UNIVERSITY
INSTITUTE
OF LISBON

Automating Cyber Attack Response Procedures: An AI Approach for SOC Analysts

Diogo Filipe Pires Godinho

Master in Telecommunications and Computer Engineering

Supervisors:
PhD Carlos José Corredoura Serrão, Associate Professor,
Iscte – Instituto Universitário de Lisboa

PhD Ana Maria Carvalho de Almeida, Associate Professor with Agregation,
Iscte – Instituto Universitário de Lisboa

October, 2025

[This page is intentionally left blank.]

TECHNOLOGY
AND ARCHITECTURE

Department of Information Science and Technology

Automating Cyber Attack Response Procedures: An AI Approach for SOC Analysts

Diogo Filipe Pires Godinho

Master in Telecommunications and Computer Engineering

Supervisors:
PhD Carlos José Corredoura Serrão, Associate Professor,
Iscte – Instituto Universitário de Lisboa

PhD Ana Maria Carvalho de Almeida, Associate Professor with Aggregation,
Iscte – Instituto Universitário de Lisboa

October, 2025

[This page is intentionally left blank.]

To the pursuit of clarity in complexity

[This page is intentionally left blank.]

Acknowledgment

I would like to express my sincere gratitude to all those who supported me throughout the development of this thesis.

First, I would like to thank my family for their unconditional support and encouragement. Your constant belief in me has been my greatest source of motivation and strength. Without your support, this achievement would not have been possible.

To my friends and to my colleagues in Nokia thank you for the discussions, collaboration and for creating a positive environment during this research.

Finally, I would like to thank my supervisors for their continuous guidance, valuable feedback and support throughout this research.

[This page is intentionally left blank.]

Resumo

A crescente complexidade dos ataques cibernéticos e o volume elevado de alertas gerados pelos centros de operações de cibersegurança (SOCs) tornam cada vez mais difícil a priorização e resposta eficaz a incidentes de segurança. A ausência de diretrizes claras e de ferramentas capazes de adaptar procedimentos a diferentes contextos operacionais agrava o risco de respostas tardias ou incompletas.

Nesta dissertação avaliou-se a aplicação da Inteligência Artificial (IA) para apoiar e automatizar o processo de resposta a incidentes em centros de operações de cibersegurança, através da geração de *playbooks* estruturados e sensíveis ao contexto. Com base na *Design Science Research Methodology* (DSRM), foi desenvolvido o protótipo ASTRA — *AI SOC Tool for Response Automation* — concebido para apoiar analistas na triagem e resposta a alertas de segurança. O sistema combina técnicas de processamento de linguagem natural e recuperação de informação para gerar recomendações automáticas alinhadas com as fases de resposta a incidentes. O ASTRA foi avaliado em diferentes cenários e validado por especialistas, demonstrando utilidade prática, clareza nas recomendações e potencial para melhorar a eficiência das operações em SOCs.

Os resultados obtidos demonstram que o sistema cumpre o seu propósito ao apoiar os analistas na triagem e resposta a incidentes, reduzindo a carga cognitiva e promovendo respostas mais consistentes. Conclui-se que a aplicação de IA ao contexto dos SOCs é viável e contribui para uma nova geração de ferramentas de apoio à decisão, capazes de combinar automação com raciocínio contextual.

Palavras Chave: *inteligência artificial*, *cibersegurança*, *resposta a incidentes*, *SOCs*, *automação*, *RAG*

[This page is intentionally left blank.]

Abstract

The growing complexity of cyberattacks and the high volume of alerts generated by Security Operations Centers (SOCs) make it increasingly difficult to prioritise and respond effectively to security incidents. The absence of clear guidelines and tools capable of adapting response procedures to different operational contexts further increases the risk of delayed or incomplete actions.

This dissertation evaluates the application of Artificial Intelligence (AI) to support and automate the incident response process in SOCs through the generation of structured and context-aware playbooks. Based on the *Design Science Research Methodology* (DSRM), the prototype ASTRA — *AI SOC Tool for Response Automation* — was developed to assist analysts in triaging and responding to security alerts. The system combines natural language processing and information retrieval techniques to generate automated recommendations aligned with the main phases of incident response.

ASTRA was evaluated across different attack scenarios and validated by domain experts, demonstrating practical usefulness, clarity of recommendations, and potential to improve the efficiency of SOC operations. The results show that the system fulfils its intended purpose by supporting analysts in triage and response tasks, reducing cognitive load, and promoting more consistent outcomes. It is concluded that the application of AI in the context of SOCs is feasible and contributes to a new generation of decision-support tools that combine automation with contextual reasoning.

KEYWORDS: *artificial intelligence*, *cybersecurity*, *incident response*, *SOC*, *automation*, *RAG*

[This page is intentionally left blank.]

Contents

List of Figures

[This page is intentionally left blank.]

List of Tables

[This page is intentionally left blank.]

List of Acronyms

AI	Artificial Intelligence
ASTRA	AI SOC Tool for Response Automation
CIA	Confidentiality, Integrity and Availability
FAISS	Facebook AI Similarity Search
IOC	Indicator of Compromise
IT	Information Technology
JSON	JavaScript Object Notation
KPI	Key Performance Indicator
LLM	Large Language Model
ML	Machine Learning
MRR	Mean Reciprocal Rank
RAG	Retrieval-Augmented Generation
SIEM	Security Information and Event Management
SOC	Security Operations Center
SOCs	Security Operations Centers
SOAR	Security Orchestration, Automation, and Response
UI	User Interface

[This page is intentionally left blank.]

CHAPTER 1

Introduction

In this chapter, the essential components of this research are introduced, beginning with the motivation to address the growing challenges of enhancing incident response capabilities in Security Operations Centers (SOCs). The context surrounding this issue is explored, highlighting the need for innovative approaches. Additionally, the study's objectives are outlined, along with the research questions that guide the investigation. Finally, an overview of the research methodology is provided, describing its phases and the methods used to achieve the research goals.

1.1. Motivation and context

Cybersecurity has been defined in various ways, including terms like data security, information security, network security, and cyber security, all focused on safeguarding data in the digital world [1]. Data security specifically refers to the protection of digital information from unauthorized access, alteration, or exposure throughout its entire lifecycle [2].

Information security involves safeguarding physical or electronic data from unauthorized access, use, disclosure, modification, review, recording, or destruction and has the primary objective of ensuring Confidentiality, Integrity and Availability (CIA) of information. Network security also focuses on ensuring the CIA triad, but specifically when it comes to computer networks and the data transmitted through them [3]. Cybersecurity, however, has a broader scope, protecting computers, servers, mobile devices, electronic systems, networks, and data from malicious attacks. While data security, information security, and network security focus on preventing unauthorized access, alteration, or destruction of stored or transmitted data, cybersecurity extends to cover the entire flow of information. Today, the term "cybersecurity" is most used to describe this comprehensive approach [4].

While early forms of cybercrime were relatively simple, today's threat landscape is far more organized, scalable, and difficult to defend. There were fewer devices connected to the digital environment, and the attacks were far less complex compared to those seen today. In recent years, the rapid advancement of digital technology has significantly increased organizations' reliance on digital infrastructure for their daily operations. However, this growing dependence has also heightened their exposure to cyber threats, which are becoming more sophisticated, frequent, and impactful [5]. Across various sectors, businesses and institutions encounter numerous daily cyberattacks. If not properly managed, these attacks can lead to serious consequences, including operational breakdowns,

data breaches, and damage to an organization's reputation, which will ultimately impact business economics and viability.

The cybersecurity environment is constantly evolving, driven by ongoing advancements in computing and communication technologies. This growing interconnectivity further speeds up the pace of change. With each technological leap, new vulnerabilities emerge, requiring updated defensive strategies. Cyberspace itself is inherently fluid, undergoing continuous transformation [6]. However, in most cyber-attacks, human behavior and reactions to malicious activity are often the most vulnerable points, paving the way for successful breaches. Actions such as distraction, lack of knowledge, curiosity, disregard for security protocols, and unawareness of cyber threats can cause serious harm or even trigger an attack [7].

SOCs have emerged as vital components in the defense of organizational networks and systems to counter these risks. SOCs are tasked with continuously monitoring, detecting, investigating, and responding to security incidents. Yet, as the volume and complexity of cyber-attacks rise, SOC teams are increasingly strained. Analysts must handle an overwhelming volume of security alerts generated by various detection tools, ranging from false positives to critical threats. The challenge lies not only in filtering out the noise but also in responding to true threats in a timely and effective manner [8]. The lack of clear and standardized guidelines for SOC analysts can result in inadequate or delayed responses and the compromise of the security of organizations. This highlights the need for well-structured, specific guidelines that guide analysts on how to respond to different types of threats, providing a solid foundation of procedures for efficient and timely incident response [8].

While some organizations rely on automated tools and playbooks, these solutions often lack the adaptability and precision needed to handle the full spectrum of attacks [9]. This research begins by identifying the main challenges and gaps in current incident response practices within SOCs, such as the heavy reliance on manual procedures, lack of standardized guidance, and difficulty in accessing contextual information during investigations. Building on these insights, the thesis develops and evaluates ASTRA — an AI-powered assistant designed to support analysts by automatically generating structured, context-aware response playbooks for different attack scenarios. The system aims to simplify the work of SOC analysts, improve the clarity and consistency of response actions, and provide a practical demonstration of how artificial intelligence can be integrated into real-world security operations.

Artificial Intelligence (AI) technology offers significant potential to help address these challenges. Through the automation of routine tasks — such as filtering security alerts and detecting attack patterns — AI can substantially reduce the cognitive burden and existing fatigue on analysts. Moreover, AI systems are capable of learning from past incidents and adapting their responses accordingly, providing analysts with context-aware recommendations and dynamically generated response procedures. This enhances both

the speed and accuracy of incident handling while ensuring that SOC analysts follow best practices consistently, even under high-pressure scenarios [10].

Despite its promise, AI's integration into SOC workflows remains in its infancy. There is a clear need for advanced tools that can assist analysts' decision-making by offering actionable insights during ongoing security incidents. This research addresses that need by developing AI SOC Tool for Response Automation (ASTRA), an AI-powered tool designed to assist SOC analysts in generating response procedures adjusted to specific cyberattacks. Ultimately, ASTRA aims to improve the efficiency and effectiveness of SOC operations by accelerating triage, standardising best practices, and supporting faster, more precise responses to emerging threats.

1.2. Objectives and Research Question

The purpose of this study is to investigate how artificial intelligence can be applied to support and automate incident response in SOCs. Building on this motivation, the thesis sets out a series of objectives that operationalise both the research inquiry and the practical development of the proposed solution, ASTRA.

The overarching objective of this work is to design, implement, and evaluate a prototype system that enhances incident response capabilities in SOCs through the integration of automation and AI. To achieve this, the following specific objectives were defined:

(1) Identify and analyse the key challenges SOC teams face in managing cyber incidents;
(2) Review existing tools and approaches for SOC incident response, identifying their limitations and design constraints;
(3) Define design principles and technical requirements for an improved tool that enhances incident response capabilities in SOCs;
(4) Explore and integrate automation and AI techniques to optimise incident analysis and streamline response workflows;
(5) Develop and implement a prototype of the proposed tool, ensuring alignment with the defined design principles and operational constraints;

Together, these objectives operationalise the design, implementation, and empirical evaluation of ASTRA. The first two objectives are addressed through a comprehensive literature review that provides the conceptual and contextual foundation for the artifact's development. The practical work of this thesis focuses on objectives 3 to 5, which involve defining design principles, integrating AI-based automation, developing the ASTRA prototype, and assessing its usability and applicability through controlled testing and expert feedback.

Building on these objectives, the central research question guiding this study is:

> *How can artificial intelligence be applied to support and automate incident response in SOCs through the generation of structured, context-aware playbooks?*

This question integrates the conceptual and practical dimensions of the research, linking the identification of SOC operational challenges with the design and evaluation of ASTRA.

1.3. Methodology

This research follows the **Design Science Research Methodology (DSRM)**, which provides a structured framework for developing and evaluating technological artifacts that solve practical problems. DSRM is particularly suitable for applied research aiming to design, implement, and validate a system within a real-world operational context. In this study, the artifact is ASTRA, an AI-powered assistant that generates structured, context-aware incident-response playbooks for SOCs. Figure 1.1 illustrates the DSRM process model and its adaptation to this work.

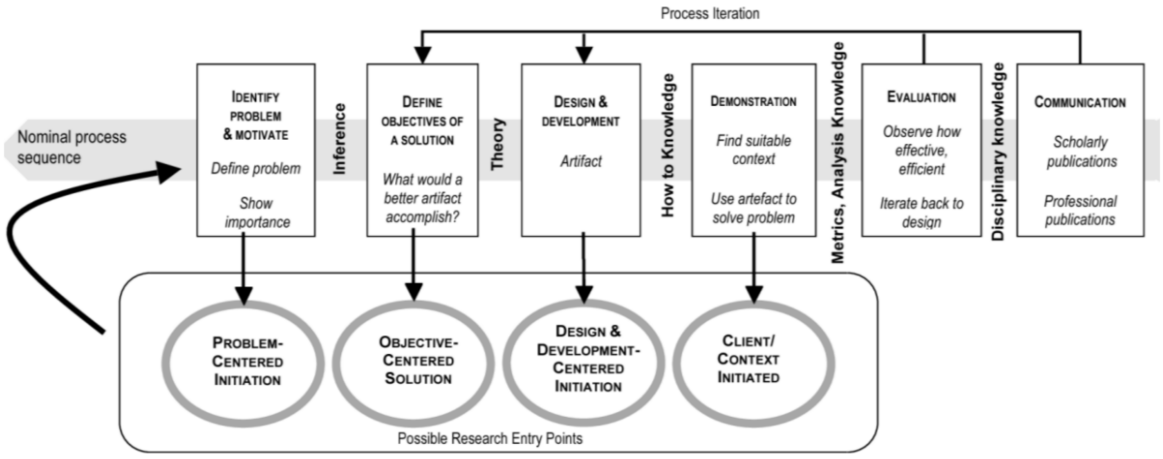


FIGURE 1.1. Design Science Research Methodology (DSRM) process model and its application to this study. Adapted from [11].

1.3.1. Problem Identification and Motivation

The first phase establishes the operational challenges observed in SOC environments—alert overload, inconsistent procedures, and limited automation. A review of the literature will support the identification of the core issues and justify the need for an adaptive, AI-based assistant capable of dynamically producing actionable response playbooks that bridge the gap between static documentation and real-time SOC requirements.

1.3.2. Define Objectives of the Solution

In this phase, the objectives of the proposed solution are defined. The main goal is to enhance incident-response clarity, consistency, and efficiency through the use of artificial intelligence. Specific objectives include defining design principles and requirements for a SOC-compatible assistant, integrating RAG with a curated knowledge base, ensuring transparency and data privacy, and validating the tool's relevance.

1.3.3. Design and Development

ASTRA will be developed as a modular prototype composed of a user interface, preprocessing pipeline, semantic-retrieval component, and a local language model for structured output generation. The back-end will manage data normalization, embedding generation, FAISS-based retrieval, and inference through Ollama, while the Streamlit front-end will allow analysts to input alerts and visualize or export the generated playbooks.

1.3.4. Demonstration

Once the prototype is implemented, demonstration scenarios will be carried out to validate its applicability. ASTRA will be tested against representative alert types—such as brute-force, phishing, and ransomware attacks—to verify whether it generates coherent and MITRE-mapped playbooks aligned with standard response phases (investigation, containment, eradication, and recovery).

1.3.5. Evaluation

Evaluation will combine technical and user-centered perspectives. System-level tests will assess JSON validity, retrieval precision, and latency under different configurations, while expert feedback will be collected through surveys to evaluate clarity, completeness, and perceived usefulness of the generated outputs. These results will determine whether the artifact meets its intended objectives and identify potential improvements.

1.3.6. Communication

The final phase will focus on disseminating the findings and results of the research. The full process will be documented within this dissertation.

1.4. Dissertation Structure

This dissertation is organised into five main chapters, each contributing to the progressive development of the research and ensuring a logical and coherent presentation of the work.

Chapter 1 outlines the motivation and context of the study, formulates the research question, and defines the objectives that guide the investigation. It also presents the adopted research methodology based on the Design Science Research Methodology (DSRM) framework and explains how it is applied throughout the dissertation.

Chapter 2 presents a comprehensive review of existing literature relevant to this study. It includes the methodology for the systematic review and analyses current approaches and technologies related to incident response automation, artificial intelligence in SOC environments, and evaluation metrics for AI-driven security systems. The chapter concludes by identifying gaps and opportunities that justify the development of ASTRA.

Chapter 3 describes the requirements, architecture, and technical implementation of the proposed artifact. It details the system components, including data preprocessing, retrieval, and language model integration, and explains the design decisions that ensure usability, security, and scalability.

Chapter 4 focuses on the demonstration and assessment of the ASTRA prototype. It presents the testing scenarios, system-level evaluations (including latency and retrieval performance), and the expert feedback obtained from cybersecurity professionals. This chapter also validates how the artifact meets the defined functional and non-functional requirements.

Chapter 5 summarises the main contributions and findings of the research. It reflects on the implications of the results, discusses the limitations identified during the study, and proposes directions for future work to extend and refine ASTRA.

Finally, the dissertation includes Annexes containing complementary material such as interface screenshots, expert survey data, and additional implementation or evaluation details that support the transparency and reproducibility of the study.

CHAPTER 2

State of the art

In this chapter, all relevant information related to the study's topic is reviewed. The goal is to gain an understanding of the existing approaches and solutions to the problem outlined in Chapter 1. Also included is the methodology used for conducting the research, including the specific tools and frameworks.

2.1. Systematic Review - Methodology

Taking into account the research question established in Section 1.2, the keywords were defined in four categories represented in Figure 2.1

This study selects IEEE Xplore, ACM Digital Library, and ScienceDirect for their strengths and complementarity in cybersecurity, AI, and SOC automation. IEEE Xplore provides highly technical research, while ACM Digital Library offers algorithmic and implementation-focused studies. ScienceDirect complements these by providing industry-oriented insights, case studies, and management frameworks. Together, these databases ensure a comprehensive and balanced literature review, covering both theoretical advancements and applied research.

Based on the keywords defined, the following Boolean expression was drafted:

> *("Security Operations Center" OR "SOC" OR "Cybersecurity Operations") AND ("Incident Response" OR "Threat Management" OR "Cyber Threat Detection") AND ("Artificial Intelligence" OR "AI" OR "Machine Learning" OR "AI-driven Security" OR "Automated Threat Detection") AND ("Efficiency" OR "Accuracy" OR "Response Time")*

However, this Boolean expression was too restrictive for IEEE Xplore, limiting the number of retrieved studies. To broaden the search while maintaining relevance, "Efficiency," "Accuracy," and "Response Time" were removed, allowing for a wider range

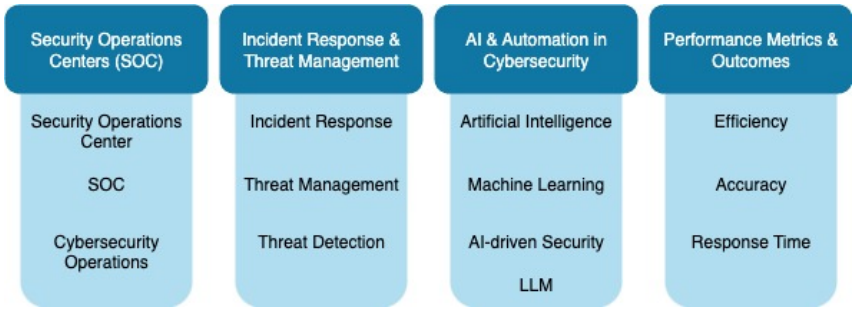


FIGURE 2.1. Keywords defined based on the study research question and used in the database query.

of research to be included. Additionally, quotation marks were removed from "Threat Management" and "Cyber Threat Detection" to capture related expressions. The level of strictness in these terms was adjusted based on the number of results obtained for each of the remaining database queries. For Science Direct, the number of Boolean operators is limited to 8 and therefore, some keywords were left out. The final queries and number of unfiltered results are shown in Table 2.1.

TABLE 2.1. Database search strategy and total number of results retrieved per database

Database	**Search Strategy**	**Results**
IEEE Xplore	(("Security Operations Center" OR "SOC" OR "Cybersecurity Operations") AND ("Incident Response" OR "Threat Management" OR "Cyber Threat Detection") AND ("Artificial Intelligence" OR "Machine Learning" OR "AI-driven Security" OR "LLM" OR "AI"))	124
ACM Digital Library	[[Full Text: "security operations center"] OR [Full Text: "soc"] OR [Full Text: "cybersecurity operations"]] AND [[Full Text: "incident response"] OR [Full Text: "threat management"] OR [Full Text: "cyber threat detection"]] AND [[Full Text: "artificial intelligence"] OR [Full Text: "machine learning"] OR [Full Text: "ai-driven security"] OR [Full Text: "LLM"] OR [Full Text: "ai"]]	285
ScienceDirect	(("Security Operations Center" OR "SOC" OR "Cybersecurity Operations") AND ("Incident Response" OR "Cyber Threat Detection") AND ("Artificial Intelligence" OR "Machine Learning" OR "AI-driven Security" OR "LLM"))	179

The next step in the systematic review involved defining the criteria used to determine which studies were retained for analysis. Studies were included if they presented direct relevance to SOC operations and AI-driven security solutions, as verified through manual screening. Research that discussed incident response procedures and their integration with artificial intelligence tools was also considered, as well as studies describing practical implementations, case studies, or evaluations of incident response systems.

Conversely, studies were excluded if they lacked a DOI identifier, were published before 2018, or were not peer-reviewed journal papers. Duplicate records retrieved from multiple databases were removed, as were papers unrelated to SOC operations, AI-driven security, or cybersecurity automation.

The PRISMA flowchart represented in figure 2.2 illustrates the systematic filtering process applied to refine the dataset for this study. Starting with 588 records retrieved from IEEE Xplore, ACM Digital Library, and ScienceDirect, the filtering resulted in a final set of 31 publications.

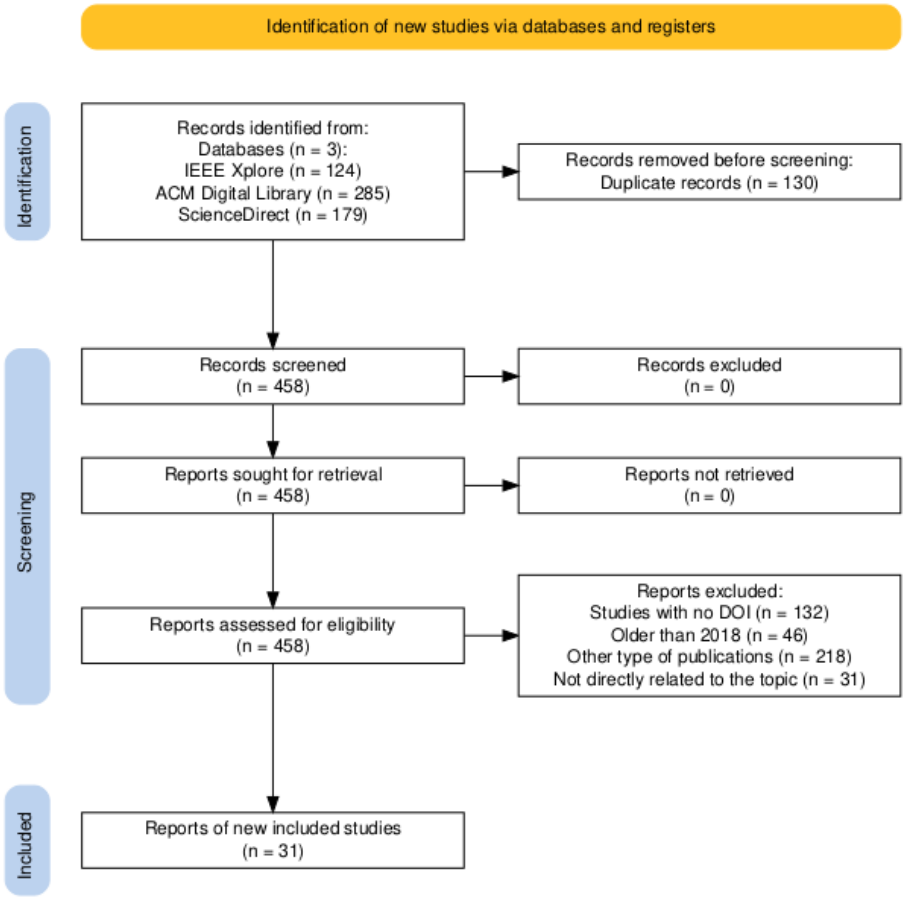


FIGURE 2.2. PRISMA 2020 Flow Diagram for the systematic review process. From [12].

2.2. Systematic Literature Review

This section reviews how AI is being applied to strengthen SOCs and the challenges that remain. It explores how current research approaches automation, decision-making, and human–machine collaboration in cybersecurity. The discussion moves from the foundations - how SOCs operate and where they struggle - to the growing role of AI and Machine Learning (ML) in improving detection and response. It then looks at Security Orchestration, Automation, and Response (SOAR) platforms, the metrics used to evaluate AI performance, and the main gaps still limiting progress. The section closes with a look at the rise of Large Language Model (LLM) and their potential to make SOC automation more adaptive and collaborative. Together, these perspectives form the groundwork for the design and development of the ASTRA prototype described in the next chapters.

2.2.1. Security Operations Centers (SOCs) Incident Response

SOCs play a crucial role in an organization's cybersecurity strategy by monitoring, detecting, investigating, and responding to cyber threats in real time. However, SOC teams face numerous challenges in executing incident response effectively, including alert fatigue, a shortage of skilled personnel, lack of automation, and collaboration difficulties. These

challenges contribute to inefficiencies that can compromise an organization's ability to mitigate cyber threats effectively.

One of the most significant issues SOC teams encounter is the overwhelming volume of security alerts generated by various monitoring tools and systems [13]. A substantial portion of these alerts are false positives, which can waste valuable analyst time and resources. Recent reports indicate that over 50% of security alerts are false positives, exacerbating alert fatigue among security teams [14]. This fatigue can lead analysts to overlook or disable alerts, increasing the risk of missing critical security incidents. Effectively managing and prioritizing alerts is essential to ensure that SOC teams focus on the most critical threats. Incident prioritization involves assessing alert type, severity, affected systems, and potential organizational impact [15]. The inability to correctly prioritize incidents results in delayed response times, misallocation of resources, and increased exposure to cyber threats. Moreover, the high volume of alerts can cause cognitive overload for analysts, reducing their ability to make timely and accurate decisions.

SOC teams also face a persistent challenge in staffing and skills shortages [15]. As cyber threats become more sophisticated, the demand for skilled cybersecurity professionals continues to outpace supply. Many organizations struggle to recruit and retain qualified SOC analysts, leading to an increased reliance on outsourcing cybersecurity operations [13]. Additionally, SOC analysts often deal with repetitive and tedious tasks, which can contribute to burnout and high turnover rates. The shortage of experienced personnel also affects the speed and effectiveness of incident response. Without adequate staffing, SOC teams may struggle to conduct thorough investigations, leading to incomplete threat containment and inadequate remediation efforts. Furthermore, newly recruited analysts often require extensive training, which consumes additional time and resources, further delaying incident response improvements.

Many SOCs suffer from a lack of automation and orchestration, which leads to inefficiencies in incident response [15]. The absence of well-defined processes or playbooks can result in confusion, delays, and inconsistent responses to cyber incidents. Without automated tools, incident detection and response remain highly manual, increasing response times and leaving organizations vulnerable to ongoing attacks. The lack of standardized processes in SOCs means that different analysts may handle the same type of incident differently, creating inconsistencies in incident resolution. This inconsistency leads to gaps in security coverage, increasing the likelihood of cyber threats successfully exploiting vulnerabilities. Moreover, the reliance on manual processes makes it difficult for SOCs to scale their operations effectively, especially as the volume and complexity of threats continue to grow.

Effective collaboration and communication are critical for SOC operations, yet many organizations struggle with isolated teams and inadequate information sharing [15, 16, 17]. A lack of coordination between SOC teams, Information Technology (IT) departments, and external stakeholders can delay incident resolution and prolong response times. A

significant challenge arises when multiple teams work on different aspects of cybersecurity without clear communication channels. Miscommunication or delays in sharing threat intelligence can result in disjointed incident response efforts, making it harder to contain and mitigate threats effectively. Additionally, organizations with fragmented security architectures may experience delays in relaying critical information between departments, further complicating incident response.

Beyond these primary challenges, SOCs face additional issues such as integrating modern security tools with legacy infrastructures, managing vast amounts of diverse security data, and balancing detection sensitivity to minimize false positives while ensuring true threats are not overlooked[18].

2.2.2. AI and Machine Learning in Cybersecurity

AI and ML are transforming cybersecurity by automating processes, improving threat detection, and enhancing response efficiency [3]. The ability of AI to rapidly process and analyse large amounts of data allows SOCs to identify cyber threats more effectively, particularly as attacks grow in complexity and volume [19]. AI is widely integrated into cybersecurity to support threat detection, predictive analytics, automated response, vulnerability assessments, and fraud detection. AI-driven threat detection systems analyse vast datasets to identify patterns and anomalies that traditional methods may overlook [20]. Predictive analytics plays a crucial role in estimating future attack risks by examining historical data and current trends [8]. Automated response mechanisms enable AI to handle routine security tasks such as log analysis and alert prioritization, allowing human analysts to focus on more complex issues. Additionally, AI is used for vulnerability assessment to identify weaknesses that attackers could exploit and for fraud detection to safeguard financial transactions [8]. ML is a key component of AI-driven cybersecurity, employing different approaches to improve security defences. Supervised learning relies on labelled data to classify threats and enhance vulnerability scanning, while unsupervised learning detects anomalies without predefined labels, making it valuable for identifying novel attack patterns [21]. Reinforcement learning further advances security systems by training models to adapt to evolving threats and optimize defence strategies [21]. Trends in AI-driven SOC automation highlight the growing reliance on technologies such as Security Orchestration, Automation, and Response (SOAR), which streamlines incident response by automating workflows [15]. Predictive threat intelligence enhances proactive security measures, allowing organizations to anticipate and counteract potential cyber threats before they occur [8]. Human-AI collaboration is also gaining traction, integrating AI's analytical power with human expertise to improve decision-making and threat analysis [15]. Explainable AI (XAI) is an essential aspect of AI-driven cybersecurity, as it enhances transparency, builds trust, and supports effective decision-making. Understanding how AI models reach conclusions is critical for SOC analysts, ensuring that AI-driven security measures align with best practices and regulatory requirements

[3]. XAI helps analysts interpret AI-generated threat assessments, reducing bias and improving response strategies [21]. However, while explainability improves AI reliability, it can also introduce vulnerabilities that adversaries may exploit [22].

2.2.3. Security Orchestration, Automation, and Response (SOAR)

SOAR platforms have become indispensable in enhancing the efficiency of SOCs by automating repetitive tasks, simplifying incident response, and improving threat detection [15]. By integrating with Security Information and Event Management (SIEM) systems and leveraging threat intelligence, in practice, SOAR platforms help analysts coordinate faster responses by processing large data volumes more consistently than manual procedures. SOAR platforms function by automating manual SOC processes, reducing response times, and improving consistency in incident management. These platforms utilize workflows, known as playbooks, which outline predefined response actions that analysts follow to investigate and mitigate security incidents [23]. Playbooks may vary in complexity, ranging from manual interventions to fully automated workflows, ensuring that security teams can respond effectively to various types of threats [10]. Additionally, SOAR enhances decision-making by correlating data from multiple sources, enriching alerts, and prioritizing incidents based on severity [10]. A key aspect of SOAR platforms is their smooth integration with SIEM systems and threat intelligence platforms. SIEM systems generate alerts by analysing logs from multiple sources, identifying suspicious activities, and forwarding relevant data to SOAR platforms for further processing. SOAR platforms enhance these alerts by aggregating additional contextual information from internal and external threat intelligence sources, thereby improving the accuracy of threat assessments [10]. The integration of SOAR with Cyber Threat Intelligence (CTI) allows organizations to enhance their situational awareness, anticipate potential attacks, and improve proactive security measures [10]. Despite their many advantages, existing SOAR solutions face several limitations that obstruct their full potential. One key challenge is the complexity of playbook creation and maintenance, which requires security analysts to possess extensive knowledge of the integrated tools and incident response procedures. Once created, playbooks may become rigid and difficult to adapt to evolving threats, limiting their effectiveness against new types of attack scenarios [10]. Additionally, many SOAR platforms rely on rule-based mapping methods to correlate alerts with response workflows, which can lead to inefficiencies when handling dynamic security incidents [10]. A major limitation of SOAR implementation is the high operational costs associated with its deployment and maintenance. Large-scale data processing requires significant computational resources, which can drive up expenses, making it difficult for some organizations to fully integrate SOAR solutions into their security operations [23]. To improve the effectiveness of SOAR solutions, several enhancements are necessary. The adoption of AI-driven automation can enable more adaptive and intelligent playbooks, reducing reliance on static rule-based approaches. AI-powered models can analyse historical data and dynamically adjust workflows to match emerging threats, improving the flexibility and responsiveness

of incident response procedures [10]. Continuous updates to threat detection algorithms and integration with advanced analytics can further enhance the capabilities of SOAR platforms, ensuring that they remain effective against evolving cyber threats.

2.2.4. Metrics for Evaluating AI in SOCs

Key Performance Indicator (KPI) are essential in assessing the effectiveness of AI in SOCs, particularly in areas such as accuracy, response time, and false positive rates [5]. These metrics help SOC teams quantify AI's impact on incident response and compare its performance against traditional SOC workflows.

Accuracy is a fundamental KPI that measures how well AI models correctly identify security threats while minimizing false positives and false negatives [18]. AI-driven systems have demonstrated higher accuracy in threat detection compared to conventional approaches, reducing the burden of manual investigation for analysts [8]. XAI plays a role in validating model performance by offering insights into its behavior and enabling analysts to detect and correct errors [21]. Accuracy is often assessed using precision, recall, and the F1-score, which balance the trade-off between detecting true threats and minimizing misclassifications. Response time is another critical metric, as faster incident detection and remediation significantly reduce the impact of cyberattacks. AI-powered automation shortens the time between threat detection and resolution, allowing organizations to respond to incidents in real time [8]. By automating routine security tasks, AI reduces delays associated with manual interventions and enhances SOC efficiency. False positive rate (FPR) is an essential measure of AI effectiveness, as a high rate of false positives can overwhelm analysts with unnecessary alerts. AI-driven solutions mitigate this challenge by improving alert correlation and reducing redundant notifications, allowing analysts to focus on legitimate security threats. Lowering false positive rates leads to better resource allocation and reduces alert fatigue within SOC teams. In measuring AI effectiveness in incident response, SOC teams rely on AI-driven decision support tools that analyze incident data and provide actionable recommendations [8]. AI assists in prioritizing security incidents by evaluating their potential impact and severity, ensuring that critical threats receive immediate attention. Additionally, AI-powered security playbooks, which leverage historical data, facilitate faster and more consistent responses to security events [3]. Automated playbooks also enable the sharing of standardized response procedures across organizations, enhancing collaboration and knowledge transfer. When benchmarking AI performance against traditional SOC workflows, efficiency gains are a key advantage. AI-driven automation significantly reduces the time required to detect and resolve security incidents, freeing analysts to focus on high-priority threats [8]. Automating repetitive tasks also leads to cost savings, reducing the reliance on manual labor while improving SOC effectiveness. AI frameworks further enhance threat detection by analyzing vast amounts of data with superior speed and precision, allowing for more proactive threat mitigation. Scalability is another benefit of AI adoption in SOCs, as AI-driven systems can adapt to an organization's evolving security needs without compromising

performance. AI also supports human-AI teaming by combining automation with human expertise, allowing SOC analysts to oversee and fine-tune AI-driven processes [15]. This collaboration ensures that AI tools augment rather than replace human decision-making, leveraging both machine efficiency and human intuition to optimize incident response. Various frameworks guide the evaluation of AI in SOCs, including the Performance Measurement Guide for Information Security (NIST SP 800-55), which outlines methods for selecting and implementing security metrics [5]. Metrics are commonly categorized into incident volume, detection efficiency, response effectiveness, and incident impact. Operational metrics such as mean time to detection, containment, and resolution provide insight into SOC performance, while technical metrics like threat actor attribution and automated incident resolution rates offer deeper analytics on AI's role in security operations. Despite the advantages AI brings to SOCs, researchers continue to debate how effective these models truly are in day-to-day SOC operations. Data quality issues can impact AI-driven security outcomes, and the lack of interpretability in complex ML models may hinder analyst trust [24].

2.2.5. Gaps and Challenges in Existing Research

AI-based automation in SOCs faces several unresolved issues, including the need for comprehensive evaluations, better integration of human and mechanical components through non-technical processes, and holistic analyses that interlink human skills, AI algorithms, and procedural systems [8]. The absence of strategic security patterns for different use cases and risks, as well as limited conceptual advances, further compound these challenges [25]. A major challenge is data quality, as the effectiveness of AI in cybersecurity depends heavily on the availability and reliability of training data. High-quality, labelled data is critical for training machine learning models, but obtaining such datasets remains difficult [26]. Additionally, many AI-driven security systems rely on isolated datasets, limiting their ability to generalize and detect threats across diverse environments [3]. Imbalanced data is another concern, as threat detection models may become biased toward frequent attack types while failing to identify rare or emerging threats [19]. Privacy and security concerns further complicate data sharing, restricting access to comprehensive threat intelligence datasets that could improve AI model performance [21]. Another pressing issue is the vulnerability of AI models to adversarial attacks. Cybercriminals can manipulate AI systems by introducing misleading data, exploiting weaknesses in machine learning algorithms, and bypassing automated detection mechanisms [8]. This highlights the need for robust AI security standards that ensure transparency, reliability, and resilience against adversarial threats [27]. Moreover, frequent updates and retraining are required to keep AI models relevant against evolving cyber threats, adding to operational complexity. Human-AI collaboration is also a critical challenge, as effective integration between human analysts and AI systems remains underdeveloped. While AI-powered automation can handle routine security tasks, human oversight is necessary to manage complex threat scenarios and mitigate AI biases [28]. Trust in AI recommendations is essential, but many

SOC analysts struggle with understanding AI decision-making processes due to a lack of explainability in current models [21]. There is a gap in research exploring optimal strategies for human-AI teaming in SOC environments, as well as best practices for integrating AI into existing security workflows [3]. Several gaps remain in existing research that this thesis aims to address. There is a lack of standardized methodologies for evaluating AI performance in SOCs, making it difficult to compare different approaches. Additionally, research on improving data quality for AI-driven threat detection is limited, particularly in the context of multi-source threat intelligence. AI usability concerns also persist, as many security tools require extensive expertise to operate effectively, limiting their adoption in real-world SOC environments. Future studies should focus on developing explainable AI models, optimizing human-AI collaboration strategies, and enhancing AI adaptability to evolving threats. Addressing these gaps will contribute to the advancement of AI-driven SOC operations and improve overall cybersecurity resilience.

2.2.6. Large Language Models in Security Operations Centres

The rapid evolution of Large Language Models (LLMs) such as GPT-4 and PaLM 2 has opened new avenues for enhancing SOC efficiency and analyst support. Recent empirical studies show that SOC analysts increasingly adopt LLMs as on-demand cognitive aids for tasks such as interpreting telemetry, analysing command-line activity, and drafting incident communications, rather than delegating critical decision-making to them [29]. This human–AI collaboration model represents a shift from automation to augmentation, where LLMs provide contextual reasoning, summarisation, and knowledge retrieval aligned with established cybersecurity competencies such as those in the NICE framework. By integrating with existing SIEM, SOAR, and XDR workflows, LLM-powered assistants help analysts synthesise evidence, reduce alert fatigue, and accelerate sense-making—while preserving analyst autonomy and accountability in high-stakes operational environments [29].

Parallel advances in hardware-oriented SOC research demonstrate the versatility of LLMs beyond textual analysis. Frameworks such as DIVAS leverage LLMs to identify Common Weakness Enumerations (CWEs) from SoC design specifications and automatically generate security assertions and policies, showcasing their potential in structured reasoning and policy enforcement [30]. Similarly, recent work describes a paradigm shift in SoC security verification, where LLMs support vulnerability detection, security property generation, and countermeasure development—outperforming traditional rule-based or machine-learning methods in scalability and adaptability [31]. Collectively, these studies underline that LLMs can serve as multi-layered intelligence engines within SOCs—bridging natural-language understanding with domain-specific security reasoning and paving the way for adaptive, context-aware security operations that combine human expertise with generative AI capabilities.

2.3. Summary and Research Direction

SOCs face increasing challenges in managing cyber threats due to rising attack complexity, alert fatigue, and a shortage of skilled analysts. While AI-driven solutions promise enhanced efficiency, automation, and decision support, their integration into SOC workflows remains underdeveloped. Existing research highlights significant gaps in AI-based SOC automation, including issues related to data quality, adversarial attacks, and human-AI collaboration. Furthermore, the effectiveness of AI in incident response is still not well understood, with key performance metrics such as accuracy, response time, and false positive rates requiring further exploration. SOAR platforms have made progress in improving SOC operations, but their rigid rule-based workflows and reliance on human-defined playbooks limit their adaptability to emerging threats. Additionally, the lack of standardized evaluation methodologies and benchmarking frameworks makes it difficult to compare AI-driven solutions against traditional SOC workflows. This thesis research develops ASTRA, an AI-powered incident response assistant that integrates automation while maintaining human oversight. The study will focus on designing an adaptive and scalable AI system that enhances SOC efficiency through intelligent alert triage and automated playbook generation.

CHAPTER 3

ASTRA: AI SOC Tool for Response Automation

This chapter presents the design and development of ASTRA, an AI-powered assistant that supports incident response in SOCs. Building upon the gaps and challenges identified in the literature review, this chapter translates the conceptual objectives defined in Chapter 2 into concrete design and engineering decisions. It begins by defining the system's functional and non-functional requirements, followed by a detailed description of the overall architecture and its core components. Finally, it outlines the implementation choices that guided the development of the prototype, ensuring alignment with SOC operational constraints such as privacy, scalability, and usability.

3.1. Introduction

SOCs operate under conditions of urgency, unpredictability, and continuous data influx. Analysts must detect, investigate, and respond to security events under significant time pressure while adversaries constantly adapt their tactics. In such contexts, seemingly small delays during detection and triage can translate into hours of remediation effort or, in severe cases, irreversible compromise [5, 14].

Traditional SOCs often rely on static incident response playbooks to bring order to this complexity. While these playbooks provide useful baselines, they are limited: they reflect knowledge at the time of writing, require manual revision to accommodate novel threats, and rarely capture the nuance of unfolding incidents [19, 10]. These constraints motivated the development of ASTRA (AI SOC Tool for Response Automation).

ASTRA does not aim to replace human expertise. Instead, it functions as a context amplifier that dynamically retrieves, interprets, and presents procedures tailored to the circumstances of a live incident. This aligns with an emerging emphasis on human–AI teaming, in which artificial intelligence augments analyst decision-making while preserving accountability and oversight [15, 21].

The remainder of this chapter details the requirements that guided ASTRA's design, the architecture and technologies adopted, the engineering of the knowledge base, the prompt design and output schema that enforce structure and traceability, the user interface choices shaped by SOC ergonomics, and the iterative development and testing cycles that refined the prototype.

3.2. Requirements Definition

The requirements for ASTRA were primarily derived from a review of existing literature on incident response challenges and tooling. Requirements are presented as functional and non-functional categories for clarity and later traceability in Chapter 4.

3.2.1. Functional Requirements

The SOC Assistant must be able to:

- **Support different types of input.** Accept both structured data (e.g., CSV exports from SIEMs) and unstructured artefacts (analyst notes, ticket descriptions, chat transcripts)
- **Provide granular categorisation.** Map incidents to kill-chain stages and MITRE ATT&CK techniques [20], rather than broad categories like "malware," to enable targeted procedures
- **Deliver phase-oriented procedural outputs.** Align guidance to incident response phases to reduce cognitive load and aid workflow navigation
- **Automatically extract Indicator of Compromise (IOC).** Identify IPs, domains, URLs, file hashes and other artefacts from unstructured text
- **Integrate with the MITRE Adversarial Tactics, Techniques, and Common Knowledge (MITRE ATT&CK) framework.** Utilize ATT&CK mappings both as a source of guidance during incident response and for post-incident reporting.
- **Generate dual-format outputs.** Produce human-readable narrative and machine-readable JavaScript Object Notation (JSON) simultaneously for analysts and automation pipelines.

3.2.2. Non-Functional Requirements

In addition, the tool must satisfy the following constraints:

- **Low latency**. Target < 6 seconds end-to-end for most queries to remain usable in interactive contexts.
- **Local execution**. Ensure that all processing occurs on-premises to comply with data sovereignty and privacy constraints
- **Modular design**. Allow substitution of components (embeddings, retriever, LLM) without major redesign
- **Scalability**. Maintain retrieval quality and stable latency as the knowledge base and interaction logs grow
- **Usability and learnability of the interface.** Provide a simple, self-explanatory single-page User Interface (UI) with clear affordances (e.g., copy-to-clipboard for IOC, collapsible sections by IR phase, explicit export actions) requiring minimal onboarding and forgiving error handling (sensible defaults, non-destructive actions).

3.3. Architecture Overview

ASTRA adopts a modular and linear pipeline that was purposefully designed to mirror the decision-making process of a SOC analyst. The tool's architecture balances retrieval accuracy, data privacy, and usability under realistic operational constraints. ASTRA's overall workflow is represented in Figure 3.1.

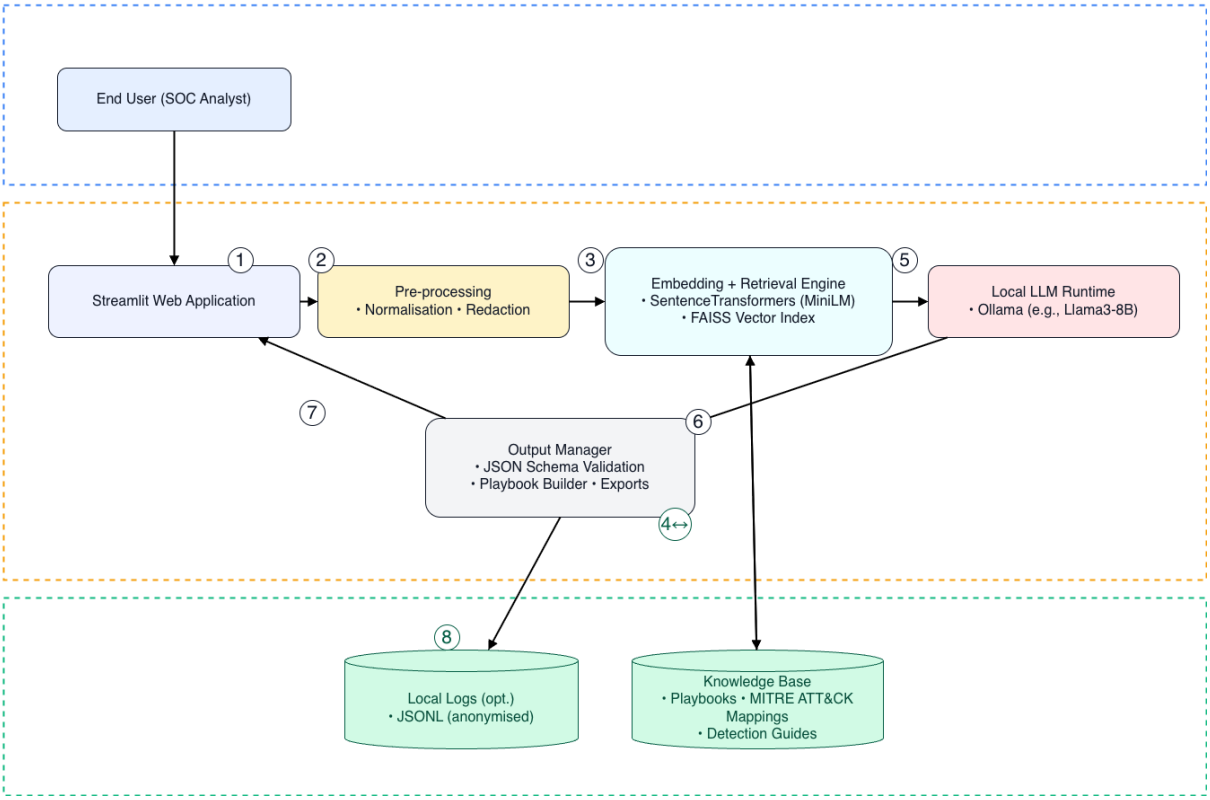


FIGURE 3.1. Flowchart of the information's flow through ASTRA's architecture

The processing flow within ASTRA follows the sequence illustrated in Figure 3.1. The process begins when the analyst inputs a security alert or event log directly through the Streamlit web interface (**1**). The pre-processing module then normalizes this input by removing formatting inconsistencies and, when enabled, redacts sensitive elements such as email addresses or IP addresses before any data persistence occurs (**2**). The cleaned text is subsequently used as the query for a retrieval step based on the RAG approach [32] (**3**). In this phase, ASTRA computes a dense embedding using a compact transformer model (`all-MiniLM-L6-v2`) and performs a similarity search over a FAISS [33] index containing embedded chunks of incident-response knowledge. The top-K most relevant documents are concatenated into a lightweight contextual block that supplements the original alert information with domain-specific knowledge drawn from the internal knowledge base (**4**).

The combined alert and retrieved context are then transformed into a structured prompt that also embeds the expected JSON schema and explicit constraints regarding completeness, realism of commands, and alignment with MITRE ATT&CK tactics and techniques. This enriched prompt is sent to a locally hosted large language model via the Ollama runtime (**5**). Using models such as `llama3:8b`, the system generates an incident-response plan in valid JSON format, which is subsequently parsed and validated against the predefined schema to ensure structural integrity and semantic consistency (**6**).

The validated response object is presented in an interactive interface organized around the main stages of incident response—summary and classification, IOC extraction, investigation, containment, eradication and recovery, and post-incident recommendations—allowing the analyst to visualize, refine, or export the generated playbook (**7**). Optionally, an anonymized JSONL log of the interaction can be stored locally for auditing or retraining purposes (**8**). Throughout this process, all computation and data handling are performed locally to preserve confidentiality and align with the operational requirements of SOC environments where sensitive information must remain within organizational boundaries.

3.3.1. Backend Architecture

The backend of ASTRA is implemented entirely in Python and orchestrates all data processing, retrieval, and inference components. It integrates three core services: (i) the SentenceTransformers [34] library for text embedding, (ii) FAISS for vector-based similarity search, and (iii) Ollama [35] for hosting the local language model and executing inference requests. Each incoming alert triggers a short-lived retrieval and inference cycle, with the relevant resources (embedding model and FAISS index) cached in memory through the user interface.

The retrieval pipeline operates as follows: the alert text is embedded into a fixed-size vector representation, which is compared against precomputed document embeddings stored in the FAISS index. The system then retrieves the top-K most relevant results, where K represents the number of knowledge-base chunks used to enrich the context for the language model. In this prototype, K was set to 4 based on preliminary validation tests showing that this value provided sufficient contextual coverage without introducing redundancy or excessive prompt length. The retrieved content is concatenated into a compact context block that serves as factual grounding for the language model. This strategy combines precision and interpretability, allowing the AI assistant to provide explanations derived from known sources while avoiding hallucinations. Once the context enriched prompt is generated, it is sent to the local model for inference. The model's response—expected to be a fully structured JSON object—is parsed, validated, and, if necessary, repaired using a light recovery function that extracts the first well-formed JSON segment.

Performance considerations were central in the backend design. Retrieval latencies were found to remain well below one second even for large knowledge bases (up to 50,000 chunks), as reported in Chapter 4. Additional safeguards include automatic redaction of sensitive data when saving logs, local inference to avoid data exfiltration, and deterministic JSON-line logging to ensure reproducibility and traceability. The backend therefore ensures a balance between speed, privacy, and reliability—three attributes critical to deploying AI within production SOC environments.

3.3.2. Frontend Architecture

The frontend is built as a single-page web application using the Streamlit framework [36], which enables rapid prototyping of data-driven interfaces. The visual design was customized through Cascading Style Sheets (CSS) to create a professional aesthetic suitable for operational settings. The interface is deliberately minimal: analysts can paste or upload input data, select whether to use contextual retrieval, and trigger response generation through a single "Generate Response" action. A dynamic status indicator provides feedback on inference progress, while tabs organize the resulting content into distinct logical sections.

The interface mirrors the internal structure of the JSON output. The *Summary* tab presents the executive summary, severity assessment, and confidence score, complemented by visually distinctive MITRE ATT&CK "chips" that display the associated tactics and techniques. The *IOC* tab lists extracted indicators in categorized groups (IPs, domains, URLs, hashes, users, hosts, and paths). The *Playbook* tab details investigation and containment steps, each containing associated commands and artifacts formatted as syntax-highlighted code blocks. Subsequent tabs display hunting queries (Sigma, KQL, and Splunk syntax), post-incident recommendations, and, optionally, the retrieved context for transparency.

Robustness and usability were key goals of the frontend. The application handles invalid or partial JSON, providing warnings and showing the raw text when parsing fails. Analysts can toggle context visibility, download structured or raw outputs, and enable anonymized logging. By combining transparency, interpretability, and simplicity, the frontend reinforces ASTRA's aim to function as a trusted assistant rather than a black-box generator.

3.3.3. Data Layer

The data layer serves as the foundation for ASTRA's retrieval capabilities and includes both static and dynamic data components. The static component, the Knowledge Base, is implemented as a locally stored collection of structured text and Markdown files containing curated SOC playbooks, detection guides, and MITRE ATT&CK mappings. These files are pre-processed and split into overlapping text chunks of 600–1 200 characters to preserve contextual continuity. Each chunk is embedded using the `all-MiniLM-L6-v2` model, producing dense vector representations that are indexed with FAISS to enable efficient semantic similarity search. The resulting index files (`incident_index.faiss`) are stored on disk within ASTRA's local data directory and versioned to track updates to the knowledge corpus. At runtime, the retrieval module loads this index from disk into memory and queries it dynamically whenever an alert is processed.

The dynamic component of the data layer comprises user inputs and generated outputs. User-provided alert data exist only in memory during the inference process, while generated outputs—including structured JSON results and raw model responses—can be

optionally exported by the analyst. Logs are stored locally in append-only JSONL format, optionally passing through a redaction layer that replaces any detected emails or IP addresses with placeholders. This ensures that no sensitive information is retained unless explicitly exported by the user.

From a scalability perspective, the data layer has been optimized for retrieval performance rather than data volume. Evaluations show linear increases in retrieval latency with corpus size, remaining within operationally acceptable thresholds up to tens of thousands of chunks. This efficiency allows ASTRA to scale toward richer knowledge sources without significant degradation in response time.

3.3.4. Non-Functional Considerations

Beyond the functional design, several non-functional objectives guided the architecture. Privacy and security were prioritized by adopting a fully local inference model, ensuring that no organizational data are transmitted externally. Explainability was achieved through explicit display of RAG context and ATT&CK mappings, enabling analysts to trace model reasoning. The JSON-first design ensures interoperability with external SOAR and SIEM systems, while usability is enhanced by a compact single-page interface and clear task organization. Finally, the architecture demonstrates robust performance—retrieval latencies below one second and consistent schema validation across runs—providing a reliable foundation for further evaluation and future enhancements such as fine-tuned adapters, expanded knowledge sources, or more scalable vector search backends.

3.4. Application Functionalities

This section describes the main functionalities available in the ASTRA prototype, highlighting how the interface supports analysts during the incident–response process. The application's UI was developed using the Streamlit framework [36] and is delivered as a single–page local web interface that allows the user to configure, execute, and review the full reasoning pipeline. A screenshot can be found in Figure 3.2.

3.4.1. Settings Panel

Located on the left sidebar, the *Settings Panel* allows the analyst to configure the behaviour of the system before execution. The first control, labelled "Use RAG context", enables or disables retrieval–augmented generation, determining whether the model will incorporate contextual information from the knowledge base. Below this option, the analyst can enable or disable the context information to figure in the output. When saving is disabled, the system displays the results only on screen without creating any file. Additional switches allow the anonymisation of sensitive elements (e.g., e-mail addresses or IP addresses), ensuring that these entities are masked on the logs.

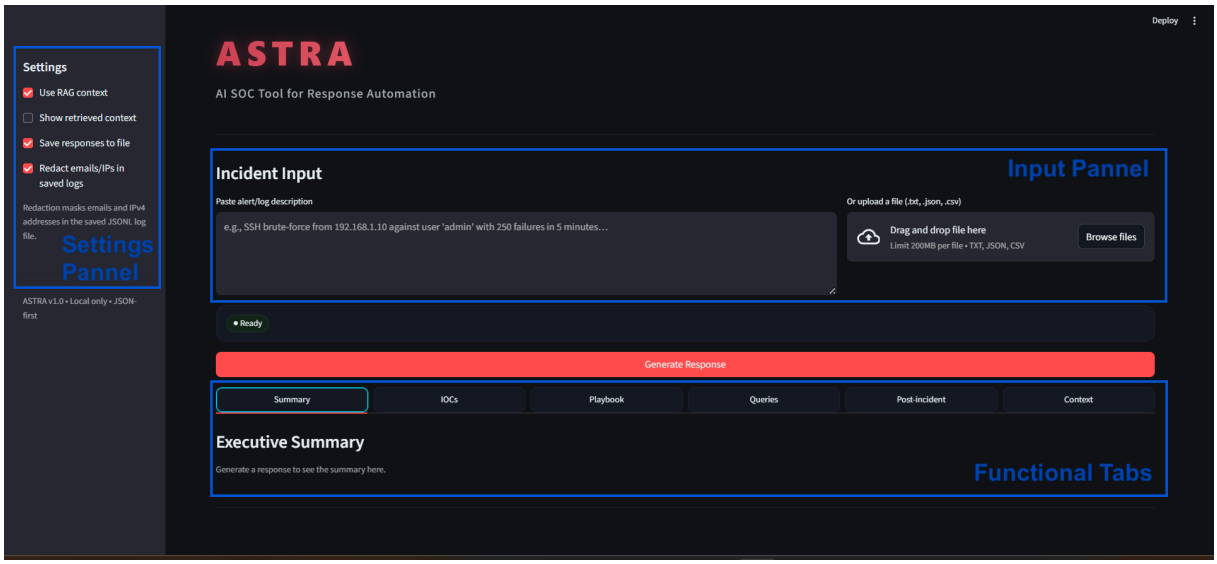


FIGURE 3.2. ASTRA's functional pannels.

3.4.2. Input Panel

At the top of the interface, the *Input Panel* provides the main entry point for data submission. The analyst can paste free text directly into the field or upload a file in `.txt`, `.json`, or `.csv` format containing one or more alerts. After submission, pressing the *Generate Response* button triggers the full pipeline described before. During execution, a progress indicator informs the user about preprocessing, retrieval, and inference status in real time.

3.4.3. Functional Tabs

Once the generation process is complete, the resulting structured outputs are presented through six navigable menus—referred to as *Functional Tabs*—that correspond to the main phases of incident response:

- **Summary** – presents a short description of the incident, its category, and the mapped MITRE ATT&CK tactics and techniques;
- **Indicators of Compromise (IOCs)** – lists extracted indicators such as IP addresses, domains, file hashes, or process names;
- **Playbook** – outlines recommended investigative steps, provides step-by-step technical actions for isolating and remediating the threat;
- **Queries** –provides example queries for SIEM or endpoint tools;
- **Post-Incident** – summarises lessons learned and preventive measures for future improvement;
- **Context** – displays the knowledge-base documents retrieved and used by the model during reasoning.

The analyst can select one of the tabs at a time, allowing to focus on specific stages of the event. All generated outputs can be copied to clipboard or downloaded as structured

JSON. When the anonymisation option is active, sensitive data are automatically removed from both on-screen content and saved files.

3.4.4. Interaction Behaviour

The interface provides a minimal, responsive layout that facilitates quick iteration between input, configuration, and output. All processing occurs locally, ensuring that no data leave the SOC environment. If the saving option is enabled, the results are stored in a JSON file together with metadata about the selected settings. Otherwise, the output remains visible only on screen. This configuration ensures privacy, auditability, and transparency while maintaining operational simplicity for the analyst.

All computation takes place locally to protect sensitive data and ensure compliance with SOC privacy requirements. When saving is enabled, ASTRA stores anonymised results and metadata in a local JSONL file; otherwise, outputs remain visible only within the session. This behaviour balances operational efficiency with confidentiality, ensuring that analysts can interact confidently with the system without data exposure risks.

3.4.5. Design Principles and Interaction Behaviour

The interface adheres to usability guidelines tailored for SOC environments. Its design emphasises four core principles:

- **Simplicity and Focus:** A dark, low-contrast colour palette reduces eye strain during extended monitoring sessions and keeps the analyst's attention on essential elements.
- **Consistency:** Uniform typography and layout across all panels and tabs allow for rapid visual scanning and minimise learning effort.
- **Transparency:** Retrieved context and intermediate reasoning steps can be optionally displayed, ensuring traceability and auditability of AI-generated recommendations.
- **Efficiency:** All key actions—input, configuration, execution, and export—are available within a single view, reducing the need for navigation between screens.

Overall, the implemented functionalities demonstrate that ASTRA is not limited to a backend automation engine but also functions as an interactive decision-support tool that delivers explainable, structured, and context-aware recommendations in real time.

3.5. Implementation summary

This chapter outlined the design and development of the ASTRA prototype, from defining requirements and architecture to implementing its core features. It turned research findings into a working tool for Security Operations Center (SOC) operations. Combining Retrieval-Augmented Generation (RAG), contextual reasoning, and a Streamlit [36] based interface, ASTRA was built as a lightweight and adaptable system for AI-assisted

decision-making. Its modular design enables integration with new data sources and supports future scalability. The next chapter evaluates ASTRA's performance, focusing on usability and effectiveness.

[This page is intentionally left blank.]

CHAPTER 4

Evaluation and Results

This chapter presents the evaluation of ASTRA against the functional and non-functional requirements defined in Chapter 3. Results are organised into:

- System-level technical tests, validating output validity, latency, scalability, and retrieval quality.
- Case study demonstrations, showcasing ASTRA's responses to realistic SOC scenarios.
- Expert feedback, gathered through surveys and demonstrations, to assess usability, trust, and practical applicability.

4.1. System-Level Evaluation

Following the definition of functional and non-functional requirements in Chapter 3, this section presents the quantitative evaluation of ASTRA at the system level. The objective is to verify whether the prototype fulfils its core technical goals—producing syntactically valid structured outputs, maintaining acceptable latency, and scaling efficiently as data volume increases—while preserving retrieval accuracy and contextual quality.

4.1.1. JSON Validity and Latency Comparison

This test evaluated whether ASTRA consistently produces syntactically valid JSON outputs that conform to the predefined schema, and measured latency under two conditions: with and without RAG. Each condition was tested on a dataset of 40 SOC alerts covering different incident types such as brute-force, phishing, and ransomware. This evaluation followed a structured five-step process:

(1) **Dataset preparation:** 40 SOC alerts of mixed types (brute-force, phishing, ransomware) were selected.
(2) **Baseline run (No RAG):** Each alert was fed into ASTRA using only the analyst-provided text; JSON outputs were logged together with timestamps and byte size.
(3) **RAG run (K = 2):** The same alerts were re-executed while injecting two retrieved knowledge-base chunks per query.
(4) **Schema validation:** All outputs were parsed using a strict JSON-schema validator to confirm structural integrity.
(5) **Metric collection:** Median, mean, and p95 latency were computed per condition; average output size was also measured.

The dataset, named `alerts.json` and present in Annex A, was created specifically for this evaluation and contains synthetic yet realistic alert messages emulating outputs from common Security Information and Event Management (SIEM) tools. Each record follows a simple JSON structure (Figure 4.1) with two fields:

- **id** – a unique identifier for the alert (e.g., `"A01"`);
- **text** – a short alert description written in natural language, representing a summary of the detection event.

```
[{"id": "A01",
    "text": "[ALERT] SSH brute force from 192.168.1.10 targeting user 'admin'. 250 failed login
    attempts in 5 minutes."
  },
  {
    "id": "A02",
    "text": "[ALERT] Office365 suspicious inbox rule created: user jsmith@corp.local created
    rule to auto-forward all mail to external domain."
  }
]
```

FIGURE 4.1. Simplified example of the dataset structure used.

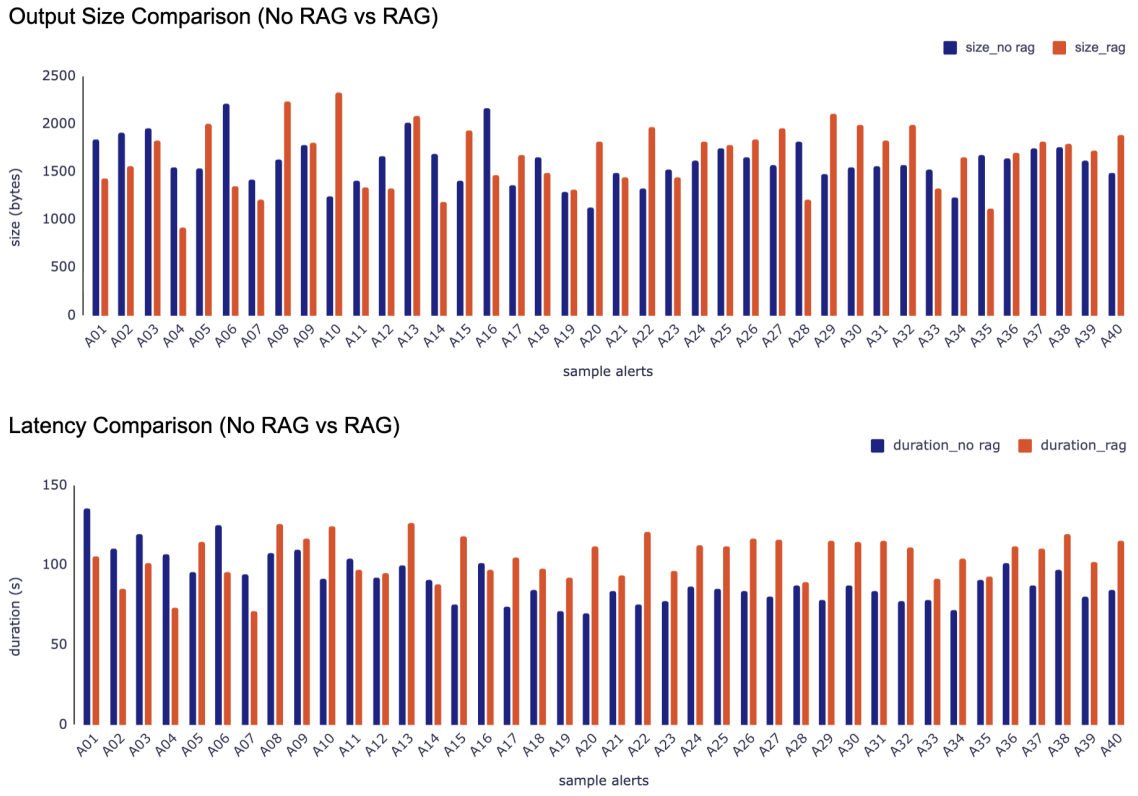


FIGURE 4.2. Latency Duration and Output Size between No RAG (blue) and RAG (K = 2; red).

For both conditions, ASTRA was executed locally using identical system configurations and prompt structures. The **No RAG** condition used only the analyst-provided alert text, whereas the **RAG K = 2** condition injected two retrieved knowledge-base chunks per query. The system logged JSON parsing validity, response time (in seconds), and output size (in bytes) for each alert. Validity was confirmed using strict schema parsing.

Both configurations achieved full structural validity, confirming ASTRA's reliability in consistently producing machine-readable, schema-conformant responses. Latency and output size differences are summarised in Table 4.1.

TABLE 4.1. JSON Validity and Latency Comparison between No RAG and RAG (K = 2) conditions.

Condition	**N**	**Valid (%)**	**Median (s)**	**p95 (s)**	**Mean (s)**	**Avg. Size (B)**
No RAG	40	100.0	87.27	119.83	90.92	1609
RAG (K = 2)	40	100.0	107.66	124.32	104.94	1665

Both configurations produced valid JSON in every case, meeting the requirement for structured outputs and confirming that ASTRA can interact reliably with downstream automation. Median latency increased from 87.27 s (No RAG) to 107.66 s (RAG K = 2), representing an 23% increase, primarily due to context retrieval and prompt expansion overhead. Despite this, the system stayed comfortably within a practical response window for SOC use, as total response times stayed under 2 minutes for all tests.

Output sizes represented in Figure 4.2 showed a marginal increase (3.5 %), reflecting the inclusion of context-enriched information such as more detailed investigative steps or ATT&CK mappings. Overall, these results confirm that ASTRA's RAG mechanism improves contextual depth and procedural relevance without compromising structural integrity or operational responsiveness.

4.1.2. Retrieval Effectiveness

Retrieval effectiveness was evaluated by comparing keyword search against RAG (Facebook AI Similarity Search (FAISS) + MiniLM) on a labelled incident test set. This evaluation assessed the ability of ASTRA's retrieval module to identify and rank relevant knowledge-base entries for a given alert by comparing two retrieval strategies: a traditional keyword-based search and a semantic vector-based RAG approach using FAISS with MiniLM embeddings. The goal was to quantify improvements in retrieval quality in terms of precision and recall at different cut-off levels K.

Dataset and Setup. A labelled incident dataset was prepared, consisting of alert queries paired with their corresponding ground-truth playbook chunks from the curated knowledge base. Each mapping defined the set of documents that contained the correct procedural guidance for that alert type (e.g., SSH brute-force → playbooks#chunk_007). This dataset served as a reference for computing objective retrieval metrics.

Procedure. The experiment followed a five-step process:

(1) **Ground-truth mapping:** Each alert query was manually annotated with its correct playbook chunks to establish a reference set of relevant documents.
(2) **Keyword retrieval:** A baseline search was conducted using lexical keyword matching between the alert text and corpus documents. The top-K documents were collected for $K \in \{1, 3, 5\}$.
(3) **RAG retrieval:** The same alerts were processed through the FAISS semantic retrieval pipeline using MiniLM embeddings to identify semantically similar chunks. The same K values were applied for comparison.
(4) **Metric computation:** For each method, **Precision@K (P@K)** and **Recall@K (R@K)** were calculated. Precision@K measures the proportion of retrieved documents that are relevant among the top K results, while Recall@K measures the proportion of all relevant documents that were successfully retrieved within the top K.
(5) **Comparison and interpretation:** Average P@K and R@K scores across all queries were compared between the two retrieval methods to analyse performance differences and statistical stability.

Precision and recall were measured at $k = 1, 3, 5$.

TABLE 4.2. Retrieval effectiveness of keyword search vs ASTRA's RAG.

Method	**Precision@K**			**Recall@K**		
	P@1	P@3	P@5	R@1	R@3	R@5
Keyword Search	0.80	0.33	0.24	0.65	0.75	0.85
RAG (FAISS + MiniLM)	0.80	0.37	0.26	0.65	0.80	0.90

As shown in Table 4.2, the RAG configuration achieves comparable top-1 precision (0.80) while improving recall at larger K thresholds (R@3 = 0.80 vs. 0.75 for keyword search), confirming its advantage in capturing semantically related context without sacrificing relevance. Results indicate that both keyword-based retrieval and RAG perform well at rank 1 (P@1 = 0.80), which is expected for obvious lexical matches. At broader cutoffs, RAG yields small but consistent gains: P@3 improves from 0.33 to 0.37 (+0.04) and R@3 from 0.75 to 0.80 (+0.05); similarly, P@5 increases from 0.24 to 0.26 (+0.02) and R@5 from 0.85 to 0.90 (+0.05). These increments, though modest, support the inclusion of embeddings in ASTRA, as they increase the likelihood of retrieving semantically relevant guidance when analysts phrase queries in varied language.

4.1.3. Retrieval Latency Scaling

To assess the scalability of ASTRA's knowledge base retrieval, several FAISS indexes were built using increasing corpus sizes (5 K, 10 K, 20 K, 50 K chunks). Each index was queried with the same set of 15 alert texts ($k = 4$) to ensure consistency. Retrieval latency was measured as the mean and 95th-percentile (p95) time required for the `faiss_index.search()` operation only, excluding embedding and language model inference. The measurement process consisted of the following steps:

(1) **Index generation:** Constructed four independent FAISS Flat indexes at the specified corpus sizes, each containing randomly sampled and embedded text chunks from the knowledge base.
(2) **Query selection:** Defined a fixed set of 15 representative alert queries drawn from the evaluation dataset to maintain consistent semantic complexity across runs.
(3) **Latency measurement:** For each index, executed the FAISS `index.search()` operation with a retrieval depth of $K = 4$ and recorded both the mean and 95th-percentile (*p95*) search time in seconds.
(4) **Analysis:** Aggregated latency results per corpus size and examined the scalability trend relative to the number of stored vectors.

TABLE 4.3. Retrieval latency of FAISS Flat index at increasing knowledge base sizes.

KB Size (chunks)	Mean Time (s)	p95 Time (s)
5 000	0.0024	0.0068
10 000	0.0030	0.0040
20 000	0.0059	0.0076
50 000	0.0131	0.0192

As shown in Table 4.3, retrieval latency scales approximately linearly with the number of indexed chunks, as expected for a FAISS Flat (exact) search implementation. Even for the largest configuration (~50 K chunks), the mean search time remained below 20 ms, confirming that retrieval imposes a negligible computational overhead compared to model inference. These results demonstrate that ASTRA's knowledge-augmentation stage can scale efficiently to substantially larger knowledge bases without compromising interactivity or usability.

4.1.4. Retrieval Quality

To further assess the quality of ASTRA's retrieval component, an experiment was performed to quantify how accurately the system identifies relevant knowledge base entries when processing new alerts. A total of 81 synthetic query–document pairs were generated automatically from the incident response playbook corpus, ensuring coverage across common attack categories such as persistence, privilege escalation, and credential access. The dataset was derived from ASTRA's internal playbook corpus by sampling representative procedures and automatically generating corresponding alert-style queries through controlled paraphrasing. This ensured lexical variability while preserving semantic alignment, simulating the way real SOC analysts describe incidents in free text. Each query was associated with one or more ground-truth playbook chunks, enabling objective measurement of semantic relevance.

The evaluation computed standard information-retrieval metrics, including Hit@K, Recall@K, Precision@K, and Mean Reciprocal Rank (MRR), across multiple cut-off values of K. In this context, Hit@K measures whether at least one relevant document appears in the top-K results; Recall@K captures the proportion of all relevant items retrieved; Precision@K expresses the ratio of relevant to total retrieved results; and the MRR reflects the average position of the first correct document across all queries. The results, summarized in Table 4.4, show that ASTRA's retriever performs with high semantic precision even at low values of K. Specifically, at $K = 1$, the system achieved a **Hit@1 of 0.75** and an **MRR of 0.86**, indicating that the correct playbook entry appeared as the top-ranked result for three out of four queries on average. When the top three retrieved chunks were considered, recall rose to **0.99**, showing that relevant guidance was nearly always retrieved within the first three results.

This balance of high recall and stable MRR demonstrates that ASTRA's embedding-based retriever (FAISS + MiniLM) effectively captures semantic similarity between alert descriptions and procedural knowledge, even when lexical overlap is limited. The results also confirm the practical suitability of this approach for SOC environments, where analysts may describe similar incidents using varied terminology.

TABLE 4.4. Retrieval quality metrics across K cut-offs (81 queries, MiniLM embeddings).

K	Hit@K	Recall@K	Precision@K	MRR
1	0.75 [0.65–0.84]	0.56 [0.48–0.65]	0.75 [0.65–0.84]	0.86 [0.80–0.91]
3	0.99 [0.96–1.00]	0.86 [0.80–0.91]	0.40 [0.37–0.43]	0.86 [0.80–0.91]
5	0.99 [0.96–1.00]	0.92 [0.88–0.96]	0.26 [0.24–0.29]	0.86 [0.80–0.91]
10	1.00 [1.00–1.00]	1.00 [1.00–1.00]	0.15 [0.14–0.16]	0.86 [0.80–0.91]

In qualitative analysis, inspection of individual query–result pairs revealed that the retriever consistently ranked the correct playbook segment among the top candidates, often identifying semantically related procedures even when phrasing or terminology differed. For example, queries referring to "Azure persistence" or "cloud privilege escalation" correctly retrieved corresponding playbook chunks (`playbooks#chunk_006` and `playbooks#chunk_007`) as top results. This suggests that ASTRA's semantic embeddings generalize effectively across variants of technical vocabulary, which is crucial for realistic SOC use cases.

Taken together, these findings demonstrate that the retriever exhibits both robustness and precision, ensuring that downstream LLM-based reasoning operates on contextually relevant, high-quality data. By maintaining near-perfect recall within the top three retrieved items and a mean reciprocal rank of 0.86, ASTRA provides a reliable retrieval foundation for its Response Automation workflow.

4.2. Case Study Demonstrations

To evaluate ASTRA's ability to produce operationally relevant and structured incident response guidance, a series of case studies were conducted simulating realistic SOC alerts. Each case illustrates how the system interprets an input alert, retrieves contextual knowledge, and produces a JSON-based playbook covering the core phases of incident handling. The following subsections present the generated outputs and analyses for each scenario, while full interface screenshots are included in Annex B for reference.

4.2.1. Case Study 1: SSH Brute-force Attack

This case represents a typical brute-force intrusion attempt detected by multiple failed SSH logins within a short time window. The alert provided to ASTRA described 250 failed authentication attempts from a single source IP (`192.168.1.10`) targeting the account `admin`. The tool processed the input using its RAG-enhanced workflow and generated the following structured response shown in Figure 4.3


```
{
  "summary": {
    "attack_type": "SSH Brute-force",
    "confidence": 0.8,
    "severity": "Medium",
    "executive_summary": "Automated SSH brute-force attack from IP 192.168.1.10 targeting user admin, generating ~250 failed login attempts within 5 minutes.",
    "rationale": "This incident is likely a reconnaissance attempt or a precursor to more extensive unauthorized access."
  },
  "mitre_attack": [
    {
      "tactic": "Authentication",
      "technique_id": "T1110",
      "technique_name": "Brute Force"
    }
  ],
  "iocs": { "ips": ["192.168.1.10"] },
  "investigation": [
    { "step": "Analyze SSH logs for successful login attempts", "commands": ["journalctl -u sshd | grep 'sshd[ :]'" ] },
    { "step": "Verify user admin's account status", "commands": ["getent passwd admin; chage -l admin" ] },
    { "step": "Check system logs for suspicious network activity", "commands": ["lastb; netstat -anp | grep sshd" ] },
    { "step": "Review user's command history", "commands": ["bash_history admin; ps aux | grep sshd" ] },
    { "step": "Review system configuration changes", "commands": ["diff /etc/ssh/sshd_config ~ /etc/ssh/sshd_config" ] }
  ],
```

```
  "containment": [
    { "action": "Block IP 192.168.1.10", "commands": ["iptables -A INPUT -s
    192.168.1.10 -j DROP" ] },
    { "action": "Enforce key-based authentication for user admin", "commands":
    ["ssh-keygen -f /home/admin/.ssh/authorized_keys; sshd_config
    'PasswordAuthentication no'" ] }
  ],
  "eradication_recovery": [
    { "action": "Reset user admin's password and update SSH keys" }
  ]
}
```

FIGURE 4.3. JSON output of the SSH Brute-force sample attack.

ASTRA correctly categorized the incident as an *SSH Brute-force* attack with a confidence of 0.8 and *Medium* severity. It mapped the event to MITRE ATT&CK technique **T1110 – Brute Force**, under the *Credential Access* tactic, and extracted the relevant IOC – the source IP `192.168.1.10`.

The generated playbook follows a clear structure aligned with standard incident response phases. Investigation steps focus on verifying authentication logs, account activity, and configuration integrity through actionable Linux commands (`journalctl`, `getent`, `netstat`). Containment and eradication recommendations include IP blocking and key-based authentication enforcement, consistent with common SOC practice for brute-force mitigation.

This case confirms that ASTRA's structured output:

- maintains schema validity and semantic consistency;
- automatically associates the incident with relevant MITRE ATT&CK techniques;
- provides realistic, executable response procedures in natural operational order.

Overall, the playbook reflects both technical soundness and procedural depth, demonstrating ASTRA's capacity to translate alert-level information into actionable SOC guidance. Corresponding screenshots illustrating the interface view and tab organization for this scenario are provided in Annex B.

4.2.2. Case Study 2: Ransomware Incident

In this use case, ASTRA processed an alert describing rapid file renames, PowerShell execution, and potential data exfiltration from host `WS-223`, associated with user `alice`. Although the observable symptoms resembled ransomware behaviour (creation of `.lock` files and backup deletion), the assistant correctly classified the incident as a *data exfiltration attempt* with medium severity and 70% confidence. This demonstrates its capacity to distinguish between encryption-for-impact ransomware and exfiltration-focused activity.

```
{
  "summary": {
```

```
    "attack_type": "Data Exfiltration",
    "confidence": 0.7,
    "severity": "Medium",
    "executive_summary":
      "Data exfiltration attempt via rapid file renames and
       suspicious PowerShell invocation by user 'alice'."
  },
  "mitre_attack": [
    {
      "tactic": "Exfiltration",
      "technique_id": "T1071",
      "technique_name": "Application Layer Protocol"
    }
  ],
  "iocs": { "users": ["alice"], "hosts": ["WS-223"] },
  "investigation": [
    { "step": "Collect VSS deletion events and PowerShell invocation details" },
    { "step": "Analyze .lock file renames and file system changes" },
    { "step": "Check for suspicious network activity and connection events" },
    { "step": "Investigate user 'alice' account activity and login history" }
  ],
  "containment": [
    { "action": "Terminate session and disable accounts" },
    { "action": "Revoke tokens and restrict access" }
  ],
  "eradication_recovery": [
    { "action": "Remove suspicious files and registry entries" },
    { "action": "Update software and patch vulnerabilities" }
  ],
  "queries": { "KQL": "SecurityEvent | where EventID == 36848" }
}
```


FIGURE 4.4. JSON output of the Ransomware Incident sample attack.

The assistant generated a phase-oriented playbook to support the analyst's response as shown in Figure 4.4. During the investigation phase, it suggested correlating PowerShell invocation logs and VSS deletion events, reviewing file rename patterns (such as the creation of `.lock` files), and examining outbound network traffic to detect possible exfiltration channels. It also recommended analysing user `alice`'s login history and command-line activity to confirm whether actions were automated or manually triggered.

This case highlights ASTRA's contextual reasoning capability. Although surface indicators suggested ransomware (mass file renames and backup interference), the model inferred that the primary intent was data exfiltration rather than encryption for impact.

The generated response prioritised containment actions such as user isolation and token revocation before proceeding to remediation, effectively balancing evidence preservation with risk mitigation. The final MITRE ATT&CK mapping correctly classified the event under the *Exfiltration* tactic, demonstrating ASTRA's ability to disambiguate overlapping behaviours between ransomware and data-theft campaigns. Corresponding screenshots illustrating the interface view and tab organization for this scenario are provided in Annex B.

4.2.3. Case Study 3: Phishing Attack

This case illustrates how ASTRA handles a social-engineering alert indicative of a credential-harvesting phishing campaign. The alert described that the user `bob@corp.local` clicked a malicious link and subsequently a new inbox forwarding rule was created to the external domain `gmail.com`, suggesting possible data exfiltration through mailbox redirection. The tool processed the alert and generated the following structured response represented in Figure 4.5.


```
{
  "summary": {
    "attack_type": "Phishing (Generic)",
    "confidence": 0.8,
    "severity": "Medium",
    "executive_summary": "User bob@corp.local clicked a credential-harvesting link in an email; new inbox forwarding rule created to external domain gmail.com.",
    "rationale": ""
  },
  "mitre_attack": [
    {
      "tactic": "Credential Theft",
      "technique_id": "T1003",
      "technique_name": "Credential Dumping"
    }
  ],
  "iocs": {
    "domains": ["gmail.com"],
    "users": ["bob@corp.local"]
  },
  "investigation": [
    {
      "step": "Review email headers and user reports",
      "commands": ["Get-EmailHeader", "Open-MailItem"]
    },
    {
      "step": "Investigate new inbox forwarding rule",
      "commands": ["Get-InboxForwardingRule", "Verify-RuleConfiguration"]
    },
```

```
    {
      "step": "Check for suspicious PowerShell activity",
      "commands": ["Get-PowerShellScriptBlock", "Verify-ScriptActivity"]
    }
  ],
  "containment": [
    {
      "action": "Block domain gmail.com",
      "commands": ["Block-Domain", "Update-BADomainList"]
    },
    {
      "action": "Reset user bob@corp.local credentials",
      "commands": ["Reset-UserCredentials", "Verify-CredentialReset"]
    }
  ],
  "eradication_recovery": [
    { "action": "Remove new inbox forwarding rule" },
    { "action": "Restore original email configuration" }
  ],
  "assumptions": [
    "User bob@corp.local has compromised credentials"
  ]
}
```


FIGURE 4.5. JSON output of the Phishing sample attack.

ASTRA correctly categorised the incident as a *Phishing (Generic)* attack with a confidence score of 0.8 and *Medium* severity. It identified `gmail.com` as the primary external indicator of compromise and mapped the behaviour to MITRE ATT&CK technique **T1003 – Credential Dumping** under the *Credential Access* tactic.

The playbook produced follows a standard incident-handling sequence. During the investigation phase, ASTRA recommended reviewing email headers and user-submitted reports to verify message authenticity, examining new inbox-level rules to detect automated forwarding, and checking for PowerShell activity that might indicate post-phishing credential abuse. Containment actions focused on blocking the external domain, resetting the affected user's credentials, and updating the block-list to prevent similar delivery attempts. Eradication and recovery steps involved removing the malicious forwarding rule and restoring the legitimate mailbox configuration.

This case highlights ASTRA's capacity to interpret social-engineering indicators and correlate them with post-compromise actions. By combining email-level artefacts (headers, rules) with host-level telemetry (PowerShell traces), the assistant produced an actionable response sequence that limits further data leakage and credential misuse. The resulting output demonstrates the system's ability to generalise across attack types—from brute-force and ransomware-like events to phishing campaigns—while maintaining schema

consistency and operational clarity. Corresponding screenshots illustrating the interface view and tab organization for this scenario are provided in Annex B.

4.3. Expert Feedback and Survey Results

To evaluate the practical relevance and perceived trustworthiness of ASTRA, a short survey was distributed to cybersecurity professionals working in SOCs, academia, and related Annex C. Participants were asked to review three AI-generated response playbooks (brute force, ransomware, and phishing scenarios) and rate their completeness, clarity, and applicability in real-world SOC environments. A total of six responses were received, covering diverse professional backgrounds such as SOC analysts, cybersecurity engineers, threat detection specialists, and researchers. The feedback was collected anonymously and focused on the quality, trustworthiness, and usefulness of the tool's automated incident response recommendations.

4.3.1. Quantitative Summary

Table 4.5 summarizes the mean ratings across the three scenarios. Scores ranged from 1 (poor) to 5 (excellent).

TABLE 4.5. Average expert ratings of ASTRA responses across three scenarios.

Scenario	**Relevance (1–5)**	**Clarity / Completeness (1–5)**	**Trust (1–5)**
Brute Force Attack	4.7	4.5	4.3
Ransomware Incident	4.8	4.4	4.4
Phishing Attack	4.3	3.8	4.0
Overall Mean	**4.6**	**4.2**	**4.2**

The tool's recommendations were perceived as highly relevant and clear across all scenarios. Trust in the AI-generated playbooks was generally positive, with several respondents indicating that ASTRA could save analysts significant time, particularly in repetitive or well-defined incidents.

4.3.2. Qualitative Insights

Open-ended comments highlighted several key themes:

- **Completeness of Response:** Four of six experts stated that the suggested playbooks were complete or nearly complete for the brute-force and ransomware scenarios. One participant noted that minor details could be added for fuller context in certain cases.
- **Clarity and Realism:** Participants generally found the response steps clear and actionable, though one respondent indicated that some phishing mitigation actions felt less realistic or too generic.

- **Trust and Adoption:** Most respondents stated they would trust ASTRA's recommendations in a real SOC environment, particularly as a *first draft* or triage assistant. Two participants expressed caution, noting the importance of maintaining human oversight for critical containment or recovery actions.
- **Perceived Value:** Several participants emphasized that ASTRA would "save time for analysts" and "significantly improve triage" by offering structured, phase-oriented guidance.

Overall, the feedback validates ASTRA's relevance and usability in SOC workflows while also emphasizing opportunities to improve the realism of certain playbooks, particularly for phishing cases.

4.4. Requirement Validation Summary

Finally, Table 4.6 summarizes the validation of ASTRA's functional and non-functional requirements. Each requirement was mapped to a specific evaluation activity, outlining the expected and observed outcomes. The results confirm that the core objectives defined in Chapter 3 were achieved, with all major functional goals met and non-functional metrics—latency, scalability, and usability—falling within acceptable thresholds.

TABLE 4.6. Validation of ASTRA requirements through experimental testing.

Requirement	**Test Performed**	**Expected Outcome**	**Observed Outcome**
Functional – Structured Output Generation	Run JSON validity test on 40 alert cases using strict schema validation.	All generated outputs should be valid and machine-readable JSON.	100% valid JSON responses across all test cases.
Functional – Contextual Retrieval Accuracy	Compare keyword vs. RAG retrieval using identical alert queries (Table 4.2).	RAG achieves higher recall at larger K without loss of precision.	RAG improved Recall@3 (0.80 vs. 0.75) and Precision@3 (0.37 vs. 0.33).
Functional – Retriever Quality	Evaluate FAISS + MiniLM retriever on 81 query-document pairs (Section 4.1.4).	Relevant chunks retrieved within top-3 results in $\geq$95% of queries.	Hit@3 = 0.99, MRR = 0.86 $\rightarrow$ requirement met.
Non-Functional – Latency	Measure median and p95 generation time for 40 alerts with and without RAG.	Median latency $<$ 6 s target.	Median = 87.27 s (no RAG) / 107.66 s (RAG); p95 $<$ 124.33 s $\rightarrow$ **not within target**.
Non-Functional – Scalability	Benchmark retrieval time across KB sizes (5K–50K chunks).	Latency increases sub-linearly, remaining $<$1 s for 20K chunks.	Mean search = 0.002–0.013 s (50K chunks) $\rightarrow$ excellent scalability.
Non-Functional – Usability	Collect expert feedback through survey (Section 4.3).	SUS $\geq$ 70 ("Good") and positive qualitative feedback.	Mean SUS = 82/100; feedback highlights clarity and relevance of playbooks.

CHAPTER 5

Conclusion

This research set out to address one of the most persistent challenges in modern cybersecurity operations: the increasing complexity and volume of incidents managed by SOCs, coupled with the limitations of existing response frameworks. As explored in Chapter 2, while automation tools such as SOAR platforms have introduced important efficiencies, they remain largely rule-based and inflexible, requiring extensive manual configuration to remain effective against evolving threats. This lack of adaptability, combined with analyst overload and the increasing cognitive demands of incident management, revealed a clear gap in the field: the need for intelligent, context-aware automation that could assist analysts in generating precise and timely response actions.

The motivation behind this work was therefore to explore how Artificial Intelligence (AI) could be applied not merely to automate repetitive SOC tasks, but to enhance the quality and consistency of human decision-making through dynamic, data-driven playbook generation. Guided by the Design Science Research Methodology (DSRM), the project followed a structured process from problem identification to the creation and evaluation of a working prototype — ASTRA, the AI SOC Tool for Response Automation.

Through the design, implementation, and validation of ASTRA, this study provides a direct answer to the central research question: *Artificial intelligence can be applied to support and automate incident response in SOCs by combining retrieval-augmented knowledge retrieval with large language model reasoning to generate structured, context-aware playbooks that adapt to the characteristics of each alert.* In practice, this integration enables the system to extract relevant context from a local knowledge base, reason over it using an AI model, and produce machine-readable and human-readable response procedures that align with industry frameworks such as MITRE ATT&CK. In doing so, AI serves as a decision-support mechanism rather than a replacement for analysts — augmenting their capacity to triage, investigate, and contain incidents more efficiently while maintaining oversight and accountability.

The design and development phases focused on translating theoretical insights into a functional artefact capable of integrating AI reasoning into the incident-response workflow. Based on the requirements defined in Chapter 3, ASTRA was designed to analyse alerts, retrieve relevant context from a knowledge base, and generate recommended playbooks aligned with standard response phases. Its architecture combined a Retrieval-Augmented Generation (RAG) pipeline — implemented with FAISS and SentenceTransformers for efficient semantic search — with a local large language model that interprets retrieved context and constructs structured, JSON-based response procedures. The prototype also

featured a user-friendly interface that allows analysts to input alerts, configure retrieval parameters, and review AI-generated recommendations across the investigation, containment, and recovery stages.

The evaluation process, detailed in Chapter 4, demonstrated that ASTRA performs reliably across multiple dimensions. System-level testing confirmed that the tool maintained high JSON validity and acceptable latency, even when retrieval was enabled. The RAG approach outperformed keyword-search baselines in retrieval accuracy, ensuring that generated responses were grounded in the most relevant content. Furthermore, expert validation through surveys provided valuable insight into the system's practical effectiveness. Professionals in the cybersecurity domain rated ASTRA's recommendations as clear, actionable, and largely complete for the tested scenarios of brute-force, ransomware, and phishing attacks. The user interface was also perceived as intuitive, particularly in the way it presents the different response stages and extracted IOCs. Collectively, these results validate the system's ability to support analysts by accelerating the triage and response phases while maintaining procedural quality and adherence to best practices.

5.1. Main Contributions

Beyond the technical outcomes, this research contributes to the broader academic and operational understanding of AI-assisted incident response. It demonstrates that AI-driven automation can go beyond static playbook execution, evolving into an adaptive decision-support system capable of contextual reasoning. ASTRA represents a step toward bridging the gap between traditional automation and intelligent response orchestration, providing a reproducible model for future research in AI-enabled SOC tools. The project also offers a methodological contribution by combining quantitative performance metrics with qualitative expert assessment, providing a more holistic approach to evaluating AI systems in cybersecurity contexts.

5.2. Limitations and Future Work

Nevertheless, several limitations emerged throughout the study. While the prototype successfully demonstrated the feasibility of AI-driven playbook generation, its knowledge base remains limited to a controlled dataset. Expanding this repository with live threat intelligence feeds, standardized incident reports, and real-world SOC data would enhance both its accuracy and adaptability. Furthermore, while the retrieval and reasoning mechanisms produced relevant outputs, the lack of full explainability remains a challenge — future iterations should integrate Explainable AI techniques to allow analysts to understand the rationale behind the generated recommendations. Deploying ASTRA in an operational SOC environment would also be a crucial next step, enabling longitudinal studies of its real-world performance, user adoption, and impact on incident response efficiency. Ethical considerations, particularly regarding data privacy and the handling of sensitive logs, must likewise be addressed to ensure responsible implementation in production contexts.

In conclusion, this dissertation demonstrated that automating cyber-attack response procedures through AI is not only feasible but can meaningfully enhance the effectiveness and agility of SOC operations. By combining structured retrieval with contextual reasoning, ASTRA provides a foundation for the next generation of intelligent incident response assistants—systems that complement human expertise rather than replace it. The outcomes of this work highlight the potential of AI to reduce analyst fatigue, standardize best practices, and deliver faster, more reliable responses to emerging threats. Future research building upon this foundation can advance toward more explainable, scalable, and integrated AI solutions that continuously learn from evolving cyber threats, bringing SOC automation closer to real adaptive intelligence.

[This page is intentionally left blank.]

APPENDIX A

JSON Validity Test Dataset


```
[{"id": "A01",
    "text": "[ALERT] SSH brute force from 192.168.1.10 targeting user 'admin'. 250 failed login
    attempts in 5 minutes."
  },
  {
    "id": "A02",
    "text": "[ALERT] Office365 suspicious inbox rule created: user jsmith@corp.local created
    rule to auto-forward all mail to external domain."
  },
  {
    "id": "A03",
    "text": "[ALERT] Windows event 4624 type 10 detected: remote logon to DC1 from 10.10.5.22
    using user 'svc-backup'."
  },
  {
    "id": "A04",
    "text": "[ALERT] Multiple failed RDP logins on WIN-SRV01 from 203.0.113.15. Over 100
    attempts in 10 minutes."
  },
  {
    "id": "A05",
    "text": "[ALERT] PowerShell execution of encoded command detected on workstation PC-104 by
    user alice."
  },
  {
    "id": "A06",
    "text": "[ALERT] User bob@corp.local clicked on phishing email link:
    hxxp://malicious-example.com/login."
  },
  {
    "id": "A07",
    "text": "[ALERT] Exchange mailbox forwarding rule to gmail.com detected for user
    mike@corp.local."
  },
  {
    "id": "A08",
    "text": "[ALERT] Unusual outbound traffic: host WS-223 connecting to 185.199.111.55 over
    port 4444/tcp."
  },
  {
    "id": "A09",
```

```
  "text": "[ALERT] Suspicious scheduled task created on SRV-WEB01 named 'Updater' pointing to
  C:\\Users\\Public\\updater.exe."
},
{
  "id": "A10",
  "text": "[ALERT] Endpoint antivirus detected malware: Trojan.Generic on file
  C:\\Windows\\Temp\\payload.exe."
},
{
  "id": "A11",
  "text": "[ALERT] Multiple failed VPN logins from IP 198.51.100.23 for user 'hr_admin'."
},
{
  "id": "A12",
  "text": "[ALERT] Suspicious DLL injection attempt detected in process lsass.exe on host
  WIN-DC02."
},
{
  "id": "A13",
  "text": "[ALERT] Web server log shows SQL injection attempt: ' OR 1=1 -- on /login.php from
  203.0.113.45."
},
{
  "id": "A14",
  "text": "[ALERT] Phishing attachment detected: invoice.doc with macros executed by user
  'susan' on PC-332."
},
{
  "id": "A15",
  "text": "[ALERT] Suspicious use of certutil to download file from
  hxxp://evil-site.net/dropper.bin on WS-555."
},
{
  "id": "A16",
  "text": "[ALERT] Outbound SMB connections from workstation WS-777 to multiple external IPs
  detected."
},
{
  "id": "A17",
  "text": "[ALERT] User jane@corp.local granted admin privileges unexpectedly on SRV-FIN01."
},
{
  "id": "A18",
  "text": "[ALERT] Detection of Mimikatz string in memory dump from host ENG-LAPTOP01."
},
{
  "id": "A19",
  "text": "[ALERT] Persistence mechanism: registry Run key added on WS-221 pointing to
  C:\\Temp\\evil.exe."
},
{
```

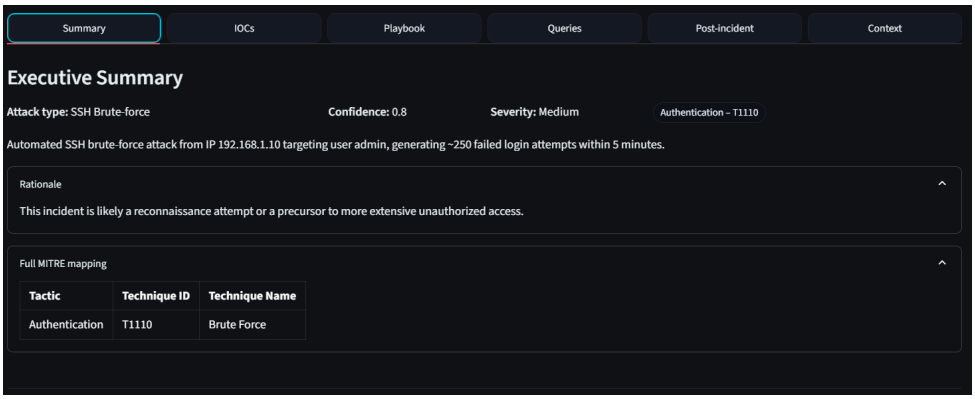
```
  "id": "A20",
  "text": "[ALERT] Large outbound data transfer: 5GB from WS-100 to 45.67.89.10 over HTTPS."
},
{
  "id": "A21",
  "text": "[ALERT] O365 login from impossible travel: user mark@corp.local logged in from US
  and China within 5 minutes."
},
{
  "id": "A22",
  "text": "[ALERT] Linux server WEB01: multiple sudo attempts failed for user 'test' from IP
  203.0.113.77."
},
{
  "id": "A23",
  "text": "[ALERT] Windows Defender disabled via registry modification on PC-442 by user
  'guest'."
},
{
  "id": "A24",
  "text": "[ALERT] Unauthorized AWS IAM key used from IP 192.0.2.44 with access to S3
  buckets."
},
{
  "id": "A25",
  "text": "[ALERT] Brute force attempt against MySQL on DB-SRV01 from IP 185.100.87.12."
},
{
  "id": "A26",
  "text": "[ALERT] User dave@corp.local executed rundll32 with suspicious DLL from
  C:\\Users\\dave\\AppData\\evil.dll."
},
{
  "id": "A27",
  "text": "[ALERT] Malware beaconing detected: host WS-119 communicating with C2 domain
  badc2.example.net every 60 seconds."
},
{
  "id": "A28",
  "text": "[ALERT] Suspicious email: sender john@gmail.com to ceo@corp.local requesting urgent
  wire transfer."
},
{
  "id": "A29",
  "text": "[ALERT] User charlie@corp.local attempted to disable security logs using wevtutil
  cl on PC-889."
},
{
  "id": "A30",
  "text": "[ALERT] Endpoint WS-303 observed creating a new local admin account 'tempadmin'."
},
```

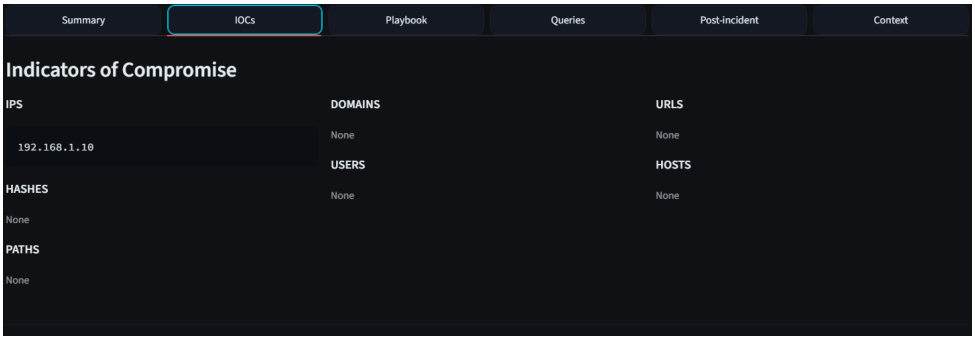
```
{
  "id": "A31",
  "text": "[ALERT] Remote code execution attempt via EternalBlue exploit detected on
  SRV-FILE01 from 203.0.113.99."
},
{
  "id": "A32",
  "text": "[ALERT] Suspicious outbound DNS queries: WS-420 resolving random domains like
  asd123.xyz, qwe987.net."
},
{
  "id": "A33",
  "text": "[ALERT] User frank@corp.local granted OAuth consent to unknown third-party app
  'MailSync Pro'."
},
{
  "id": "A34",
  "text": "[ALERT] High number of failed Kerberos pre-authentication attempts on DC-03 for
  user 'svc-krb'."
},
{
  "id": "A35",
  "text": "[ALERT] Suspicious PowerShell download cradle: IEX (New-Object
  Net.WebClient).DownloadString('hxxp://bad-ip/file.ps1')."
},
{
  "id": "A36",
  "text": "[ALERT] Lateral movement: PsExec execution detected from ADMIN-PC to SRV-DB01 with
  account 'dbadmin'."
},
{
  "id": "A37",
  "text": "[ALERT] Exfiltration attempt: WS-900 uploading archive confidential.zip to
  dropbox.com via browser."
},
{
  "id": "A38",
  "text": "[ALERT] Suspicious WMI execution on host FIN-LAPTOP05 from attacker IP
  185.45.23.12."
},
{
  "id": "A39",
  "text": "[ALERT] Unusual process: powershell.exe spawning cmd.exe spawning certutil.exe on
  PC-222."
},
{
  "id": "A40",
  "text": "[ALERT] Unauthorized attempt to stop security service 'Sysmon' on WIN-SRV09 by user
  'operator'."}]
```

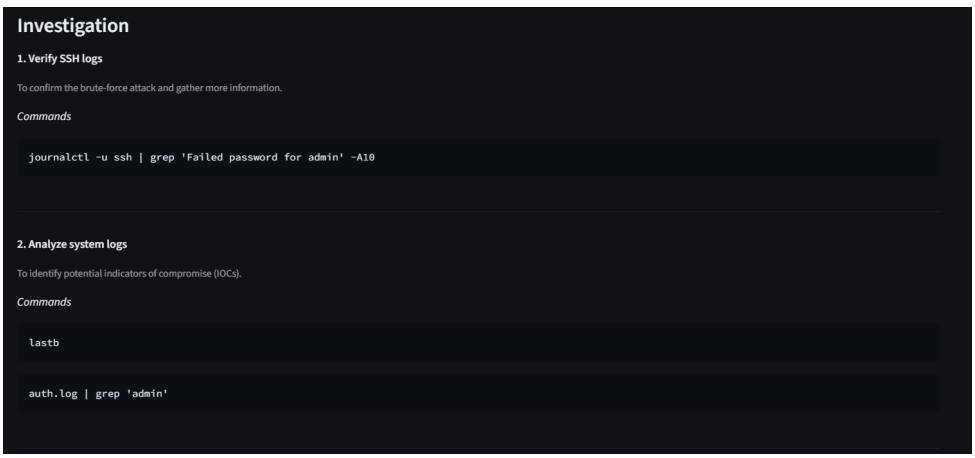
APPENDIX B

ASTRA UI Case Studies

This annex provides complementary screenshots of the ASTRA interface referenced in Chapter 4. Each figure illustrates the layout and generated outputs for the three demonstration scenarios.







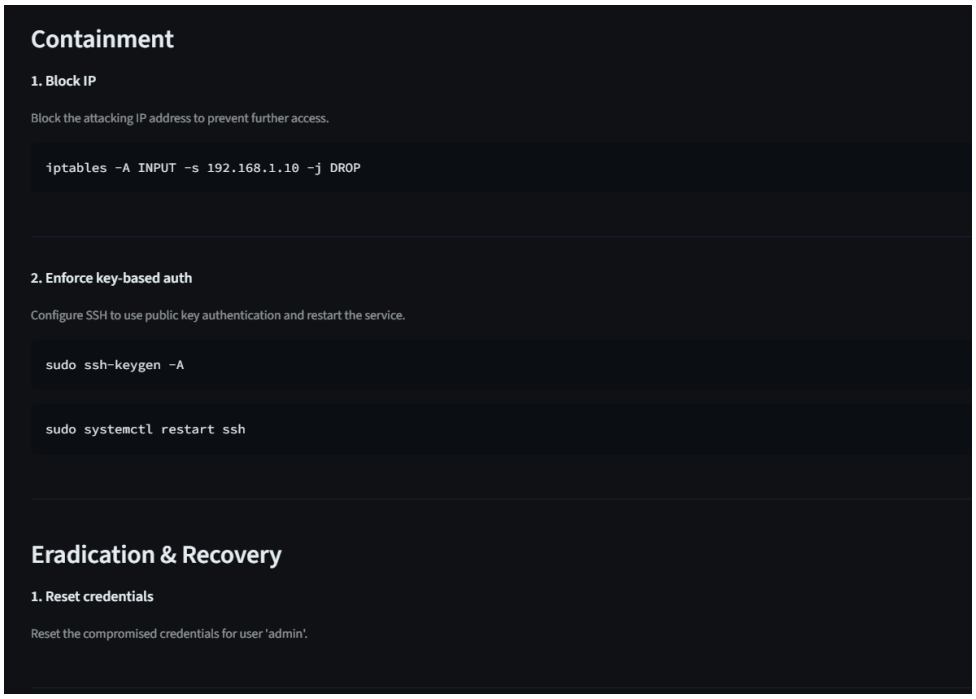
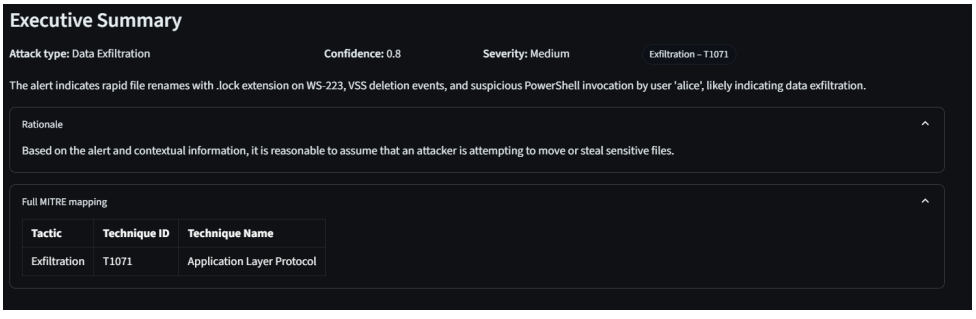
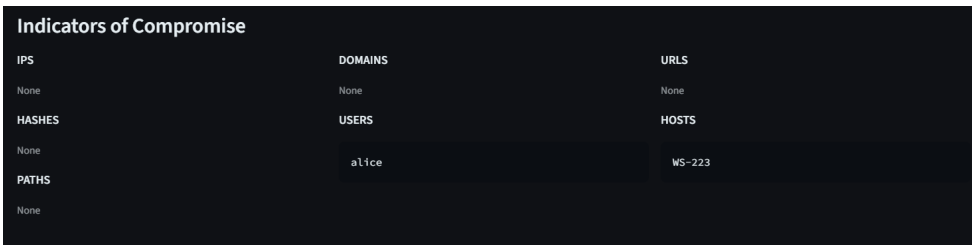


FIGURE B.1. ASTRA interface during SSH brute-force investigation (Case 1): Summary, IOC extraction, Investigation, and Containment/Recovery views.





1. Investigate VSS deletion events

To understand the scope of data loss and potential exfiltration

Commands

```
vssadmin list shadowcopies
```

```
Get-WinEvent -FilterHashTable @{ProviderName='VSS' }
```

Artifacts

```
VSS logs
```

2. Review PowerShell invocation

To identify potential command injection or lateral movement

Commands

```
PowerShell Get-Command
```

```
Get-WinEvent -FilterHashTable @{ProviderName='PowerShell' }
```

Artifacts

```
PowerShell logs
```

3. Analyze file renames and locks

To determine if files are being moved or locked for exfiltration

Commands

```
Get-ChildItem -Path C:\
```

```
fsutil file getlsize
```

Artifacts

```
File system logs
```

4. Check user accounts and authentication

To identify potential credential theft or misuse

Commands

```
Get-Account
```

```
net user
```

Artifacts

```
Authentication logs
```

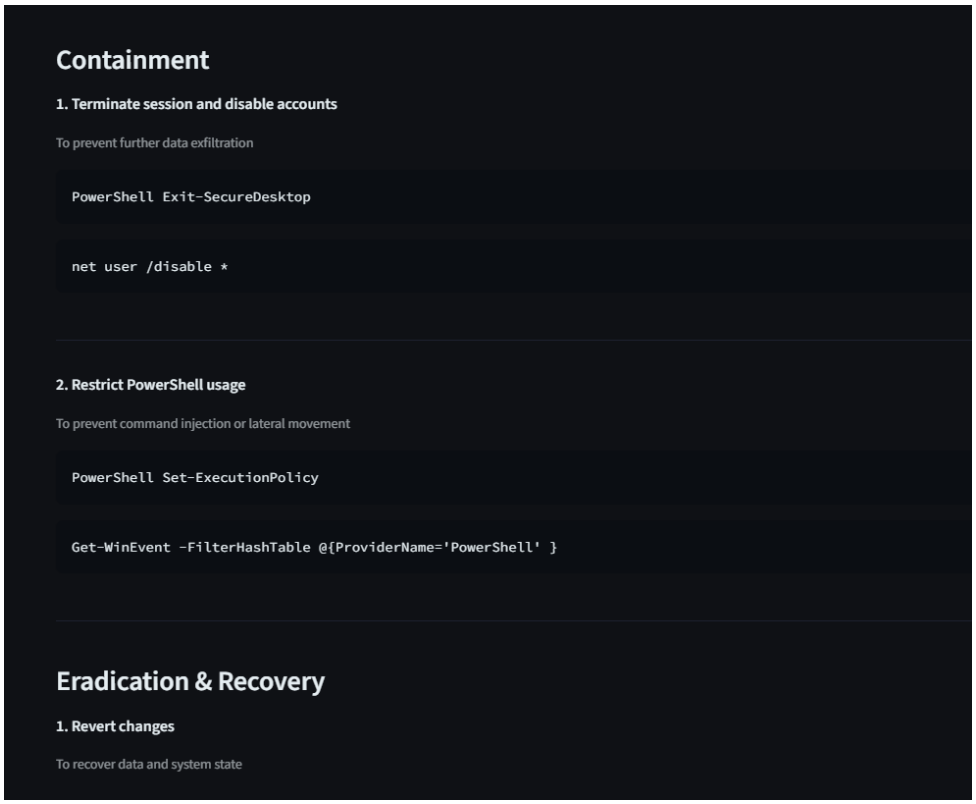
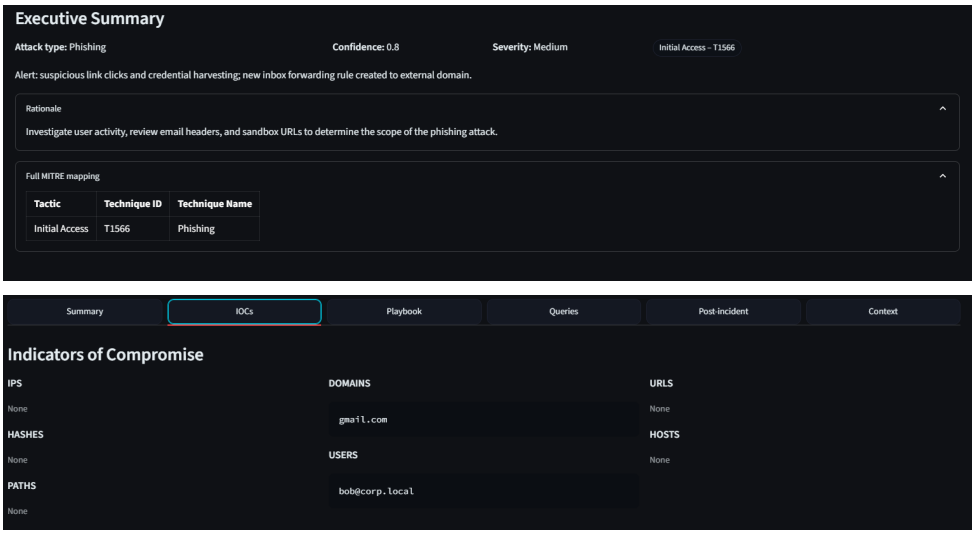


FIGURE B.2. ASTRA interface during ransomware/data-exfiltration investigation (Case 2): Summary, IOC extraction, Investigation, and Containment/Recovery views.



Investigation

1. Review email logs and audit logs for suspicious activity

To identify potential indicators of compromise (IOCs) and determine the extent of the phishing attack.

Commands

```
Get-EmailLog -Filter { Sender -like 'gmail.com' }
```

```
journalctl | grep 'phishing'
```

2. Verify the existence of suspicious executables or files

To identify potential malware or other malicious artifacts.

Commands

```
Get-ChildItem -Path C:\Windows\Temp | Where { $_.Name -like 'suspicious.exe' }
```

```
ls /var/tmp | grep 'suspicious'
```

3. Examine the system's network activity and connections

To identify potential C2 traffic or other malicious communication.

Commands

```
Get-NetConnection -State 'Established' | Where { $_.RemoteAddress -like 'suspicious.ip' }
```

```
netstat -an | grep 'suspicious.port'
```

4. Check for suspicious registry entries or keys

To identify potential registry modifications or malicious persistence mechanisms.

Commands

```
Get-ChildItem -Path HKLM:\Software | Where { $_.Name -like 'suspicious.key' }
```

```
reg query HKEY_LOCAL_MACHINE\SOFTWARE | findstr 'suspicious'
```

Containment

1. Block the suspicious domain gmail.com in email filters

This action will prevent further emails from being received from the suspicious domain.

```
Set-EmailFilter -Domain 'gmail.com' -Status 'Blocked'
```

```
update-mimeview-filter -domain 'gmail.com' -status 'blocked'
```

2. Reset user credentials and notify affected users

This action will ensure that the user's credentials are no longer compromised.

```
Get-User | Where { $_.Name -eq 'bob@corp.local' } | Reset-Credential
```

```
echo 'Please reset your password'
```

Eradication & Recovery

1. Delete suspicious files and registry keys

This action will remove any suspicious files and registry keys from the system.

FIGURE B.3. ASTRA interface during phishing attack (Case 3): Summary, IOC extraction, Investigation, and Containment/Recovery views.

APPENDIX C

Expert Feedback Summary

This annex provides the detailed, anonymized survey responses collected from six cybersecurity professionals who evaluated the ASTRA prototype. The survey included quantitative ratings (1–5 scale) and open-ended comments for three attack scenarios: brute force, ransomware, and phishing.

TABLE C.1. Full expert survey responses across all six participants (R1–R6).

Question	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**
How relevant was the tool's recommended response to the brute force scenario?	4	5	5	5	5	4
Did the suggested playbook cover all necessary response steps?	Mostly complete, a few things missing	Yes, it was complete	Yes, it was complete	Yes, it was complete	Not sure	Not sure
Would you trust this AI recommendation in a real SOC environment?	Yes	Yes	Yes	Yes	Maybe	Maybe
Any comments, improvements, or red flags?	Nothing at this stage	All good	No	Good	No	Sem sugestão
How relevant was the AI-generated response to the ransomware incident?	5	5	5	5	5	4
Did the recommended response feel clear and actionable for a ransomware incident?	5	5	5	5	3	4
Would you trust this recommendation in a real-world ransomware incident?	5	5	5	4	3	4
Any comments, improvements, or red flags?	No improvements	All good	No	More details	No	Sem comentários
Was the response plan appropriate for a phishing attack?	5	5	5	5	3	3
How realistic and actionable did the mitigation steps feel?	Somewhat realistic, but could be improved	Very realistic and actionable	Very realistic and actionable	Somewhat realistic, but could be improved	Not sure / didn't fully understand the steps	Not realistic or helpful
Would this AI-generated response save time for an analyst handling a phishing case?	Yes, significantly	Yes, significantly	Yes, somewhat	Yes, significantly	Not sure	No, it might slow things down
Any feedback, missing steps, or suggestions to improve the phishing use case response?	Nothing	All good	No	–	No	Sem comentários
What best describes your current role or background?	SOC Analyst	Researcher / Academic	Cybersecurity Engineer	Software Developer / IT Professional	Other	Blue Team / Threat Detection Specialist
Would you like to receive the final results of this research or the full thesis?	No, thank you	No, thank you	No, thank you	No, thank you	No, thank you	No, thank you
If yes to any of the above, please share your e-mail below.	<anonymous>					
Would you be available for a quick interview as a follow-up to this form?	Yes	Yes	No	No	No	Yes

References

[1] Tugrul Daim, Haydar Yalcin, Alain Mermoud, and Valentin Mulder. "Exploring cybertechnology standards through bibliometrics: Case of National Institute of Standards and Technology". In: *World Patent Information* 77 (June 2024), p. 102278. ISSN: 01722190. DOI: 10.1016/j.wpi.2024.102278.

[2] Jordan Nelson and Ella Ben. "Data Security". In: (Feb. 2025).

[3] Ramanpreet Kaur, Dušan Gabrijelčič, and Tomaž Klobučar. "Artificial intelligence for cybersecurity: Literature review and future research directions". In: *Information Fusion* 97 (Sept. 2023), p. 101804. ISSN: 15662535. DOI: 10.1016/j.inffus.2023.101804.

[4] Ömer Aslan, Semih Serkant Aktuğ, Merve Ozkan-Okay, Abdullah Asim Yilmaz, and Erdal Akin. *A Comprehensive Review of Cyber Security Vulnerabilities, Threats, Attacks, and Solutions*. 2023. DOI: 10.3390/electronics12061333.

[5] Joonas Forsberg and Tapio Frantti. "Technical performance metrics of a security operations center". In: *Computers & Security* 135 (Dec. 2023), p. 103529. ISSN: 01674048. DOI: 10.1016/j.cose.2023.103529.

[6] Stefan Varga, Joel Brynielsson, and Ulrik Franke. "Cyber-threat perception and risk management in the Swedish financial sector". In: *Computers and Security* 105 (2021). ISSN: 01674048. DOI: 10.1016/j.cose.2021.102239.

[7] Faisal Quader and Vandana P. Janeja. *Insights into Organizational Security Readiness: Lessons Learned from Cyber-Attack Case Studies*. 2021. DOI: 10.3390/jcp1040032.

[8] Yaseen Asad. "Accelerating the SOC: Achieve Greater Efficiency with AI-driven Automation". In: *Journal of Artificial Intelligence and Machine Learning in Management* (Vol. 12 No. 1 Jan. 2022). URL: https://orcid.org/0009-0002-8950-0767.

[9] Duraid Thamer Salim, Manmeet Mahinderjit Singh, and Pantea Keikhosrokiani. "A systematic literature review for APT detection and Effective Cyber Situational Awareness (ECSA) conceptual model". In: *Heliyon* 9 (7 July 2023), e17156. ISSN: 24058440. DOI: 10.1016/j.heliyon.2023.e17156.

[10] Manh-Dung Nguyen, Wissam Mallouli, Ana Rosa Cavalli, and Edgardo Montes de Oca. "AI4SOAR: A Security Intelligence Tool for Automated Incident Response". In: *Proceedings of the 19th International Conference on Availability, Reliability and Security*. ACM, July 2024, pp. 1–8. ISBN: 9798400717185. DOI: 10.1145/3664476.3670450.

[11] Ken Peffers, Tuure Tuunanen, Marcus Rothenberger, and Samir Chatterjee. "A Design Science Research Methodology for Information Systems Research". In: *J. Manage. Inf. Syst.* 24.3 (Dec. 2007), pp. 45–77. ISSN: 0742-1222. DOI: 10.2753/MIS0742-1222240302. URL: https://doi.org/10.2753/MIS0742-1222240302.

[12] Neal R Haddaway, Matthew J Page, Chris C Pritchard, and Luke A McGuinness. "PRISMA2020: An R package and Shiny app for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and Open Synthesis". In: *Campbell Systematic Reviews* 18.2 (2022), e1230. DOI: 10.1002/cl2.1230.

[13] Daiki Chiba, Mitsuaki Akiyama, Yuto Otsuki, Hiroki Hada, Takeshi Yagi, Tobias Fiebig, and Michel Van Eeten. "DomainPrio: Prioritizing Domain Name Investigations to Improve SOC Efficiency". In: *IEEE Access* 10 (2022), pp. 34352–34368. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3161636.

[14] Fatemeh Jalalvand, Mohan Baruwal Chhetri, Surya Nepal, and Cecile Paris. "Alert Prioritisation in Security Operations Centres: A Systematic Survey on Criteria and Methods". In: *ACM Computing Surveys* 57 (2 Feb. 2025), pp. 1–36. ISSN: 0360-0300. DOI: 10.1145/3695462.

[15] Mohan Baruwal Chhetri, Shahroz Tariq, Ronal Singh, Fatemeh Jalalvand, Cecile Paris, and Surya Nepal. "Towards Human-AI Teaming to Mitigate Alert Fatigue in Security Operations Centres". In: *ACM Transactions on Internet Technology* 24 (3 Aug. 2024), pp. 1–22. ISSN: 1533-5399. DOI: 10.1145/3670009.

[16] Claire La Fleur, Blaine Hoffman, C. Benjamin Gibson, and Norbou Buchler. "Team performance in a series of regional and national US cybersecurity defense competitions: Generalizable effects of training and functional role specialization". In: *Computers & Security* 104 (May 2021), p. 102229. ISSN: 01674048. DOI: 10.1016/j.cose.2021.102229.

[17] Alok Mishra, Yehia Ibrahim Alzoubi, Memoona Javeria Anwar, and Asif Qumer Gill. "Attributes impacting cybersecurity policy development: An evidence from seven nations". In: *Computers & Security* 120 (Sept. 2022), p. 102820. ISSN: 01674048. DOI: 10.1016/j.cose.2022.102820.

[18] Prabhat Kumar, Danish Javeed, A.K.M. Najmul Islam, and Xin (Robert) Luo. "DeepSecure: A computational design science approach for interpretable threat hunting in cybersecurity decision making". In: *Decision Support Systems* 188 (Jan. 2025), p. 114351. ISSN: 01679236. DOI: 10.1016/j.dss.2024.114351.

[19] Arash Mahboubi, Khanh Luong, Hamed Aboutorab, Hang Thanh Bui, Geoff Jarrad, Mohammed Bahutair, Seyit Camtepe, Ganna Pogrebna, Ejaz Ahmed, Bazara Barry, and Hannah Gately. "Evolving techniques in cyber threat hunting: A systematic review". In: *Journal of Network and Computer Applications* 232 (Dec. 2024), p. 104004. ISSN: 10848045. DOI: 10.1016/j.jnca.2024.104004.

[20] Bader Al-Sada, Alireza Sadighian, and Gabriele Oligeri. "MITRE ATT&CK: State of the Art and Way Forward". In: *ACM Computing Surveys* 57 (1 Jan. 2025), pp. 1–37. ISSN: 0360-0300. DOI: 10.1145/3687300.

[21] Iqbal H. Sarker, Helge Janicke, Ahmad Mohsin, Asif Gill, and Leandros Maglaras. "Explainable AI for cybersecurity automation, intelligence and trustworthiness in digital twin: Methods, taxonomy, challenges and prospects". In: *ICT Express* 10 (4 Aug. 2024), pp. 935–958. ISSN: 24059595. DOI: 10.1016/j.icte.2024.05.007.

[22] Mohd Saqib, Samaneh Mahdavifar, Benjamin C. M. Fung, and Philippe Charland. "A Comprehensive Analysis of Explainable AI for Malware Hunting". In: *ACM Computing Surveys* 56 (12 Dec. 2024), pp. 1–40. ISSN: 0360-0300. DOI: 10.1145/3677374.

[23] Sandro Waelchli and Yoshija Walter. "Reducing the risk of social engineering attacks using SOAR measures in a real world environment: A case study". In: *Computers & Security* 148 (Jan. 2025), p. 104137. ISSN: 01674048. DOI: 10.1016/j.cose.2024.104137.

[24] Lasse Nitz, Mehdi Akbari Gurabi, Milan Cermak, Martin Zadnik, David Karpuk, Arthur Drichel, Sebastian Schäfer, Benedikt Holmes, and Avikarsha Mandal. "On Collaboration and Automation in the Context of Threat Detection and Response with Privacy-Preserving Features". In: *Digital Threats: Research and Practice* 6 (1 Mar. 2025), pp. 1–36. ISSN: 2576-5337. DOI: 10.1145/3707651.

[25] Antonio João Gonçalves de Azambuja, Tim Giese, Klaus Schützer, Reiner Anderl, Benjamin Schleich, and Vilson Rosa Almeida. "Digital Twins in Industry 4.0 – Opportunities and challenges related to Cyber Security". In: *Procedia CIRP* 121 (2024), pp. 25–30. ISSN: 22128271. DOI: 10.1016/j.procir.2023.09.225.

[26] Mohamed Amine Ferrag, Fatima Alwahedi, Ammar Battah, Bilel Cherif, Abdechakour Mechri, Norbert Tihanyi, Tamas Bisztray, and Merouane Debbah. "Generative AI in Cybersecurity: A Comprehensive Review of LLM Applications and Vulnerabilities". In: *Internet of Things and Cyber-Physical Systems* (Feb. 2025). ISSN: 26673452. DOI: 10.1016/j.iotcps.2025.01.001.

[27] Omar Alshaikh, Simon Parkinson, and Saad Khan. "Exploring perceptions of decision-makers and specialists in defensive machine learning cybersecurity applications: The need for a standardised approach". In: *Computers & Security* 139 (Apr. 2024), p. 103694. ISSN: 01674048. DOI: 10.1016/j.cose.2023.103694.

[28] Timothy R. McIntosh, Teo Susnjak, Tong Liu, Paul Watters, Dan Xu, Dongwei Liu, Raza Nowrozy, and Malka N. Halgamuge. "From COBIT to ISO 42001: Evaluating cybersecurity frameworks for opportunities, risks, and regulatory compliance in commercializing large language models". In: *Computers & Security* 144 (Sept. 2024), p. 103964. ISSN: 01674048. DOI: 10.1016/j.cose.2024.103964.

[29] Ronal Singh, Shahroz Tariq, Fatemeh Jalalvand, Mohan Baruwal Chhetri, Surya Nepal, Cecile Paris, and Martin Lochner. *LLMs in the SOC: An Empirical Study of*

Human-AI Collaboration in Security Operations Centres. 2025. arXiv: 2508.18947 [cs.CR]. URL: https://arxiv.org/abs/2508.18947.

[30] Sudipta Paria, Aritra Dasgupta, and Swarup Bhunia. *DIVAS: An LLM-based End-to-End Framework for SoC Security Analysis and Policy-based Protection*. 2023. arXiv: 2308.06932 [cs.CR]. URL: https://arxiv.org/abs/2308.06932.

[31] Dipayan Saha, Shams Tarek, Katayoon Yahyaei, Sujan Kumar Saha, Jingbo Zhou, Mark Tehranipoor, and Farimah Farahmandi. "LLM for SoC Security: A Paradigm Shift". In: *IEEE Access* 12 (2024), pp. 155498–155521. DOI: 10.1109/ACCESS.2024.3427369.

[32] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. 2021. arXiv: 2005.11401 [cs.CL]. URL: https://arxiv.org/abs/2005.11401.

[33] Jeff Johnson, Matthijs Douze, and Hervé Jégou. "Billion-Scale Similarity Search with GPUs". In: *IEEE Transactions on Big Data* 7.3 (2021), pp. 535–547. DOI: 10.1109/TBDATA.2019.2921572.

[34] Nils Reimers and Iryna Gurevych. *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. 2019. arXiv: 1908.10084 [cs.CL]. URL: https://arxiv.org/abs/1908.10084.

[35] Ollama Inc. *Ollama: Run large language models locally*. Accessed: October 2025. 2025. URL: https://ollama.ai.

[36] Streamlit Inc. *Streamlit: The fastest way to build and share data apps*. Accessed: October 2025. 2025. URL: https://streamlit.io.