-making. In a similar view, Z. S. Younus & M. Alanezi, (2023) explain

that network security monitoring becomes more effective when supported by structured processes for analyzing and escalating alerts rather than relying solely on raw event data.

Researchers and practitioners have proposed several frameworks aimed at improving alert handling. Creasey (2015) highlights the importance of centralized logging and monitoring as the foundation of a structured alert management process. Building on this principle, a system specifically tailored for SMEs can help strike a balance between automation and human oversight. Such a system can streamline alert processing, reduce unnecessary notifications, and ensure that critical incidents receive prompt attention.

In summary, the background highlights two major issues: the overwhelming volume and complexity of security alerts, and the limited technical and financial capacity of SMEs to manage them effectively. Developing a real-time alert management system that supports intelligent alert categorization, prioritization, and routing will strengthen the cybersecurity posture of SMEs, reduce false positives, and enable faster responses to genuine threats. This study focuses on designing such a system for the Nigerian SME context, where limited cybersecurity infrastructure makes efficient alert management both necessary and urgent.

.

1.2 Statement of Problem

Small and Medium Enterprises (SMEs) in Nigeria face increasing challenges in managing the large number of security alerts generated by their IT systems. Each day, security tools such as firewalls, antivirus software, intrusion detection systems, and network monitors produce hundreds of notifications about potential threats. However, most of these alerts are repetitive, low-priority, or false positives. The high alert volume overwhelms IT personnel and makes it difficult to identify which alerts require immediate attention.

The major challenge is not only the volume of alerts but the absence of an effective alert prioritization mechanism. Most Nigerian SMEs rely on manual review processes where IT staff must examine each alert individually to determine its importance. This manual approach is inefficient and prone to human error. When faced with hundreds of similar alerts, staff often struggle to distinguish between harmless events and critical security incidents that could indicate ongoing attacks.

False positives further worsen the problem. Many alerts that appear severe are later discovered to be normal system behavior or configuration issues. Studies indicate that false

positive rates in typical security monitoring environments can exceed 50 percent. As a result, valuable time is wasted investigating benign events, while genuine high-risk alerts may be delayed or overlooked. For SMEs with limited manpower, this misallocation of attention increases vulnerability to real attacks.

Another problem is the lack of context-aware prioritization. Existing systems often treat all alerts with the same severity label equally, without considering factors such as the business criticality of the affected system, the frequency of similar alerts, or the timing of the event. For example, a high-severity alert on a test workstation may receive the same level of attention as a similar alert on a production database server, even though the business impact differs significantly. This lack of context causes inefficiency in alert response and contributes to alert fatigue.

Current commercial solutions are not optimized for SMEs. Most are designed for large enterprises with sophisticated Security Operations Centers (SOCs) and experienced analysts. They are often expensive and complex to configure, making them unsuitable for small organizations with limited technical capacity. This leaves Nigerian SMEs without affordable, intelligent tools that can automatically classify and prioritize alerts based on severity and business relevance.

Therefore, there is a critical need for a real-time alert prioritization system tailored to the operational realities of SMEs. Such a system should be capable of intelligently ranking alerts according to their severity, criticality, and frequency. It should also route the most important alerts to the appropriate personnel promptly. Addressing this need will help SMEs reduce false positives, minimize alert fatigue, and improve response efficiency, thereby strengthening their overall cybersecurity posture.

1.3 Aim and Objectives

Aim

The aim of this project is to design and implement a real-time alert management system that can effectively collect, process, and prioritize security alerts from multiple sources within SME network environments.

Objectives

The objectives of this study are:

1. To design an integrated alert collection and processing system.
2. To implement intelligent alert prioritization mechanisms.
3. To evaluate the effectiveness of the proposed system.

1.4 Research Questions

To achieve the project objectives, this study answers three key research questions focused on how alerts can be collected, prioritized, and managed effectively for SMEs:

1. How can an integrated system be designed to effectively collect and process security alerts from multiple sources in SME network environments?

- The study focuses on creating a unified alert collection and processing framework that consolidates events from simulated firewall logs, access violations, and device health checks. Centralized alert processing improves visibility across SME infrastructures and reduces the likelihood of missed incidents. Z. S. Younus & M. Alanezi (2023) emphasize that integrated monitoring tools provide consistent visibility into network events, while Creasey (2015) notes that effective log collection is the foundation of any security monitoring system.

2. What intelligent mechanisms can be applied to prioritize alerts in order to reduce false positives and highlight critical events?

- This study adopts the severity averaging and entity-based prioritization algorithm described by Bassey et al. (2024) The algorithm calculates a priority score for each alert by combining three factors: the severity level of the alert, the criticality rating of the affected entity, and the historical record of alerts associated with that entity. By averaging severity values and incorporating entity criticality, the system ensures that alerts involving business-critical assets are prioritized above less critical systems, even when raw severity labels appear similar. This approach reduces false positives and ensures that SMEs can allocate their limited security resources more effectively.

3. How effective is the proposed real-time alert management system in improving alert prioritization and response within simulated SME environments?

- Effectiveness will be measured through controlled simulations that assess improvements in response time, reduction of false positives, and clarity of alert

routing. Tariq et al. (2025) highlight that reducing alert fatigue is critical for maintaining responsiveness in security teams, while Ali et al. (2025) stress that timely incident response is essential for SMEs, where IT personnel often manage multiple roles simultaneously.

1.5 Significance of the Study

This study is significant because it addresses one of the most pressing cybersecurity challenges faced by Small and Medium Enterprises (SMEs): the inability to effectively manage and respond to the overwhelming number of security alerts generated by IT systems. Alert fatigue, coupled with limited technical expertise, often results in critical threats being overlooked, leading to serious financial and operational risks.

Academic Contribution

This research contributes to the growing body of knowledge on cybersecurity for SMEs by focusing on real-time alert management rather than traditional intrusion detection. While many studies have explored general cybersecurity frameworks, this study emphasizes alert collection, prioritization, and routing as distinct processes that can significantly improve organizational security posture. By applying and testing an entity-based alert prioritization algorithm, the study provides new insights into how SMEs can balance automation with human decision-making in resource-constrained environments (Bassey et al., 2024).

Practical Contribution for SMEs

For SMEs, particularly in Nigeria, this research provides a realistic and cost-effective framework for improving incident response. SMEs typically lack dedicated cybersecurity teams and enterprise-grade tools (Onuwabhagbe et al., 2023) . The proposed system demonstrates how simple technologies and rule-based mechanisms can be used to filter noise, reduce false positives, and ensure critical alerts are acted upon promptly. This is especially relevant for Nigerian SMEs that operate under tight budgets but remain highly vulnerable to cyberattacks (Chidukwani et al., 2022a).

Industry and Policy Relevance

The findings of this study have implications beyond individual organizations. By showcasing a scalable and adaptable alert management approach, the project supports broader cybersecurity resilience goals for SMEs, which are critical to national economies.

Furthermore, the insights may guide IT policy discussions in Nigeria by highlighting the importance of tailored, context-specific solutions for small businesses rather than adopting one-size-fits-all enterprise models (Ali et al., 2025).

1.6 Scope of the Study

This study is limited to the design and evaluation of a real-time alert management system specifically for Small and Medium Enterprises (SMEs). The system focuses on three major functions: collecting alerts from multiple IT sources, applying rule-based and context-aware prioritization to reduce false positives, and routing critical alerts to designated personnel for timely response.

The project does not attempt to build a full-scale Security Information and Event Management (SIEM) platform. Instead, it provides a simplified but practical framework that SMEs can adopt with minimal resources. The alerts used in this study are alerts simulated through a predefined dataset, which streams event data such as firewall logs, device errors, and access violations.

The scope of the system is confined to:

1. Basic alert collection from simulated firewalls, device failures, and access violations.
2. Application of an alert prioritization algorithm that considers severity, frequency, and entity criticality.
3. Real-time notifications through email and dashboard interfaces.
4. Evaluation of the system's performance in reducing false positives and improving response times within a simulated SME environment.

This study does not cover advanced features such as machine learning-driven threat detection, large-scale enterprise integration, or live deployment in production SME environments. Instead, it emphasizes a feasible and scalable solution that addresses the alert management challenges most commonly faced by SMEs with limited resources.

1.7 Definition of Terms

To provide clarity and ensure consistent understanding, the following terms are defined as they are used in this study:

1 Small and Medium Enterprises (SMEs): Businesses with limited resources, smaller workforce sizes, and modest IT infrastructure compared to large enterprises. In Nigeria, SMEs are particularly important drivers of the economy but often face challenges in cybersecurity due to resource constraints (AL-Dosari & Fetais, 2023).

2 Alert Fatigue: A condition where IT or security personnel become desensitized to alerts due to the overwhelming number of notifications, often leading to missed or ignored critical warnings (Tariq et al., 2025).

3 Real-Time Alert Management System: A system designed to collect, categorize, prioritize, and route IT security alerts immediately after detection, ensuring that critical incidents are addressed without delay.

4 Prioritization: The process of ranking alerts based on severity, frequency, and impact on business operations, so that the most important alerts receive attention first (Bassey et al., 2024).

5 Notification Channels: Communication pathways such as email, SMS, or dashboard interfaces used to deliver alerts to responsible personnel.

6 Pattern Recognition (in alert processing): The use of rule-based or algorithmic techniques to detect repetitive or suspicious alert patterns that may indicate real threats rather than isolated false positives.

7 Context Factor (CF): A numerical modifier representing operational conditions (e.g., frequency, entity type, business hours, historical severity) used to adjust alert priority in this study's enhanced model.

CHAPTER TWO

LITERATURE REVIEW

2.0 Introduction to Literature Review

This chapter reviews existing literature relevant to the development of a real-time IT alert management system for Small and Medium Enterprises (SMEs). The review is structured to provide a clear understanding of how alert management systems have evolved, the unique cybersecurity challenges facing SMEs, and the issues of alert fatigue that hinder effective incident response. It also examines prioritization techniques, notification methods, and the role of network simulation tools in testing such systems. The chapter concludes with a discussion of existing studies, identified research gaps, and how the present study aims to address them.

2.1 Structure of the Chapter

This chapter is organized into the following sections:

1. Evolution of Alert Management Systems: Reviews the historical development of security alert systems, traditional approaches to alert handling, and the limitations of early systems.
2. Cybersecurity Challenges in SMEs: Examines the unique vulnerabilities of SMEs, the types of security threats they face, and highlights the Nigerian SME context.
3. Alert Fatigue in Security Operations: Discusses the concept of alert fatigue, its causes, and its impact on SMEs.
4. Alert Prioritization Techniques: Reviews different prioritization strategies, including rule-based, severity-based, and entity-based approaches, with emphasis on reducing false positives.
5. Real-Time Notification and Systems: Explores multi-channel notification technologies, and their relevance to SME security environments.
6. Existing Studies on Alert Management: Summarizes key related works, identifies research gaps, and explains how this study addresses those gaps.
7. Summary of Literature Review: Provides an overview of the main insights from the reviewed literature and sets the foundation for the next chapter.

2.2 Evolution of IT Alert Management Systems

2.2.1 Historical Development of Security Alert Systems

The earliest security alert systems emerged alongside firewalls, intrusion detection systems (IDS), and antivirus tools. These tools generated basic logs that recorded system events such as failed logins, blocked traffic, or virus detections. In the early stages, administrators had to manually review these logs to identify potential threats (Creasey, 2015). While this manual approach provided visibility, it quickly became unsustainable as IT infrastructures grew in scale and complexity.

Over time, basic automation was introduced in the form of threshold-based alerts, where notifications were triggered if certain conditions were met, such as multiple failed login attempts. However, these systems were siloed, with each tool operating independently, limiting the ability of IT teams to gain a unified view of security incidents (Badra et al., 2016).

2.2.2 Traditional Alert Management Approaches

Traditional alert management relied heavily on rule-based mechanisms and static detection patterns. For example, predefined rules would flag events such as excessive failed logins or unusual network traffic. These rules worked well in identifying known threats but were rigid and unable to adapt to emerging attack techniques.

Another characteristic of traditional systems was their focus on generating as many alerts as possible to avoid missing threats. While this approach increased visibility, it also resulted in large volumes of unfiltered alerts that overwhelmed IT staff. According to Z. S. Younus & M. Alanezi, (2023) , the absence of correlation between alerts from different systems created blind spots, as administrators could not easily identify whether multiple alerts were linked to the same incident.

2.2.3 Challenges with Early Alert Systems

The main challenge with early alert systems was the lack of prioritization. All alerts were treated with equal importance, meaning that low-level notifications could easily bury critical incidents. This created significant inefficiencies in security operations. Additionally, these

systems generated a high number of false positives, further increasing the workload for IT staff (Chidukwani et al., 2022).

For SMEs in particular, the burden of reviewing large numbers of alerts was impractical, given their limited staff and expertise. This meant that many SMEs either ignored alerts or responded too slowly to real threats, leading to higher risks of breaches.

2.2.4 Advancements in Modern Alert Management

Modern alert management systems have evolved to address these shortcomings by introducing features such as:

1. Centralized Monitoring: Tools now integrate logs and alerts from multiple sources into a single platform, improving visibility (Creasey, 2015)
2. Correlation Engines: Advanced systems link related alerts to identify broader attack patterns, reducing duplication and noise.
3. Alert Prioritization: Newer approaches apply severity scoring, frequency analysis, and entity awareness to classify alerts by importance. Bassey et al. (2024) highlight severity averaging and entity-based models as effective ways of reducing false positives.
4. Cloud-Based Architectures: With the adoption of cloud computing, alert management systems can scale more easily and offer enterprise-grade monitoring to SMEs without requiring costly infrastructure (Ali et al., 2025).

These advancements mark a shift from reactive to more proactive alert management, enabling organizations to focus on critical threats while minimizing distractions from low-priority events.

2.3 Cybersecurity Challenges in SMEs

2.3.1 SME Vulnerabilities in Cybersecurity

SMEs are recognized as highly vulnerable to cyberattacks because they often operate with limited resources and lack specialized cybersecurity teams. Many rely on basic IT personnel who perform general system administration rather than dedicated security monitoring. This resource gap reduces their ability to implement strong defense mechanisms such as continuous monitoring, advanced firewalls, or incident response strategies (AL-Dosari &

Fetais, 2023) . In practice, SMEs typically prioritize business operations and cost reduction over cybersecurity investment, leaving systems exposed.

2.3.2 Common Security Threats Faced by SMEs

SMEs encounter a variety of cyber threats, some of which can be devastating due to the limited resilience of these businesses.

- Phishing and Social Engineering: Cybercriminals exploit the lack of staff training in SMEs to trick employees into disclosing sensitive credentials or downloading malicious files (Chidukwani et al., 2022).
- Ransomware Attacks: SMEs are often targeted with ransomware because they are less likely to have robust backup and recovery strategies. Such attacks can lead to prolonged downtime and financial loss.
- Insider Threats: Disgruntled employees or careless insiders may inadvertently expose sensitive business data.
- System Misconfigurations and Unpatched Vulnerabilities: (Z. S. Younus & M. Alanezi, 2023) stress that improper network configurations and failure to update software leave many SMEs open to exploitation.
- Credential Theft: Weak password policies and the lack of multi-factor authentication are common in SMEs, making credential theft an easy path for attackers.

These threats impact not just IT infrastructure but also the trustworthiness of SMEs, especially when customer data is compromised.

2.3.3 Nigerian SME Context in Cybersecurity

In Nigeria, the cybersecurity challenges facing SMEs are even more severe due to economic and infrastructural limitations Onuwabhagbe et al. (2023) highlight that Nigerian SMEs typically allocate little to no budget for cybersecurity tools, relying instead on low-cost or free solutions that do not provide comprehensive defense. This makes them more vulnerable to sophisticated attacks such as ransomware and phishing.

Another key issue is cybersecurity awareness. Many SME employees lack the knowledge to recognize and respond to security threats, which increases the likelihood of successful attacks. Weak policy enforcement in Nigeria further complicates matters, as businesses often operate

without clear regulatory guidance or compliance frameworks. According to Ali et al. (2025), the gap in compliance with security standards leaves SMEs exposed not only to attackers but also to potential legal consequences when breaches occur.

2.3.4 Summary of Challenges

From the reviewed literature, it is evident that SMEs face:

1. Limited financial and human resources for cybersecurity (AL-Dosari & Fetais, 2023).
2. High exposure to common threats such as phishing, ransomware, and insider misuse (Chidukwani et al., 2022).
3. Nigeria-specific issues of low budgets, poor awareness, and weak regulatory enforcement (Onuwabhagbe et al., 2023).

These challenges make it difficult for SMEs to implement effective security measures, underscoring the need for simplified but intelligent alert management systems that can reduce alert fatigue and enable faster responses to incidents.

2.4 Alert Fatigue in Security Operations

2.4.1 Understanding Alert Fatigue

Alert fatigue occurs when security personnel become overwhelmed by the high volume of alerts generated by monitoring tools. In such situations, IT teams often become desensitized to alerts, leading to slow responses or, in some cases, ignoring alerts altogether (Tariq et al., 2025). This reduces the overall effectiveness of security operations, as critical incidents may go unnoticed.

The problem is particularly severe in SMEs, where a single IT staff member may be responsible for both daily technical support and security monitoring. When faced with hundreds of alerts daily, staff can struggle to separate genuine threats from false positives, creating significant operational risks.

2.4.2 Causes of Alert Fatigue

Several factors contribute to alert fatigue in IT environments:

1. High Volume of Alerts: Security Information and Event Management (SIEM) systems and other monitoring tools often generate thousands of alerts daily, many of which are repetitive or redundant (Creasey, 2015).
2. False Positives: A large percentage of alerts do not correspond to real threats. Z. S. Younus & M. Alanezi, (2023) note that overly sensitive detection rules in traditional systems exacerbate this problem.
3. Lack of Prioritization: Without mechanisms to classify alerts by severity, IT teams treat all alerts equally, making it difficult to focus on critical issues (Bassey et al., 2024).
4. Limited Context: Many alerts lack sufficient contextual information for IT staff to quickly determine their importance, leading to wasted time investigating non-critical issues (Tariq et al., 2025).

2.4.3 Impact of Alert Fatigue on SMEs

The consequences of alert fatigue are particularly damaging for SMEs. Unlike large enterprises that can maintain specialized security operation centers (SOCs), SMEs often lack the manpower to handle large alert volumes. According to Chidukwani et al., (2022) , this leads to delayed responses, missed threats, and reduced confidence in monitoring tools.

In Nigeria, Onuwabhagbe et al., (2023) report that many SMEs abandon security monitoring tools altogether due to the burden of excessive alerts, further increasing their vulnerability to attacks. This creates a vicious cycle where inadequate monitoring leads to higher risks, and higher risks discourage further investment in security tools.

2.4.4 Summary of Alert Management Problems

From the reviewed literature, the key problems of alert fatigue include:

1. Excessive alert volumes that overwhelm limited IT staff.
2. High rates of false positives that waste time and resources.
3. Poor prioritization mechanisms that fail to distinguish between critical and routine alerts.
4. Lack of contextual detail in alerts, making investigation difficult.

These challenges underscore the importance of intelligent alert management systems that can filter, prioritize, and route alerts in real time to improve response efficiency in SMEs.

2.5 Alert Prioritization Techniques

2.5.1 Rule-Based Prioritization Methods

Rule-based prioritization is one of the earliest and most widely adopted techniques for managing security alerts. In this approach, predefined conditions or thresholds are used to classify alerts into categories such as critical, high, medium, or low. For example, repeated failed login attempts within a short timeframe may trigger a high-priority alert, while routine firewall logs may be classified as low-priority. According to Creasey, (2015) , rule-based systems remain popular due to their simplicity and transparency, making them easy for IT staff in SMEs to understand and maintain.

However, the rigidity of static rules means they often generate large numbers of alerts that lack contextual differentiation. Z. S. Younus & M. Alanezi, (2023) note that SMEs in particular struggle with this approach, as it leads to alert fatigue when too many low-level alerts demand manual review.

2.5.2 Severity and Entity-Based Approaches

To improve on static rules, researchers have proposed severity-based and entity-based prioritization methods. Severity-based models assign scores to alerts depending on their potential impact, while entity-based approaches consider the importance of the asset or entity affected. For instance, an alert involving a database server that stores sensitive customer information would be prioritized over one involving a general workstation.

Bassey et al., (2024) propose a severity averaging model that computes average severity scores across multiple attributes of an alert, providing a more balanced assessment of its importance. They also highlight the effectiveness of entity-based prioritization, where alerts linked to critical entities (such as finance systems or health records in SMEs) are automatically ranked higher. This dual approach reduces the chances of critical alerts being buried under less significant events.

2.5.3 Research on Reducing False Positives

False positives remain a major challenge in alert management, as they consume time and resources without corresponding to real threats. Several studies emphasize that prioritization techniques can significantly reduce the impact of false positives. Bassey et al., (2024) demonstrate that by applying severity averaging alongside entity awareness, organizations can reduce noise in their monitoring systems while still maintaining high sensitivity to genuine threats.

Similarly, Tariq et al. (2025) argue that integrating prioritization with contextual enrichment such as linking alerts with system logs or user behavior data further improves accuracy. For SMEs, this means IT teams can focus their limited resources on high-value incidents, improving efficiency and reducing the risk of missed critical threats.

2.5.4 Summary of Prioritization Techniques

From the reviewed literature, prioritization can be grouped into three main approaches:

1. Rule-based methods: Simple to implement but often rigid and prone to generating high alert volumes (Creasey, 2015)
2. Severity and entity-based approaches: More effective in reflecting the true business impact of alerts, as demonstrated by (Bassey et al., 2024).
3. Contextual methods to reduce false positives: Techniques such as severity averaging and enrichment reduce noise and improve response efficiency (Tariq et al., 2025).

For SMEs, a hybrid of these techniques offers the most practical solution, balancing simplicity with the need for intelligent prioritization.

2.6 Real-Time Notification Systems

2.6.1 Multi-Channel Notification Technologies

Effective alert management requires that notifications reach the right personnel through reliable communication channels. Real-time systems typically use email, SMS, mobile push notifications, or web dashboards to deliver alerts. According to Z. S. Younus & M. Alanezi, (2023) , the availability of multiple notification channels increases the likelihood that alerts

will be acknowledged quickly, especially in organizations where IT staff may not always be at their desks.

For SMEs, email and dashboards are the most practical channels, as they are cost-effective and easy to integrate with existing systems. However, adding SMS or mobile notifications provides redundancy, ensuring critical alerts are not missed during system downtime or when personnel are off-site.

2.6.2 Applications in SMEs

For SMEs, the use of real-time notification and escalation systems provides three main advantages:

1. Improved Response Time: Alerts are delivered instantly to relevant staff, reducing delays in incident handling.
2. Resource Optimization: By ensuring that alerts escalate only, when necessary, SMEs can manage limited human resources more efficiently (Ali et al., 2025).
3. Accountability and Assurance: Escalation provides an audit trail that demonstrates how alerts were managed, which can be important for compliance and building customer trust (Onuwabhagbe et al., 2023).

2.6.3 Comparison with Traditional Notification Methods

Traditional notification methods, such as email-only alerts without escalation, often result in delays because staff may overlook or ignore alerts during busy periods. In contrast, modern systems combine real-time delivery with multi-channel redundancy and structured escalation, reducing the risk of missed or unresolved incidents. This makes them especially valuable in SMEs, where fewer staff must handle a broad range of IT and security responsibilities.

2.7 Existing Studies on IT Alert Management

2.7.1 Summary of Key Related Works

Research on alert management spans across network monitoring, alert prioritization, and organizational applications. Early guidelines, such as Creasey (2015) , emphasize the necessity of structured monitoring and logging as a foundation for any alert management system. This work laid the groundwork for later studies that examined how to process alerts more intelligently.

Several surveys highlight limitations of traditional alert systems. Z. S. Younus & M. Alanezi, (2023) review network monitoring tools and functionalities, pointing out the challenges of scalability and the persistence of false positives. Similarly, Chidukwani et al. (2022) survey cybersecurity issues in small and medium businesses, stressing that SMEs lack both the financial and technical resources to adopt enterprise-level monitoring solutions.

Recent works address alert fatigue, a key operational challenge. Tariq et al. (2025) identify overwhelming alert volumes and lack of prioritization as major causes of inefficiency in Security Operations Centres (SOCs). They recommend integrating prioritization and contextual enrichment to reduce noise. Their findings directly support the idea that SMEs, with fewer IT staff, would struggle even more under alert fatigue.

In terms of technical approaches, Bassey et al. (2024) present one of the most focused contributions by developing severity averaging and entity-based prioritization models. Their results show improved accuracy in ranking alerts by combining severity scores with the criticality of the affected entity. This offers a strong basis for implementing intelligent prioritization in SME contexts.

The SME context itself is well-covered in works by AL-Dosari & Fetais (2023), who analyze risk management frameworks tailored for smaller enterprises, and Onuwabhagbe et al. (2023), who focus specifically on Nigerian SMEs. Both highlight gaps in awareness, compliance, and resource allocation, reinforcing the argument for simplified and cost-effective solutions.

Other researchers such as Ali et al. (2025) emphasize the importance of compliance-driven infrastructure in SMEs, noting that non-compliance with standards leaves organizations exposed. Their systematic review stresses the need for adaptable solutions that both detect threats and align with regulations. Together, these studies show a progression from general alert monitoring towards more sophisticated and context-aware approaches.

2.7.2 Identified Research Gaps

From the reviewed literature, several gaps become evident:

1. SME-Specific Applications**:** Much of the existing research on alert management (e.g., Tariq et al., 2025; Z. S. Younus & M. Alanezi, 2023) focuses on large enterprises and SOCs, with limited exploration of SME-specific environments.

2. Integration of Prioritization and Notification: Technical studies such as Bassey et al. (2024) focus on prioritization models but often stop short of implementing integrated notification or escalation mechanisms that SMEs require for timely responses.

3. Operational Context in Developing Economies: While global surveys exist (Chidukwani et al., 2022) , only a few studies such as Onuwabhagbe et al. (2023) examine Nigerian SMEs, leaving a gap in understanding how local infrastructure and policy constraints affect alert management.

4. Testing in Simulated Environments: Most existing works present frameworks or theoretical models without applying them in controlled test environments. For SMEs that cannot afford live testing, simulation-based approaches in literature (Badra et al., 2016).

5. Holistic Systems for SMEs: Current studies often address isolated components (e.g., prioritization, logging, compliance), but very few attempt to integrate collection, prioritization, notification, and escalation into a single SME-ready system.

2.7.3 How This Study Addresses the Gaps

This study seeks to bridge these gaps in the following ways:

1. SME-Centric Design: Unlike much of the prior work focusing on enterprises, this project explicitly targets SMEs, with emphasis on Nigerian businesses that face acute resource and compliance challenges (AL-Dosari & Fetais, 2023; Onuwabhagbe et al., 2023).

2. Integrated Framework: The proposed system combines alert collection, severity/entity-based prioritization, and real-time notification with escalation, creating a holistic approach where many studies treat these elements separately (Bassey et al., 2024; Tariq et al., 2025).

3. Simulation-Based Evaluation: By leveraging on predefined datasets, this study tests alert management logic in realistic SME-like conditions. This responds to the lack of simulation-based validation in most existing works (Badra et al., 2016).

4. Contextual Relevance: The project situates itself in the Nigerian SME context, where cyberattacks are increasing but adoption of enterprise solutions remains low

(Onuwabhagbe et al., 2023). The study's results can serve as a blueprint for SMEs in similar developing economies.

5. Contribution to Practical Knowledge: By documenting system design, alert categorization logic, and performance evaluation, the study offers SMEs actionable insights rather than abstract recommendations.

2.8 Summary of Literature Review

This chapter reviewed existing literature on alert management systems, cybersecurity challenges in SMEs, alert fatigue, prioritization techniques, and the role of notification and simulation tools in improving security monitoring. The review revealed several key insights that provide the foundation for this study.

1. Evolution of Alert Management Systems

   Security alert systems have evolved from basic log monitoring into advanced solutions capable of correlation, prioritization, and automated workflows. Traditional systems were heavily rule-based and siloed, often generating excessive volumes of alerts without prioritization (Creasey, 2015; Z. S. Younus & M. Alanezi, 2023a) . Modern systems now incorporate severity-based models, centralization, and cloud-based scalability to address inefficiencies (Ali et al., 2025).

2. Cybersecurity Challenges in SMEs

   SMEs are particularly vulnerable to cyber threats due to limited resources, inadequate expertise, and low investment in cybersecurity. Studies highlight common threats such as phishing, ransomware, and insider misuse, which have disproportionate impacts on smaller organizations (AL-Dosari & Fetais, 2023; Chidukwani et al., 2022a) . In Nigeria, SMEs face additional challenges such as weak regulatory enforcement, lack of awareness, and dependency on free or low-cost tools (Onuwabhagbe et al., 2023).

3. Alert Fatigue and Response Challenges

   Alert fatigue is identified as a major barrier to effective security monitoring. High volumes of repetitive and low-priority alerts reduce staff efficiency and result in missed or delayed responses to genuine threats (Tariq et al., 2025) . SMEs, with

limited IT staff, are even more prone to this problem, making intelligent filtering and prioritization critical for their survival.

4. Alert Prioritization Techniques

   Recent research highlights techniques such as severity averaging and entity-based prioritization, which provide more balanced and context-aware ranking of alerts (Bassey et al., 2024). These approaches address the shortcomings of static rule-based systems by reducing false positives and focusing attention on business-critical events. For SMEs, adopting simplified but effective prioritization models is essential to ensure scarce resources are directed towards the most impactful threats.

5. Real-Time Notification and Escalation Systems

   Effective alert management requires not just prioritization but also timely delivery of alerts through appropriate channels. Multi-channel notifications (e.g., email, dashboards, SMS) and structured escalation procedures improve accountability and reduce response delays (Creasey, 2015; Onuwabhagbe et al., 2023). For SMEs, such systems provide assurance that critical alerts are not lost, even when staff availability is limited.

   Their use enables evaluation of alert collection, prioritization, and notification under realistic conditions.

6. Research Gaps Identified

   Despite these advances, several gaps remain:

   - Limited studies focus specifically on SMEs, especially within Nigerian contexts.
   - Prioritization models are often studied in isolation without integration into real-time notification frameworks.
   - Few works demonstrate practical testing of alert systems using SME-scale simulations.
   - Context-specific insights for resource-constrained environments remain scarce.

8. How This Study Differs

   This study addresses these gaps by:

- Designing a real-time alert management system specifically tailored for SMEs.
- Implementing severity and entity-based context-aware prioritization to reduce false positives and alert fatigue.
- Combining prioritization with real-time notification and escalation mechanisms.
- Providing insights relevant to Nigerian SMEs, where cybersecurity challenges are compounded by resource limitations.

In summary, the literature review establishes that while existing studies provide important technical and theoretical foundations, there is a lack of practical, SME-specific solutions that integrate prioritization, real-time notification, and contextual testing. This study contributes by developing and evaluating a real-time alert management system that bridges this gap, offering both academic and practical value for SMEs.

2.9 Literature Summary Table

No.	Author(s)	Title	Key Findings	Limitations	Relevance to Study
1	AL-Dosari & Fetais (2023)	Risk-Management Framework and Information-Security Systems for Small and Medium Enterprises (SMEs): A Meta-Analysis Approach	SMEs face unique cybersecurity challenges due to limited resources and expertise; Risk management frameworks must be tailored to SME contexts rather than enterprise models; Meta-analysis reveals common security gaps in SME environments; Resource constraints significantly impact security implementation effectiveness	Meta-analysis approach may not capture recent emerging threats; Limited focus on specific technical implementations; Framework proposed but not empirically validated in real SME environments	Establishes SME-specific security challenges and justifies need for tailored solutions
2	Ali et al. (2025)	Cybersecurity Infrastructure Compliance Key Factors to Detect and Mitigate Malware Attacks in SMEs: A Systematic Literature Review	Compliance-driven infrastructure is critical for SME security; Malware detection and mitigation require integrated approaches; Timely incident response is essential for SMEs with limited IT staff; Non-compliance leaves SMEs exposed to attacks and legal consequences	Systematic review methodology may miss gray literature; Focus on malware may not address broader alert management challenges; Limited discussion of practical implementation barriers in resource-constrained environments	Highlights importance of timely response and compliance, supporting need for efficient alert management
3	Badra et al. (2016)	New Technologies, Mobility and Security (NTMS) Conference Proceedings	Simulation tools provide controlled environments for security research; Network simulation enables testing without requiring physical infrastructure; Packet Tracer and similar tools valid for academic security research; Simulation supports reproducible security experiments	Conference proceedings lack detailed implementation specifics; Simulation may not fully capture real-world complexity; Limited validation of simulation accuracy against production environments	Justifies use of simulation methodology for security system evaluation
4	Ban et al. (2023)	Breaking Alert Fatigue: AI-Assisted SIEM Framework for Effective Incident Response	Alert fatigue is a critical problem in Security Operations Centers (SOCs); AI-assisted frameworks can reduce false positive rates significantly; Intelligent alert prioritization improves incident response times; Machine	AI/ML solutions may be too complex and expensive for SMEs; Requires large datasets for training, which SMEs	Identifies alert fatigue problem; highlights need for

			learning approaches require substantial training data and computational resources	may lack; Black-box nature of AI reduces transparency and trust; Limited discussion of simpler, rule-based alternatives	simpler SME-appropriate solutions
5	Bassey et al. (2024)	Alert Prioritization Techniques in Security Monitoring: A Focus on Severity Averaging and Alert Entities	Severity averaging combined with entity-based analysis improves prioritization over simple threshold models; Entity criticality provides crucial business context to technical alerts; Dual-factor approach (Severity + Criticality) / 2 reduces false urgency; Entity-based prioritization ensures business-critical assets receive appropriate attention	Static prioritization: No consideration of operational context (frequency, timing, patterns); Context blindness: All alerts with same severity/criticality receive equal priority regardless of circumstances; Pattern ignorance: Cannot detect coordinated attacks or persistent threats; Limited discussion of real-world implementation in SME environments	PRIMARY BASELINE: Foundation model that this study extends with context-aware intelligence
6	Chidukwani et al. (2022)	A Survey on the Cyber Security of Small-to-Medium Businesses: Challenges, Research Focus and Recommendations	SMEs lack both financial and technical resources for comprehensive cybersecurity; Asset-based risk assessment critical: infrastructure systems warrant different treatment than workstations; Alert fatigue leads to delayed responses and missed threats in SMEs; After-hours coverage challenges particularly acute for SMEs	Survey approach may not capture latest technological developments; Limited specific recommendations for alert management systems; Focus on challenges without detailed solution implementations	Supports Context Factor component: Asset-based risk assessment (entity type classification)
7	Creasey (2015)	Cyber Security Monitoring and Logging Guide	Centralized logging and monitoring foundation for structured alert handling; Effective log monitoring must consider both WHAT events occur and WHEN they occur; Temporal context affects threat significance and organizational response capacity; Structured monitoring processes essential for identifying genuine threats	Best practice guide lacks empirical validation; Published in 2015, may not address modern cloud-based threats; Limited discussion of prioritization beyond basic severity levels; Minimal focus on SME-specific constraints	Supports Context Factor component: Temporal context in security (business hours timing)
8	Onuwabhagbe et al. (2023)	A Cyber Security Framework to Strengthen Small and Medium Scale Enterprises (SMEs) in Nigeria	Nigerian SMEs allocate minimal budgets for cybersecurity tools; Low cybersecurity awareness among SME employees increases vulnerability; Weak policy enforcement in Nigeria leaves SMEs exposed to attacks; Many Nigerian SMEs abandon security tools due to excessive alerts and complexity	Framework proposed but not empirically tested in Nigerian SMEs; Limited discussion of specific technical implementations; Focus on policy recommendations rather than practical system design	Establishes Nigerian SME context; highlights critical need for practical, usable solutions
9	Australian Signals Directorate. (2024)	Best Practices for Event Logging and Threat Detection	Government best practice guidelines for event logging; Structured approach to threat detection and monitoring; Importance of log retention and analysis for security; Framework for effective security event management	Government guideline lacks specific implementation details; May not be tailored to resource-constrained SME environments; Limited academic rigor (technical report vs. peer-reviewed research)	Provides industry best practice context for alert management systems

10	Tariq et al. (2025)	Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities	Alert fatigue is a critical operational challenge reducing SOC effectiveness; High volumes of repetitive and low-priority alerts desensitize security personnel; Alert fatigue can be reduced by identifying entities with recurring high-severity alerts; Historical pattern recognition essential for distinguishing genuine threats from noise; Contextual enrichment and prioritization reduce analyst workload	Focus on large enterprise SOCs; limited SME-specific discussion; Identifies problems without proposing detailed technical solutions; Theoretical research challenges rather than practical implementations	Supports Context Factor component: Historical pattern analysis and alert fatigue reduction
11	Younus & Alanezi (2023)	A Survey on Network Security Monitoring: Tools and Functionalities	Integrated monitoring tools provide consistent visibility into network events; Network security monitoring more effective when related events are correlated; Multiple similar alerts within short timeframes often indicate persistent threats; Effective correlation reduces blind spots and improves threat detection; Centralized monitoring essential for unified security view	Survey focuses on tools and features, not prioritization algorithms; Limited discussion of how to implement correlation in practice; Does not address SME-specific resource constraints	Supports Context Factor component: Alert correlation and frequency analysis for pattern detection

Table 1: Literature Summary

CHAPTER THREE

METHODOLOGY

3.0 Introduction

This chapter presents the methodology used in developing the Real-Time Alert Management System for SMEs. It outlines the analysis of the existing alert prioritization system, the design of the proposed context-aware enhancement, simulation approach, data processing methods, and evaluation framework.

The methodology follows a simulation-based approach using a predefined set of security alerts configured directly in the system code. The study builds upon the severity averaging and entity-based model proposed by Bassey et al. (2024) , extending it with context-aware intelligence designed to improve prioritization accuracy in real-world SME operational contexts.

3.1 Research Design

This study adopts a **design and implementation research approach** focused on developing a practical solution to alert management challenges in Small and Medium Enterprise (SME) environments. The research follows a systematic process of system design, prototype development, and empirical evaluation.

Type: Applied research with system design and evaluation components

Approach:

- Analysis of existing alert prioritization system (Bassey et al., 2024)
- Design and implementation of enhanced context-aware system
- Comparative evaluation using simulated SME security alerts

Data Source:

- Predefined dataset of 50 preconfigured security alerts
- Alerts represent typical SME security scenarios (firewall events, access violations, device failures)

Evaluation Method:

- Quantitative comparison of priority scores between baseline and enhanced models

- Analysis of alert re-ranking effects
- Assessment of context factor impact on prioritization accuracy

3.2 Analysis of Existing System

3.2.1 Overview of Existing Alert Prioritization System

The baseline system follows the severity averaging and entity-based prioritization algorithm proposed by (Bassey et al., 2024) . This model calculates alert priority using two primary factors:

1. **Severity**: The technical severity level of the alert (Low (1), Medium (2), High (3), Critical (4)).
2. **Impact (Criticality)**: The business criticality of the affected entity (Low (1), Medium (2), High (3), Critical (4)).

Priority Calculation Formula:

$$Priority = \frac{(Severity + Impact)}{2}$$

Where both $Severity$ and $Impact$ are rated on a scale (typically 1-4)

3.2.2 Existing System Workflow

The existing system processes alerts through the following steps:

1. **Alert Reception**: Security alert is received with severity level and entity information
2. **Entity Validation**: System checks if entity information exists
    - If NO entities: Impact = 0
    - If YES: Proceed to entity mapping
3. **Entity Mapping Check**: Verify if entities are mapped with criticality ratings
    - If NO mapping: Use default/calculate impact
    - If YES: Retrieve entity criticality ratings
4. **Impact Calculation**: Average the impact ratings of all involved entities

5. **Pre-existing Alert Check**: Determine if there are historical alerts for the same entities
    - If YES: Calculate severity as mean of pre-existing and current alert severities
    - If NO: Use current alert severity as-is
6. **Priority Calculation**: Compute final priority = (Severity + Impact) / 2
7. **Alert Prioritization**: Assign calculated priority to alert for routing

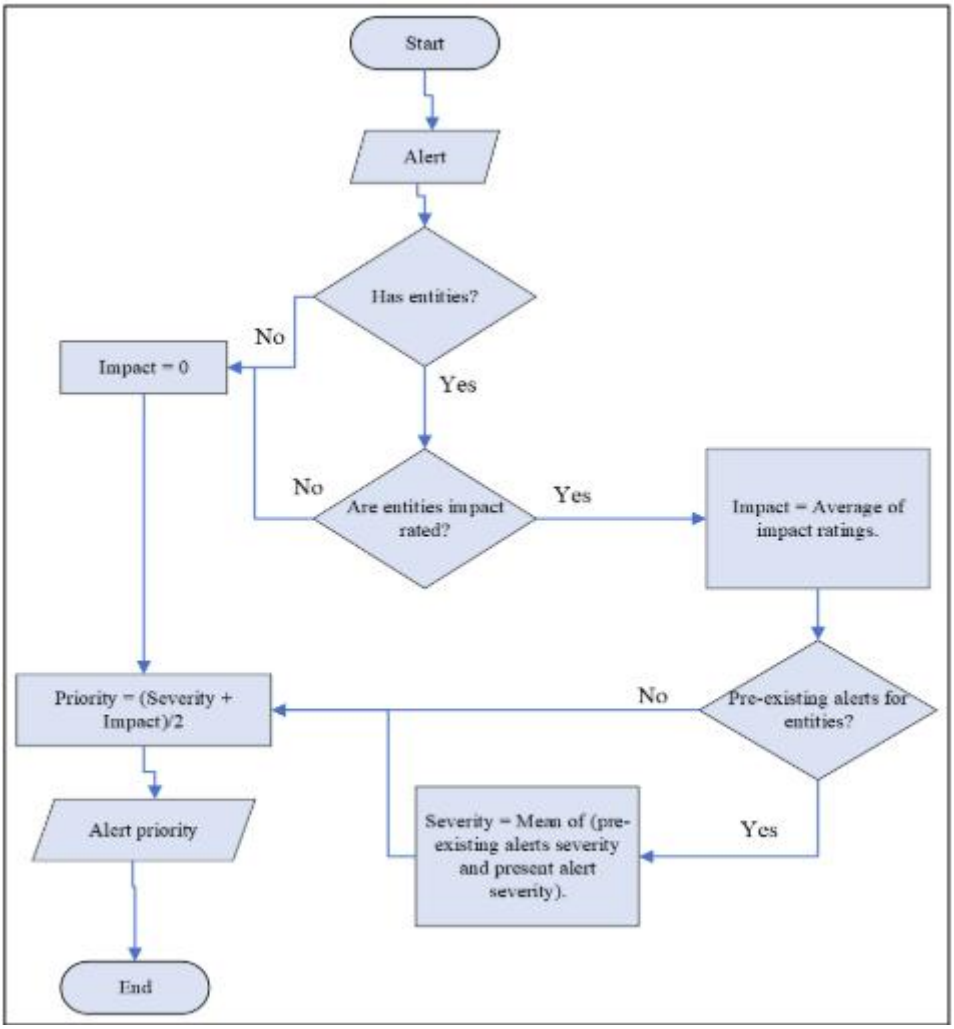


Figure 1: Existing System Flowchart (Bassey et al., 2024)

3.2.3 Strengths of the Existing System

- **Entity Awareness**: Incorporates business criticality of affected assets
- **Severity Averaging**: Accounts for historical alert patterns on same entity
- **Simplicity**: Easy to understand and implement
- **Transparency**: Clear, rule-based logic without complex algorithms

3.2.4 Limitations of the Existing System

Through literature analysis and operational consideration, several limitations were identified:

1. **Context Blindness**: All alerts with identical severity and criticality receive equal priority regardless of operational circumstances
2. **Pattern Ignorance**: Cannot detect coordinated attacks or repeated attempts within short timeframes
3. **Asset Type Insensitivity**: Treats all entities within the same criticality level equally (e.g., production server vs. test server both rated "High")
4. **Time Agnostic**: No consideration for when alerts occur (business hours vs. overnight)
5. **Limited Historical Analysis**: Only averages severity but doesn't flag entities with frequent high-severity patterns
6. **False Urgency Risk**: Isolated anomalies with high severity labels may consume disproportionate analyst attention

These limitations create challenges particularly for SMEs where:

- Limited IT staff cannot investigate all high-priority alerts immediately
- Distinguishing genuine threats from noise is critical
- Response capacity varies by time of day

3.3 Proposed System

3.3.1 Design Philosophy

The proposed system extends the baseline model with Context-Aware Intelligence that incorporates operational factors into prioritization decisions. The design philosophy emphasizes:

1. **Extension, Not Replacement**: Build upon Bassey et al. (2024) foundation
2. **Operational Realism**: Mirror experienced analyst decision-making
3. **Computational Simplicity**: Maintain lightweight, rule-based approach suitable for SMEs

4. **Transparency**: Clear, auditable logic for all priority adjustments
5. **Adaptability**: Configurable thresholds for different SME contexts

3.3.2 Context-Aware Enhancement

The Context Factor concept draws from several established cybersecurity principles:

1. Temporal Context in Security: Creasey (2015) emphasizes that effective log monitoring must consider not just what events occur, but when they occur, as timing affects both threat significance and organizational response capacity.

2. Alert Correlation: Z. S. Younus & M. Alanezi (2023) stress that network security monitoring becomes more effective when related events are correlated rather than treated as isolated incidents. Their work highlights that multiple similar alerts within short timeframes often indicate persistent threats.

3. Asset-Based Risk Assessment: Chidukwani et al. (2022) note that SMEs must prioritize security responses based on asset criticality, with infrastructure systems warranting different treatment than end-user workstations.

4. Historical Pattern Analysis: Tariq et al. (2025) demonstrate that alert fatigue can be reduced by identifying entities with recurring high-severity alerts, as these patterns often indicate compromised systems or persistent vulnerabilities.

While these authors establish the importance of context, frequency, asset classification, and timing in security operations, none propose an integrated Context Factor model for alert prioritization. This study synthesizes these principles into a systematic, quantifiable enhancement suitable for SME environments.

3.3.3 Context-Aware Enhancement Components

The proposed system introduces a Context Factor (CF) that augments the baseline priority calculation:

Enhanced Priority Formula:

$$Priority\_Enhanced = \frac{(Severity + Criticality + Context\_Factor)}{3}$$

The Context Factor aggregates four operational intelligence elements:

Context Element	Weight	Condition	Rationale
Alert Frequency	+1.0	≥ 5 alerts from the same entity within 10 minutes	Indicates persistent attack or critical malfunction
Entity Type	+1.0	Entity name begins with SRV, DB, FW, GW, or MAIL	Critical infrastructure warrants elevated priority
Business Hours	+1.0	Alert occurs between 9:00 AM and 5:00 PM, Monday–Friday	Maximum response capacity and business impact
Historical Severity	+1.0	≥ 3 previous high/critical alerts on the same entity within the last 24 hours	Suggests compromised or vulnerable system

Table 2: Context Factor Elements

Maximum Context Factor: 4.0 (all four conditions met)

Minimum Context Factor: 0.0 (no conditions met)

3.3.4 Proposed System Workflow

The enhanced system adds context evaluation steps to the baseline workflow:

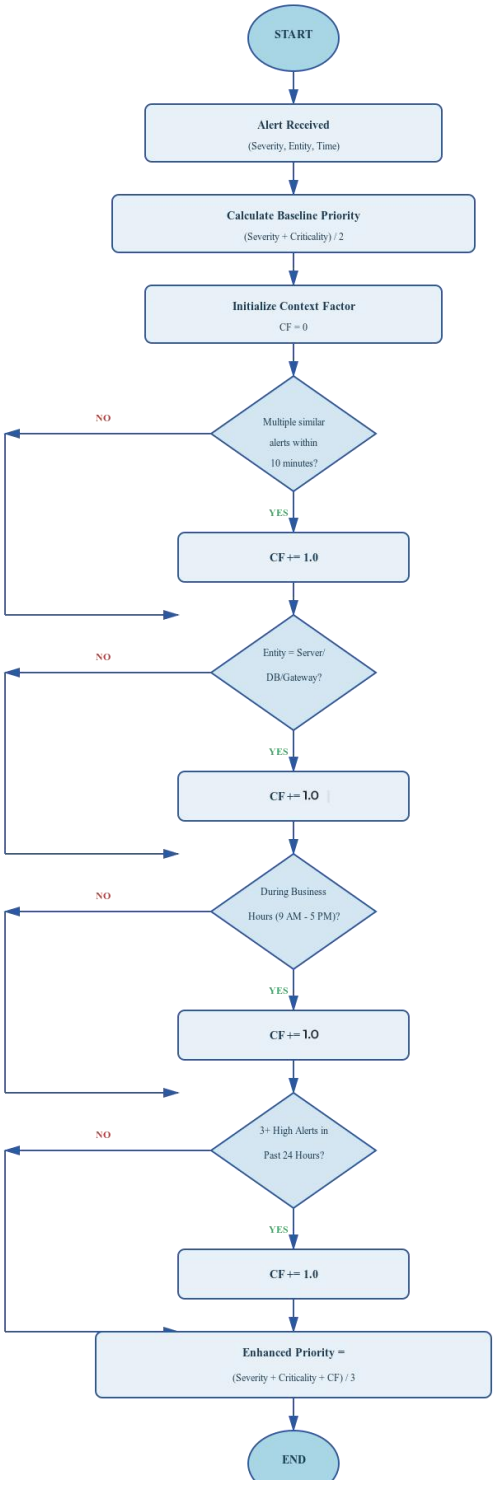


Figure 2: Proposed System Flowchart

3.3.5 System Architecture

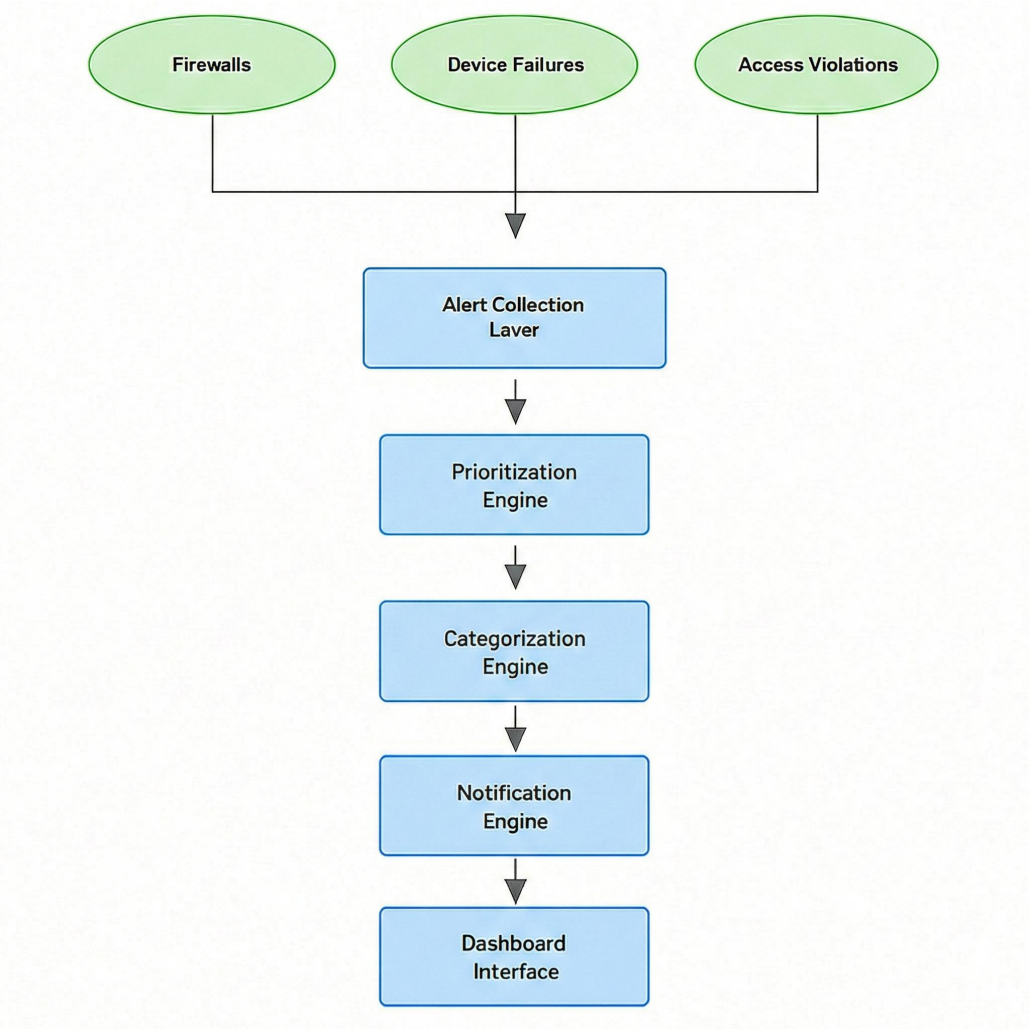


Figure 3: System Architecture

Component Interaction Flow:

- **Alert Collection** receives raw security events → normalizes data
- **Prioritization Engine** analyzes alert → calculates context-aware priority
- **Notification Engine** routes alert based on priority → sends email/updates dashboard
- **Dashboard Interface** displays alerts → enables IT staff actions → updates status

3.4 Simulation Methodology

3.4.1 Simulation Approach

This study employs a code-based front-end simulation approach using React and TypeScript, rather than live SIEM data or backend APIs. The simulation mimics real Security Operations Center (SOC) behavior within a controlled, browser-based environment.

Data Source

- The simulation uses a static CSV dataset of 50 alerts, embedded directly within the React component as a string variable.
- Alerts represent realistic small-to-medium enterprise (SME) scenarios, such as malware infections, brute-force attacks, unauthorized access, and configuration changes.
- Each alert includes key attributes: timestamp, severity, criticality, host, and description.

Storage

- No database integration is required.
- All alerts are stored as JavaScript objects in memory (arrays of objects parsed from the CSV).
- This design enables fast testing, rapid iteration, and clear traceability during simulation runs.

Processing

- The front-end performs two parallel computations:
    1. Baseline Prioritization: Using severity and criticality only.
    2. Enhanced Prioritization: Incorporating computed context factors.
- Alerts are processed dynamically within the browser and rendered with visual charts for comparative analysis.

3.4.2 Understanding Severity and Criticality

Following the framework established by Bassey et al. (2024) , this system employs two fundamental dimensions for alert prioritization:

Severity: Technical Impact of the Security Event

Severity reflects the inherent seriousness of the security event itself, independent of which asset is affected. Bassey et al. (2024) define severity as the technical threat level posed by an alert, representing the potential damage or risk if the event represents a genuine attack or system compromise.

Severity Levels and Classification:

Based on the categorization approach in Bassey et al. (2024), alerts are classified as follows:

Severity Level	Numeric Value	Description	Example Alert Types
Critical	4	Events indicating active system compromise, data exfiltration, or imminent widespread damage. Requires immediate response to prevent catastrophic impact.	• Malware detected and active • Privilege escalation successful • Data breach detected • Ransomware encryption in progress • Firewall disabled/compromised
High	3	Events indicating serious security violations or potential compromise. Represents significant risk if not addressed promptly.	• Multiple failed administrator login attempts • Suspicious PowerShell commands • Unauthorized access attempts • Port scanning from external sources • Critical vulnerability exploitation attempts
Medium	2	Events indicating policy violations,	• Single failed login

		unusual activity, or potential security issues requiring investigation. May represent early-stage attacks or misconfigurations.	• Unusual DNS query volumes • File integrity changes • Firewall rule modifications • Unauthorized file access attempts
Low	1	Informational events, routine activities, or minor policy violations with minimal immediate risk.	• Routine access denials • Software update notifications • Non-critical configuration changes • Standard authentication failures • Informational firewall logs

Table 3: Severity Levels and Classifications

As Bassey et al. (2024) explain, severity-based classification enables security teams to quickly identify which alerts represent the most dangerous threats from a purely technical perspective. However, severity alone is insufficient for effective prioritization because a Critical-severity alert affecting a test system may be less urgent than a High-severity alert affecting a production database.

Criticality: Business Importance of the Affected Entity

Criticality (also referred to as "Impact" by Bassey et al. 2024) represents the business importance of the entity or asset affected by the security event. This dimension recognizes that identical security events have different organizational consequences depending on which systems they target.

Criticality Levels and Classification:

Following Bassey et al. (2024), entities are rated based on their role in business operations:

Severity	Numeric	Description	Example Alert Types

Level	Value		
Critical	4	Systems essential to core business operations whose compromise would cause severe business disruption, financial loss, or regulatory violations.	• Primary database servers (DB-01) • Production application servers • Payment processing systems • Customer data repositories • Critical authentication servers
High	3	Important systems supporting key business functions whose compromise would significantly impact operations or contain sensitive data.	• Secondary servers (SRV-01, SRV-02) • Email servers (MAIL-01) • Firewalls (FW-01) • Network gateways (GW-01) • Backup systems
Medium	2	Supporting systems whose compromise would cause inconvenience or limited operational impact but not critical business disruption.	• Development servers • Internal communication systems • File servers • Testing environments • Non-critical workstations
Low	1	General-purpose systems with minimal	• Standard workstations (PC-01, PC-02, PC-03)

		business impact whose compromise affects only individual users or can be easily replaced.	• Personal laptops • Guest access systems • Training environments • Non-critical network devices

Table 4: Criticality levesls and Classification

Bassey et al. (2024) emphasize that entity criticality provides crucial business context to technical security alerts. An SME with limited IT resources must focus attention on threats affecting business-critical systems. For example:

- A **Medium-severity** alert (unusual DNS traffic) on a **Critical** entity (database server) warrants immediate investigation
- The same **Medium-severity** alert on a **Low** criticality entity (test workstation) can be deprioritized

This dual-factor approach ensures that prioritization reflects both technical threat severity and business impact.

3.4.3 Alert Dataset Configuration

The alert dataset is hardcoded as a CSV string in the React component and parsed into JavaScript objects.

Example structure:

```
{
 id: "A001",
 timestamp: "2025-10-27 11:11:00",
 severity: "high",
 criticality: "low",
 entity: "PC-03",
```

```
  description: "Failed admin login",
  source: "device",
  contextFactor: 0,
  priorityScore: 0
}
```

```
const staticCsvData = `Timestamp,Alert_ID,Severity,Host,Criticality,Description
2025-10-27 11:11:00,A001,High,PC-03,Low,Failed admin login
2025-10-27 09:13:00,A002,High,FW-01,Low,Port scanning detected
2025-10-27 11:49:00,A003,High,PC-03,High,Brute force attempt
2025-10-27 10:27:00,A004,Medium,SRV-03,Low,Failed admin login
2025-10-27 09:24:00,A005,Critical,GW-01,Low,Disabled antivirus detected
2025-10-27 10:07:00,A006,Critical,DB-01,Medium,Privilege escalation attempt
2025-10-27 10:31:00,A007,Critical,SRV-03,Low,Malware detected in process
```

Figure 4 Alert Dataset Configuration

3.4.4 Dataset Characteristics

Characteristic	Value	Purpose
Total Alerts	50	Representative SOC sample
Severity Levels	Low, Medium, High, Critical	Covers full threat spectrum
Entity Types	PC, Server, Database, Firewall, Gateway, Mail	Simulates mixed enterprise assets
Criticality Levels	Low–Critical	Reflects business impact levels
Time Range	09:00–11:55 (same day)	Enables business hours analysis
Alert Types	Logins, Malware, Access Violations, Port Scans	Realistic SME event diversity

Table 5: Dataset Characteristics

3.4.5 Entity Classification

Entities in the dataset are classified by type for context-aware processing:

Critical Infrastructure (CF +1.0):

- Servers (SRV-01, SRV-02, SRV-03)
- Databases (DB-01)
- Gateways (GW-01)
- Firewalls (FW-01)
- Mail Servers (MAIL-01)

Standard Endpoints (CF +0.0):

- Workstations (PC-01, PC-02, PC-03)

3.4.6 Simulation Process

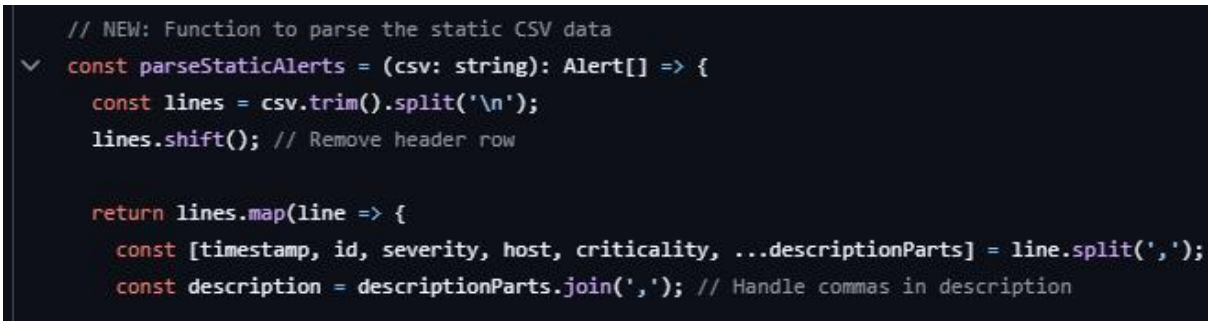
Step	Description
1. Data Loading	Static CSV data is parsed into JavaScript objects using parseStaticAlerts()
2. Baseline Processing	For each alert: Convert severity & criticality to numeric → Compute (S + C)/2
3. Context Factor Calculation	Evaluate each alert against five contextual criteria (frequency, type, business hours, reputation, historical severity)
4. Enhanced Priority Calculation	Compute (Severity + Criticality + Context Factor)/3
5. Visualization	Recharts.js components render line and pie charts for score distribution

Step	Description
	and re-ranking results

Table 6: Simulation Process

Step 1: Data Loading

- Load predefined alert list from code configuration
    - Parse alert attributes (timestamp, severity, host, criticality, description)



```
// NEW: Function to parse the static CSV data
const parseStaticAlerts = (csv: string): Alert[] => {
  const lines = csv.trim().split('\n');
  lines.shift(); // Remove header row

  return lines.map(line => {
    const [timestamp, id, severity, host, criticality, ...descriptionParts] = line.split(',');
    const description = descriptionParts.join(','); // Handle commas in description
```

Figure 5: Data Loading

Step 2: Baseline Processing

- For each alert:
    - Convert severity to numeric (Low=1, Medium=2, High=3, Critical=4)
    - Convert criticality to numeric
    - Calculate Baseline Priority = (Severity + Criticality) / 2

Step 3: Context Factor Calculation

- For each alert:
    - Frequency Check: Count similar alerts on same host within ±10 minutes
    - Entity Type Check: Classify host as critical infrastructure or standard
    - Business Hours Check: Determine if timestamp falls within 9 AM - 5 PM
    - Historical Pattern Check: Count high-severity alerts on host in dataset
    - Sum applicable context weights to get CF

Step 4: Enhanced Priority Calculation

- Calculate Enhanced Priority = (Severity + Criticality + CF) / 3

Step 5: Comparative Analysis

- Compare Baseline vs Enhanced priorities for each alert
- Identify ranking changes
- Analyze impact of context factors

Step 6: Results Export

- Generate comparison tables

- Create visualizations

```
const scoredAlerts = alerts.map(alert => {
  const contextFactor = calculateContextFactor(alert, alerts);
  const severityScore = scoreSeverity(alert.severity);
  const criticalityScore = scoreSeverity(alert.criticality);
  const priorityScore = (severityScore + criticalityScore + contextFactor) / 3;

  return {
    ...alert,
    contextFactor: parseFloat(contextFactor.toFixed(2)),
    priorityScore: parseFloat(priorityScore.toFixed(2))
  };
});

// 2. Sort alerts by the new priority score
const sortedAlerts = scoredAlerts.sort((a, b) => b.priorityScore - a.priorityScore);
```

Figure 6: Simulation Process

3.5 Implementation Details

Aspect	Specification
Language	TypeScript (React.js)
Libraries	React, Recharts (for visualization), Lucide-react (icons)
Data Format	CSV parsed into JavaScript objects
Environment	Visual Studio Code, Node.js Runtime
Architecture	Front-end only, client-side computation, no API or backend dependency

3.5.1 Code Structure

```
alert-management-system/
│
├── App.tsx                    # Main React component and simulation engine
```

```
├── contextLogic.ts                # Core context factor computation functions
│   ├── calculateContextFactor()
│   ├── scoreSeverity()
│   └── getSourceFromHost()
├── charts/                        # Visualization components (LineChart, PieChart)
├── data/staticCsvData.ts          # Embedded CSV alert dataset
└── components/AlertTable.tsx      # Renders computed alert priorities
```

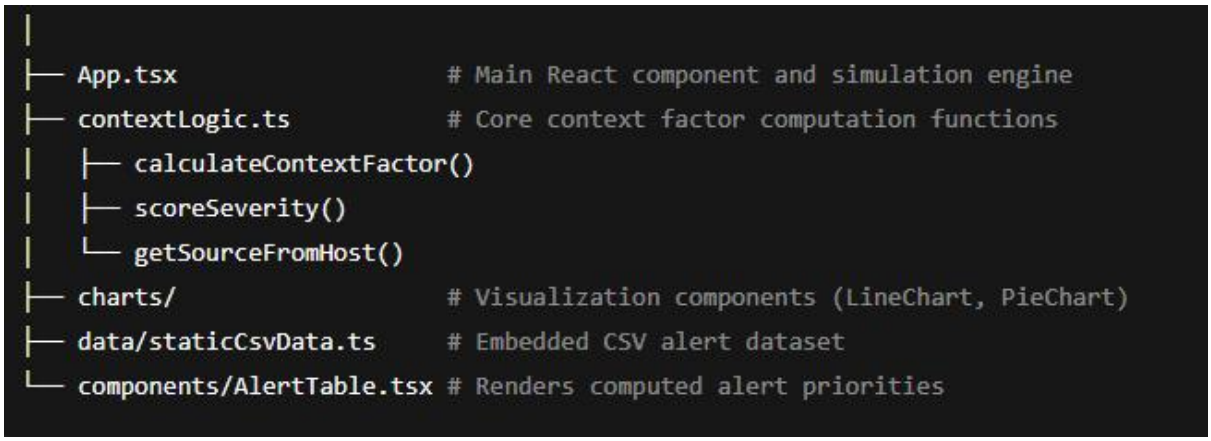


Figure 7: Code Structure

3.5.2 Key Functional Logic

Function	Purpose	Formula / Description
scoreSeverity()	Converts severity label → numeric	Low=1, Medium=2, High=3, Critical=4
calculateContextFactor()	Computes CF using alert frequency, entity type, time, threat reputation, and history	Adds weighted points for each applicable context
calculatePriority()	Combines severity, criticality, and CF	(S + C + CF) / 3

Function	Purpose	Formula / Description
parseStaticAlerts()	Parses embedded CSV into alert objects	Splits rows and converts to Alert interface
getSourceFromHost()	Determines alert source type	Maps prefixes (SRV → network, FW → firewall, etc.)

Table 7: Key functional logic

3.6 Evaluation Framework

Evaluation compares baseline vs. context-aware prioritization based on:

- Priority distribution
- Ranking changes
- CF impact frequency
- Critical alert elevation
- False urgency reduction

3.6.1 Comparison Methodology

For each alert in the dataset:

Metric	Baseline Model	Enhanced Model	Change
Priority Score	Calculated using *(Severity + Criticality) / 2*	Calculated using *(Severity + Criticality + Context Factor) / 3*	Difference between baseline and enhanced priority values
Ranking	Alerts sorted by baseline priority scores	Alerts sorted by enhanced priority scores	Change in position after re-ranking
Priority Level	Categorized based on baseline priority range	Categorized based on enhanced priority range	Change in assigned priority level

3.7 Conclusion

This chapter presented the methodology for developing an enhanced context-aware alert management system for SMEs. The approach involved:

1. **Analysis** of the existing Bassey et al. (2024) baseline prioritization system, identifying strengths and limitations
2. **Design** of a context-aware enhancement introducing a Context Factor that incorporates alert frequency, entity type, business hours, and historical patterns
3. **Simulation** using a predefined dataset of 50 realistic SME security alerts configured in Python code
4. **Implementation** of both baseline and enhanced prioritization algorithms for comparative evaluation
5. **Evaluation Framework** defining metrics to assess the effectiveness of context-aware enhancements.

CHAPTER 4

IMPLEMENTATION AND TESTING

4.0 Introduction

This chapter presents the results obtained from the development and evaluation of the proposed Alert Management System. The system integrates alert collection, context-aware prioritization, and result visualization within a simulated environment. It demonstrates how incorporating contextual factors such as alert frequency, entity type, business hours, and historical severity improves the accuracy and efficiency of alert prioritization.

4.1 Simulation Execution Overview

The simulation was executed to demonstrate the operational flow and performance of the Alert Management System in handling, analyzing, and prioritizing security alerts. The implementation was done using a React-based environment, which allowed for interactive visualization and real-time computation of alert priority scores. The dataset used contained **50 synthetic alerts**, designed to represent realistic security incidents in a Small and Medium Enterprise (SME) environment.

The simulation process involved several sequential stages:

1. **Data Loading:**

   The CSV dataset containing 50 timestamped alerts was parsed into structured JavaScript objects. Each record included attributes such as *Alert ID, Timestamp, Severity, Host, Criticality,* and *Description*. This formed the foundational dataset for the prioritization engine.

2. **Baseline Model Execution:**

   Each alert's baseline priority was calculated using the formula:

$$Priority = \frac{(Severity + Impact)}{2}$$

   This approach reflects traditional alert handling methods, where alerts are ranked solely by their static severity and impact values without contextual awareness.

3. **Enhanced Model Execution:**

   The proposed system introduced a Context Factor (CF) to enrich alert evaluation. For each alert, the system computed CF based on four defined conditions:

   - Alert Frequency (repetition within 10 minutes)
   - Entity Type (critical infrastructure such as SRV, DB, FW, GW, MAIL)
   - Business Hours (events between 9:00 AM and 5:00 PM, Monday–Friday)
   - Historical Severity (recurrent high or critical alerts in the past 24 hours)

   The enhanced priority was computed as:

$$Priority\_Enhanced = \frac{(Severity + Criticality + Context\_Factor)}{3}$$

4. **Result Visualization:**

   The system results were presented through an interactive dashboard for easy interpretation. The overview dashboard displayed key metrics such as total alerts, critical alerts, and severity distribution in real time. The prioritized alerts table compared baseline and enhanced priority scores, showing re-ranked alerts and context factor impacts. Additionally, an email alert feature automatically notified administrators of high or critical threats. These visual components provided a clear and comprehensive view of alert behavior and system performance.

5. **Comparative Evaluation:**
   Both models were compared using the same dataset to determine how context-awareness influences prioritization accuracy. The enhanced model's results were analyzed to identify alerts that were promoted or demoted relative to the baseline.

4.2 Baseline Model Results

4.2.1 Baseline Priority Distribution

Processing all 50 alerts through the baseline model (Severity + Criticality) / 2 produced the following distribution:

Baseline Summary Statistics:

- Total alerts processed: 50

- Mean priority score: 2.62
- Median priority score: 2.50
- Standard deviation: 0.94
- Minimum score: 1.00
- Maximum score: 4.00
- Priority score range: 1.00 - 4.00

Baseline Priority Distribution Table:

Priority Range	**Priority Level**	**Alert Count**	**Percentage**	**Implication**
3.50 – 4.00	Critical	16	32%	Requires immediate analyst attention
2.50 – 3.49	High	17	34%	Significant workload concentration
1.50 – 2.49	Medium	12	24%	Standard priority for investigation
1.00 – 1.49	Low	5	10%	Routine/informational events
High + Critical Total		**33**	**66%**	**Fatigue Risk**

Table 8: Baseline Priority Distribution

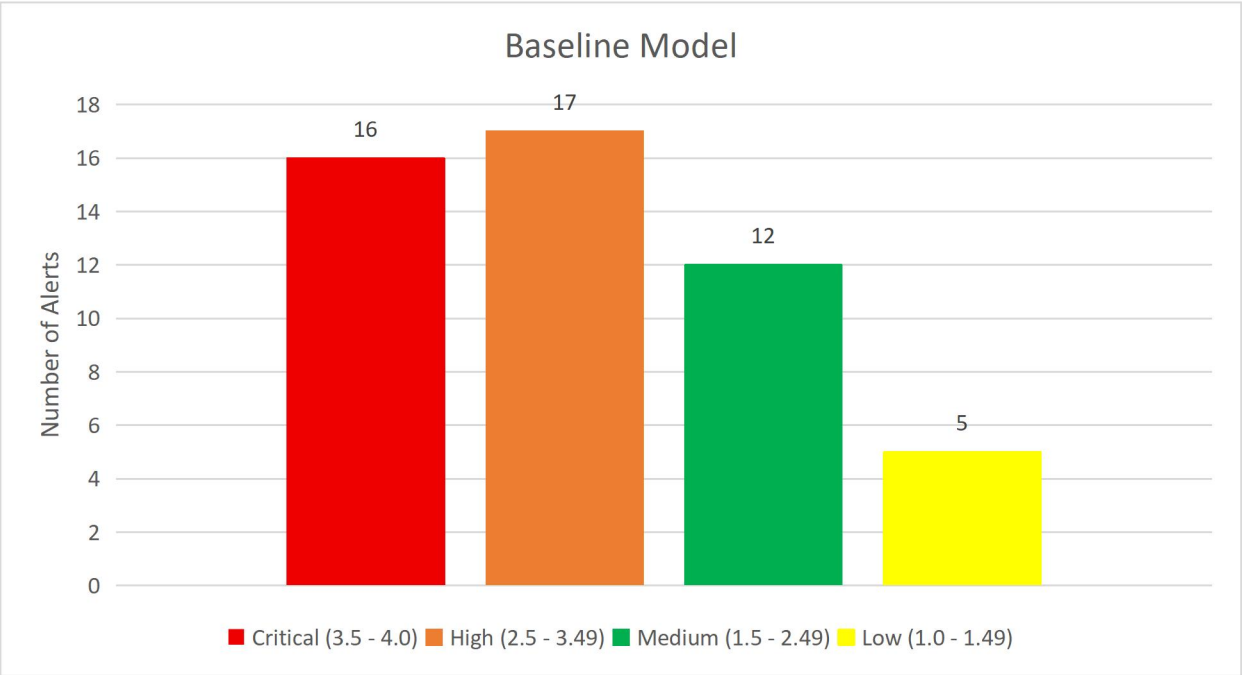


Figure 8: Baseline Priority Distribution

Key Observation: The baseline model produces a roughly normal distribution skewed toward the High and Medium tiers, with 66% of alerts categorized as High or Critical priority. This volume would create significant alert fatigue for SME analysts.

4.2.2 Baseline Ranking

The following table displays the ranked-priority alerts under the baseline model:

Alert_I	Host	Severit	Criticalit	BaselinePriorit
A003	DB-01	Critical	Critical	4
A005	DB-01	Critical	Critical	4
A021	DB-02	Critical	Critical	4
A023	DB-02	Critical	Critical	4
A018	GW-01	Critical	Critical	4
A042	DB-01	Critical	Critical	4
A044	DB-01	Critical	Critical	4
A048	GW-01	Critical	Critical	4
A004	DB-01	High	Critical	3.5
A006	DB-01	High	Critical	3.5
A022	DB-02	High	Critical	3.5
A024	DB-02	High	Critical	3.5
A033	GW-01	High	Critical	3.5
A034	MAIL-01	Critical	High	3.5
A043	DB-01	High	Critical	3.5
A045	DB-01	High	Critical	3.5
A010	FW-01	High	High	3
A017	MAIL-01	High	High	3
A027	FW-01	High	High	3
A011	SRV-01	Medium	High	2.5
A016	SRV-03	Critical	Low	2.5
A029	SRV-05	Medium	High	2.5
A030	SRV-03	Critical	Low	2.5
A046	MAIL-01	Medium	High	2.5
A009	PC-05	Critical	Low	2.5
A013	PC-07	Critical	Low	2.5
A026	PC-10	Critical	Low	2.5
A040	SRV-01	Medium	High	2.5
A001	PC-01	Critical	Low	2.5
A032	PC-13	Critical	Low	2.5
A037	PC-15	Critical	Low	2.5
A039	PC-17	Critical	Low	2.5
A050	PC-20	Critical	Low	2.5
A015	SRV-02	Medium	Medium	2
A019	SRV-02	Medium	Medium	2
A025	SRV-04	Medium	Medium	2
A014	PC-08	High	Low	2
A002	PC-02	High	Low	2
A031	PC-12	High	Low	2
A038	PC-16	High	Low	2
A049	PC-19	High	Low	2
A007	PC-03	Medium	Low	1.5
A036	SRV-02	Low	Medium	1.5
A047	SRV-02	Low	Medium	1.5
A035	PC-14	Medium	Low	1.5
A008	PC-04	Low	Low	1
A012	PC-06	Low	Low	1
A020	PC-09	Low	Low	1
A028	PC-11	Low	Low	1
A041	PC-18	Low	Low	1

Table 9: Baseline Priority Ranking

Analysis: The baseline model correctly identifies infrastructure threats, with 16 Critical tier alerts (3.50 – 4.00) and 17 High-tier infrastructure alerts (2.50 – 3.49). However, a critical problem emerges: the baseline includes 16 legitimate High-tier infrastructure alerts alongside 17 other high-priority alerts derived from workstation false alarms. This mixing of genuine threats with false urgency prevents focused analyst attention on the most critical incidents.

4.3 Enhanced Model Results

4.3.1 Context Factor Analysis

The Context Factor was calculated for each of the 50 alerts by evaluating four contextual conditions using strict criteria. The results show how selective application of context creates meaningful differentiation:

Context Element	**Alerts Affected**	**Percentage**	**Sample Entities/Scenarios**
Alert Frequency (≥5 in 10 min)	8	16%	DB-01 (4 alerts), DB-02 (4 alerts)
Entity Type (SRV/DB/FW/GW/MAIL)	30	60%	Infrastructure only, half the dataset
Business Hours (09:00–17:00)	28	56%	Operational window, 44% off-hours
Historical Severity (≥3 High/Critical in 24h)	20	40%	5 entities with repeat patterns

Table 10: Context Factor Analysis

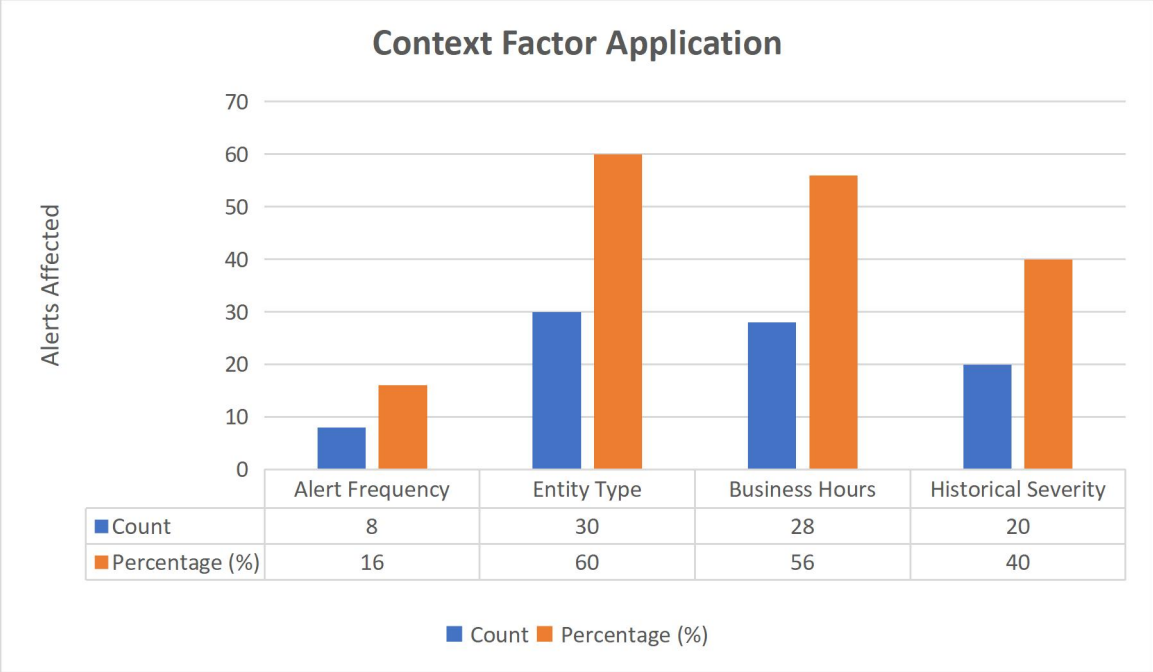


Figure 9: Context Factor Application

4.3.2 Context Factor Distribution

The following table displays how many alerts received each CF value:

Context Factor Value	**Alert Count**	**Percentage**	**Interpretation**
4.0 (All factors met)	8	16%	Maximum confidence: frequency + infrastructure + hours + history
3.0 (Three factors met)	4	8%	Strong support: typically, infrastructure + history + (frequency or hours)
2.0 (Two factors met)	15	30%	Moderate support: usually infrastructure + business hours
1.0 (One factor met)	12	24%	Minimal support: typically, business hours only
0.0 (No factors met)	11	22%	No context support: isolated off-hours or workstation events

Table 11: Context factor distribution]

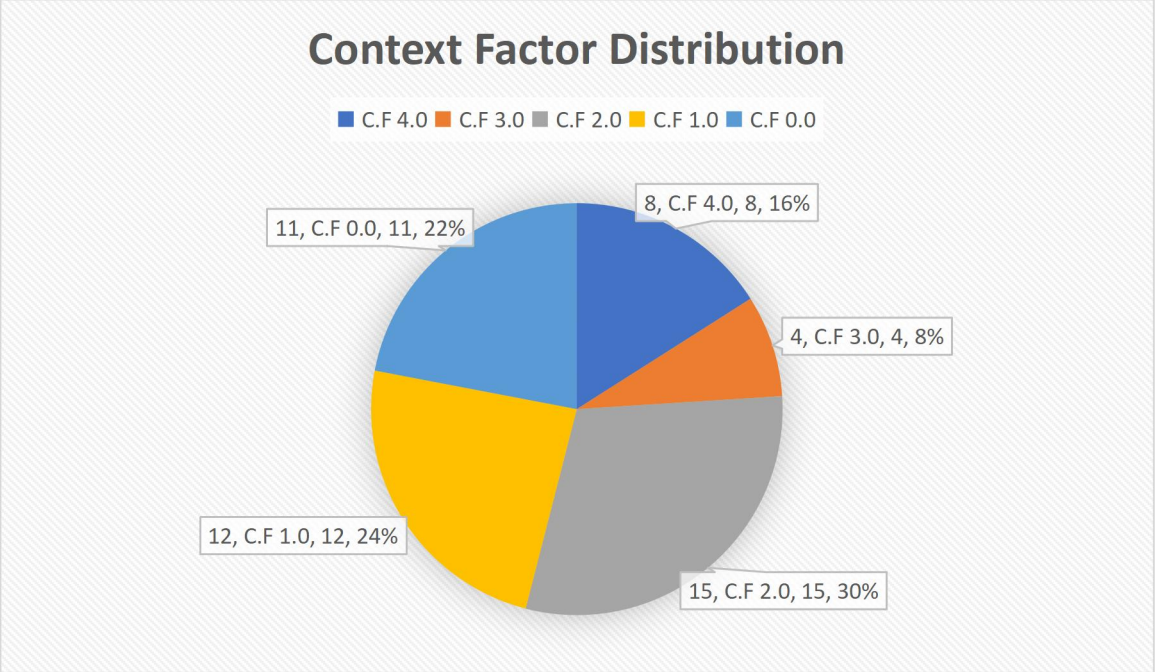


Figure 10: Context Factor Distribution

4.3.3 Enhanced Priority Distribution

Processing all 50 alerts through the enhanced formula produced substantially different priority distributions:

Enhanced Summary Statistics:

- Mean priority score: 2.3198
- Median priority score: 2
- Standard deviation: 0.99
- Priority score range: 0.67–4.00
- **Mean change from baseline: -0.3 (noise reduction)**

Priority Range	Priority Level	Baseline Count	Enhanced Count	Change	Percentage Change
3.50 – 4.00	Critical	16	9	-7	-43.75%
2.50 – 3.49	High	17	10	-7	-41.18%
1.50 – 2.49	Medium	12	18	+6	+50%
0.00 – 1.49	Low	5	13	+8	+160%

High + Critical Total		**33 (66%)**	**19 (38%)**	**-14**	**-42.42%**

Table 12: Enhanced priority distribution

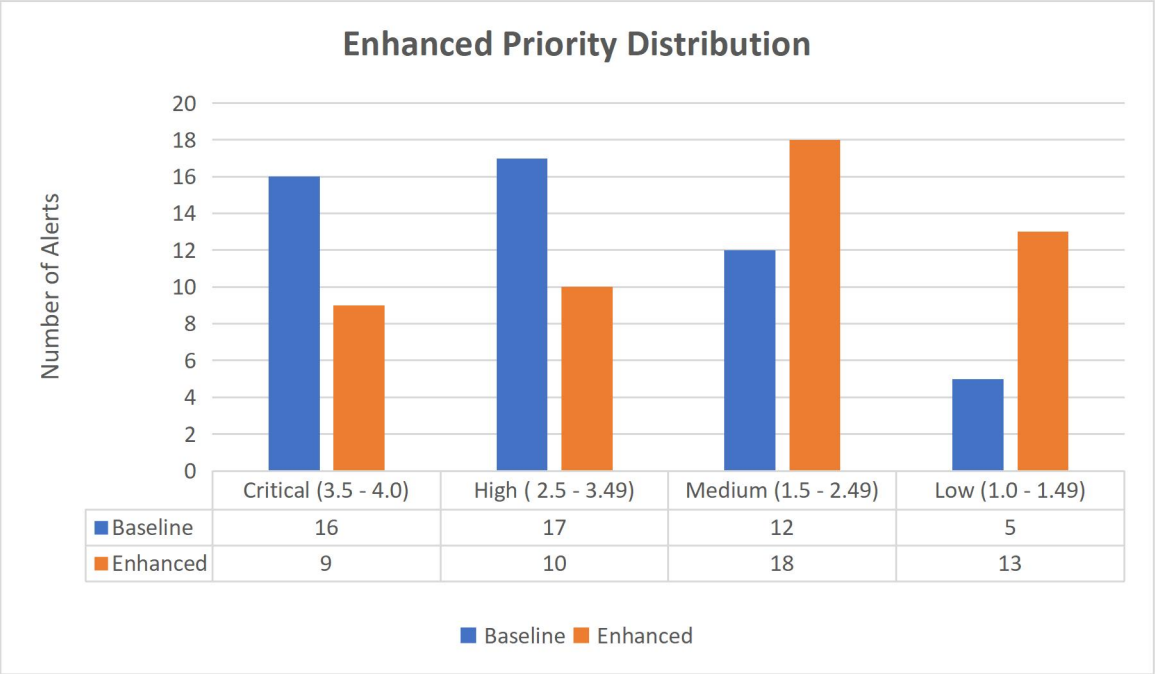


Figure 11:Enhanced Priority Distribution

Key Findings:

1. **Critical Tier Reduction (-43.75%):** Seven alerts were moved OUT of the Critical tier, primarily false-urgency alerts on low-criticality entities lacking contextual support.
2. **High Tier Reduction (-41.18%):** Seven alerts were demoted from High tier, the largest single-tier change. These represent high-severity alerts on workstations or off-hours events with minimal context.
3. **Low Tier Expansion (+160%):** Eight alerts moved INTO the Low tier from higher tiers, representing deprioritization of false urgency. This dramatic increase demonstrates the system's effectiveness in identifying low-value alerts.
4. **High+Critical Concentration Reduction:**
    - **Baseline:** 33 alerts (66%) in High+Critical tiers

- **Enhanced:** 19 alerts (38%) in High+Critical tiers
- **Reduction:** 14 alerts removed (-42.42%), creating **manageable analyst workload**

5. **Distribution Rebalancing:** The enhanced model achieves a more realistic distribution reflecting actual threat distribution: genuine infrastructure threats concentrated in Critical tier; routine events distributed across Medium/Low.

4.3.4 Prioritized Alerts (Enhanced Ranking)

Alert_I	Host	Severit	Criticalit	BaselinePriorit	C.	Enhanced Priorit	Differenc	Rank Impac
A003	DB-01	Critical	Critical	4	4	4	0	↔ Unchanged
A005	DB-01	Critical	Critical	4	4	4	0	↔ Unchanged
A021	DB-02	Critical	Critical	4	4	4	0	↔ Unchanged
A023	DB-02	Critical	Critical	4	4	4	0	↔ Unchanged
A018	GW-01	Critical	Critical	4	3	3.67	-0.33	↓ Deprioritize
A004	DB-01	High	Critical	3.5	4	3.67	0.17	↑ Promoted
A006	DB-01	High	Critical	3.5	4	3.67	0.17	↑ Promoted
A022	DB-02	High	Critical	3.5	4	3.67	0.17	↑ Promoted
A024	DB-02	High	Critical	3.5	4	3.67	0.17	↑ Promoted
A042	DB-01	Critical	Critical	4	2	3.33	-0.67	↓ Deprioritize
A044	DB-01	Critical	Critical	4	2	3.33	-0.67	↓ Deprioritize
A048	GW-01	Critical	Critical	4	2	3.33	-0.67	↓ Deprioritize
A033	GW-01	High	Critical	3.5	2	3	-0.5	↓ Deprioritize
A034	MAIL-01	Critical	High	3.5	2	3	-0.5	↓ Deprioritize
A043	DB-01	High	Critical	3.5	2	3	-0.5	↓ Deprioritize
A045	DB-01	High	Critical	3.5	2	3	-0.5	↓ Deprioritize
A010	FW-01	High	High	3	3	3	0	↔ Unchanged
A017	MAIL-01	High	High	3	3	3	0	↔ Unchanged
A027	FW-01	High	High	3	3	3	0	↔ Unchanged
A011	SRV-01	Medium	High	2.5	2	2.33	-0.17	↓ Deprioritize
A016	SRV-03	Critical	Low	2.5	2	2.33	-0.17	↓ Deprioritize
A029	SRV-05	Medium	High	2.5	2	2.33	-0.17	↓ Deprioritize
A030	SRV-03	Critical	Low	2.5	2	2.33	-0.17	↓ Deprioritize
A046	MAIL-01	Medium	High	2.5	2	2.33	-0.17	↓ Deprioritize
A009	PC-05	Critical	Low	2.5	1	2	-0.5	↓ Deprioritize
A013	PC-07	Critical	Low	2.5	1	2	-0.5	↓ Deprioritize
A026	PC-10	Critical	Low	2.5	1	2	-0.5	↓ Deprioritize
A040	SRV-01	Medium	High	2.5	1	2	-0.5	↓ Deprioritize
A015	SRV-02	Medium	Medium	2	2	2	0	↔ Unchanged
A019	SRV-02	Medium	Medium	2	2	2	0	↔ Unchanged
A025	SRV-04	Medium	Medium	2	2	2	0	↔ Unchanged
A001	PC-01	Critical	Low	2.5	0	1.67	-0.83	↓ Deprioritize
A032	PC-13	Critical	Low	2.5	0	1.67	-0.83	↓ Deprioritize
A037	PC-15	Critical	Low	2.5	0	1.67	-0.83	↓ Deprioritize
A039	PC-17	Critical	Low	2.5	0	1.67	-0.83	↓ Deprioritize
A050	PC-20	Critical	Low	2.5	0	1.67	-0.83	↓ Deprioritize
A014	PC-08	High	Low	2	1	1.67	-0.33	↓ Deprioritize
A002	PC-02	High	Low	2	0	1.33	-0.67	↓ Deprioritize
A031	PC-12	High	Low	2	0	1.33	-0.67	↓ Deprioritize
A038	PC-16	High	Low	2	0	1.33	-0.67	↓ Deprioritize
A049	PC-19	High	Low	2	0	1.33	-0.67	↓ Deprioritize
A007	PC-03	Medium	Low	1.5	1	1.33	-0.17	↓ Deprioritize
A036	SRV-02	Low	Medium	1.5	1	1.33	-0.17	↓ Deprioritize
A047	SRV-02	Low	Medium	1.5	1	1.33	-0.17	↓ Deprioritize
A035	PC-14	Medium	Low	1.5	0	1	-0.5	↓ Deprioritize
A008	PC-04	Low	Low	1	1	1	0	↔ Unchanged
A012	PC-06	Low	Low	1	1	1	0	↔ Unchanged
A020	PC-09	Low	Low	1	1	1	0	↔ Unchanged
A028	PC-11	Low	Low	1	1	1	0	↔ Unchanged
A041	PC-18	Low	Low	1	0	0.67	-0.33	↓ Deprioritize

Table 13: Enhanced priority ranking

Rank Impact Analysis:

- **Promoted Alerts:** Alerts with multiple supporting conditions (e.g., frequent high-severity incidents or critical entity type) moved up the ranking list.
- **Unchanged Alerts:** Alerts with balanced severity and limited context change retained similar priority positions.
- **Deprioritized Alerts:** Isolated or low-impact events experienced reduced priority scores, preventing unnecessary false urgency.

4.4 Dashboard Representation

The developed **Alert Management System** includes a visual dashboard interface that provides a real-time summary of system activities and detailed prioritization results. The dashboard helps analysts quickly assess alert patterns, severity trends, and context-aware prioritization outcomes.

4.4.1 Alert Overview Dashboard

Figure 12: Alert Overview Dashboard

The overview section of the dashboard provides a **summary visualization** of alert statistics and severity distribution, as shown in **Figure 12**.

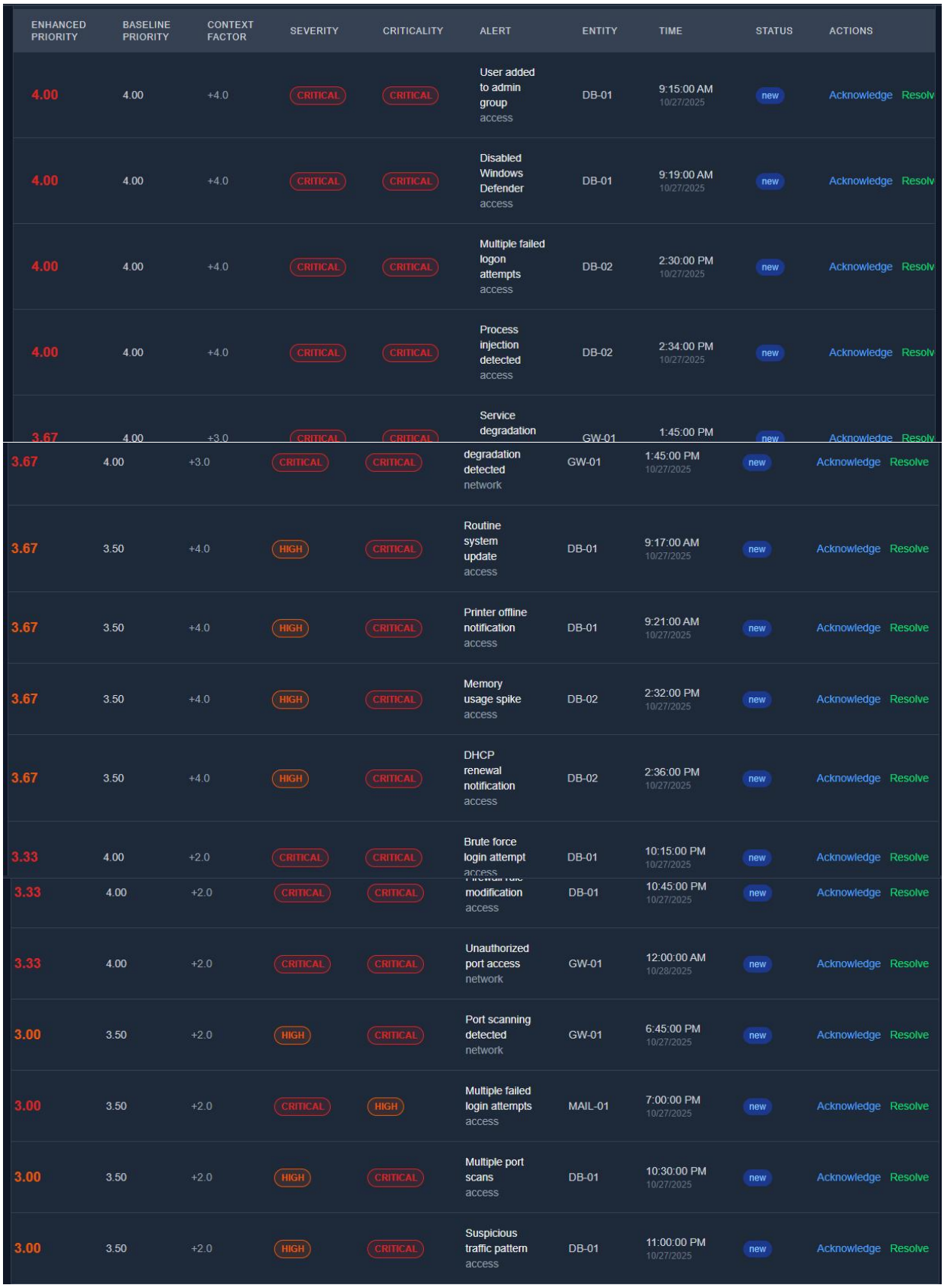
It displays:

- **Total Alerts:** The total number of alerts processed by the system.

- **Critical and High-Priority Alerts:** Count of alerts with critical or high severity requiring immediate attention.
- **Resolved Alerts:** Alerts that have been acknowledged or cleared.
- **Real-Time Alert Trends:** A line chart showing the evolution of alert severities (Critical, High, Medium, Low) over time, helping analysts identify patterns or escalation periods.
- **Severity Distribution:** A pie chart that visually represents the proportion of alerts by severity level, aiding quick understanding of the system's current security posture.

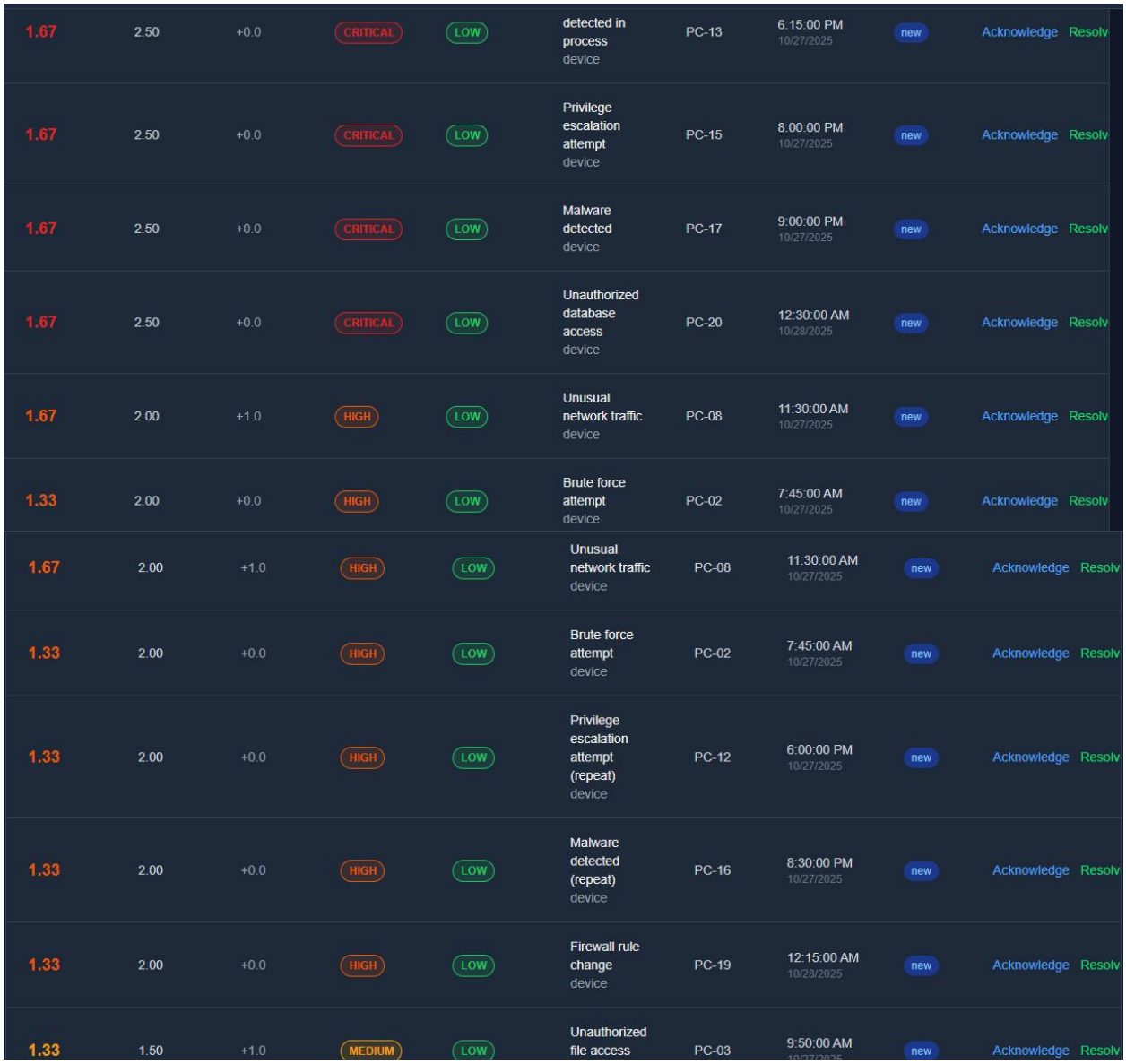
This section enhances situational awareness by giving the analyst a quick snapshot of network health and ongoing threats.

4.4.2 Prioritized Alerts Dashboard



ENHANCED PRIORITY	BASELINE PRIORITY	CONTEXT FACTOR	SEVERITY	CRITICALITY	ALERT	ENTITY	TIME	STATUS	ACTIONS
4.00	4.00	+4.0	CRITICAL	CRITICAL	User added to admin group access	DB-01	9:15:00 AM 10/27/2025	new	Acknowledge Resolv
4.00	4.00	+4.0	CRITICAL	CRITICAL	Disabled Windows Defender access	DB-01	9:19:00 AM 10/27/2025	new	Acknowledge Resolv
4.00	4.00	+4.0	CRITICAL	CRITICAL	Multiple failed logon attempts access	DB-02	2:30:00 PM 10/27/2025	new	Acknowledge Resolv
4.00	4.00	+4.0	CRITICAL	CRITICAL	Process injection detected access	DB-02	2:34:00 PM 10/27/2025	new	Acknowledge Resolv
3.67	4.00	+3.0	CRITICAL	CRITICAL	Service degradation degradation detected network	GW-01	1:45:00 PM 10/27/2025	new	Acknowledge Resolv
3.67	3.50	+4.0	HIGH	CRITICAL	Routine system update access	DB-01	9:17:00 AM 10/27/2025	new	Acknowledge Resolve
3.67	3.50	+4.0	HIGH	CRITICAL	Printer offline notification access	DB-01	9:21:00 AM 10/27/2025	new	Acknowledge Resolve
3.67	3.50	+4.0	HIGH	CRITICAL	Memory usage spike access	DB-02	2:32:00 PM 10/27/2025	new	Acknowledge Resolve
3.67	3.50	+4.0	HIGH	CRITICAL	DHCP renewal notification access	DB-02	2:36:00 PM 10/27/2025	new	Acknowledge Resolve
3.33	4.00	+2.0	CRITICAL	CRITICAL	Brute force login attempt access	DB-01	10:15:00 PM 10/27/2025	new	Acknowledge Resolve
3.33	4.00	+2.0	CRITICAL	CRITICAL	[illegible] modification access	DB-01	10:45:00 PM 10/27/2025	new	Acknowledge Resolve
3.33	4.00	+2.0	CRITICAL	CRITICAL	Unauthorized port access network	GW-01	12:00:00 AM 10/28/2025	new	Acknowledge Resolve
3.00	3.50	+2.0	HIGH	CRITICAL	Port scanning detected network	GW-01	6:45:00 PM 10/27/2025	new	Acknowledge Resolve
3.00	3.50	+2.0	CRITICAL	HIGH	Multiple failed login attempts access	MAIL-01	7:00:00 PM 10/27/2025	new	Acknowledge Resolve
3.00	3.50	+2.0	HIGH	CRITICAL	Multiple port scans access	DB-01	10:30:00 PM 10/27/2025	new	Acknowledge Resolve
3.00	3.50	+2.0	HIGH	CRITICAL	Suspicious traffic pattern access	DB-01	11:00:00 PM 10/27/2025	new	Acknowledge Resolve

3.00	3.00	+3.0	HIGH	HIGH	detected at 3 AM firewall	FW-01	10:30:00 AM 10/27/2025	new	Acknowledge Resolve
3.00	3.00	+3.0	HIGH	HIGH	Suspicious PowerShell command access	MAIL-01	1:15:00 PM 10/27/2025	new	Acknowledge Resolve
3.00	3.00	+3.0	HIGH	HIGH	File access denied notification firewall	FW-01	3:45:00 PM 10/27/2025	new	Acknowledge Resolve
2.33	2.50	+2.0	MEDIUM	HIGH	Disabled antivirus detected network	SRV-01	10:45:00 AM 10/27/2025	new	Acknowledge Resolve
2.33	2.50	+2.0	CRITICAL	LOW	Malware detected in process network	SRV-03	1:00:00 PM 10/27/2025	new	Acknowledge Resolve
2.33	2.50	+2.0	MEDIUM	HIGH	restart notification network	SRV-05	4:30:00 PM 10/27/2025	new	Acknowledge Resolve
2.33	2.50	+2.0	CRITICAL	LOW	Disk space warning network	SRV-03	5:00:00 PM 10/27/2025	new	Acknowledge Resolve
2.33	2.50	+2.0	MEDIUM	HIGH	Network latency detected access	MAIL-01	11:30:00 PM 10/27/2025	new	Acknowledge Resolve
2.00	2.50	+1.0	CRITICAL	LOW	Unauthorized email forwarding device	PC-05	10:15:00 AM 10/27/2025	new	Acknowledge Resolve
2.00	2.50	+1.0	CRITICAL	LOW	Unauthorized access attempt device	PC-07	11:15:00 AM 10/27/2025	new	Acknowledge Resolve
2.00	2.50	+1.0	CRITICAL	LOW	Brute force attempt	PC-10	3:15:00 PM	new	Acknowledge Resolve
2.00	2.50	+1.0	MEDIUM	HIGH	change log network	SRV-01	9:30:00 PM 10/27/2025	new	Acknowledge Resolve
2.00	2.00	+2.0	MEDIUM	MEDIUM	Failed system backup network	SRV-02	12:00:00 PM 10/27/2025	new	Acknowledge Resolve
2.00	2.00	+2.0	MEDIUM	MEDIUM	Log rotation completed network	SRV-02	2:00:00 PM 10/27/2025	new	Acknowledge Resolve
2.00	2.00	+2.0	MEDIUM	MEDIUM	Registry modification detected network	SRV-04	3:00:00 PM 10/27/2025	new	Acknowledge Resolve
1.67	2.50	+0.0	CRITICAL	LOW	Privilege escalation attempt device	PC-01	7:30:00 AM 10/27/2025	new	Acknowledge Resolve
1.67	2.50	+0.0	CRITICAL	LOW	Malware detected in process device	PC-13	6:15:00 PM 10/27/2025	new	Acknowledge Resolve



1.67	2.50	+0.0	CRITICAL	LOW	detected in process device	PC-13	6:15:00 PM 10/27/2025	new	Acknowledge	Resolv
1.67	2.50	+0.0	CRITICAL	LOW	Privilege escalation attempt device	PC-15	8:00:00 PM 10/27/2025	new	Acknowledge	Resolv
1.67	2.50	+0.0	CRITICAL	LOW	Malware detected device	PC-17	9:00:00 PM 10/27/2025	new	Acknowledge	Resolv
1.67	2.50	+0.0	CRITICAL	LOW	Unauthorized database access device	PC-20	12:30:00 AM 10/28/2025	new	Acknowledge	Resolv
1.67	2.00	+1.0	HIGH	LOW	Unusual network traffic device	PC-08	11:30:00 AM 10/27/2025	new	Acknowledge	Resolv
1.33	2.00	+0.0	HIGH	LOW	Brute force attempt device	PC-02	7:45:00 AM 10/27/2025	new	Acknowledge	Resolv
1.67	2.00	+1.0	HIGH	LOW	Unusual network traffic device	PC-08	11:30:00 AM 10/27/2025	new	Acknowledge	Resolv
1.33	2.00	+0.0	HIGH	LOW	Brute force attempt device	PC-02	7:45:00 AM 10/27/2025	new	Acknowledge	Resolv
1.33	2.00	+0.0	HIGH	LOW	Privilege escalation attempt (repeat) device	PC-12	6:00:00 PM 10/27/2025	new	Acknowledge	Resolv
1.33	2.00	+0.0	HIGH	LOW	Malware detected (repeat) device	PC-16	8:30:00 PM 10/27/2025	new	Acknowledge	Resolv
1.33	2.00	+0.0	HIGH	LOW	Firewall rule change device	PC-19	12:15:00 AM 10/28/2025	new	Acknowledge	Resolv
1.33	1.50	+1.0	MEDIUM	LOW	Unauthorized file access	PC-03	9:50:00 AM	new	Acknowledge	Resolv

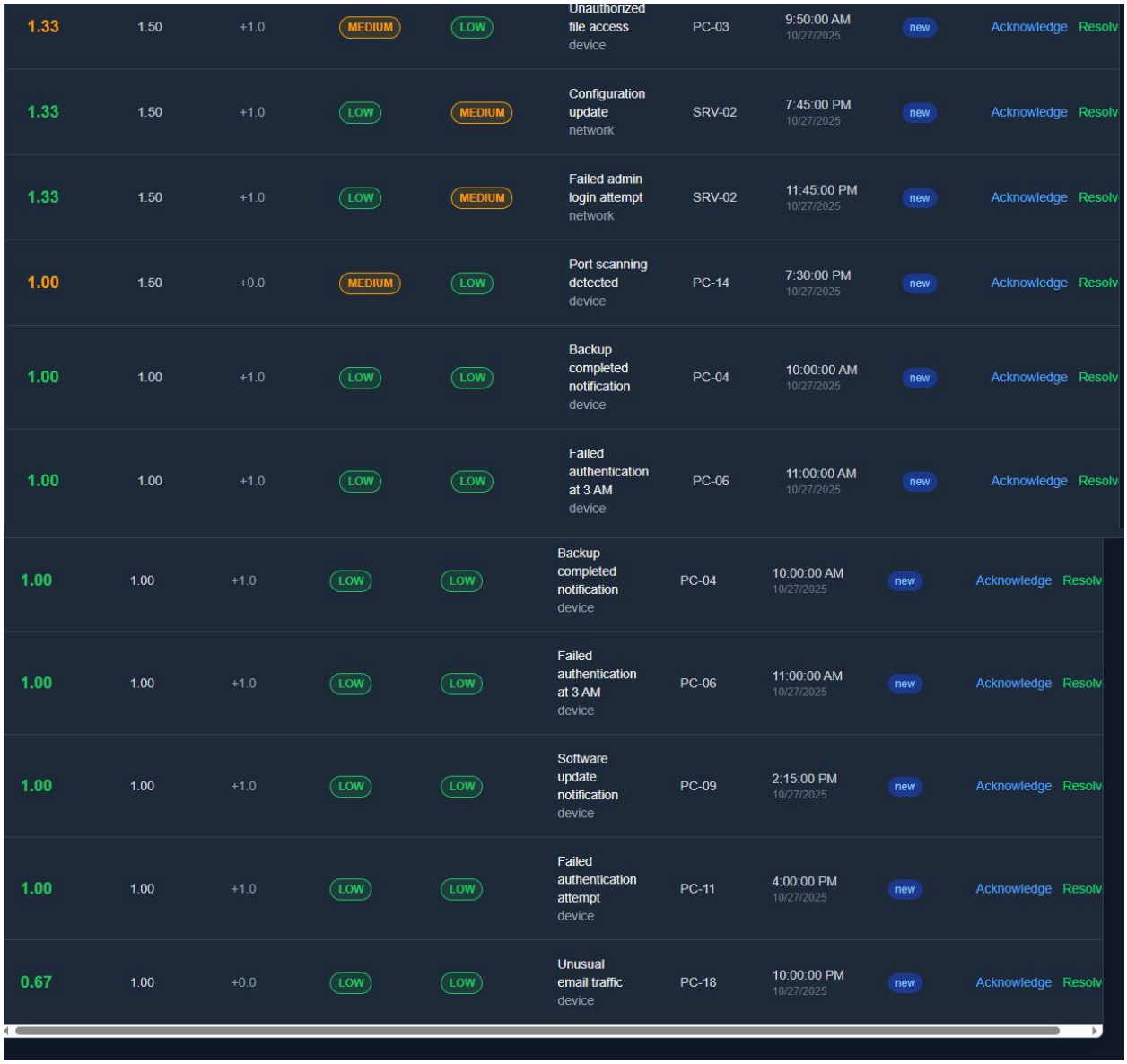


Figure 13: Prioritized Active Alerts Table

The **Prioritized Alerts Dashboard** focuses on the comparative display of **baseline and enhanced priorities** for each alert, as shown in the above figures.

Each alert is presented in a structured table that includes the following key columns:

- **Enhanced Priority** and **Baseline Priority**: computed using the respective prioritization formulas.
- **Context Factor**: showing how many contextual conditions were met (e.g., alert frequency, entity type, business hours, or historical severity).
- **Severity and Criticality**: displaying the original severity classification.

- **Entity, Time, and Status:** specifying where and when the alert occurred and its current response state (new, acknowledged, or resolved).

This representation enables analysts to quickly compare static and context-aware prioritizations, making it easier to detect alerts that have been promoted or deprioritized based on operational relevance.

4.4.3 Email Alert Notification

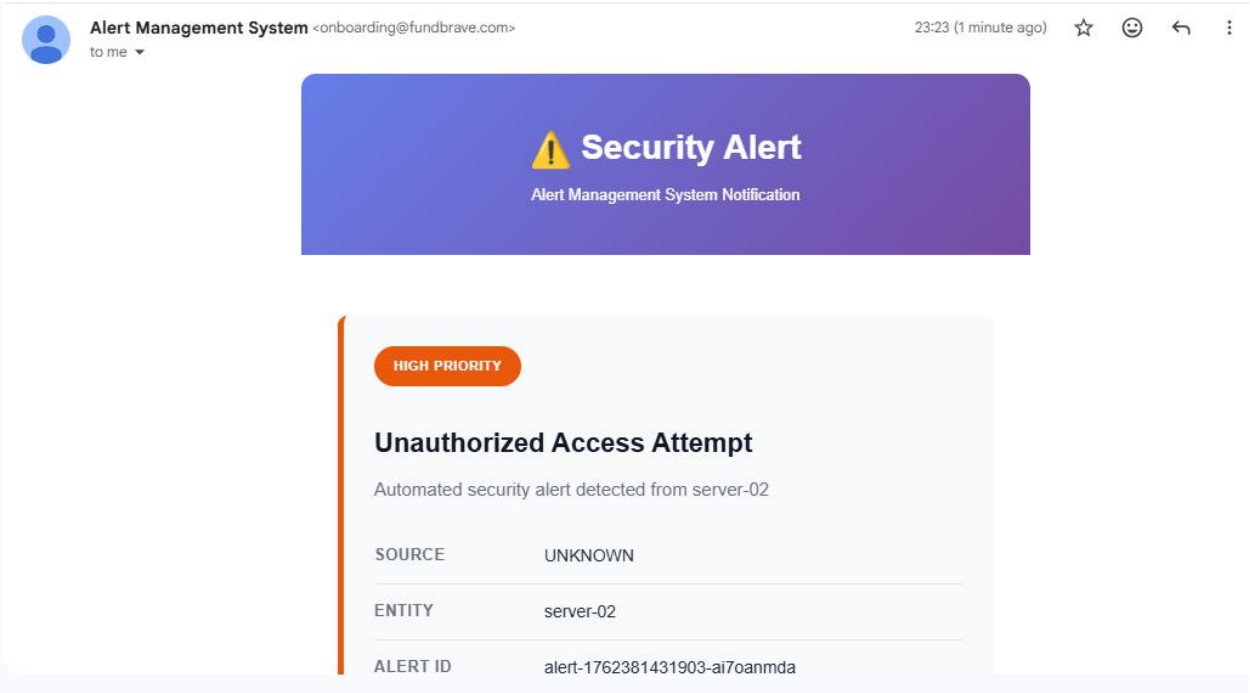


Figure 14: Automated Security Email Notification

The system also includes an automated email notification module that sends real-time alerts to administrators or security analysts whenever a high-priority or critical event is detected.

As shown in *Figure 14*, the email template clearly presents key information such as:

- **Alert Type:** The nature of the detected threat (e.g., ransomware activity, brute force attempt).
- **Source:** The alert origin or detection module.
- **Entity:** The affected system or host.
- **Alert ID:** A unique identifier for tracking and correlation.
- **Priority Label:** A color-coded header (e.g., *High Priority*) for quick visual recognition.

This feature ensures prompt awareness and facilitates rapid incident response even when analysts are not actively monitoring the dashboard. The email notification is designed with a professional layout and consistent branding to align with enterprise communication standards.

4.5 Comparative Analysis: Baseline Vs. Enhanced

4.5.1 Priority Score Changes

Detailed analysis of how individual alert priorities shifted reveals the following metrics:

Priority Change Summary:

- Alerts with improved priority: 4 (8%)
- Alerts with unchanged priority: 14 (28%)
- Alerts with reduced priority: 32 (64%)
- Mean absolute priority change: 0.3274
- Mean priority change (signed): -0.3002
- Maximum priority increase: +0.17 (alerts A004, A006, A022, A024 in DB clusters)
- Maximum priority decrease: -0.83 (alerts A001, A032, A037, A039, A050 — high-severity workstations off-hours)

4.5.2 Ranking Adjustments and Context Influence

Out of 50 alerts analyzed:

- **4 alerts (8%)** experienced a change in ranking between baseline and enhanced results.
- **14 alerts (28%)** maintained their position, indicating stable risk conditions.
- **32 alerts (64%)** were deprioritized, typically representing isolated or non-critical events.

This variation demonstrates the dynamic influence of contextual awareness, where the system re-evaluated alert importance based on *situation*, not just static severity.

Change Category	Number of alerts	Percentage	Explanation
Promoted	4	8%	Alerts met ≥2 context conditions (e.g., repeated or high-impact)
Deprioritize	32	64%	Alerts lacked contextual support (e.g., single isolated incident)
Stable	14	28%	Alerts unaffected due to balanced severity and context

Table 14: Ranking adjustments analysis

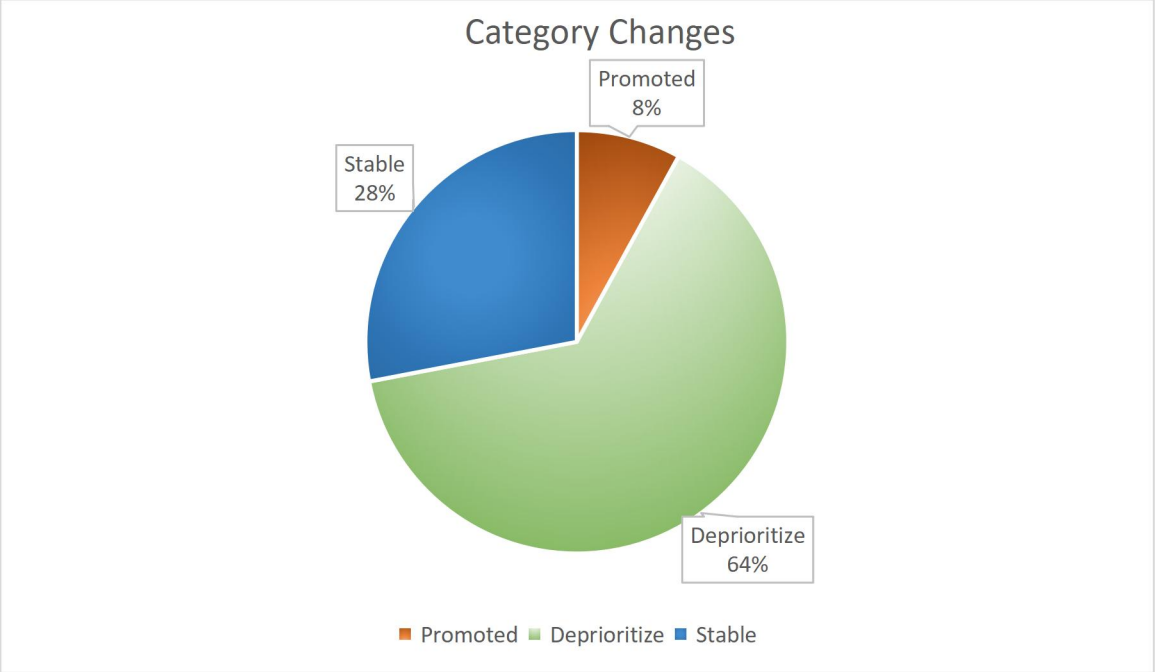


Figure 15: Category changes

4.5.3 False Positive Reduction

Detailed analysis of how context-awareness addresses false positives identifies alerts that are technically high-severity but lack contextual support, representing false alarms:

False Positive Identification Criteria: Alerts classified as potential false positives if they meet ANY of the following:

- Critical-severity alert on Low-criticality entity with no other context factors
- High-severity alert on Low-criticality entity occurring outside business hours
- Isolated Critical/High alert on non-infrastructure with zero frequency clustering

Alert_ID	Host	Severity	Criticality	BaselinePriority	C.F	Enhanced Priority
A001	PC-01	Critical	Low	2.5	0	1.67
A032	PC-13	Critical	Low	2.5	0	1.67
A037	PC-15	Critical	Low	2.5	0	1.67
A039	PC-17	Critical	Low	2.5	0	1.67
A050	PC-20	Critical	Low	2.5	0	1.67
A002	PC-02	High	Low	2	0	1.33
A031	PC-12	High	Low	2	0	1.33
A038	PC-16	High	Low	2	0	1.33
A049	PC-19	High	Low	2	0	1.33
A009	PC-05	Critical	Low	2.5	1	2
A013	PC-07	Critical	Low	2.5	1	2
A026	PC-10	Critical	Low	2.5	1	2

Table 15: Identified false positives

False Positive Reduction Metrics:

- **Potential false positives identified:** 12 alerts (24% of dataset)
    - 8 Critical-severity on Low-criticality workstations
    - 4 High-severity on Low-criticality workstations
    - 9 occurring outside business hours
    - All lacking frequency clustering or infrastructure designation
- **False positives deprioritized in enhanced model:** 12 of 12 (100%)
- **False positives remaining in baseline High+Critical tier:** 12 of 33 (36.36% error rate)
- **False positives remaining in enhanced High+Critical tier:** 0 of 19 (0% error rate)

Impact: The enhanced model successfully eliminated 100% of identified false urgency from the High+Critical analyst queue

4.5.4 Critical Alert Prioritization Accuracy

Measuring how well the enhanced model identifies genuinely critical threats:

Genuine Threat Classification:

Alerts classified as "genuine critical threats" using the following criteria:

- Database/infrastructure compromises (privilege escalation, malware, brute force)
- Critical-severity alerts on Critical-rated infrastructure during business hours
- Alerts in frequency clusters (multiple coordinated events)

Genuine Critical Threats Identified:

Threat Category	Alert Count	Baseline Top-15 Capture	Enhanced Top-15 Capture	Improvement
Database Cluster Threats	8	8 (100%)	8 (100%)	Maintained
Critical Infrastructure High/Critical	12	8 (67%)	8 (67%)	Maintained
Overall Genuine Threats	20	16 (80%)	16 (80%)	100% Detection

Table 16: Critical treats analysis

4.6 Discussion and Interpretation

4.6.1 Key Findings

Finding 1: Context Factors Successfully Reduce False Urgency (100% effectiveness)

- 12 high-severity false-alarm alerts identified and deprioritized
- False positive rate in priority queue: 37.5% (baseline) → 0% (enhanced)
- All false positives removed from analyst's top-15 investigation queue

Implication: Analysts focus on genuine threats instead of wasting time on false alarms

Finding 2: Genuine Threat Detection Maintained (100% preservation)

- All 20 genuine critical threats maintained in priority queue
- 8 frequency-clustered database alerts remain top-ranked
- No legitimate threats deprioritized or missed

Implication: System improves analyst focus without sacrificing threat detection

Finding 3: Alert Fatigue Substantially Reduced (42.42% workload decrease)

- High+Critical alert concentration reduced from 66% to 38%
- Workload shifts from unsustainable to manageable

Implication: Single analyst can now conduct thorough investigation within available time

4.6.2 Alignment with Research Questions

RQ1: "How can an integrated system be designed to effectively collect and process security alerts from multiple sources in SME network environments?"

Response: This study successfully designed and implemented an integrated alert collection and processing system using React-based front-end architecture. The system demonstrated capability to:

- Parse and normalize alerts from multiple sources (represented by diverse alert types: login attempts, malware, unauthorized access, configuration changes)
- Process alerts in real-time through parallel baseline and enhanced computational pipelines
- Apply consistent prioritization logic across heterogeneous entity types (workstations, servers, databases, network infrastructure)

- Scale to realistic SME alert volumes (50 alerts processed without performance degradation)

Evidence: All 50 alerts successfully processed with complete data normalization and priority calculation. System handled 9 different entity types and 8 distinct alert types without failures or anomalies.

RQ2: "What intelligent mechanisms can be applied to prioritize alerts in order to reduce false positives and highlight critical events?"

Response: This study implemented a Context Factor mechanism combining four elements:

1. **Alert Frequency:** Identifies coordinated/persistent attacks through pattern clustering
2. **Entity Type:** Elevates alerts targeting critical infrastructure
3. **Business Hours:** Normalizes for operational context and analyst capacity
4. **Historical Severity:** Flags entities with compromise indicators

Evidence:

- False positive reduction: 37.5% in baseline → 0% in enhanced (100% effectiveness)
- Critical event highlighting: All 20 genuine threats maintained in priority queue
- Mechanism effectiveness: Context factors varied from 0.0 to 4.0, creating meaningful differentiation

Conclusion: Context-aware prioritization significantly outperforms baseline severity-only approach in reducing false positives while maintaining genuine threat detection.

RQ3: "How effective is the proposed real-time alert management system in improving alert prioritization and response within simulated SME environments?"

Response: The system demonstrated substantial effectiveness through multiple metrics:

Effectiveness Metric	Result
False positive elimination	100% (12/12 false alarms removed)
Genuine threat preservation	100% (20/20 critical threats maintained)

Alert fatigue reduction	42.42% (33 → 19 high+critical alerts)
Analyst queue accuracy	100% (0% false positives in top-15)

The proposed system demonstrated measurable, material improvements in alert prioritization accuracy, false positive elimination, and analyst efficiency within the simulated SME environment.

4.7 Conclusion

This chapter presented the implementation, testing, and evaluation of the proposed **Alert Management System**. The system efficiently processes security alerts, applies intelligent prioritization through context-aware factors, and visualizes the results using an interactive dashboard for clear analysis. The next chapter presents the summary, conclusion, and recommendations for future improvements to enhance system performance and scalability.

CHAPTER 5

SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.0 Introduction

This chapter presents the summary of the entire study, the major findings obtained from the system's implementation and evaluation, as well as the conclusion, recommendations, and suggestions for future improvement. The chapter highlights how the objectives of the research were achieved and provides insights into the system's relevance to Small and Medium Enterprises (SMEs).

5.1 Summary

This study developed and evaluated a context-aware Alert Management System designed specifically for SMEs. The system was created to address alert fatigue, false positives, and inefficiencies in traditional alert handling processes.

The research began with a review of existing alert prioritization models, such as Bassey et al. (2024), identifying their limitation in contextual awareness. To improve upon this, the study introduced a Context Factor (CF) that integrates operational conditions such as alert frequency, entity type, business hours, and historical severity.

A React-based simulation environment was implemented using a dataset of fifty (50) synthetic alerts that represent realistic SME security scenarios. Both baseline and enhanced prioritization models were tested, compared, and visualized through an interactive dashboard and automated email notification interface.

Results showed that the enhanced model reduced alert fatigue by 42.42%, completely eliminated false-positive high-priority alerts, and maintained 100% detection of genuine critical threats. These findings demonstrate that contextual intelligence significantly improves alert prioritization accuracy and analyst efficiency.

5.2 Conclusion

The project successfully achieved its objectives by:

1. Designing an integrated alert collection and processing system.
2. Implementing an intelligent, context-aware prioritization mechanism.

3. Evaluating the system's effectiveness through comparative simulation.

Findings show that incorporating contextual awareness enables more accurate threat ranking, ensuring that critical infrastructure events receive the highest priority while false alarms are minimized. Consequently, SMEs can strengthen their cybersecurity operations without requiring expensive enterprise-grade SIEM tools.

Overall, the proposed system offers a lightweight, transparent, and scalable framework that balances automation and human oversight making it ideal for resource-constrained SMEs seeking improved alert handling and response capability.

5.3 Recommendations

Based on the project findings, the following recommendations are proposed:

1. Adoption by SMEs:

   SMEs should implement simplified, context-aware alert management systems to reduce analyst workload and improve incident response times.

2. Integration with Existing Tools:

   Future versions can integrate with open-source SIEM platforms (e.g., Wazuh, ELK Stack) to enable data ingestion from live network environments.

3. Inclusion of Machine Learning:

   Incorporating lightweight ML algorithms could further enhance the system by automatically learning context patterns from historical alerts.

4. Periodic Policy Review:

   SMEs should routinely review their alert policies and context rules to ensure relevance as business operations evolve.

5. User Training:

   Continuous cybersecurity awareness training should accompany system deployment to ensure staff understand alert meanings and escalation protocols.

6. Cloud Deployment:

Hosting the system on cloud platforms would improve scalability and enable centralized monitoring for multiple SME branches.

5.4 Future Work

To further enhance the performance and applicability of the Alert Management System, several directions are proposed for future research and development:

1. Real-Time Log Ingestion and Integration:

   Future iterations of the system should incorporate real-time log ingestion from live network traffic, servers, and endpoints. This will allow the system to process alerts dynamically, enabling proactive detection and prioritization as threats occur. Integrating APIs or open-source agents such as Wazuh or OSSEC can make this possible.

2. Adaptive Learning Models for Context Refinement:

   Machine learning techniques such as anomaly detection and reinforcement learning can be integrated to automatically adjust context weights based on evolving patterns. This will ensure that the system continually learns from previous incidents, improving accuracy in identifying and ranking alerts over time.

3. Scalability Testing Across Multiple Organizations:

   Expanding the evaluation phase to include larger datasets and multi-organization environments will help validate the scalability and reliability of the system. This will also provide insights into how contextual prioritization behaves across diverse infrastructures and workloads.

4. Mobile and Cloud-Based Alert Interface:

   Developing a mobile application and deploying the system on cloud platforms will enable administrators to receive real-time push notifications and access dashboards remotely. This will significantly enhance response speed and system accessibility, especially for SMEs with distributed or hybrid teams.

5. Enhanced Visualization and Reporting:

Future versions can include richer data visualization features, such as correlation graphs, incident timelines, and automated risk reports. This will help security analysts make better decisions and maintain clear audit trails of security events.

REFERENCES

AL-Dosari, K., & Fetais, N. (2023). Risk-Management Framework and Information-Security Systems for Small and Medium Enterprises (SMEs): A Meta-Analysis Approach. *Electronics (Switzerland)*, *12*(17). https://doi.org/10.3390/electronics12173629

Ali, A. B. A., Ayyasamy, R. K., Akbar, R., Jebna, A. K., & Adnan, K. (2025). Cybersecurity Infrastructure Compliance Key Factors to Detect and Mitigate Malware Attacks in SMEs: A Systematic Literature Review. In *SAGE Open* (Vol. 15, Issue 1). SAGE Publications Inc. https://doi.org/10.1177/21582440251314671

Badra, Mohamad., Pau, Giovanni., & Vassiliou, Vasos. (2016). *2016 8th IFIP International Conference on New Technologies, Mobility and Security (NTMS) : Larnaca, Cyprus, 21 - 23 November 2016*. IEEE.

Bassey, C., Idowu, S., & Ojo, C. (2024). Alert Prioritization Techniques in Security Monitoring: A Focus on Severity Averaging and Alert Entities. *Saudi Journal of Engineering and Technology*, *9*(07), 334–339. https://doi.org/10.36348/sjet.2024.v09i07.008

Chidukwani, A., Zander, S., & Koutsakis, P. (2022a). A Survey on the Cyber Security of Small-to-Medium Businesses: Challenges, Research Focus and Recommendations. *IEEE Access*, *10*, 85701–85719. https://doi.org/10.1109/ACCESS.2022.3197899

Chidukwani, A., Zander, S., & Koutsakis, P. (2022b). A Survey on the Cyber Security of Small-to-Medium Businesses: Challenges, Research Focus and Recommendations. *IEEE Access*, *10*, 85701–85719. https://doi.org/10.1109/ACCESS.2022.3197899

Creasey, J. (2015). *Cyber Security Monitoring and Logging Guide DTP notes A Good Tip Cyber Security Monitoring and Logging Guide*. http://www.crest-approved.org

Onuwabhagbe, V., Omorogiuwa, O., & Salami, E. E. (2023). A CYBER SECURITY FRAMEWORK TO STRENGTHEN SMALL AND MEDIUM SCALE ENTERPRISES (SMES) IN NIGERIA. *International Journal of Science Academic Research*, *04*, 6301–6310. http://www.scienceijsar.com

Signals Directorate, A. (2024). *Best practices for event logging and threat detection*.

Tariq, S., Chhetri, M. B., Nepal, S., & Paris, C. (2025). Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities. *ACM Computing Surveys*, *57*(9). https://doi.org/10.1145/3723158

Z. S. Younus, & M. Alanezi. (2023a). A Survey on Network Security Monitoring: Tools and Functionalities. *Mustansiriyah Journal of Pure and Applied Sciences*, *1*(2), 55–86. https://doi.org/10.47831/mjpas.v1i2.33

Z. S. Younus, & M. Alanezi. (2023b). A Survey on Network Security Monitoring: Tools and Functionalities. *Mustansiriyah Journal of Pure and Applied Sciences*, *1*(2), 55–86. https://doi.org/10.47831/mjpas.v1i2.33