

Article

Toward Robust Security Orchestration and Automated Response in Security Operations Centers with a Hyper-Automation Approach Using Agentic Artificial Intelligence

Ismail ¹, Rahmat Kurnia ² , Zilmas Arjuna Brata ² , Ghitha Afina Nelistiani ², Shinwook Heo ² ,
Hyeongon Kim ²  and Howon Kim ^{1,*} 

¹ School of Computer Science and Engineering, Pusan National University, Busan 46241, Republic of Korea; ismail@pusan.ac.kr

² SmartM2M. Co., Ltd., 701, 702, Building A, Centum Skybiz, 97 Centumjungang-ro, Haeundae-gu, Busan 48058, Republic of Korea; rahmat@smartm2m.co.kr (R.K.); zilmas@smartm2m.co.kr (Z.A.B.); githa@smartm2m.co.kr (G.A.N.); shinwookheo@smartm2m.co.kr (S.H.); hyeongonkim@smartm2m.co.kr (H.K.)

* Correspondence: howonkim@pusan.ac.kr

Abstract: The evolving landscape of cybersecurity threats demands the modernization of Security Operations Centers (SOCs) to enhance threat detection, response, and mitigation. Security Orchestration, Automation, and Response (SOAR) platforms play a crucial role in addressing operational inefficiencies; however, traditional no-code SOAR solutions face significant limitations, including restricted flexibility, scalability challenges, inadequate support for advanced logic, and difficulties in managing large playbooks. These constraints hinder effective automation, reduce adaptability, and underutilize analysts' technical expertise, underscoring the need for more sophisticated solutions. To address these challenges, we propose a hyper-automation SOAR platform powered by agentic-LLM, leveraging Large Language Models (LLMs) to optimize automation workflows. This approach shifts from rigid no-code playbooks to AI-generated code, providing a more flexible and scalable alternative while reducing operational complexity. Additionally, we introduce the IVAM framework, comprising three critical stages: (1) Investigation, structuring incident response into actionable steps based on tailored recommendations, (2) Validation, ensuring the accuracy and effectiveness of executed actions, (3) Active Monitoring, providing continuous oversight. By integrating AI-driven automation with the IVAM framework, our solution enhances investigation quality, improves response accuracy, and increases SOC efficiency in addressing modern cybersecurity threats.

Keywords: security operation center; incident-response; hyper-automation; SOAR; agentic-LLM



Academic Editors: Zia Ush
Shamszaman, Shareeful Islam,
Stefano Silvestri and Xiaokun Zhang

Received: 26 February 2025

Revised: 3 April 2025

Accepted: 21 April 2025

Published: 29 April 2025

Citation: Ismail; Kurnia, R.; Brata, Z.A.; Nelistiani, G.A.; Heo, S.; Kim, H.; Kim, H. Toward Robust Security Orchestration and Automated Response in Security Operations Centers with a Hyper-Automation Approach Using Agentic Artificial Intelligence. *Information* **2025**, *16*, 365. <https://doi.org/10.3390/info16050365>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Effective security operations are essential for safeguarding digital assets in an era of rapidly evolving cyber threats. The Security Operations Center (SOC) is the cornerstone of an organization's defense strategy, continuously monitoring, detecting, and responding to security incidents. However, SOC teams are increasingly overwhelmed by the rising volume, complexity, and sophistication of security alerts, leading to operational inefficiencies and delayed responses [1]. These challenges underscore the urgent need for innovative solutions that streamline workflows and enhance SOC effectiveness.

Security Orchestration, Automation, and Response (SOAR) platforms have emerged as a transformative solution to these challenges. By integrating diverse security tools, automating repetitive tasks, and orchestrating complex workflows, SOAR platforms improve the speed and accuracy of incident response [2,3]. As a critical component of modern SOC, these platforms enable organizations to manage an expanding threat landscape while maintaining operational efficiency [4,5]. Additionally, SOAR systems allow SOC teams to focus on high-value tasks, such as advanced threat hunting and strategic decision-making, by automating routine activities [6,7].

Despite their advantages, designing and implementing effective SOAR playbooks presents several challenges. These include ensuring the accuracy and relevance of automated actions [8], maintaining compatibility with diverse security tools [9], and continuously updating playbooks to address emerging threats [10]. Adopting best practices for playbook development and deployment is crucial to maximizing the benefits of SOAR platforms.

Key Contributions

To address the inherent limitations of traditional SOAR (Security Orchestration, Automation, and Response) implementations, this study introduces a novel framework for hyper-automation within SOAR systems. The proposed methodology employs AI-driven agents to autonomously construct adaptive workflows, each customized to the specific attributes of individual security events. This dynamic capability significantly enhances the precision, scalability, and contextual relevance of automated incident response. At the core of the framework lies a hyper-automation architecture underpinned by agentic artificial intelligence and Large Language Models (LLMs). The principal contributions of this research are as follows:

1. **The IVAM Framework:** The Investigation–Validation–Active Monitoring (IVAM) framework is introduced, incorporating MITRE ATT&CK for adversarial mapping, the NIST Cybersecurity Framework for standardization, and Quantitative Risk Assessment (QRA) for structured validation. This integration delivers a rarely seen level of structured, risk-driven incident response.
2. **Agentic LLM-Based SOAR Architecture:** A novel SOAR framework is proposed that leverages autonomous LLM agents capable of dynamically generating, adapting, and executing contextual playbooks. This approach surpasses traditional static or no-code/low-code models by enabling zero-shot task execution and tool orchestration.
3. **Adaptive Multi-Agent System:** The capabilities of a multi-step reasoning AI agent are demonstrated, featuring shared memory, external tool integration, and human-in-the-loop functionality. This enables evolution from reactive task automation to proactive, self-adjusting threat mitigation.

This innovation addresses the critical need for scalability and adaptability in SOC operations, particularly within an increasingly complex threat landscape.

2. Background & Related Works

2.1. Evolution of Cybersecurity Automation

The integration of automation in Security Operations Centers (SOCs) has undergone significant evolution. Early efforts focused on rule-based expert systems and signature-based detection mechanisms that provided foundational capabilities for detecting and responding to known threats. These classical AI methods, such as decision trees, Bayesian classifiers, and expert systems, offered deterministic outputs but lacked adaptability to evolving threats. Examples include early implementations of IBM QRadar and ArcSight platforms that employed predefined correlation rules for event detection.

As threats became more complex and diverse, machine-learning techniques began to enhance SOC capabilities. Traditional algorithms such as Support Vector Machines (SVMs), Random Forests, and clustering methods improved anomaly detection but often required extensive feature engineering and struggled with real-time contextual understanding.

Modern approaches now emphasize hyper-automation, integrating AI with real-time data analysis to orchestrate responses dynamically. These systems are powered by Large Language Models (LLMs) and agentic AI, enabling them to interpret unstructured data, adapt to new threat patterns, and automate multi-step workflows across complex environments.

2.2. Comparative Analysis of Traditional and Contemporary SOAR Architectures

Integrating automation within Security Operations Centers (SOCs) has become essential for enhancing both operational efficiency and analytical effectiveness. By streamlining routine tasks, automation enables human analysts to allocate more time to complex threat analysis and strategic decision-making. This synergy between human expertise and automated systems not only accelerates incident response but also improves the precision of threat detection and mitigation [11]. Emerging paradigms such as no-code, low-code, and hyper-automation each introduce distinct benefits and limitations, collectively shaping the trajectory of SOC operations.

Traditional Security Orchestration, Automation, and Response (SOAR) platforms have primarily leveraged no-code and low-code environments to broaden accessibility and reduce development overhead. While these platforms democratize automation, they frequently encounter limitations, including restricted customization, difficulty in maintaining large and complex playbooks, and limited adaptability to evolving threat scenarios.

In contrast, contemporary SOAR solutions leverage LLMs to dynamically generate code, enabling adaptive and scalable automation. Unlike their static predecessors, these AI-powered systems integrate contextual awareness and reasoning to respond in real time.

2.2.1. No-Code Automation

No-code platforms have transformed software development by enabling individuals with minimal or no programming expertise to create functional applications. These platforms leverage visual interfaces, drag-and-drop tools, and pre-built components, democratizing software creation and accelerating digital transformation [12].

Frameworks such as Security Orchestration, Automation, and Response (SOAR) illustrate how no-code solutions can thrive in hyper-connected environments like the Internet of Blended Environment (IoBE) [13]. By integrating automation with threat intelligence and dynamic responses, SOAR enhances scalability and security, addressing challenges across diverse domains.

No-code tools empower “citizen developers” to drive digital innovation within their organizations, eliminating the need for extensive coding expertise. This approach accelerates development cycles, reduces costs, and fosters adaptability, key requirements in today’s fast-paced business landscape [12]. By supporting rapid deployment and iterative processes, no-code platforms align with the growing emphasis on agility and innovation.

However, despite these advantages, no-code platforms also present certain limitations. The lack of flexibility and customization in pre-built components may restrict the development of complex, highly specialized applications.

2.2.2. Low-Code Automation

Traditional risk management systems, often reliant on manual processes and legacy software, struggle to address the complexities and evolving nature of modern organizational risks [14,15]. Low-code automation has emerged as a transformative solution,

enabling organizations to streamline processes, enhance compliance, and swiftly adapt to regulatory changes [16].

Low-code platforms are software development environments that require minimal hand-coding, allowing both technical and non-technical users to create and deploy applications with greater speed and efficiency [17]. By leveraging visual interfaces and pre-built components, low-code solutions reduce development time, lower operational costs, and foster innovation [18,19]. These capabilities are particularly valuable in risk management, where timely and accurate data-driven decision-making is essential.

Additionally, low-code platforms facilitate rapid prototyping and iterative development, enabling businesses to adapt quickly to changing market demands. This agility empowers teams to experiment with new ideas, refine processes, and stay ahead of industry trends without the constraints of extensive coding [20].

Furthermore, the integration of generative AI amplifies the potential of low-code platforms. By automating repetitive tasks, generating code snippets, and enabling natural language interactions, AI-driven low-code environments provide unprecedented customization and scalability. This synergy allows organizations to not only accelerate workflows but also tailor solutions to unique project requirements, paving the way for personalized and impactful applications [20,21].

However, despite these advantages, low-code platforms are not without limitations. Their reliance on pre-built components may restrict flexibility, particularly for highly specialized applications. Additionally, vendor lock-in can pose challenges as organizations outgrow platform capabilities, necessitating careful consideration of long-term scalability and interoperability.

2.2.3. Hyper-Automation

Hyper-automation is rapidly transforming the modern industries by integrating advanced technologies such as Artificial Intelligence (AI), Machine Learning (ML), Robotic Process Automation (RPA), and the Internet of Things (IoT) [22]. Unlike traditional automation, hyper-automation takes a holistic approach to automating complex business processes, enabling organizations to achieve unprecedented efficiency, accuracy, and innovation [23].

The rise of hyper-automation is driven by the increasing need for agility and scalability in a highly competitive global market. As organizations face growing pressure to adapt to evolving customer expectations and technological advancements, hyper-automation provides a strategic solution. By combining cognitive and operational capabilities, it empowers businesses to streamline workflows, enhance decision-making, and optimize resource utilization [24,25].

Recent studies highlight the transformative potential of hyper-automation across various sectors, including manufacturing, healthcare, finance, and logistics [26,27]. For instance, in manufacturing, hyper-automation facilitates the integration of smart factories, enabling real-time monitoring and predictive maintenance [28].

2.2.4. The Integration of ML and AI Capabilities into Modern SOAR Architectures

Security Orchestration, Automation, and Response (SOAR) platforms integrate a variety of security tools and data sources, enabling organizations to automate repetitive tasks, orchestrate workflows, and streamline incident response processes [29]. By consolidating data and automating critical functions, these platforms significantly reduce the time required to address cybersecurity incidents, thereby protecting assets and minimizing potential damage [30].

Recent advancements in Artificial Intelligence (AI) and Machine Learning (ML) have further enhanced SOAR capabilities, enabling intelligent threat detection, adaptive response

mechanisms, and improved decision-making [31]. By analyzing vast amounts of security data in real-time, AI and ML algorithms detect patterns and anomalies that may indicate potential threats. This allows security teams to respond more effectively and efficiently while reducing the cognitive burden on analysts.

As illustrated in Figure 1, the integration of AI/ML capabilities into SOAR systems aligns with the SANS PICERL framework [32], enhancing automated threat detection and incident response processes. The figure highlights how AI-driven SOAR solutions facilitate proactive security operations by leveraging machine learning for anomaly detection, threat intelligence correlation, and automated remediation workflows.

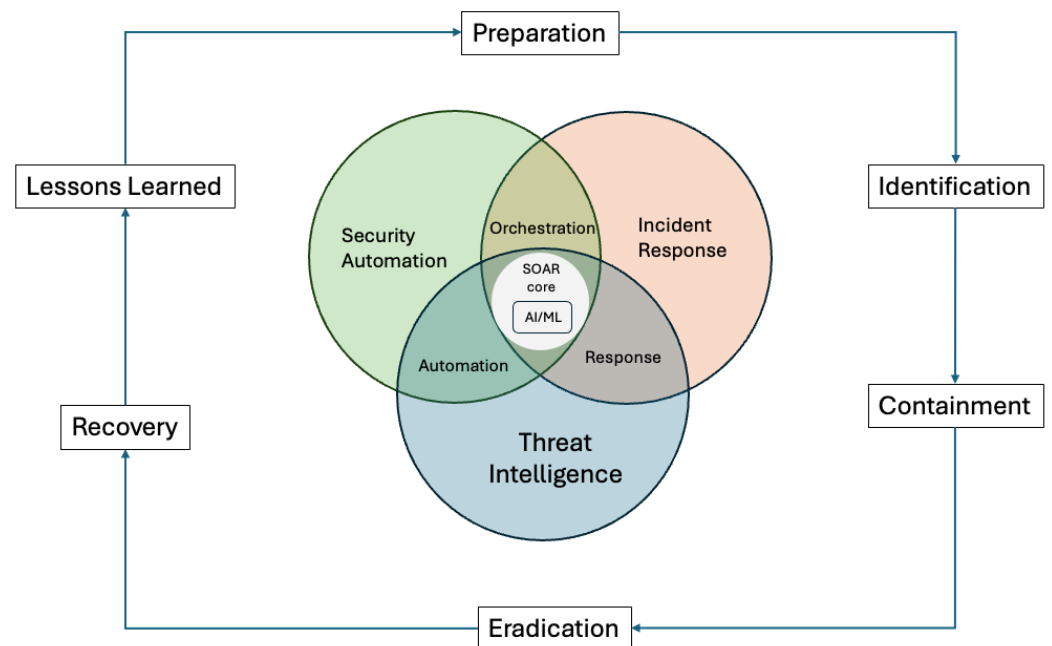


Figure 1. Integration of AI/ML into SOAR platforms aligned with the SANS PICERL framework. (adapted from [31]).

The integration of Artificial Intelligence (AI) and Machine Learning (ML) into Security Orchestration, Automation, and Response (SOAR) platforms has been widely explored in recent research. For instance, PHOENI2X, a European Cyber Resilience Framework, leverages AI-assisted orchestration to enhance business continuity and incident response for Operators of Essential Services (OESs) [33]. This framework highlights AI's role in streamlining complex workflows and enabling seamless collaboration among stakeholders.

Similarly, IC-SECURE introduces a deep-learning-based system designed to assist security analysts in automating playbook generation, thereby improving incident response efficiency and reducing reliance on manual intervention [34]. Other significant contributions include behavioral honeypots, which adapt dynamically to attackers' techniques to gather threat intelligence, and frameworks like APIRO, which optimize the integration and interoperability of security tools within SOAR platforms [35].

Recent innovations in SOAR platforms have addressed critical cybersecurity challenges, demonstrating their potential to redefine security operations through automation and orchestration. Among these advancements, AI-driven playbook generation, as shown by IC-SECURE, has automated the creation of tailored incident response strategies, ensuring consistent execution during cyber incidents while adapting to evolving threats [34]. Similarly, the PHOENI2X framework has revolutionized collaborative security operations by leveraging AI-assisted orchestration, facilitating real-time information exchange and

decision-making. These innovations exemplify how SOAR platforms are transforming cybersecurity practices and strengthening organizational resilience [33].

Despite these advancements, challenges remain in adopting and implementing SOAR platforms. Organizations often struggle with tool interoperability and regulatory compliance, hindering seamless integration into existing security infrastructures. Additionally, the dynamic nature of modern cyber threats, such as those observed in the Internet of Blended Environments (IoBE), necessitates adaptive and scalable solutions [35]. These challenges underscore the need for continued innovation in SOAR architectures and methodologies to ensure their effectiveness in mitigating evolving cybersecurity risks.

2.3. Hyper-Automation and Agent-Based AI Systems

Hyper-automation extends beyond traditional task automation by integrating cognitive technologies such as LLMs, Robotic Process Automation (RPA), and agentic AI. These systems autonomously manage, analyze, and respond to incidents while learning from historical data. The integration of agentic AI in SOAR allows for modular and goal-directed automation, which is especially crucial in IoT-driven environments.

Recent studies highlight the expanding role of agentic AI in both defensive and offensive security contexts. For instance, Valencia [36] introduces ReaperAI, an autonomous agent that leverages Large Language Models (LLMs) like GPT-4 to identify, exploit, and analyze vulnerabilities, demonstrating the feasibility of AI-driven offensive security strategies. Oesch et al. [37] further examine the strategic implications of agentic AI in the context of a cyber arms race, underscoring its dual-use potential for automating both attack and defense mechanisms and raising critical questions about global cyber stability. On the defensive front, Khan et al. [38] explore the risks associated with agentic AI systems' access to sensitive databases, detailing threats such as unauthorized data extraction and the exploitation of system vulnerabilities stemming from autonomous behavior. Complementing these perspectives, Kaheh et al. [39] present Cyber Sentinel, a dialogue-based agent designed to explain cyber threats and execute mitigations using GPT-4, highlighting the practical application of conversational agents in streamlining security operations. Collectively, these studies emphasize the transformative potential of agentic AI in enhancing cyber resilience through real-time adaptation, autonomy, and strategic reasoning.

2.4. Frameworks for Constructing an Effective Incident Response

Cyber incidents such as ransomware attacks, phishing campaigns, and Advanced Persistent Threats (APTs) highlight the need for robust Incident Response Plans (IRPs) [40] that can effectively detect, respond to, and recover from security breaches. Constructing an effective IRP requires a deep understanding of organizational risks and the application of proven frameworks that provide structured methodologies for incident management.

Several established frameworks have been developed to guide organizations in building resilient and efficient IRPs, each offering unique strengths tailored to different operational, strategic, and technical needs.

For instance, the NIST Cybersecurity Framework (CSF) provides comprehensive guidance across its five core functions:

- Identify: Recognizing and assessing security risks.
- Protect: Implementing safeguards to mitigate potential threats.
- Detect: Continuously monitoring for security incidents.
- Respond: Taking immediate action upon detecting threats.
- Recover: Restoring affected systems and minimizing impact.

This framework ensures a holistic approach to incident response and cybersecurity management [41].

Similarly, the MITRE ATT&CK Framework categorizes adversarial actions into three distinct components:

- Tactics: Represent the adversary's overall objectives.
- Techniques: Describe the methods used to achieve those objectives.
- Procedures: Outline specific implementations of these techniques.

This level of granularity provides security teams with a clear understanding of real-world attack methodologies, improving their ability to anticipate, detect, and mitigate cyber threats [42].

2.5. Risk Assessment in Cybersecurity Domain

2.5.1. Quantitative Risk Assessment

The increasing complexity and prevalence of cybersecurity threats necessitates systematic, data-driven approaches to assess and mitigate risks. Quantitative Risk Analysis (QRA) is a methodology that evaluates risks using numerical measures, allowing decision-makers to estimate the likelihood and potential impact of specific events. Compared to qualitative methods, which often lack sufficient rigor for guiding critical security decisions, QRA provides a precise and actionable framework [43].

QRA applies probabilistic models to evaluate both the uncertainty and consequences of risks in a given scenario. The fundamental objective is to quantify risk exposure by combining the likelihood of threat events with their corresponding impacts. A widely adopted formula for this is:

$$\text{Risk} = \text{Likelihood} \times \text{Impact}. \quad (1)$$

In practice, this concept is extended to assess the Total Risk Exposure (TRE) [44], aggregating all identified risks as:

$$\text{TRE} = \sum_{i=1}^n P_i \cdot I_i \quad (2)$$

where P_i represents the probability of the i -th threat event and I_i denotes the corresponding impact or cost of the event.

Building on these foundations, recent studies such as that of [45] introduced formal models to estimate the probability of data breaches and the associated costs over defined periods. These models integrate empirical data, such as historical breach records and industry benchmarks, to provide realistic and actionable estimates. By adopting such methodologies, organizations can prioritize risk mitigation strategies based on potential cost-benefit tradeoffs.

Additionally, integrating multifaceted classification approaches has further refined the quantitative assessment of cybersecurity risks. Ref. [46] proposed a systematic, extendable, and modular model for assessing information systems, offering a quantitative analysis grounded in a well-defined classification scheme. This approach underscores the importance of adaptability and modularity in addressing the ever-evolving nature of cybersecurity threats.

Advanced QRA techniques [43] often account for the dynamic nature of threats. For example, time-dependent stochastic models are used to represent risk as a function of time t :

$$\text{Risk}(t) = \int_{t_0}^{t_1} \lambda(t) \cdot I(t) dt, \quad (3)$$

where $\lambda(t)$ represents the time-dependent hazard rate and $I(t)$ denotes the impact as a function of time. These methods enable organizations to assess the effectiveness of mitigation strategies under evolving threat conditions.

By leveraging QRA methodologies, organizations can derive actionable insights, strategically allocate resources to high-priority vulnerabilities, and strengthen their adaptive capacity to counteract evolving cybersecurity threats.

2.5.2. Large Language Models (LLMs) and Agentic AI in Cybersecurity Domain

The evolution of Artificial Intelligence (AI) has significantly transformed various sectors, with Large Language Models (LLMs) and agentic AI emerging as pivotal innovations in cybersecurity. Characterized by advanced natural language understanding and autonomous decision-making, these models are reshaping how cyber threats are detected, analyzed, and mitigated.

Prominent LLMs, including OpenAI's GPT series [47], Google's Bard [48], Anthropic's Claude [49], Meta's LLaMA [50], Cohere's Command R [51], and open-source models such as Hugging Face's BLOOM [52], EleutherAI's GPT-NeoX [53], DeepSeek-R1 [54], and Qwen [55], excel in processing and generating human-like text. Within cybersecurity, these models enhance threat intelligence by analyzing large volumes of unstructured data to uncover patterns indicative of malicious activity [56]. Furthermore, their ability to generate human-readable analyses and reports aids security analysts in efficiently interpreting complex threat landscapes.

A fundamental strength of LLMs lies in their adaptability. Fine-tuning them for specific cybersecurity tasks such as phishing detection, malware analysis, and vulnerability assessment can significantly improve detection accuracy while reducing human error. For example, Bommasani et al. [57] illustrate how foundation models enhance response times and support data-driven decision-making in security operations.

LLMs have evolved beyond their original language-based applications into comprehensive AI agents capable of multi-step reasoning, external tool integration, and system collaboration. This transformation has been driven by two key factors: advancements in prompting strategies and external capability integration.

Techniques such as Chain-of-Thought (CoT) and chain-of-density prompting enable LLMs to reveal their intermediate reasoning steps, thereby enhancing both interpretability and problem-solving performance [58]. These models improve accuracy and transparency by systematically decomposing complex tasks, making them more reliable for security applications.

Furthermore, LLM-based agents can now interface with external APIs, sensors, and actuators. Recent research demonstrates that LLMs can autonomously execute thousands of real-world tools by generating function calls in controlled environments [59]. Additionally, they can plan high-level actions for robotic systems without requiring extensive task-specific training [60]. This external integration marks the emergence of "embodied GPT" systems, where language models serve as interactive agents with both virtual and physical interfaces [61].

These advancements underscore the growing role of LLMs and agentic AI in cybersecurity, enabling automated threat detection, adaptive defense mechanisms, and intelligent decision-making in increasingly complex digital environments.

As a result, these models are increasingly seen as general-purpose AI agents capable of planning, reasoning, and decision-making [56,62]. They can be adapted to interactive environments, generate high-level policies, and exhibit consistency of style and coherence. Recent work highlights the potential to leverage LLMs for large-scale shared au-

tonomy and zero-shot tool usage, expanding their capabilities in practical and real-world settings [63,64].

Agentic AI represents a significant advancement in cybersecurity, shifting from traditional reactive measures to proactive threat management. As shown in Figure 2, the unique anatomy of agentic AI comprises key components that enable autonomous learning and real-time adaptation. Unlike conventional systems that rely heavily on predefined rules and patterns, agentic AI leverages advanced machine-learning algorithms to anticipate potential threats. This proactive capability allows it to neutralize threats in their early stages, preventing breaches before they occur. According to Mitchell et al. [65], deploying agentic AI in cybersecurity enables real-time monitoring and adaptive responses, significantly reducing the window of vulnerability.

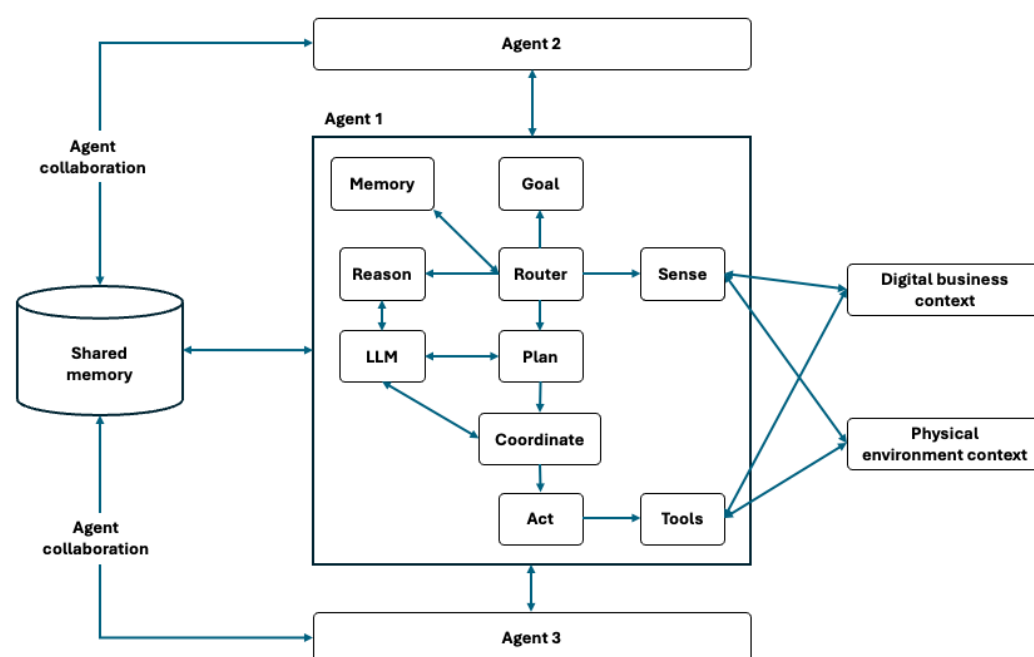


Figure 2. Agentic AI anatomy (adapted from [66]).

In addition to predictive capabilities, agentic AI systems are adept at orchestrating complex security workflows. For example, these systems can coordinate the deployment of patches, conduct penetration tests, and simulate attack scenarios to bolster an organization's defense mechanisms [67]. Cybersecurity frameworks can achieve unparalleled synergy by integrating LLMs with agentic AI, combining linguistic precision with autonomous operational efficiency.

An additional layer of interoperability emerges when integrating short-lived ontologies and High-Level Architecture (HLA) frameworks into agent-based systems. As demonstrated by Zacharewicz et al. [68], short-lived ontologies enable on-the-fly, non-persistent data exchange among autonomous agents. This technique aligns with agentic AI's real-time adaptability, enhancing system-level collaboration across distributed enterprise environments. By structuring information flow through temporary ontologies and orchestrating simulation agents via HLA standards, cybersecurity agents can flexibly exchange knowledge and respond dynamically to evolving threats without relying on fixed semantic schemas.

Recent research underscores the synergy between LLMs and agentic AI in addressing complex cybersecurity challenges. For instance, Paul et al. [69] demonstrated how LLMs can efficiently parse and interpret threat intelligence reports, enabling faster incident response times. Similarly, Zhou et al. [70] highlight the role of agentic AI in automating

vulnerability assessments, reducing manual intervention, and improving accuracy. These advancements are complemented by studies such as Radanliev et al. [71], which explore the ethical implications of deploying autonomous AI systems in cybersecurity, emphasizing the need for transparency and accountability.

Another critical area of research involves mitigating adversarial attacks on AI systems. Goodfellow et al. [72] introduced foundational concepts of adversarial examples, which continue to influence strategies for securing AI models. More recently, Johnson et al. [73] proposed novel defense mechanisms that leverage the capabilities of LLMs to detect and neutralize adversarial inputs in real-time, marking a significant step forward in AI-driven cybersecurity.

Despite the significant advancements in SOAR technologies, several critical gaps persist in the existing body of literature. Many implementations remain largely reactive, lacking mechanisms that support proactive and adaptive automation strategies. While no-code and low-code platforms have expanded accessibility, they often sacrifice flexibility, limiting the development of code-generative systems capable of nuanced decision-making. Interoperability continues to be a major challenge, as the orchestration of heterogeneous security tools is frequently hampered by incompatible APIs and disparate data formats. Moreover, current models commonly overlook adaptive knowledge representation techniques, such as vector-based memory and Retrieval-Augmented Generation (RAG), which are essential for preserving contextual continuity across incidents. To address these limitations, we propose a novel LLM-driven SOAR framework that leverages agentic AI and continuous learning mechanisms to enhance the precision, adaptability, and scalability of cybersecurity operations.

3. Methodology

3.1. The IVAM Framework

Traditional incident response methods often struggle to adapt to dynamic attack vectors and emerging threat tactics. To enhance the accuracy and comprehensiveness of incident response, this study introduces the IVAM Framework, a structured approach that translates incident response instructions into a systematic technical flow. As illustrated in Figure 3, the IVAM Framework ensures a more methodical and effective response strategy.

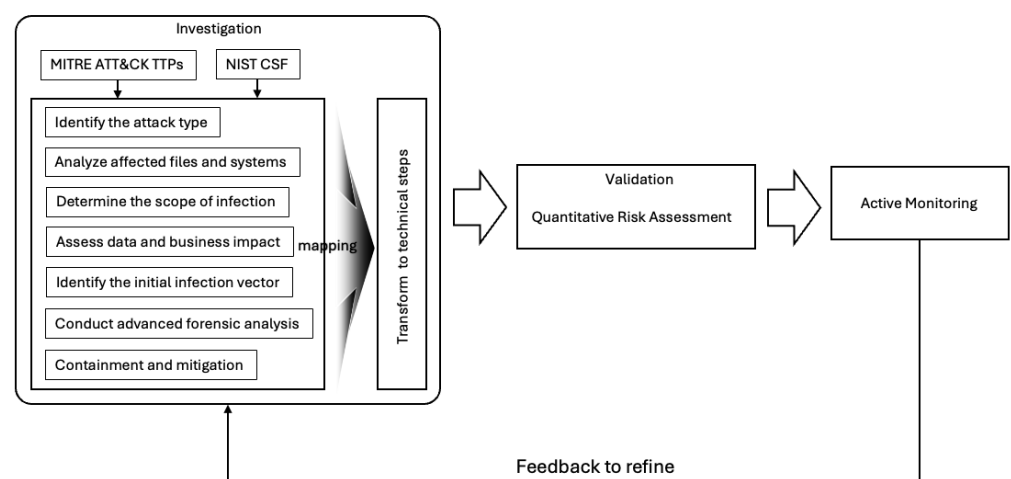


Figure 3. IVAM Framework translating incident response instructions into a systematic technical flow.

The IVAM Framework integrates three well-established cybersecurity methodologies to ensure a more structured and intelligence-driven response:

- MITRE ATT&CK Knowledge Base for mapping Tactics, Techniques, and Procedures (TTPs),
- NIST Cybersecurity Framework (CSF) for prioritization and procedural standardization,
- Quantitative Risk Assessment (QRA) for structured risk evaluation.

By leveraging these methodologies, the IVAM Framework provides a systematic, adaptable, and intelligence-driven approach to incident response, enhancing each phase of the response lifecycle. It is structured into three interdependent stages: Investigate, Validate, and Active Monitoring. Each stage focuses on a distinct aspect of the incident response lifecycle while informing and reinforcing the others.

3.1.1. Investigation Phase

The investigative process involves collecting data from logs, Security Information and Event Management (SIEM) systems, endpoint telemetry, and threat intelligence feeds. Forensic analysis identifies root causes and threat vectors, prioritizing high-value targets and critical vulnerabilities. To enhance precision, the investigation integrates the MITRE ATT&CK and NIST frameworks [74], mapping collected data into structured technical steps.

Identify the Attack Type

Initially, it is essential to collect indicators by gathering details from system alerts, user reports of suspicious logs, and automated detection tools such as Endpoint Detection and Response (EDR), Security Information and Event Management (SIEM), and Intrusion Detection Systems (IDSs). This comprehensive data collection lays the foundation for understanding potential security incidents.

Once the data are collected, the next step is to classify the attack. This involves using signatures, patterns, or threat intelligence to categorize the nature of the attack, be it malware, phishing, or a web application attack. Accurate classification is crucial for determining the appropriate response strategy.

After classification, assessing the severity of the attack is imperative. This assessment evaluates the risk based on factors such as the complexity of the attack, the capabilities of the attacker, and the initial impact observed. Understanding severity helps prioritize response efforts and effectively allocate resources to mitigate threats.

Analyze Affected Files and Systems

In the process of investigating potential security incidents, a thorough examination of system modifications is essential. This begins with reviewing file changes to identify any unusual modifications, unexpected file extensions, or unauthorized registry alterations. Such anomalies can be indicative of malicious activity or system compromise.

Concurrently, it is important to scrutinize system behavior by investigating logs and processes for anomalies. This includes monitoring for signs of privilege escalation, where unauthorized users gain elevated access rights, or abnormal resource usage that could suggest the presence of malicious processes or unauthorized applications consuming excessive system resources.

To further ensure system integrity, it is crucial to evaluate security tools. Using forensic tools can aid in detecting compromised files or executables, providing a deeper analysis of the system's state and uncovering hidden threats that may not be evident through standard monitoring.

By systematically reviewing file changes, monitoring system behavior, and employing forensic evaluations, organizations can effectively identify and respond to security incidents, thereby maintaining the integrity and security of their systems.

Determine the Scope of Infection

Identifying compromised endpoints is crucial for timely intervention in cybersecurity. Once these endpoints are identified, it is essential to monitor for lateral movement within the network. This involves analyzing authentication records and system-to-system connections to detect unauthorized access or the spread of malicious activity. Monitoring for discrepancies in administrative tasks and unusual login activities can provide insights into potential lateral movement.

Additionally, analyzing network traffic plays a vital role in threat detection. Reviewing logs from firewalls, proxies, and Intrusion Detection/Prevention Systems (IDS/IPS) can help identify signs of data exfiltration or command-and-control communications. Tools like Wireshark and tcpdump can be utilized to examine network connections to and from the system, aiding in detecting suspicious activities.

By systematically identifying compromised endpoints, monitoring for lateral movement, and analyzing network traffic, organizations can enhance their ability to detect, understand, and respond to security incidents efficiently.

Assess Data and Business Impact

In the aftermath of a security incident, conducting a thorough assessment is crucial to understanding its implications and determining appropriate response measures. This process involves classifying the data, assessing the operational impact, and evaluating compliance risks.

- **Classify Data:** Begin by identifying whether sensitive or regulated data, such as Personally Identifiable Information (PII), financial records, or intellectual property, was accessed or exfiltrated during the breach. Understanding the type of data compromised is essential for assessing potential risks and determining necessary remediation steps.
- **Assess Operational Impact:** Evaluate the extent of business disruption caused by the incident. This includes measuring downtime, loss of productivity, and the resources required for recovery efforts. Understanding the operational impact helps in prioritizing response actions and allocating resources effectively.
- **Evaluate Compliance Risks:** Identify any legal obligations arising from the breach, such as those under the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), or Payment Card Industry Data Security Standard (PCI-DSS). Determine if notifications to affected individuals or regulatory bodies are necessary, and ensure compliance with relevant laws to mitigate potential legal consequences.

Identify the Initial Infection Vector

This process involves inspecting communication channels, reviewing access logs, investigating web applications, and considering physical access points.

Inspect Communication Channels: Begin by examining email gateways for signs of phishing attempts or malicious attachments. Phishing remains a prevalent method for attackers to gain unauthorized access, making it crucial to monitor and filter incoming emails effectively. Implementing robust email filtering solutions and educating employees on recognizing phishing attempts can significantly reduce this risk.

- **Review Access Logs:** Analyze logs from Remote Desktop Protocol (RDP), Virtual Private Network (VPN), and Secure Shell (SSH) services to detect unauthorized access or brute-force attempts. Unusual login times, failed login attempts, and access from unfamiliar IP addresses can indicate potential security breaches. Regularly reviewing these logs helps in the early detection of unauthorized activities.
- **Investigate Web Applications:** Examine server logs for any signs of exploitation, such as abnormal requests or injection attacks. Web applications are common targets for attackers seeking vulnerabilities to exploit, necessitating regular monitoring and prompt patching of identified issues. Utilizing Web Application Firewalls (WAFs) and conducting regular security assessments can enhance protection.
- **Consider Physical Access:** Assess the potential for security breaches through physical means, including the use of removable media or insider threats. Unauthorized physical access to systems can lead to data theft or the introduction of malicious software, highlighting the need for strict access controls and monitoring. Implementing measures such as surveillance systems, access badges, and security personnel can mitigate these risks.

By systematically inspecting communication channels, reviewing access logs, investigating web applications, and considering physical access points, organizations can effectively identify vulnerabilities and respond to security incidents, thereby maintaining the integrity and security of their systems.

Conduct Advanced Forensic Analysis

Following a security incident, it is essential to conduct a thorough forensic examination to identify the root causes and understand the adversary's methods. This process involves several key steps:

- **Perform Forensic Analysis:** Begin by conducting memory forensics and disk analysis to uncover the root causes of the incident. Memory forensics involves capturing and analyzing the contents of a computer's volatile memory (RAM) to identify malicious processes, open network connections, and other artifacts that may not be present on the disk. Disk analysis complements this by examining the file system and storage media for malicious files, logs, and other persistent indicators of compromise. Tools such as The Sleuth Kit can assist in this analysis.
- **Map to MITRE ATT&CK:** Utilize the MITRE ATT&CK framework to identify the Tactics, Techniques, and Procedures (TTPs) employed by the adversaries. This globally accessible knowledge base categorizes adversary behaviors observed in real-world attacks, aiding in understanding and anticipating potential threat actions.
- **Correlate with Threat Intelligence:** Compare the findings from your forensic analysis against known attack groups or malware families. By correlating observed TTPs with threat intelligence reports, you can attribute the attack to specific adversaries and understand their motivations and capabilities. This correlation enhances your organization's ability to defend against future attacks by informing proactive security measures.

Containment and Mitigation

In the event of a security incident, implementing immediate and effective response measures is crucial to mitigate damage and restore system integrity. The following steps outline key actions to be taken:

- **Isolate Affected Systems:** Promptly remove compromised endpoints from the network to prevent the spread of malicious activity. This containment strategy is essential to limit further damage and is a critical component of incident response frameworks.
- **Block Malicious Entities:** Update security tools, such as firewalls and intrusion prevention systems, to block identified malicious IP addresses, domains, and file hashes. This proactive measure helps prevent further exploitation by known threats.
- **Secure and Patch:** Apply the latest security updates to all systems to address vulnerabilities exploited during the incident. Reset compromised credentials to prevent unauthorized access and enforce hardened configurations to enhance system defenses.
- **Implement Best Practices:** Enforce the principle of least privilege by ensuring users have only the access necessary for their roles. Implement multi-factor authentication to add an extra layer of security and establish network segmentation to contain potential threats and limit their movement within the network.

By executing these steps, organizations can effectively contain security incidents, mitigate their impact, and strengthen defenses against future threats.

3.1.2. Validation Phase

Once actions have been implemented, the framework employs Quantitative Risk Assessment (QRA) methods. AI agents estimate the probable impacts of specific incidents, including financial, reputational, or operational impacts.

Challenges of Asset Valuation

Accurate asset valuation presents significant challenges in modern, interconnected systems. Many organizations rely on intangible assets such as intellectual property, customer trust, or operational uptime, which lack fixed monetary equivalents. Additionally, asset values fluctuate due to market dynamics, reputational impacts, and external conditions, making static valuation methods unreliable.

Furthermore, financial and asset-specific data are often sensitive or confidential. Including such details in risk assessments can introduce security and privacy risks. Asset valuation may impose unnecessary burdens for smaller organizations or those with limited resources, hindering effective risk management [75,76].

Focus on Relative Quantification

Excluding asset value shifts, QRA moves toward relative risk quantification, prioritizing comparative risk assessment over absolute financial estimates. Here, risk is framed as a function of likelihood and impact, where impact is evaluated across operational, reputational, and other relevant dimensions.

This approach aligns with the practical needs of AI agents, which prioritize actionable insights over precise financial estimations. By simplifying assessment processes and emphasizing agility, this method enables security teams to make rapid response and recovery decisions without exhaustive asset valuation.

Addressing Information Constraints

In practical scenarios, obtaining accurate and up-to-date asset valuation data for risk assessment poses significant challenges:

- **Intangible Risks:** Factors such as reputational damage and regulatory penalties often lack direct ties to specific assets, making their valuation complex.

- **Supply Chain Vulnerabilities:** Involving external stakeholders complicates comprehensive asset valuation due to varying data availability and reliability across the supply chain.
- **Dynamic Operational Environments:** Rapid changes in operations can lead to fluctuating asset values, rendering static estimates unreliable and necessitating continuous monitoring.

These challenges underscore the need for adaptable and robust risk assessment methodologies that can accommodate the inherent uncertainties in asset valuation. To establish a Quantitative Risk Assessment (QRA) approach aligned with the NIST Cybersecurity Framework (CSF), the methodology typically involves calculating risk as a function of likelihood and impact. Since asset value information is excluded, the focus is on relative quantification rather than on absolute financial values. Here is a structured formula and breakdown:

$$R = P \cdot I$$

where:

R: Relative Risk,

P: Probability of the incident (likelihood of occurrence),

I: Impact of the incident in terms of financial, reputational, or operational effects.

$$R = (L \times I) \times (1 - E)$$

where:

L: Likelihood of the incident occurring,

I: Impact of the incident in terms of financial, reputational, or operational effects,

E: Mitigation effectiveness (proportion of the impact mitigated).

By excluding asset value, QRA acknowledges these constraints and provides a streamlined, scalable approach for assessing risks. The relative metrics derived from likelihood and impact facilitate timely and actionable decision-making, even in resource-constrained environments.

3.1.3. Active Monitoring

The final stage underlines the continuous, iterative nature of effective incident response. Active Monitoring ensures that any intelligence gained and actions taken are fed back into the system in real time. If emergent threats or anomalies are detected, the cycle begins anew with an updated Investigate phase.

In practice, Active Monitoring involves a culture of constant refinement, where lessons learned feed directly into ongoing risk assessments and strategy, ensuring that the security posture continues to evolve in parallel with the threat landscape.

By weaving these three stages together, Investigate, Action, Validate, and Active Monitoring, the IVAM framework aims to deliver a balanced and repeatable approach to incident mitigation. Each phase draws on globally accepted standards and methodologies, ensuring that decisions are not made in isolation but rather anchored in risk-based insights, threat intelligence, and robust validation techniques. Over time, organizations implementing this framework can expect to reduce blind spots, streamline their response processes, and strengthen their overall resilience against adversarial tactics.

3.2. Agentic AI Security Response Construction

Agentic AI is an intelligent system that can autonomously analyze, decide, and execute security actions with minimal or no direct human intervention. Unlike traditional AI,

agentic AI possesses self-governing capabilities, enabling adaptive decision-making and orchestrated security operations across distributed environments.

A key advancement in this field is agentic AI security response, which integrates AI-driven autonomous agents capable of proactive and coordinated responses to cyber threats. One essential aspect of this response mechanism is Security Response Orchestration, which involves coordinating multiple security tools, processes, and workflows to streamline incident detection, analysis, and mitigation.

Table 1 outlines the core responsibilities of the SOAR (Security Orchestration, Automation, and Response) Advisor Assistant, a key component in the AI-driven agentic security response. This AI-powered assistant enhances cybersecurity operations by performing various critical functions:

- Incident Response Analysis & Generation: Analyzes log data and problem reports to detect security threats using industry-standard frameworks.
- Incident Mitigation & Resolution: Provides mitigation strategies aligned with NIST CSF 2.0 and MITRE ATT&CK and generates remediation steps, including playbook automation.
- Automation & Technical Guidance: Ensures security responses follow SOAR best practices and offers step-by-step technical procedures.
- Security Research & Advisory: Utilizes vector databases and security repositories to provide evidence-based security recommendations.
- Conversational Efficiency & Memory: Engages in professional, context-aware interactions while maintaining conversation history for enhanced accuracy.

Table 1. System prompt format for SOAR advisor assistant.

System Prompt Format
1. Incident Response Analysis & Generation: <ul style="list-style-type: none">- Analyze log data and problem reports.- Identify security threats using industry-standard frameworks.
2. Incident Mitigation & Resolution: <ul style="list-style-type: none">- Provide mitigation strategies aligned with NIST CSF 2.0 and MITRE ATT&CK.- Generate remediation steps, including playbook automation.
3. Automation & Technical Guidance: <ul style="list-style-type: none">- Offer step-by-step response procedures.- Ensure technical flow follows SOAR best practices.
4. Security Research & Advisory: <ul style="list-style-type: none">- Utilize vector databases and security repositories.- Provide evidence-based guidance.
5. Conversational Efficiency & Memory: <ul style="list-style-type: none">- Engage professionally and contextually with users.- Maintain conversation history for improved accuracy.

This structured approach allows agentic AI-driven security systems to autonomously manage cybersecurity incidents, reducing response times and improving the overall effectiveness of security operations.

3.2.1. End-to-End Mitigation Workflow Powered AI-Agent

The mitigation workflow, as depicted in Figure 4, follows a structured process for detecting, analyzing, and mitigating cyber threats by integrating AI-driven recommendations with human verification. This approach ensures a balance between automation and expert oversight, optimizing security responses while minimizing potential risks.

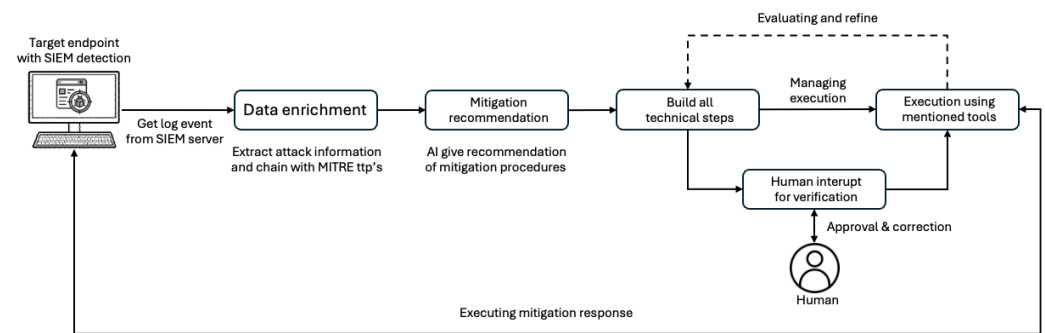


Figure 4. Proposed incident response end-to-end workflow.

The process begins with Log Event Collection, where a target endpoint equipped with SIEM detection identifies a security event. The event log, which contains crucial details about the detected attack, is then retrieved from the SIEM server.

Once collected, the data are moved to the Data Enrichment module. Here, the system extracts critical attack information and maps it to MITRE TTPs (Tactics, Techniques, and Procedures) to better understand the attack’s technical characteristics. This enrichment process ensures that security teams comprehensively view the threat.

Using this enriched data, the Mitigation Recommendation phase leverages an AI-driven system to generate a precise response plan. These recommendations align with the MITRE ID technical procedures, ensuring that the proposed mitigation steps are accurate and relevant to the nature of the attack.

The next phase, Building Technical Steps, involves formulating an action plan that outlines the necessary steps for mitigation. These steps serve as a structured guide to ensure an effective response.

Before execution, the workflow incorporates a Human Verification and Approval stage, where a security analyst reviews the proposed steps. The analyst can approve, modify, or refine the plan, ensuring that the response is both effective and appropriate for the situation.

Once verified, the process moves to Execution of Mitigation, where the approved steps are carried out using dedicated security tools. This may involve applying security controls, blocking malicious activities, or deploying patches to neutralize the threat.

After the mitigation steps are executed, the process moves into the Evaluation and Refinement stage, where the effectiveness of the implemented measures is carefully assessed. This phase is crucial for determining whether the sent commands successfully neutralized the threat or require further improvement to achieve optimal performance.

If the instructions executed effectively mitigate the issue without unintended consequences, they are recorded as validated response actions for future use. However, if adjustments are needed, the system refines the approach by modifying commands, tuning execution parameters, or restructuring the mitigation sequence. This iterative process helps identify the most efficient and reliable response strategies based on real-world results.

By continuously analyzing whether each executed instruction needs improvement or has already achieved success, this phase ensures that the system consistently evolves and adapts, enhancing overall threat response performance over time.

3.2.2. Agentic AI Building Blocks and Function

Figure 5 illustrates a proposed multi-agent system architecture designed to automate and enhance security incident response workflows. The proposed system encompasses three main layers: (i) planning and data management, (ii) modeling organization and task decomposition, and (iii) tooling and execution. This layered approach ensures modularity,

facilitates scalability, and provides robust interfaces to external systems, such as a Security Information and Event Management (SIEM) platform.

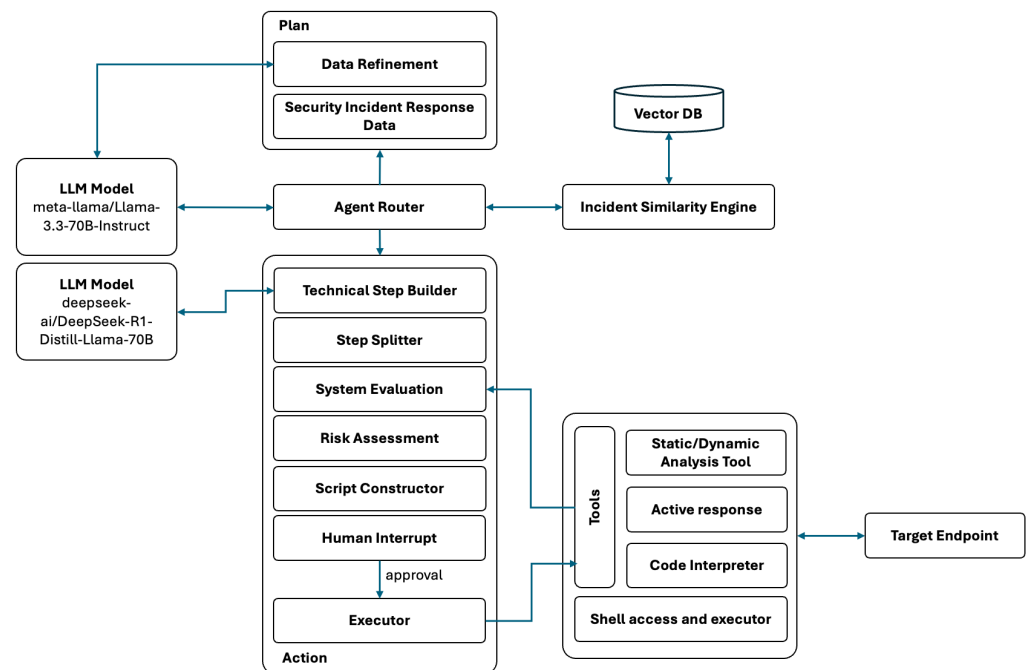


Figure 5. Agentic AI high-level system architecture for automated security incident response.

Leveraging LLMs for Agent-Powered Security Systems

DeepSeek-R1-Distill-Llama-70B [77] and Meta’s Llama-3.3-70B-Instruct [78] represent significant advancements in large language model development, each contributing uniquely to the field of artificial intelligence. DeepSeek-R1-Distill-Llama-70B is a distilled version of DeepSeek’s R1 model, fine-tuned from the Llama-3.3-70B-Instruct base model [79]. This model leverages knowledge distillation to retain robust reasoning capabilities while enhancing efficiency, achieving superior performance on complex mathematics and coding tasks compared to larger models.

On the other hand, Meta’s Llama-3.3-70B-Instruct is a text-only, instruction-tuned model optimized for multilingual dialogue use cases. It outperforms many open-source and closed-chat models on common industry benchmarks, providing enhanced performance relative to its predecessors [78]. Collectively, these models exemplify the rapid progress in developing efficient, high-performing language models capable of handling complex reasoning and multilingual tasks.

Planning and Data Management

At the core of the system is an advanced data refinement and management process, ensuring all security events and contextual information are pre-processed for downstream analysis. The system integrates the following:

- **Data Refinement:** The system cleanses, labels, and structures incoming event data for efficient processing.
- **Security Incident Response Data:** Historical logs, real-time event monitoring, and domain-specific threat intelligence inform analysis and decision-making.
- **Vector Database (Vector DB):** A specialized database storing the embeddings of security incidents, enabling rapid retrieval of past events with similar characteristics. This capability aids in contextualizing new threats and supports proactive response measures.

Model Orchestration and Task Decomposition

An effective security response requires intelligent task management and model orchestration, achieved through the following:

- **Agent Router:** An agent component that routes queries and tasks to the appropriate Large Language Model (LLM) based on predefined rules, model specialization, or real-time performance metrics. It also utilizes the incident similarity engine, retrieving relevant cases from the vector DB to enhance contextual understanding.
- **LLM Models:** Two specialized LLMs power the system:
 - LLM Model Llama-3.3-70B-Instruct: Focuses on broad security policies, general text processing, and strategic threat mitigation.
 - LLM Model deepseek-ai/DeepSeek-R1-Distill-Llama-70B: Designed for deep, domain-specific security analysis and advanced query resolution.

Tooling and Execution Component

As illustrated in Figure 6, the Tooling and Execution component of the proposed system is designed to systematically manage security incidents through a structured pipeline, ensuring that each response is thorough, auditable, and effective.

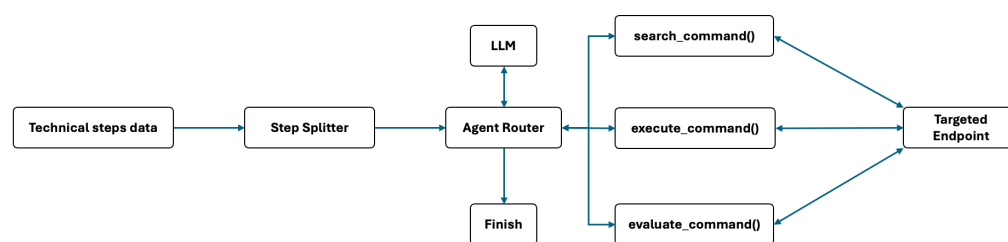


Figure 6. Agent workflow illustrating tool integration and result evaluation.

Technical Step Builder

The Technical Step Builder is a structured pipeline that decomposes an overarching response strategy into smaller, auditable tasks, ensuring a systematic and efficient approach to incident management. Aligned with established incident management frameworks, this methodology emphasizes structured processes for effectively handling security threats.

The process begins with the step splitter, which breaks down large, complex tasks into manageable subtasks. This decomposition allows teams to address each component systematically, reducing the risk of oversight and improving execution clarity.

Following this, the system evaluation phase assesses the environment, identifying critical assets and evaluating system health metrics. A built-in feedback loop strengthens this step, collecting insights from previous response actions on targeted endpoints. These evaluations enable the AI agent to refine its strategies, optimizing mitigation efforts for both effectiveness and resource efficiency.

Subsequently, the risk assessment stage determines the potential impact and severity of the incident or proposed actions. Highlighting associated risks ensures informed decision-making, allowing teams to implement protective measures that minimize operational and security disruptions.

To accelerate the response process, the script constructor automates the creation of scripts for threat mitigation and incident response. Standardized execution through automation is essential, particularly during time-sensitive incidents that require swift and precise actions. Before execution, a Human Interrupt phase introduces expert review,

refinement, and approval of the proposed actions. This step integrates human judgment and contextual understanding, ensuring necessary adjustments for nuances that automated systems might overlook.

Implementing this structured pipeline enhances incident response efficiency, enabling precise threat management while balancing automation and human oversight.

Agent Executor

The executor plays a critical role in executing approved tasks through a suite of specialized tools. These tools facilitate comprehensive analysis, real-time mitigation, and system modifications, ensuring an efficient and effective incident response process. A key component of this system is the integration of static and dynamic analysis tools, which examine files and binaries to detect potential malicious content. These tools identify threats by analyzing behavioral patterns and code structures, enabling early detection and response.

Beyond detection, the executor employs active response mechanisms that automate real-time mitigations. This includes blocking malicious IPs, isolating compromised hosts, and deploying necessary patches to neutralize threats before they escalate. The system incorporates a code interpreter capable of executing or interpreting code dynamically to enhance adaptability. This feature allows for flexible incident response strategies, ensuring rapid adaptation to new and evolving threats.

For more direct intervention, the executor component of the system provides shell-level access, enabling real-time command-line execution of critical system modifications. This capability empowers security teams to implement immediate changes when necessary, thereby reinforcing the system's responsiveness and operational agility. By combining advanced orchestration tools with targeted automation, the executor ensures a seamless and proactive security posture, one that minimizes risks while preserving both system integrity and performance.

Complementing the executor's capabilities is the WebSocket Agent Client, a lightweight, Command-Line Interface (CLI) utility designed to serve as a remote execution endpoint within a distributed computing environment. Its principal function is to establish and sustain a persistent WebSocket connection with a centralized execution service, continuously listen for incoming command payloads, execute those commands securely within the local host context, and transmit structured execution results back to the orchestration layer in real time. This design paradigm facilitates low latency, bidirectional communication between the control plane and underlying infrastructure, thereby enabling a robust and extensible execution pipeline that can operate reliably across heterogeneous systems.

During initialization, the agent parses command-line arguments to extract its unique identifier denoted `agent_id` alongside the target WebSocket server address. It then constructs the appropriate endpoint URL and attempts to establish a stable connection via the Gorilla WebSocket library. Should the connection attempt fail, the agent terminates gracefully while providing a descriptive error log. Once the connection is successfully established, the agent enters a continuous execution loop wherein it awaits and processes command instructions. Each received command is executed within a POSIX-compliant shell, and the resulting output, comprising standard output, standard error, and a status indicator, is encapsulated in a structured JSON response. The logic underpinning this operational workflow is formally articulated in Algorithm 1.

Once connected, the agent enters a continuous listening loop, wherein it awaits messages transmitted from the server. Each message is interpreted as a shell command that the agent is expected to execute locally. The `executeCommand` function is responsible for handling the execution logic. It invokes the command using a POSIX-compliant shell (sh

-c) and captures the combined standard output and standard error streams. A status flag is then assigned based on the success or failure of the command execution.

Algorithm 1: WebSocket Agent Client (Command Executor)

```

Input: Command-line arguments: agent ID, server host
Output: Execution result returned to WebSocket server
Fn() agentID ← parse -id flag
host ← parse -host flag (default: localhost:8080)
if agentID is empty then
  | exit with error message
end
url ← "ws://" + host + "/ws/agent/" + agentID
conn ← open WebSocket connection to url
if connection fails then
  | exit with connection error
end
Print "Connected as agent ID"
while true do
  message ← read WebSocket message
  if error then
    | log and break
  end
  Print "Received command"
  stdout, stderr, status ← message
  response ← { agentID, status, stdout, stderr }
  json ← marshal response to JSON
  send json to server over WebSocket
  if send error then
    | log and break
  end
  Print "Sent result to server"
end
close WebSocket connection

Fn(command) cmd ← execute shell command using sh -c command
stdout ← combined output from command
if command fails then
  | log error
  | return stdout, error message, "error"
end
sleep for 1 second
return stdout, "", "success"

```

After execution, the agent encapsulates the results, including the agent identifier, execution status, standard output, and any error messages, into a structured JSON payload. This payload is then transmitted back to the server over the existing WebSocket connection. The use of structured response objects ensures that downstream systems can reliably parse and act upon the execution outcome.

The agent supports fault-tolerant behavior through consistent error logging and connection handling. Should any read or write operation on the WebSocket stream fail due to network disruption or server-side disconnection, the agent logs the issue and exits

cleanly. To simulate realistic latency and prevent resource exhaustion, a short artificial delay is introduced post execution.

This agent design facilitates real-time, bidirectional command execution and monitoring, making it a suitable component within a broader Security Orchestration, Automation, and Response (SOAR) framework; remote operations system; or cloud-based infrastructure automation platform.

3.3. System Automation Flow

Integrating a vector database into an incident response system significantly enhances efficiency and effectiveness. By storing incident data, historical records, and corresponding mitigation processes, the system can autonomously respond to recurring threats. When a security incident occurs, the database stores its characteristics and mitigation steps as vector representations. This approach captures complex relationships and patterns within the data, ensuring valuable insights are retained for future use.

Upon encountering a new incident, the system converts its attributes into a vector and searches the database for similar cases. Identifying incidents with comparable features can quickly retrieve effective response strategies, reducing analysis time and accelerating decision-making. Once a match is found, the system automatically retrieves the corresponding mitigation steps and adapts them to the current context. This includes fine-tuning parameters such as the target endpoint and necessary dependencies, enabling a precise and efficient response without manual intervention.

This methodology streamlines the incident response process, ensuring faster, more consistent, and highly accurate resolutions by leveraging previously validated solutions. Vector databases are particularly valuable in managing the complexity and scale of modern cybersecurity data, offering efficient high-dimensional querying essential for real-time threat detection and response in large-scale systems.

3.4. Agent Implementation Overview and Flow Logic

The core of the Agent AI architecture is a state transition graph that systematically routes incident input through a sequence of modular processing nodes, each responsible for a distinct cognitive or executional function. At the highest level of abstraction, the system ingests an input, whether provided by a user or generated by a machine, and classifies it into one of several predefined categories: Extracted Data (ED), Incident Response (IR), structured JSON logs, or general free-form queries.

The architecture is intentionally modular, comprising a chain of explicitly defined computational components that collectively enable automated, adaptive, and explainable incident analysis and remediation. As illustrated in Algorithm 2, each node in the graph serves a dedicated role within the pipeline, ranging from input triage and semantic classification to command generation, agent-based execution, and post-execution reporting. This section provides a comprehensive exposition of these functional components, with emphasis on their orchestration and conditional interactions within the system's control flow.

3.4.1. Routing and Input Classification

At the entry point of the workflow lies the `router_node`, a pivotal component responsible for the initial triage and classification of user provided inputs. This node discerns the structure and intent of the input, categorizing it as general free-form text, an incident response directive, or a structured JSON-based log. In cases where the input lacks sufficient structure or is deemed conversational, it is routed to a fallback mechanism, the `chatbot_node`, which generates human-like responses through a language model without invoking the entire analytical pipeline. In contrast, structured inputs trigger downstream components designed for deeper semantic parsing and technical processing.

Algorithm 2: Agent Graph Flow

```

Data: Input from user or log
Result: Incident resolved and report generated
Start with input
Call router_node(input)
Fn(router_node(input)) classify ← determine_input_type(input)
if classify is "ED" then
  | return incident_response_node(input)
else if classify is "IR" then
  | return technical_steps_node(input)
else if classify is "JSON" then
  | return incident_similarity_node(input)
else
  | return chatbot_node(input)
end
Fn(incident_similarity_node(input)) similar ←
  check_incident_memory(input)
if similar then
  | return technical_similarity_node
else if early_stop then
  | return END
else
  | return extracted_problem_node(input)
end
Fn(extracted_problem_node(input)) problem ← extract_problem(input)
return incident_response_node(problem)
Fn(incident_response_node(problem)) response ←
  generate_incident_response(problem)
return technical_steps_node(response)
Fn(technical_steps_node(response)) steps ←
  generate_technical_steps(response)
return splitted_steps_node(steps)
Fn(splitted_steps_node(steps)) list ← split_steps(steps)
return human_interrupt_node(list)
Fn(human_interrupt_node(list)) if more_steps then
  | return script_capture_node(current_step)
else
  | return report_node()
end
Fn(script_capture_node(step)) commands ← capture_script(step)
return executor_node(commands)
Fn(executor_node(commands)) foreach command in commands do
  | result ← run_command(command)
  | if result = ERROR then
  | | return human_interrupt_node(current_step)
  | end
end
if last_step then
  | return report_node()
else
  | return script_capture_node(next_step)
end
Fn(report_node()) report ← compile_final_report()
save and index report
return END

```

3.4.2. Similarity-Based Retrieval

For structured inputs, particularly those formatted as JSON logs, Agent AI implements a retrieval-augmented reasoning mechanism through the `incident_similarity_node`. This node leverages dense vector representations and similarity search techniques to identify previously encountered incidents that exhibit high contextual overlap with the current input. Upon identification of a sufficiently similar prior case, the system retrieves the associated resolution steps and responses, thereby obviating the need for redundant analysis. This capability significantly improves response efficiency, supports case-based reasoning, and promotes knowledge reuse within the incident resolution corpus.

3.4.3. Problem Extraction and Response Generation

In scenarios where no prior incident is deemed sufficiently similar, the system transitions into a generative reasoning phase. The `extracted_problem_node` employs a large language model to abstract and articulate the underlying problem from the input, translating raw logs or text into a well-defined problem statement. This extracted information serves as the foundation for the `incident_response_node`, which formulates a high-level mitigation strategy tailored to the described issue. The strategy is subsequently transformed into executable remediation steps by the `technical_steps_node`, which synthesizes detailed technical actions from the prior abstraction. This sequential transformation, from problem recognition to action planning, is the backbone of the adaptive intelligence of the system.

3.4.4. Stepwise Execution and Control Flow

Following the generation of technical instructions, the system initiates a stepwise execution pipeline. The `splitted_steps_node` decomposes the technical response into a sequence of discrete, logically ordered steps. Prior to execution, the `human_interrupt_node` offers an opportunity for human validation or intervention, ensuring oversight in critical workflows. Once validated, each step is passed to the `script_capture_node`, where it is converted into command-line instructions or script templates. These commands are then executed by the `executor_node`, which orchestrates interaction with external tools or environments through agent-based mechanisms.

Execution outcomes are evaluated using an LLM-powered result classifier that determines the success or failure of each command based on predefined criteria or learned heuristics. Successful steps lead to continued progression through the pipeline, whereas failures trigger either a retry sequence or a return to human oversight. This design ensures a balance between automation and reliability, particularly in complex operational contexts.

3.4.5. Post Execution Reporting and Indexing

Upon successful execution of all planned steps, or in cases of early termination, the system invokes the `report_node` to synthesize a comprehensive, human-readable summary of the incident resolution process. This report aggregates the problem description, strategic and technical responses, execution logs, and classification results into a coherent narrative. Furthermore, the incident is indexed within a persistent memory store to support future similarity retrieval and trend analysis. This closing mechanism not only improves traceability but also contributes to the continual learning and refinement of the operational knowledge base of the system.

4. Agent Validation Result

To assess the feasibility of an AI-driven Security Orchestration, Automation, and Response (SOAR) system, a pilot deployment was implemented to receive enriched security alerts, such as brute-force attack attempts from a pre-configured Security Information and

Event Management (SIEM) platform, specifically Wazuh [80]. The primary objective of this study is to examine the automation and orchestration capabilities of the response pipeline rather than the detection mechanisms themselves.

4.1. Deployment

The experimental setup was initially deployed in a virtualized environment utilizing VirtualBox on Ubuntu 22.04.5 LTS, hosted on a 13th-generation Intel Core i7 processor. Following the official Wazuh documentation [81], a proof-of-concept for brute-force attack detection was configured. Automated responses were subsequently implemented through predefined scripts, with each mitigation action evaluated using quantitative risk assessment metrics.

As illustrated in Figure 7, once an incident is identified, such as an SSH brute-force attempt involving a non-existent user, the SOAR system enriches the alert with contextual metadata, including MITRE ATT&CK Tactics, Techniques, and Procedures (TTP) classification; regulatory compliance implications; and log summaries. Following enrichment, the system autonomously executes a multi-step response. This includes blocking the offending IP address and enforcing Multi-Factor Authentication (MFA) policies via guided response scripts. Each response action is executed sequentially and monitored through a corresponding risk assessment framework, which demonstrates a measurable reduction in security risk at each stage of the response lifecycle.

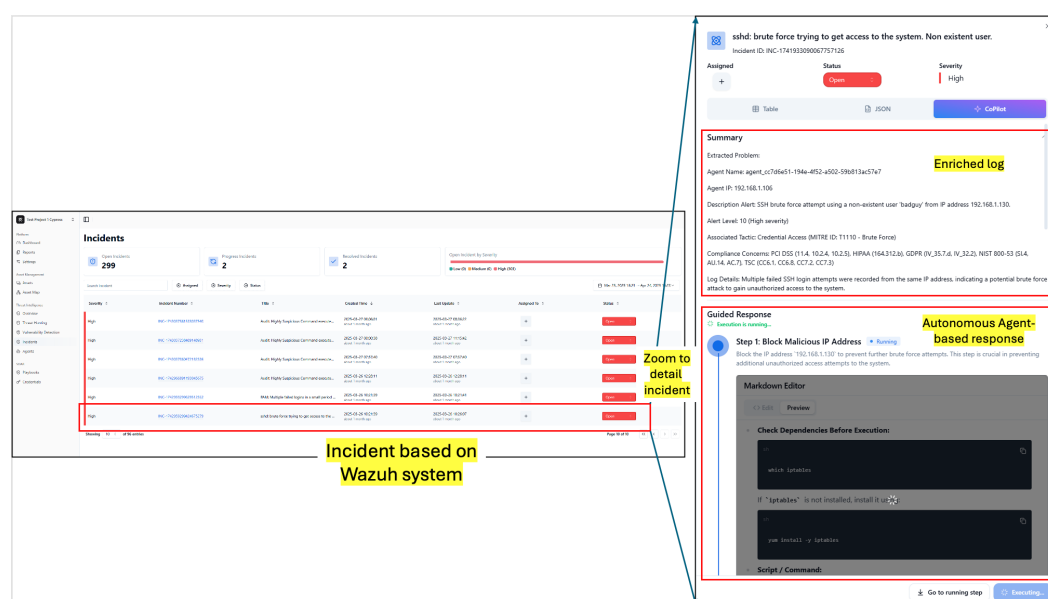


Figure 7. Pilot deployment and integration with Wazuh SIEM as detection platform.

The proposed SOAR framework was validated within a controlled, SOC (Security Operations Center)-like environment, wherein the system successfully processed real-time security alerts and autonomously executed mitigation workflows without human intervention. The results of this pilot study provide empirical evidence supporting the operational viability of an autonomous, agent-based SOAR architecture.

Subsequent iterations of this work will focus on the collection and analysis of quantitative performance metrics, including Mean Time to Respond (MTTR), the success rate of automated mitigation actions, and the extent of reduction in human analyst intervention. These metrics will be benchmarked against those of conventional SOAR platforms, thereby enabling a comprehensive evaluation of the operational efficiencies introduced by the proposed AI-driven response system.

4.2. Data Enrichment

Brute-force attacks generate extensive raw logs that can overwhelm security analysts with low-level messages and fragmented information. Transforming these raw entries into actionable intelligence requires contextual enrichment, an automated process that augments basic log data with meaningful security insights. By integrating AI-enhanced data enrichment into traditional intrusion detection systems (e.g., Wazuh), AI agents can promptly identify critical details such as attack timelines, related hosts, and suspicious behavioral patterns.

As illustrated in Figure 8, the refined data yields a clearer overview of the affected target endpoint, the specific type of brute-force attack, and standardized metadata on each event.

```

input_type : JSON
description : value
extracted_problem :
  Extracted Problem:
    Agent Name: agent-test01
    Agent IP: 192.168.1.76
    Agent Platform: Linux
    Description Alert: PAM: Multiple failed logins in a small period of time from IP
    192.168.1.130.
    Alert Level: 10 (High severity)
    Associated Tactic: Credential Access (MITRE ID: T1110 – Brute Force)
    Compliance Concerns: PCI DSS (10.2.4, 10.2.5, 11.4), HIPAA (164.312.b), TSC (CC6.1, CC6.8,
    CC7.2, CC7.3), NIST 800-53 (AU.14, AC.7, SI.4), GDPR (IV_35.7.d, IV_32.2), GP613 (7.8)
    Log Details: Full log message reads "Feb 18 05:58:47 linuxasep-VirtualBox sshd[21621]:
    pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser=
    rhost=192.168.1.130," indicating multiple failed SSH authentication attempts from the same
    source IP.
  Incident_response :
    ## Incident Summary
    - **Attack Type:** Brute Force (Multiple Failed Logins)
    - **Severity Level:** High (Alert Level 10)
    - **MITRE ATT&CK Mapping:** T1110 (Brute Force)
    - **Compliance Concerns:** PCI DSS (10.2.4, 10.2.5, 11.4), HIPAA (164.312.b), TSC (CC6.1,
    CC6.8, CC7.2, CC7.3), NIST 800-53 (AU.14, AC.7, SI.4), GDPR (IV_35.7.d, IV_32.2), GP613
    (7.8)
  
```

Figure 8. Contextually enriched brute-force attack detection output, highlighting the identified target IP, corresponding MITRE ATT&CK technique, relevant compliance requirements, and recommended incident response actions.

4.3. Agent-Based Mitigation Result

After identifying the security breach context, the next step involves feeding the extracted information into the Large Language Model (LLM) agent to generate a detailed response recommendation. Leveraging its advanced natural language processing capabilities, the LLM analyzes key aspects of the breach, such as its nature, scope, and potential vulnerabilities, and formulates a systematic course of action. As shown in Table 2, the LLM agent breaks down the response into a series of technical steps, addressing both immediate threats and long-term mitigation strategies.

Compared to the traditional Security Orchestration, Automation, and Response (SOAR) workflow outlined in the Brute-Force Investigation—Generic playbook [82], which spans 38 distinct steps, the AI-driven approach streamlines mitigation tasks into ten key actions. This optimized methodology highlights the LLM agent's ability to abstract, prioritize, and consolidate remediation efforts, focusing on essential technical and procedural elements rather than on detailing every granular sub-step. While the legacy SOAR workflow offers a highly detailed breakdown of individual tasks, the AI-driven approach enhances conceptual clarity, operational efficiency, and adaptability, ensuring that critical countermeasures are executed swiftly and effectively in response to a brute-force attack.

The AI agent offers a detailed breakdown of each step, explicitly integrating security tools and best practices. For instance, when mitigating threats, it recommends that Fail2Ban blocks malicious activity and enforces password resets as an essential security measure. In contrast, XSOAR follows a more predefined approach by implementing threshold-based

blocking and account lockouts, which, while effective, may lack adaptability to evolving security threats.

A strong emphasis on access control is evident in both approaches, particularly in enhancing Multi-Factor Authentication (MFA) and strengthening SSH security configurations. However, the AI agent provides more granular recommendations, such as using Google Authenticator for SSH access and enforcing stricter login policies, like disabling root login and limiting authentication attempts. These measures demonstrate a proactive approach to preventing unauthorized access.

Forensic analysis is another crucial component of incident response. The AI agent suggests using AIDE for file integrity checks and audit to monitor critical files, ensuring the real-time detection of unauthorized modifications. While XSOAR also focuses on deep-dive investigations, it does not specify particular tools, implying that additional customization may be required for comprehensive forensic analysis.

When it comes to remediation and system restoration, the AI agent consolidates multiple steps, such as isolating compromised systems, blocking malicious IPs, and reconfiguring security settings into a streamlined recovery process. XSOAR, on the other hand, treats remediation as a distinct phase, which may provide a structured but lengthier approach to restoring normal operations.

Incident documentation and compliance play a significant role in security management. Both methodologies emphasize thorough recordkeeping, but the AI agent explicitly aligns its response with compliance frameworks such as PCI DSS, HIPAA, NIST, GDPR, and GPG13. This focus on regulatory alignment ensures that security responses not only mitigate threats but also adhere to industry standards.

Finally, post-incident analysis and continuous improvement are essential to refining cybersecurity practices. Both XSOAR and the AI agent stress the importance of conducting debrief sessions and updating security policies based on lessons learned. However, the AI-driven approach provides a more structured mechanism for reviewing past incidents, identifying successes and failures, and incorporating these insights into future security strategies.

Table 2. Comparison of XSOAR Brute-Force Investigation—Generic and the proposed AI-agent implementation.

#	XSOAR Brute-Force Investigation—Generic	Proposed AI-Agent	Notes
1	Initial Detection & Triage Identify abnormal login attempts and confirm brute-force indicators.	Incident Summary – Attack Type: Brute Force – Severity: High – Source IP identified (192.168.1.130) – Failure patterns in logs	Both approaches emphasize quick identification of brute-force attempts. Early triage ensures correct prioritization and immediate response.
2	Gather Evidence & Analyze Logs Review system logs to confirm scope, timeline, and potential impact.	Step 2: Analyze Log Files – Reviewed <code>/var/log/auth.log</code> for failed attempts – Checked for “authentication failure” patterns	The AI agent’s procedure mirrors XSOAR’s approach by collecting evidence from relevant logs. Identifying compromised accounts or unusual sources is a shared goal.
3	Contain & Mitigate Ongoing Attack Block malicious IP addresses or isolate infected hosts.	Step 1: Isolate the Affected System – Block source IP via <code>iptables</code> – Disconnect or isolate the system if necessary	Both methods prioritize swift containment to stop the attack in progress. Blocking the malicious IP is a common immediate action.

Table 2. Cont.

#	XSOAR Brute-Force Investigation—Generic	Proposed AI-Agent	Notes
4	Implement Protective Measures Use account lockouts or IP blocking tools to thwart brute force.	Step 3: Monitor and Block Suspicious IPs – Installed and configured Fail2Ban Step 4: Change System and User Passwords – Reset root and user credentials	XSOAR’s generic playbook recommends threshold-based blocking and lockouts. The AI agent explicitly uses Fail2Ban. Password resets align with best practices for compromised accounts.
5	Strengthen Access Controls Enhance MFA and tighten SSH settings.	Step 5: Enable and Configure Two-Factor Authentication – Google Authenticator for SSH Step 6: Harden SSH Configuration – Disable root login, restrict users, reduce MaxAuthTries	Both highlight multi-factor authentication and SSH hardening as key defenses.
6	Forensic Analysis Investigate system integrity, checking for unauthorized changes or malware.	Step 7: Conduct a Forensic Analysis – AIDE for file integrity checks – auditd to monitor critical files	XSOAR’s deep-dive investigation is addressed by the AI agent’s emphasis on file integrity checks and audit logging.
7	Remediation & Restoration Return systems to secure the baseline once threats are removed.	Steps 1, 3–7 Combined – Isolation, IP blocking, reconfiguring, password resets	While XSOAR treats remediation as a distinct phase, the AI agent’s steps collectively restore normal, secure operations.
8	Documentation Record all findings, actions, and lessons learned.	Step 8: Document the Incident – Created <code>incident_report.log</code> – Logged detection, actions, and outcomes	Proper record keeping is essential for audits, compliance, and post-incident reviews. Both emphasize thorough documentation.
9	Policy Review & Compliance Check Review and update security policies for regulatory alignment.	Step 9: Review and Update Security Policies – Ensured compliance with PCI DSS, HIPAA, TSC, NIST, GDPR, GPG13	Both approaches highlight the importance of aligning policies with relevant standards. Continuous improvement is a central theme.
10	Post-Incident Analysis & Lessons Learned Conduct a debrief and refine IR processes.	Step 10: Conduct a Post-Incident Analysis – Debrief session, update plans, review successes/failures	A structured after-action review is key in both XSOAR’s process and the AI agent’s approach. Lessons learned to drive future improvements.

4.4. Brute-Force Quantitative Risk Assessment Result

A Quantitative Risk Assessment (QRA) was conducted to assess both the effectiveness and potential drawbacks of each proposed mitigation measure aimed at countering brute-force attacks. The results of this assessment are illustrated in Figure 9. The likelihood of encountering brute-force attempts is moderate to high, given their prevalence in cybersecurity incidents. In addition, the impact of a successful brute-force attack can be significant, potentially leading to unauthorized access, data breaches, and reputational damage. By combining probability and impact, the relative risk associated with brute-force attacks is assessed as high, reflecting both the frequency of these incidents and the severity of their consequences.

```

### QRA Analysis:
- **Probability (P):** The likelihood of encountering brute force attacks is moderate to high, given the prevalence of such attacks in cybersecurity incidents.
- **Impact (I):** The impact of a successful brute force attack can be significant, potentially leading to unauthorized access, data breaches, and reputational damage.
- **Relative Risk (R):**
  -  $R = P * I$ 
  - Given the moderate to high probability and significant impact, the relative risk is assessed as high.

The execution of these commands will help in identifying potential security threats and inform further mitigation strategies. The expected output includes lines from the authentication log that match the specified criteria, which can be used to assess the severity and scope of the attack.

```

Figure 9. QRA results for the brute-force use case.

Applying this QRA framework incrementally to each recommended mitigation step, such as isolating affected systems, deploying Fail2Ban, or enforcing multi-factor authentication, helps identify unintended consequences like legitimate user lockouts or performance overhead. As a result, the analysis informs whether or not the overall benefits of reducing risk justify the associated costs, enabling security teams to make data-driven decisions about prioritizing and implementing the most effective controls against brute-force threats.

4.5. AI-Driven Adaptive Error Resolution

Referencing Figure 10, the AI agent embarked on its task with precision, aiming to execute system commands seamlessly.

```

> Entering new Agent chain...
Invoking: 'execute_command_paramiko' with {'command': 'sudo apt-get install -y iptables'}

{'status': 'success', 'output': '', 'error': 'sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper(sudo: a password is required)'}
Invoking: 'correct_command' with {'wrong_command': 'sudo apt-get install -y iptables', 'error': 'sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper(sudo: a password is required)'}

Corrected command: sudo apt-get install -y iptables might need to be Correct the following command: 'sudo apt-get install -y iptables'. Error: 'sudo: a terminal is required to read the password; either use the -S option to read from standard input or configure an askpass helper(sudo: a password is required)'.
Suggest a fixed command. Based on search: [{'url': 'https://askubuntu.com/questions/872122/cant-apt-get-update-after-firewalling-with-iptables', 'content': 'sudo iptables -A INPUT -i eth0 -n state --state ESTABLISHED,RELATED -j ACCEPT By the way this rule: a Prevent HTTP DOS Attacks sudo iptables -A INPUT -p tcp --dport 80 -n limit --limit 25/minute --limit-burst 100 -j ACCEPT Will never be hit, because you bypassed it with an earlier rule: su do iptables -A INPUT -p tcp --dport 80 -j ACCEPT'}, {'url': 'https://www.silicloud.com/blog/how-to-fix-the-error-message-sudo-apt-install-in-linux/', 'content': 'Dependency Issue: Software packages may have unsatisfied dependencies, you can try fixing dependencies by using the command sudo apt-get -f install. Conflict Issue: There is a conflict between the software already installed and the software to be installed.'}, {'url': 'https://www.reddit.com/r/Ubuntu/comments/xzcruv/how_to_fix_error_sudo_aptget_update/', 'content': 'Remove the two ppa's that fail to load in software and updates - other sources. Then do a sudo apt update && sudo apt full-upgrade.'}]The command 'sudo apt-get install -y iptables' failed because it requires a password t o be entered, and this cannot be done in a non-interactive environment like this one.

To resolve this, you can either:
1. Run the command in a terminal where you can enter your password when prompted.
2. Use the -S option with 'sudo' to read the password from standard input, but this requires a secure way to provide the password.

Here's how you might run it in a terminal:

$ bash
$ sudo apt-get install -y iptables

If you are running this in a script or automated environment, you might need to configure 'sudo' to not require a password for this specific command, but be cautious as this can have security implications.

If you need further assistance, feel free to ask!

> Finished chain.
[+] Evaluate Result Command
[-] Evaluating Result Command...
[-] Executing Command: sudo iptables -A INPUT -s 192.168.1.130 -j DROP

> Entering new Agent chain...
The command 'sudo iptables -A INPUT -s 192.168.1.130 -j DROP' is intended to block incoming traffic from the IP address '192.168.1.130' by appending a rule to the 'INPUT' chain of the 'iptables' firewall.

However, this command requires superuser privileges, which means you need to enter a password when prompted. Since I can't execute commands that require interactive password input, you'll need to run this command in a terminal where you can provide the password.

Here's a breakdown of the command:

** 'sudo' **: Run the command with superuser privileges.
** 'iptables' **: The command-line utility for configuring Linux kernel firewall implemented within the Netfilter project.
** '-A INPUT' **: Appends a rule to the 'INPUT' chain, which handles incoming traffic.
** '-s 192.168.1.130' **: Specifies the source IP address to match.
** '-j DROP' **: Specifies the target action for matching packets, which in this case is to drop them.

To execute this command, open a terminal and enter it. You will be prompted to enter your password.

```

Figure 10. An AI-driven approach for diagnosing and remediating the encountered error.

However, despite its efficiency, it encountered an obstacle and a persistent authentication requirement that halted its progress. As illustrated in the figure, the AI agent engaged in a diagnostic and remediation process, iterating through possible solutions before ultimately recognizing the necessity of human intervention.

Table 3 shows the AI agent's repeated attempts to manage iptables under sudo privileges, all of which fail due to the requirement for a password. Each step concludes

with the agent indicating that human intervention or a change in sudo settings is needed before proceeding.

Table 3. AI agent execution steps.

Step	Action Taken	System Response	AI Agent Analysis	Next Steps
1	Run sudo apt-get install -y iptables	Error: sudo: a password is required	Requires sudo authentication.	User runs manually or provides password via -S.
2	Suggested authentication alternatives	User input required	Needs user interaction.	User manually enters a password.
3	Ran sudo iptables -A INPUT -s 192.168.1.130 -j DROP	Error: sudo: a password is required	Same issue: requires authentication.	Configure sudo to allow execution without password.
4	Issued a Human Intervention Request	Awaiting user action	Execution blocked by authentication.	User must execute manually or adjust sudo settings.
5	Standing by for further instructions	Ready for next attempt	Awaiting user input.	User feedback required.

Ultimately, the agent issues a Human Intervention Request and remains on standby for further instructions, highlighting the importance of valid authentication or suitably configured permissions to facilitate its automated tasks.

4.6. Mean Time to Remediate (MTTR) Results

This research evaluated Mean Time to Remediate (MTTR) across three primary remediation activities: analysis, update command execution, and compilation. The results indicated that the analysis phase required approximately 1 min and 30 s on average, while the update command phase averaged around 1 min. Similarly, the compilation stage mirrored the analysis phase, averaging approximately 1 min and 30 s.

A comparative review of the existing literature was conducted, specifically referencing recent data from a similar study [83]. According to this referenced study, MTTR values observed in alternative platforms were generally lower, averaging approximately 45 s for the analysis and compilation phases and around 30 s for command updates. This discrepancy suggests potential efficiency differences attributable to platform-specific operational methodologies.

However, a significant limitation in this comparison arises due to restricted access to detailed remediation data from alternative platforms. The closed nature of this data prevented thorough and direct comparative analysis, thereby limiting the ability to conclusively identify the precise operational causes underlying the observed MTTR discrepancies.

5. Discussion

The AI-driven mitigation system demonstrates the benefits of integrating intelligent agents into cybersecurity workflows. This experiment showcased how AI-enhanced security measures, particularly in brute-force attack scenarios, improve efficiency, precision, and adaptability compared to traditional Security Orchestration, Automation, and Response (SOAR) solutions.

A key finding is the effectiveness of data enrichment in transforming raw security logs into actionable intelligence. By contextualizing events through AI-driven analysis, the system provides a structured and interpretable dataset, reducing the cognitive load on security analysts. As illustrated in Figure 8, the enriched data includes critical attributes such as attack source identification, MITRE ATT&CK mappings, and incident response

recommendations. This enhanced visibility ensures a more targeted and informed approach to threat mitigation.

Additionally, the AI agent-based mitigation process streamlines incident response by minimizing the number of steps required to contain and neutralize security breaches. As highlighted in Table 2, the AI-driven approach condenses the 38-step SOAR workflow into ten key actions, demonstrating its ability to prioritize and optimize remediation efforts. The efficiency gains observed in the AI-driven methodology underscore the potential of leveraging Large Language Models (LLMs) for cybersecurity applications, particularly in rapidly evolving attack landscapes where swift decision-making is crucial.

A comparative assessment of the AI agent and traditional SOAR methods reveals nuanced differences in handling security incidents. While SOAR platforms rely on pre-configured workflows and structured playbooks, the AI agent introduces adaptability by incorporating dynamic threat intelligence. For instance, the AI agent recommends specific mitigation tools such as Fail2Ban for automated IP blocking and AIDE for integrity monitoring, enhancing the flexibility and robustness of the response strategy. Additionally, the agent suggests proactive security hardening techniques, such as enforcing Multi-Factor Authentication (MFA) and disabling root login for SSH, reflecting a forward-thinking approach to security management.

However, certain limitations of AI-driven mitigation must be acknowledged. While machine-learning models have a potential risk of misclassification or false positives, incorporating human intervention to validate critical steps ensures accuracy and prevents unnecessary disruptions. Implementing continuous learning and periodic validation further enhances reliability, ensuring accurate and effective security actions. Additionally, while the AI agent provides more granular and adaptive recommendations, its effectiveness depends on the quality and completeness of the underlying training data. Moreover, the interpretability of AI-generated recommendations remains a critical consideration. Security teams must understand and trust the AI agent's outputs to fully integrate them into operational workflows. Other limitations include integration with third-party tools as proven mitigation tools, limited direct access to agentless devices such as Operational Technology (OT) systems, and constraints associated with active responses utilizing WebSocket.

6. Future Works

Future improvements should focus on enhancing the model's decision-making capabilities through continuous learning and validation against real-world attack datasets. Improving explainability by incorporating transparent reasoning processes and human-in-the-loop validation mechanisms can help bridge the operational gap between AI-driven automation and expert human oversight, thereby increasing system trustworthiness and accountability.

The findings of this study demonstrate that AI-enhanced cybersecurity solutions substantially improve the efficiency and effectiveness of brute-force attack mitigation. The AI agent's ability to process high volumes of security log data, generate enriched threat intelligence, and deliver optimized response strategies positions it as a critical asset within modern security operations. Nonetheless, the validation conducted thus far remains limited in scope. Future work should pursue more comprehensive validation strategies to ensure robustness across diverse operational environments.

In addition, expanding the AI agent's capabilities to address a broader range of cyber threats, such as phishing detection, privilege escalation, and ransomware mitigation, will further support its adaptability to emerging attack vectors and adversarial tactics. These improvements are essential for advancing toward resilient, intelligent, and context-aware cybersecurity systems.

Author Contributions: Conceptualization, I.; methodology, I.; software, R.K., Z.A.B., and G.A.N.; validation, R.K.; formal analysis, I. and R.K.; data curation, I. and R.K.; writing—original draft, I., S.H., and H.K. (Hyeongon Kim); writing—review and editing, I.; visualization, I. and R.K.; supervision, H.K. (Howon Kim); project administration, H.K. (Howon Kim); funding acquisition, H.K. (Howon Kim). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (RS-2020-II201797) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data are contained within the article.

Conflicts of Interest: Authors Rahmat Kurnia, Zilmas Arjuna Brata, Ghitha Afina Nelistiani, Shinwook Heo, Hyeongon Kim are employed by SmartM2M. Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest authors declare no conflicts of interest.

References

1. Crowley, C.; Pescatore, J. Common and Best Practices for Security Operations Centers: Results of the 2019 SOC Survey. SANS Institute Information Security Reading Room, 2019. Available online: <https://www.sans.org/media/analyst-program/common-practices-security-operations-centers-results-2019-soc-survey-39060.pdf> (accessed on 12 February 2025).
2. Threat Intelligence Team. Accelerate Incident Response with SOAR. Threat Intelligence Blog, 2025. Available online: <https://www.threatintelligence.com/blog/soar-incident-response> (accessed on 7 January 2025).
3. CREST. CREST Launches Comprehensive White Paper on Maximising SOAR in the SOC—Global Security Mag Online. *Global Security Magazine Online*, 2023. Available online: <https://www.globalsecuritymag.com/crest-launches-comprehensive-white-paper-on-maximising-soar-in-the-soc.html> (accessed on 7 January 2025).
4. Logic, S. How SOAR Can Foster Efficient SecOps in Modern SOC's | Sumo Logic. Sumo Logic Blog, 2025. Available online: <https://www.sumologic.com/blog/how-soar-can-foster-efficient-secops-in-modern-socs/> (accessed on 7 January 2025).
5. Securaa. SOAR: Revolutionizing Security Operations Centers (SOC) Teams—Securaa. Securaa Blog, 2025. Available online: <https://securaa.io/soar-revolutionizing-security-operations-centers-soc-teams/> (accessed on 7 January 2025).
6. Rapid7. Automating Threat Hunting with SOAR for Faster Response Times | Rapid7 Blog. Rapid7 Blog, 2019. Available online: <https://www.rapid7.com/blog/post/2019/07/29/how-to-automate-threat-hunting-with-soar-for-faster-response-times/> (accessed on 7 January 2025).
7. D3 Security. How Smart SOAR Automates Threat Hunting | D3 Security. D3 Security Blog, 2025. Available online: <https://d3security.com/blog/how-smart-soar-automates-threat-hunting/> (accessed on 7 January 2025).
8. Saint-Hilaire, K.A.; Neal, C.; Cuppens, F.; Boulahia-Cuppens, N.; Hadji, M. Optimal Automated Generation of Playbooks. In *Data and Applications Security and Privacy XXXVIII: Proceedings of the 38th Annual IFIP WG 11.3 Conference, DBSec 2024, San Jose, CA, USA, 15–17 July 2024*; Ferrara, A.L.; Krishnan, R., Eds.; Springer Nature: Cham, Switzerland, 2024; pp. 191–199.
9. Islam, C.; Babar, M.A.; Nepal, S. Architecture-Centric Support for Integrating Security Tools in a Security Orchestration Platform. In *Proceedings of the Software Architecture, Salvador, Brazil, 16–20 March 2020*; Jansen, A.; Malavolta, I.; Muccini, H.; Ozkaya, I.; Zimmermann, O., Eds.; Springer Nature: Cham, Switzerland, 2020; pp. 165–181.
10. Zhang, Y.; Li, X.; Wang, M. Playbook-Centric Scalable SOAR System Architecture. In *Proceedings of the 2020 International Conference on Cyber Security and Protection of Digital Services (Cyber Security 2020), Dublin, Ireland, 15–17 June 2020*; IEEE: New York, NY, USA, 2020; pp. 1–8.
11. Tilbury, J.; Flowerday, S. Humans and Automation: Augmenting Security Operation Centers. *J. Cybersecur. Priv.* **2024**, *4*, 388–409. [CrossRef]
12. Kochale, K.; Boerakker, D.; Teutenberg, T.; Schmidt, T.C. Concept of flexible no-code automation for complex sample preparation procedures. *J. Chromatogr. A* **2024**, *1736*, 465343. [CrossRef] [PubMed]
13. Lee, M.; Jang-Jaccard, J.; Kwak, J. Novel Architecture of Security Orchestration, Automation and Response in Internet of Blended Environment. *Comput. Mater. Contin.* **2022**, *73*, 199–223. [CrossRef]
14. Tyagi, A. Enterprise Risk Management: Benefits and Challenges. Available online: <https://ssrn.com/abstract=3748267> (accessed on 11 January 2025).

15. Smith, D.; Fischbacher, M. The changing nature of risk and risk management: The challenge of borders, uncertainty and resilience. *Risk Manag.* **2009**, *11*, 1–12. [CrossRef]
16. Luxoft. Transforming Regulatory and GRC with Low-Code Automation Technologies | Luxoft Blog. Luxoft Blog, 2025. Available online: <https://www.luxoft.com/blog/transforming-regulatory-and-grc-with-low-code-automation-technologies> (accessed on 13 January 2025).
17. Quantzig. Top Benefits of Low-Code Platforms That Transforms Your Business in 2025. Quantzig Blog, 2025. Available online: <https://www.quantzig.com/low-code-no-code-platform/top-benefits-of-low-code-platforms> (accessed on 13 January 2025).
18. Technology, B. Low-Code Development: A Comprehensive Guide for 2025. Bacancy Technology Blog, 2025. Available online: <https://www.bacancytechnology.com/blog/low-code-development> (accessed on 13 January 2025).
19. Intelliconnectq. Low-Code Platforms for Optimizing Operational Costs. Intelliconnectq Blog, 2025. Available online: <https://intelliconnectq.com/blog/low-code-platforms-for-optimizing-operational-costs> (accessed on 13 January 2025).
20. Sido, N.; Emon, E.A.; Ahmed, E.; Supervisor, E.; Falch, M. Low/No Code Development and Generative AI. Bachelor's Thesis, Aalborg University, Copenhagen, Denmark, 2024. Available online: https://vbn.aau.dk/ws/files/717521040/LowNOCode_GenAI.pdf (accessed on 13 January 2025).
21. Desmond, M.; Duesterwald, E.; Isahagian, V.; Muthusamy, V. A No-Code Low-Code Paradigm for Authoring Business Automations Using Natural Language. *arXiv* **2022**, arXiv:2207.10648v1.
22. Rajput, A.S.; Professor, R.G.A. Hyper-automation-The next peripheral for automation in IT industries. *arXiv* **2023**, arXiv:2305.11896.
23. Quargnali, G. Low-Code/No-Code Development and Generative AI: Opportunities and Challenges. Bachelor's Thesis, Haaga-Helia University of Applied Sciences, Helsinki, Finland, 2023. Available online: https://www.theseus.fi/bitstream/handle/10024/751688/Quargnali_Giovanni.pdf (accessed on 12 January 2025).
24. Engel, C.; Ebel, P.; Leimeister, J.M. Cognitive automation. *Electron. Mark.* **2022**, *32*, 339–350. [CrossRef]
25. Engel, C.; Elshan, E.; Ebel, P.; Leimeister, J.M. Stairway to heaven or highway to hell: A model for assessing cognitive automation use cases. *J. Inf. Technol.* **2024**, *39*, 94–122. [CrossRef]
26. Matthijs Bal, P.; Davids, J.; Garcia, E.; McKnight, C.; Nichele, E.; Orhan, M.A.; van Rossenberg, Y., The Psychology of Automation and Artificial Intelligence at Work: Exploring Four Fantasies and Their Leadership Implications. In *Power, Politics and Influence: Exercising Followership, Leadership, and Practicing Politics*; Akande, A., Ed.; Springer Nature: Cham, Switzerland, 2024; pp. 575–592. [CrossRef]
27. Kusiak, A. Hyper-automation in manufacturing industry. *J. Intell. Manuf.* **2024**, *35*, 1–2. [CrossRef]
28. Fedosovsky, M.E.; Uvarov, M.M.; Aleksanin, S.A.; Pyrkin, A.A.; Colombo, A.W.; Prattichizzo, D. Sustainable Hyperautomation in High-Tech Manufacturing Industries: A Case of Linear Electromechanical Actuators. *IEEE Access* **2022**, *10*, 98204–98219. [CrossRef]
29. SANS Institute. *SANS 2020 MITRE ATT&CK Whitepaper*; Technical Report; SANS Institute: Bethesda, MD, USA, 2020. Available online: <https://www.sans.org/media/vendor/SANS-2020-MITRE-ATT&CK-Whitepaper-Sell-Sheet.pdf> (accessed on 6 February 2025).
30. Bartwal, U.; Mukhopadhyay, S.; Negi, R.; Shukla, S. Security Orchestration, Automation and Response Engine for Deployment of Behavioural Honeypots. *arXiv* **2022**, arXiv:cs.CR/2201.05326.
31. Kinyua, J.; Awuah, L. AI/ML in Security Orchestration, Automation and Response: Future Research Directions. *Intell. Autom. Soft Comput.* **2021**, *28*, 528–543. [CrossRef]
32. SANS Institute. *Incident Response Cycle*; SANS Resource; SANS Institute: Bethesda, MD, USA, 2025. Available online: <https://www.sans.org/media/score/504-incident-response-cycle.pdf> (accessed on 21 February 2025).
33. Fysarakis, K.; Lekidis, A.; Mavroeidis, V.; Lampropoulos, K.; Lyberopoulos, G.; Vidal, I.G.M.; Terés i Casals, J.C.; Luna, E.R.; Moreno Sancho, A.A.; Mavrellos, A.; et al. PHOENIX—A European Cyber Resilience Framework With Artificial-Intelligence-Assisted Orchestration, Automation & Response Capabilities for Business Continuity and Recovery, Incident Response, and Information Exchange. In *Proceedings of the 2023 IEEE International Conference on Cyber Security and Resilience (CSR)*, Venice, Italy, 31 July–2 August 2023; pp. 538–545. [CrossRef]
34. Kremer, R.; Wudali, P.N.; Momiyama, S.; Araki, T.; Furukawa, J.; Elovici, Y.; Shabtai, A. IC-SECURE: Intelligent System for Assisting Security Experts in Generating Playbooks for Automated Incident Response. *arXiv* **2023**, arXiv:cs.CR/2311.03825.
35. Sworna, Z.T.; Islam, C.; Babar, M.A. APIRO: A Framework for Automated Security Tools API Recommendation. *arXiv* **2022**, arXiv:cs.SE/2201.07959.
36. Valencia, L.J. Artificial Intelligence as the New Hacker: Developing Agents for Offensive Security. Master's Thesis, New Mexico Institute of Mining and Technology, Socorro, New Mexico, 2024.
37. Oesch, S.; Hutchins, J.; Austria, P.; Chaulagain, A. Agentic AI and the Cyber Arms Race. *IEEE Comput. Mag. Cybertrust Column* **2025**, arXiv:cs.CY/2503.04760. To be published. <http://arxiv.org/abs/2503.04760>. [CrossRef]
38. Khan, R.; Sarkar, S.; Mahata, S.K.; Jose, E. Security Threats in Agentic AI System. *arXiv* **2024**, arXiv:2410.14728.

39. Kaheh, M.; Kholgh, D.K.; Kostakos, P. Cyber Sentinel: Exploring Conversational Agents' Role in Streamlining Security Tasks with GPT-4. *arXiv* **2023**, arXiv:2309.16422.
40. Infrastructure Security Agency. CISA | Defend Today, Secure Tomorrow. CISA Website, 2021. Available online: https://www.cisa.gov/sites/default/files/publications/Incident-Response-Plan-Basics_508c.pdf (accessed on 8 January 2025).
41. *Framework for Improving Critical Infrastructure Cybersecurity*, Version 1.1; American's Cyber Defense Agency: Washington, DC, USA, 2018. [CrossRef]
42. Strom, B.E.; Applebaum, A.; Miller, D.P.; Nickels, K.C.; Pennington, A.G.; Thomas, C.B. MITRE ATT&CK®: *Design and Philosophy*; MITRE Technical Report; 2018. Available online: https://attack.mitre.org/docs/ATTACK_Design_and_Philosophy_March_2020.pdf (accessed on 8 January 2025).
43. Zadeh, A.; Lavine, B.; Zolbanin, H.; Hopkins, D. Cybersecurity risk quantification and classification framework for informed risk mitigation decisions. *Decis. Anal. J.* **2023**, *9*, 100328. [CrossRef]
44. Quinn, S.; Ivy, N.; Barrett, M.; Witte, G.; Gardner, R.K. NISTIR 8286B: *Prioritizing Cybersecurity Risk for Enterprise Risk Management*; Technical Report 8286B; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2022. [CrossRef]
45. Algarni, A.M.; Malaiya, Y.K. A consolidated approach for estimation of data security breach costs. In Proceedings of the 2016 2nd International Conference on Information Management, (ICIM), London, UK, 7–8 May 2016; pp. 26–39.
46. Jouini, M.; Rabai, L.B.A.; Khedri, R. A Quantitative Assessment of Security Risks Based on a Multifaceted Classification Approach. *Springer J. Inf. Secur.* **2020**, *20*, 493–510. [CrossRef]
47. OpenAI. GPT-4, 2023. Available online: <https://openai.com/product/gpt-4> (accessed on 6 February 2025).
48. Google. Bard, 2023. Available online: <https://bard.google.com/> (accessed on 6 February 2025).
49. Anthropic. Claude, 2023. Available online: <https://www.anthropic.com/claude> (accessed on 6 February 2025).
50. Meta AI. LLaMA: Large Language Model Meta AI, 2023. Available online: <https://ai.meta.com/blog/large-language-model-llama-meta-ai/> (accessed on 6 February 2025).
51. Cohere AI. Command R: Retrieval-Augmented Generation Model, 2023. Available online: <https://cohere.ai/command> (accessed on 6 February 2025).
52. Hugging Face BigScience. BLOOM: BigScience Large Open-Science Open-Access Multilingual Language Model, 2023. Available online: <https://huggingface.co/bigscience/bloom> (accessed on 6 February 2025).
53. EleutherAI. GPT-NeoX: A Large-Scale Autoregressive Language Model, 2023. Available online: <https://github.com/EleutherAI/gpt-neox> (accessed on 6 February 2025).
54. DeepSeek AI. DeepSeek-R1, 2023. Available online: <https://en.wikipedia.org/wiki/DeepSeek> (accessed on 6 February 2025).
55. Alibaba DAMO Academy. Qwen: Large Language Model by Alibaba, 2023. Available online: https://en.wikipedia.org/wiki/List_of_large_language_models (accessed on 6 February 2025).
56. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
57. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the Opportunities and Risks of Foundation Models. *arXiv* **2021**, arXiv:2108.07258.
58. Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Ichter, B.; Xia, F.; Chi, E.; Le, Q.; Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *arXiv* **2022**, arXiv:2201.11903.
59. Qin, Y.; Liang, S.; Ye, Y.; Zhu, K.; Yan, L.; Lu, Y.; Lin, Y.; Cong, X.; Tang, X.; Qian, B.; et al. ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs. *arXiv* **2023**, arXiv:2307.16789.
60. Nair, V.; Schumacher, E.; Tso, G.; Kannan, A. DERA: Enhancing Large Language Model Completions with Dialog-Enabled Resolving Agents. *arXiv* **2023**, arXiv:cs.CL/2303.17071.
61. Wang, H.; Zhao, L.; Chen, M. Ethical AI in Autonomous Cybersecurity: Challenges and Opportunities. *ACM Trans. Inf. Syst. Secur.* **2023**, *26*, 1–18. [CrossRef]
62. Wang, G.; Lin, Z.; Peng, J.B.; Wu, Q.; Lu, S.; Gonzalez, J.E.; Abbeel, P.; Zhou, D.S. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv* **2023**, arXiv:2305.16291.
63. Ahn, M.; Brohan, A.; Chai, Y.H.H.; Canny, J.; Goldberg, K.; McGrew, B.; Ichter, B. Can Large Language Models Be an Alternative to Robot Motion Planning? *arXiv* **2022**, arXiv:2206.05841.
64. Singh, S.; Gupta, S.; Thakur, A.; Saran, A. ProgPrompt: Generating Situated Robot Task Plans using Large Language Models. *arXiv* **2023**, arXiv:2304.05381.
65. Mitchell, E.; Brynjolfsson, E. The AI cybersecurity revolution: Leveraging autonomous decision-making for proactive threat mitigation. *Cybersecur. J.* **2022**, *10*, 45–67.
66. Arsanjani, M.S. The Anatomy of Agentic AI. Medium, 2023. Available online: <https://dr-arsanjani.medium.com/the-anatomy-of-agentic-ai-0ae7d243d13c> (accessed on 12 February 2025).
67. Singh, R.; Patel, N.; Li, Z. AI-driven cybersecurity orchestration: From predictive analytics to automated defenses. *J. Artif. Intell. Cybersecur.* **2023**, *8*, 112–138.

68. Zacharewicz, G.; Chen, D.; Vallespir, B. Short-Lived Ontology Approach for Agent/HLA Federated Enterprise Interoperability. In Proceedings of the 2009 International Conference on Interoperability for Enterprise Software and Applications, Beijing, China, 21–22 April 2009. [CrossRef]
69. Paul, S.; Alemi, F.; Macwan, R. LLM-Assisted Proactive Threat Intelligence for Automated Reasoning. *arXiv* **2025**, arXiv:2504.00428. Available online: <https://arxiv.org/abs/2504.00428> (accessed on 20 April 2025).
70. Zhou, K.; Wang, P.; Zhang, E. AI-driven Vulnerability Assessment: Automating Risk Identification and Mitigation. *J. Mach. Learn. Secur.* **2022**, *5*, 189–210.
71. Radanliev, P. AI Ethics: Integrating Transparency, Fairness, and Privacy in AI Development. *Appl. Artif. Intell.* **2025**, *39*. [CrossRef]
72. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
73. Johnson, R.; Kim, E.; Williams, D. LLM-driven Adversarial Defense: Real-Time Detection and Mitigation of Adversarial Attacks. *J. AI Secur. Res.* **2023**, *9*, 155–178.
74. Ismail; Kurnia, R.; Widyatama, F.; Wibawa, I.M.; Brata, Z.A.; Ukasyah; Nelistiani, G.A.; Kim, H. Enhancing Security Operations Center: Wazuh Security Event Response with Retrieval-Augmented-Generation-Driven Copilot. *Sensors* **2025**, *25*, 870. [CrossRef]
75. Pupentsova, S.; Gromova, E. Risk Management in Business Valuation in the Context of Digital Transformation. *Real Estate Manag. Valuat.* **2021**, *29*, 97–106. [CrossRef]
76. Metin, B.; Duran, S.; Telli, E.; Mutlutürk, M.; Wynn, M. IT Risk Management: Towards a System for Enhancing Objectivity in Asset Valuation that Engenders a Security Culture. *Information* **2024**, *15*, 55. [CrossRef]
77. DeepSeek AI. DeepSeek-R1-Distill-Llama-70B. Hugging Face, 2023. Available online: <https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-70B> (accessed on 12 February 2025).
78. Meta AI. Llama-3.3-70B-Instruct. Hugging Face, 2023. Available online: <https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct> (accessed on 12 February 2025).
79. Documentation, G. DeepSeek-R1-Distill-Llama-70B. Groq Console Docs, 2023. Available online: <https://console.groq.com/docs/model/deepseek-r1-distill-llama-70b> (accessed on 12 February 2025).
80. Wazuh. Wazuh: The Open Source Security Platform. Wazuh Official Website, 2025. Available online: <https://wazuh.com/> (accessed on 25 February 2025).
81. Wazuh. Detect Brute-Force Attack. Wazuh Documentation, 2025. Available online: <https://documentation.wazuh.com/current/proof-of-concept-guide/detect-brute-force-attack.html> (accessed on 21 February 2025).
82. Networks, P.A. Brute Force Investigation — Generic. Cortex XSOAR Documentation, 2025. Available online: <https://xsoar.pan.dev/docs/reference/playbooks/brute-force-investigation---generic> (accessed on 24 February 2025).
83. Freitas, S.; Kalajdjieski, J.; Gharib, A.; McCann, R. AI-Driven Guided Response for Security Operation Centers with Microsoft Copilot for Security. *arXiv* **2024**, arXiv:2407.09017.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.