

LLMs in the SOC: An Empirical Study of Human-AI Collaboration in Security Operations Centres

Ronal Singh
Data61, CSIRO
Melbourne, Australia

Shahroz Tariq*
Data61, CSIRO
Sydney, Australia

Fatemeh Jalalvand*
Data61, CSIRO
Melbourne, Australia

Mohan Baruwal Chhetri
Data61, CSIRO
Melbourne, Australia

Surya Nepal
Data61, CSIRO
Sydney, Australia

Cecile Paris
Data61, CSIRO
Sydney, Australia

Martin Lochner
eSentire Inc.
Waterloo, Canada

Abstract—The integration of Large Language Models (LLMs) into Security Operations Centres (SOCs) presents a transformative, yet still evolving, opportunity to reduce analyst workload through human-AI collaboration. However, their real-world application in SOC remains underexplored. To address this gap, we present a longitudinal study of 3,090 analyst queries from 45 SOC analysts over 10 months. Our analysis reveals that analysts use LLMs as on-demand aids for sensemaking and context-building, rather than for making high-stakes determinations, preserving analyst decision authority. The majority of queries are related to interpreting low-level telemetry (e.g., commands) and refining technical communication through short (1-3 turn) interactions. Notably, 93% of queries align with established cybersecurity competencies (NICE Framework), underscoring the relevance of LLM use for SOC-related tasks. Despite variations in tasks and engagement, usage trends indicate a shift from occasional exploration to routine integration, with growing adoption and sustained use among a subset of analysts. We find that LLMs function as flexible, on-demand cognitive aids that augment, rather than replace, SOC expertise. Our study provides actionable guidance for designing context-aware, human-centred AI assistance in security operations, highlighting the need for further in-the-wild research on real-world analyst-LLM collaboration, challenges, and impacts.

1. Introduction

SOCs handle diverse functions, from real-time detection and incident response to continuous improvement [1], [2], [3]. SOC analysts play a critical role in this by interpreting data and making informed decisions. They typically follow a multi-stage process, beginning with quick assessments of alert relevance [4], then checking for related signals, gathering contextual telemetry to build situational awareness [3], and evaluating evidence of compromise to make escalation decisions [5]. This reasoning workflow is supported by tooling like SIEM, SOAR, and XDR [3], [6].

*Authors contributed equally as second authors.

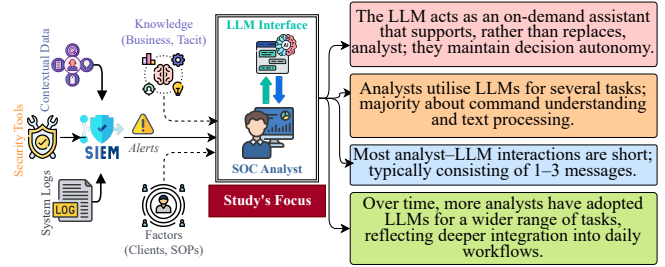


Figure 1: SOC workflow, focus of study and our insights.

Recently, LLMs have emerged as promising tools to support such SOC workflows, with applications ranging from threat intelligence to triage automation [7], [8], [9], [10]. While prior research explores human-LLM interaction in various domains and lab studies [11], [12], [13], [14], it does not capture how SOC analysts utilise LLMs in their daily workflows. Existing work, such as Copilot Guided Response (CGR) [7], shows system-level performance, not individual analysts' usage patterns.

Gaining insights into how SOC analysts utilise LLMs in practice, including the types of questions they ask and the tasks they use them for, is essential for evaluating the practical role of LLMs in operational cybersecurity (Figure 1). This study addresses that gap by analysing 3,090 queries submitted to GPT-4 during live investigations over 10 months (May 2023 to March 2024) by 45 SOC analysts at eSentire Inc., a cybersecurity firm specialising in 24/7 managed detection and response [15]. To guide this investigation, we focus on the following research question:

How do SOC analysts utilise LLMs in their daily workflows, including the specific tasks they apply them to, how these uses align with established cybersecurity frameworks (e.g., NICE), and the conversational patterns that characterise their interactions?

The answers to this question reveal the task priorities and engagement patterns of SOC analysts with LLMs, in-

forming the design of more effective LLM-based assistants as human-AI collaboration becomes increasingly common in SOC environments. We address this question through a multi-level analysis of analyst-LLM interactions across queries, tasks, and conversations.

- **Query-level:** We examine *what* analysts ask and *how* they phrase their requests, revealing their cognitive needs, the daily tasks augmented by LLMs, and alignment with frameworks such as NICE.
- **Conversation-level:** We analyse sequences of queries within investigative sessions to uncover *how* analysts structure multi-turn interactions in their daily workflows. This reveals reasoning patterns and functional roles that LLMs assume in conversations.
- **Analyst-level:** We investigate the frequency and manner in which different analysts use LLMs. This variation reveals diverse strategies for integrating LLMs and highlights how they become embedded in the daily workflows of different analysts.

We began with descriptive analysis to map overall query activity, including how frequently analysts interacted with the LLM. Then, we conducted a thematic analysis [16] to identify patterns in query phrasing, topics, and tasks. Since our research objective was to characterise analyst behaviour, not evaluate LLM performance, we focused our thematic analysis on analyst queries and used LLM responses only to contextualise intent. We grouped successive analyst queries on the same topic into single conversational units, based on topic continuity. Finally, we synthesised insights across Query, Conversation, and Analyst levels to characterise analyst-LLM interactions. Although our dataset covers a single enterprise SOC using one type of text-based LLM, it provides a valuable longitudinal view of real-world LLM use, highlighting key usage patterns and tasks. As more studies emerge, both quantitative (e.g., [7]) and qualitative (e.g., ours, [5], [17]), research and design communities will be better informed to design targeted LLM-based SOC tools.

Contributions. We make the following contributions:

- We present the **first in-depth, real-world analysis of LLM adoption in a SOC**, spanning 3,090 queries from 45 analysts over 10 months. Our analysis highlights analyst-level usage patterns across task types and individual differences.
- We develop a **structured coding framework that facilitates thematic analysis of analyst-LLM interactions** by classifying LLM queries according to task type, subject, and interaction pattern. We apply this framework to develop a deeper understanding of how LLMs assist investigative, explanatory, and communicative functions within SOC workflows.
- We provide **empirical insights into analyst behaviour and LLM utility in SOCs**. We identify three core use cases that characterise how SOC analysts engage with LLMs: (i) interpreting raw telemetry, (ii) generating and refining task-related communication content, and (iii) obtaining rapid, on-demand assistance through

concise interactions. Collectively, our analysis reveals that LLMs are becoming routine, competency-aligned aides rather than decision-makers: 93% of queries map to NICE tasks, usage has shifted from ad-hoc exploration to sustained integration, and analysts retain final judgement while leveraging LLMs for rapid sense-making. These findings guide the design of context-aware, human-centred assistants to enhance human-AI collaboration in SOCs.

- We offer **design recommendations** for AI-augmented SOCs: (1) embed analyst- and context-adaptive LLM explanations of technical artifacts to speed triage and support evidence-based decisions; (2) integrate LLM assistants for microtasks to provide in-workflow support and minimise context switching; and (3) surface evidence over recommendations for investigative tasks, preserving analyst autonomy and context.

2. Related Works

AI for SOCs. SOCs integrate multiple systems including SIEM platforms that ingest telemetry from IDS/IPS, firewalls, and EDR tools; SOAR platforms for response coordination; and supplementary tools such as User and Entity Behaviour Analytics (UEBA) and Threat Intelligence Platforms (TIPs) [18], [19]. These systems have long relied on AI to enhance operational efficiency and decision-making. Prior to the emergence of LLMs, AI was already embedded across the SOC workflow, from threat detection [20], alert triage [21], and incident response [22] to end-to-end automation [19]. SIEMs employed ML/DL for event correlation, anomaly detection, and prioritisation [18], while TIPs used NLP and graph inference to enrich contextual understanding [19]. Commercial tools such as Microsoft Sentinel, Maltego, and Splunk exemplify this integration, offering AI-driven capabilities for visualisation, triage, and orchestration [6], [7], [18], [19], [23]. While effective, these approaches were task-specific and limited in their ability to reason across diverse, unstructured contexts. The recent emergence of LLMs provides a broader, context-aware form of cybersecurity reasoning, enabling more adaptive and human-aligned support across SOC operations [23].

LLMs for Cybersecurity. Broadly, LLMs have been used to support vulnerability scanning, anomaly/intrusion detection, phishing simulation, threat intel summarisation, and privacy-preserving analysis [24], [25], [26], [27], [28], [29], [30], [31], [32], [33]. Recent works highlight the expanding role of LLMs in cybersecurity workflows to support SOC analysts. For example, Copilot Guided Response (CGR) aids triage and remediation by generating recommendations from historical threats [7]. Most of these systems are evaluated in controlled environments, with a focus on performance metrics such as speed and accuracy of decision-making. In contrast, our study offers the first empirical analysis of LLM use in a live SOC, examining how analysts engage with LLMs, what help they seek, how interactions unfold, and how adoption evolves, providing a unique human-centred view of analyst-LLM practice.

LLM Roles in Decision Support. LLMs play diverse roles in human-AI decision-making, shaped by system autonomy, user needs, and interaction structure [34], [35], [36], [37], [38]. LLMs span roles from *assisted intelligence* (retrieval and summarisation) to more collaborative forms like *augmented* and *cooperative intelligence*, which aid reasoning and joint problem solving [34], [37]. HCI research stresses aligning AI support with decision phases, whether recommending actions, predicting, proposing alternatives, or surfacing evidence [35], [36], [37], [39]. While AI suggestions can speed decisions, they may also increase over-reliance [40], [41]. Cognitive forcing techniques (e.g., delaying AI output) can reduce over-reliance but risk under-reliance [40], [42]. *Evaluative AI* provides balanced insights, highlighting pros and cons without dictating choices, thus supporting autonomy in machine-in-the-loop systems [43], [44], [45], [46]. We examine how LLMs function in practice and what *decision support* analysts seek.

Interaction Patterns and Analyst Strategies.

Despite the growing capabilities of models, interaction patterns between humans and LLMs remain underexplored. Studies show that interaction framing and modality shape user expectations, trust, and behaviour [11], [12], [13], [31], [47], [48], [49]. Recent work highlights that prompt structure shapes explainability and collaboration [13], dialogue in complex tasks involves verbal and nonverbal cues [48], and human-AI interaction spans planning to testing across varying agent roles [11]. These studies show that effective LLM integration depends on their ability to adapt to user needs, interaction design, and task context. Our study addresses this gap by analysing thousands of analyst-LLM interactions in an operational SOC, uncovering concrete interaction patterns, evolving strategies, and the situated use of LLMs as cognitive aids in high-stakes, time-sensitive environments.

3. Method

To analyse and understand SOC analysts' interactions with LLMs, we adopted a five-phase methodology, as illustrated in Figure 2, which we describe in detail below. This approach is guided by Nowell et al.'s [16] multi-phase framework for thematic analysis, encompassing: (i) familiarisation with data (phases 1-2), (ii) code generation (phases 2-3), (iii) theme identification, review, and definition (phase 4), and (iv) reporting (phase 5). We follow their iterative process and incorporate specific trustworthiness practices, including multi-coder independent coding with strong inter-rater reliability (IRR), triangulation, reconciliation, cross-verification, and systematic documentation.

PHASE 1: Exploratory Analysis of Interactions. In this phase, we performed a statistical analysis of analyst-LLM interactions to characterise overall usage patterns, such as the total number of queries, engagement frequency, and query length distributions, providing a quantitative foundation for understanding analyst behaviours before moving to deeper thematic and qualitative analysis.

PHASE 2: Data Familiarisation for Question Coding & Conversation Tagging (Partial Dataset). This phase involved three members of the research team reviewing the question-LLM response pairs multiple times to develop a shared understanding of the analyst queries as well as identify and tag conversations for thematic analysis. Since the data consisted of individual analyst-LLM interactions, the team first needed to group these into coherent conversations.

Question Coding. This step focused on developing a coding scheme to analyse the nature of the analysts' queries. The three researchers first familiarised themselves with the data by reviewing a range of question-LLM response pairs. Through discussion, the team agreed on three key attributes to enable a meaningful decomposition of the queries:

- **Query Pattern:** The structural form or phrasing of the analyst's input, abstracted to remove specific details (e.g., filenames, commands, URLs). For example, a pattern such as "*What is this [...] doing*", captures a common syntactic form regardless of the specific artifact used. This abstraction facilitates grouping similar queries and supports broader pattern analysis.
- **Query Subject:** The primary topic or focus of the question, indicating the subject matter the analyst is asking about (e.g., a command, malware, or file).
- **Task:** The underlying intent behind the query, categorised into functional areas such as *Command Understanding* or *Summarising LLM Output*.

To ensure consistent application of the agreed coding scheme, all three researchers independently coded a sample of 20 question-LLM response pairs. This sample was used to verify their initial shared understanding of coding the three attributes. This was followed by a calibration session, during which the same 20 pairs were jointly reviewed and coded to reconcile differences, refine the definitions, and develop a preliminary codebook. Once alignment was achieved, the researchers independently coded a further 310 (10%) question-response pairs to further test their collective understanding of the attributes and coding scheme before being confident in independently coding the remaining pairs. IRR, measured using Fleiss' Kappa across the three dimensions, showed strong agreement: 0.90 for Query Pattern, 0.82 for Query Subject, and 0.79 for Task, indicating substantial to near-perfect agreement. A reconciliation session helped resolve any disagreements and update the codebook.

Conversation tagging. Following a similar process described above, the next step involved grouping successive analyst queries that relate to the same topic or investigative thread and treating them as a single conversational unit by considering (a) overlapping security artifacts (e.g., commands, email rules), (b) continuity in investigative intent (e.g., both steps probing the same log entry), and (c) occurring within a 30-minute window (used only as a guideline rather than a strict cutoff). Each conversation was assigned a unique identifier (e.g., *C1*, *C2*). To ensure consistency and reliability in tagging, the three researchers independently labelled a random sample of 45 analyst questions, comprising 142 question-response pairs, then met to reconcile

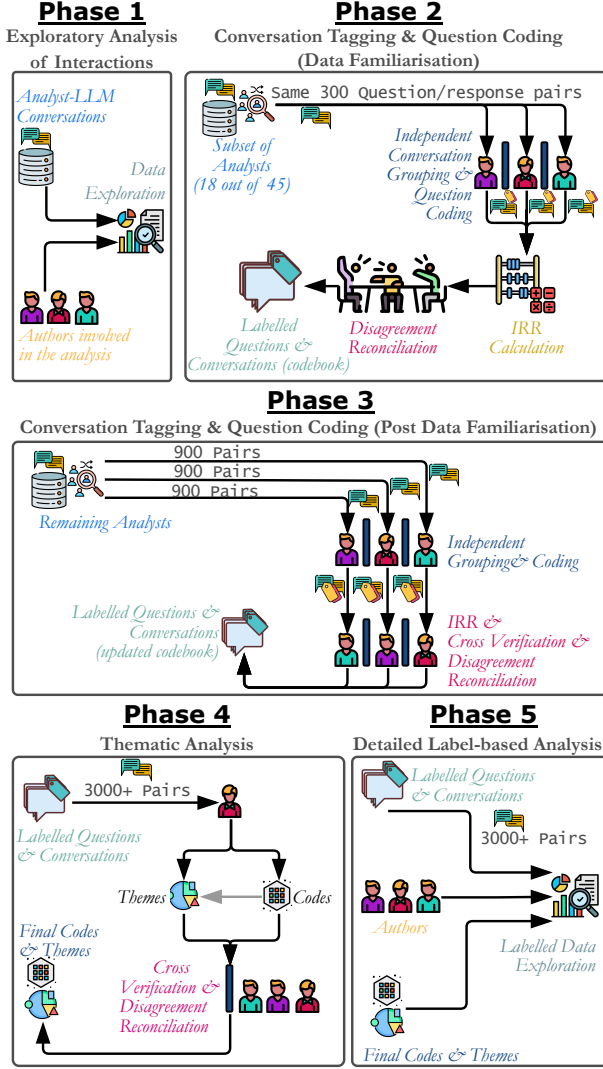


Figure 2: The five-phased approach to analyse and understand SOC analysts' interactions with LLMs.

and refine their understanding of the tagging process. Once consistent tagging practices were established, an additional 240 instances (13%) were tagged as a further verification step. Fleiss' Kappa was used to calculate IRR, resulting in a score of 0.75, indicating substantial agreement. Differences were reconciled through discussions.

PHASE 3: Remaining Dataset Question Coding & Conversation Tagging. In this phase, given acceptable agreement in Phase 2, the remaining question-LLM response pairs were divided among the three researchers, with each coding approximately 900 pairs independently. After initial coding, each researcher's annotations were cross-verified by another team member to ensure consistency and accuracy; discrepancies were resolved through discussions. To further enhance the reliability of the coding process, the lead author conducted a comprehensive review of all the coded data. Discrepancies were minimal, observed only in 4% of the

cases and resolved collaboratively through discussions.

PHASE 4: Query Level Thematic Analysis. One researcher consolidated all analyst question-LLM response pairs into a dataset and conducted query-level thematic analysis [50], as outlined in Figure 2. This phase built on earlier manual coding (PHASE 3) and introduced a hybrid approach that combined qualitative coding with data-driven clustering and thematic interpretation (PHASE 5) to surface higher-level insights. We applied semantic clustering to the coded data to support theme development. Sentence-BERT embeddings (stsb-roberta-large) and agglomerative clustering with cosine similarity [51] were used to group over 1000 distinct query patterns into semantically similar codes, supporting pattern abstraction. All clusters were manually reviewed to ensure semantic coherence, considering the original query and LLM response. Similar patterns such as *explain [x]* and *explain this [x]* were grouped. Themes were then developed by merging related clusters and assigning meaningful, high-level labels spanning across all attributes; e.g., *command* and *multiple commands* formed a broader *command-related subject* theme. For example, semantic clustering was applied to cluster 1,354 initial query pattern codes into 980 unnamed clusters, which were iteratively merged by the lead author into 147 codes and finally grouped into themes.

To ensure the robustness of these themes, we conducted a *triangulation process* [16] in which two researchers, who were not involved in the clustering, independently reviewed and refined the themes. This hybrid approach, combining coding, clustering, and human insight, balanced rigour with interpretive depth. [52], [53].

PHASE 5: Conversation and Analyst Level Thematic Analysis. In the final phase, we conducted an in-depth analysis using the final annotated dataset from PHASE 4. This involved both conversation-level and analyst-level analysis to uncover patterns at both levels. We first analysed the structure, frequency and patterns of conversations, identifying common interaction styles and task transitions. Analysts were then clustered based on usage volume, revealing distinct engagement profiles and thematic preferences. Finally, we explored behavioural and cognitive patterns, including low-engagement users, to understand diverse styles of LLM integration and potential disengagement plans.

4. Results

This section presents findings from our analysis of LLM usage. We start with an overview of SOC analysts, model information, and the dataset before diving into exploratory analysis. We then present the findings from a three-level thematic analysis covering query, conversation, and analyst levels. The dataset comprised 3,122 analyst-submitted queries to GPT-4, collected between May 18, 2023, and March 7, 2024. After removing 32 queries where the LLM could not provide a meaningful response (e.g., due to missing input or unsupported tasks), 3,090 valid queries and 532 distinct conversations remained for analysis.

| Analyst Role | Count | Responsibilities |
|----------------------|-------|---|
| SOC Analyst I | 24 | Handles initial alert triage and prioritisation, basic malware and network traffic analysis, IP/domain reputation checks, OSINT research, incident documentation, false positive tuning, and escalates complex cases. |
| SOC Analyst II | 4 | Coordinates intermediate incident response, manages specialised security tools, performs advanced log analysis and digital forensics, develops automation scripts, liaises with vendors, and reviews junior analysts' work. |
| Senior SOC Analyst | 7 | Leads advanced investigations and root cause analysis, conducts malware reverse engineering, runs threat hunting campaigns, mentors staff, optimises SIEM rules, coordinates across teams, and drives process improvements. |
| SOC Incident Handler | 1 | Collects and analyses threat intelligence, develops IOCs, attributes threat actors, monitors threat landscape, produces reports for leadership, integrates external feeds, and performs strategic risk assessments. |
| Threat Analyst | 6 | Oversees major incident response, containment, and eradication, handles stakeholder communication and coordination, facilitates post-incident reviews, executes emergency procedures, and supports continuity planning. |
| Associate Analyst | 3 | Provides entry-level monitoring, basic research, alert categorisation, report generation, and administrative support, while engaging in training and certification programs. |

TABLE 1: Roles and responsibilities of the 45 participating SOC analysts.

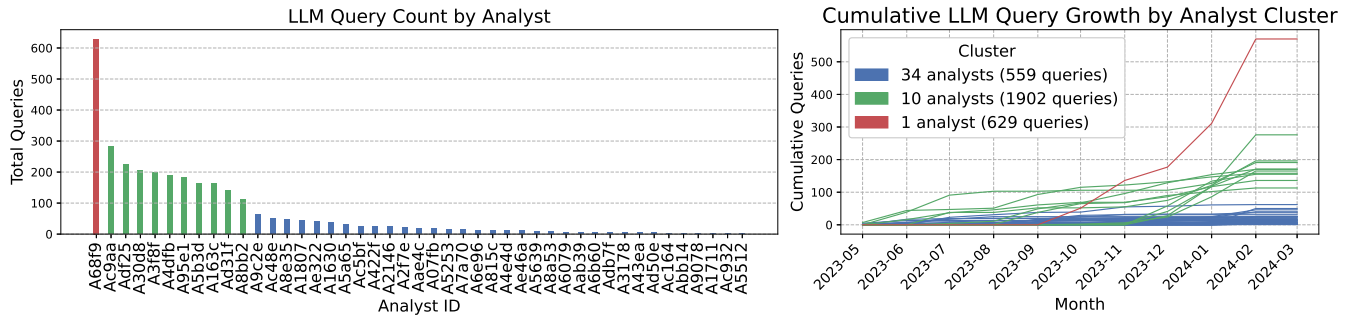


Figure 3: (Left) Query volumes vary across analysts, with heavy concentration among a few users. (Right) Overall, there is growing integration into workflows, but mostly driven by a subset of analysts (March 2024 has only 7 days of data).

4.1. Participant Overview and LLM Usage Constraints

Participants. Table 1 summarises the roles and responsibilities of the 45 SOC analysts who participated in the study. The majority were Tier 1 (SOC Analyst I), reflecting their front-line role in alert triage. Tier 2 (SOC Analyst II) and Tier 3 (Threat Analyst, Incident Handler) roles are more specialised and fewer in number, accounting for their lower representation. All participants, except for Associate Analysts (interns/co-ops, $n=3$), had 3–5 years of experience. Participation in the study was voluntary and the distribution of Tier 1, Tier 2, and Tier 3 analysts is not representative of actual staffing ratios. Analysts opted into the LLM pilot study via an internal Slack announcement, which invited exploratory use of the tool for work-related tasks.

LLM Setup. Participants accessed GPT-4 (GPT-4-0613) via a browser-based interface to an internally hosted “LLM-Gateway”,¹ which provided a control and monitoring layer for LLM usage. The gateway enabled API-based access to OpenAI’s GPT-4 Enterprise platform under a 30-day data deletion policy and a non-training agreement in place. Analysts were free to navigate to the system and use the model to support their work tasks. The model was used without fine-tuning and had no access to the internet or internal SOC systems. Although other models were available, this study focuses exclusively on GPT-4.

Data Sources. Analysts used proprietary and third-party tools to access telemetry from endpoints, networks, identities, and application layers, including EDR, NDR, SIEM/XDR, firewalls, DNS, email, identity, cloud, WAF/CDN, and SaaS applications. Participants were permitted to use only Public and Internal data, with strict policies prohibiting submission of Confidential or Highly Confidential information². Aside from this restriction, analysts were free to engage with the LLM as they saw fit. As the study focused on initial LLM engagement, all data were anonymised and devoid of personal identifiers. Accordingly, we report only aggregate usage trends, without attributing interactions to individual roles or experience levels.

4.2. Exploratory Analysis

Figure 3 (left) summarises analyst engagement with the LLM over 10 months. Usage was uneven, with a small number of analysts generating the majority of queries.

Given the skewed usage, we clustered the analysts based on their usage before analysing their long-term reliance on the LLM. We applied the Fisher-Jenks natural breaks algorithm³ to total query counts per analyst. This generated

1. <https://github.com/eSentire-Labs/LLM-Gateway>

2. Data was classified into four categories based on organisational sensitivity: Public data is freely accessible; Internal data comprises non-sensitive organisational information; Confidential data is protected by legal or contractual obligations; and Highly Confidential data is subject to the most stringent controls

3. https://en.wikipedia.org/wiki/Jenks_natural_breaks_optimization

three distinct clusters: a large group of *low-usage analysts* (Cluster 0, $n = 34$), a smaller group of *moderate users* (Cluster 1, $n = 10$), and a single *power user* (Cluster 2, $n = 1$; most active analyst). Figure 3 (right) shows that daily usage increased steadily over time, driven by a subset of highly active analysts (a stronger increase was observed among 11 analysts), suggesting deeper integration of the LLM into daily workflows. The number of active analysts also grew month on month, suggesting sustained interest and continuous adoption of the LLM.

While a single power user submitted approximately 600 queries (17% of the total dataset), usage was not limited to a few individuals. By early 2024, all 45 analysts had interacted with the system, and daily query volumes increased from fewer than 10 queries per day in May-August 2023 to over 30 queries per day in January and February 2024. We analysed the monthly usage patterns and observed that the number of active analysts grew from 4 in May 2023 to 13 in June, reaching 18 by July and steadily increasing to a peak of 25 in February 2024. By early March 2024, 16 analysts had already submitted queries, showing strong ongoing engagement.

The monthly increases include the proportion of returning analysts and new analysts rising each month (see Figure 8 in the Appendix). By early 2024, the majority of LLM interactions came from returning analysts, indicating ongoing integration of the tool into daily workflows. Most active analysts utilise the LLM in tightly clustered bursts, with a median gap of 1-2 hours between visits, suggesting its use as an on-demand aid between investigative steps.

The average query length was 25 words (standard deviation: 70), but this distribution was skewed by a few very long queries. Some analysts submitted highly verbose queries, exceeding 1000 words (due to code or script segments), while others consistently relied on concise prompts of fewer than 20 words. In contrast, LLM responses were more uniformly verbose, with a mean of 161 words (standard deviation: 97) and a relatively narrow interquartile range.

4.3. Query Level Analysis

This section discusses the thematic analysis results of *individual queries*. Each *conversation* could involve multi-step interactions. The thematic analysis was conducted on individual queries, rather than entire conversations, to retain fine-grained insights that a holistic approach might overlook. Section 4.4 presents conversation-level analysis. **The appendix includes the list of all themes; full codebook can be accessed via the link in the footnote:**⁴.

Analysts varied widely in usage patterns, with some submitting over 600 queries and others fewer than 5. To assess the robustness of our results to outlier effects following thematic analysis, we recalculated the proportion of queries assigned to each task theme after excluding (i) the most active analyst, (ii) the top 10 most active analysts, and (iii) both the most active and the 20 least active analysts.

4. https://osf.io/g8tr6/?view_only=db9c65552d364cba8c3a499bd12df589

| Themes | Theme Definition | Top Code Patterns (count,%) |
|--|---|--|
| 1 Command Understanding, Analysis & Generation (21 codes (948 samples)) | Centred on interpreting, analysing, or generating command-line inputs, including their structure, function, & potential threat relevance | Command understanding/analysis 861, 90.8% Command threat analysis 25, 2.6% Command Writing 17, 1.8% |
| 2 Text Generation & Editing (16 codes (676 samples)) | Editing & Involves the generation, rewriting, or refinement of text, including grammar correction | Editing / rewriting text 130, 19.2% Improving incident description 31, 4.6% Grammar correction 76, 11.2% |
| 3 Code, Script, and Regex Analysis (41 codes (341 samples)) | Involves writing, reading, or interpreting code, scripts, and regular expressions across programming or automation contexts | Script understanding 44, 12.9% Code writing & understanding 128, 37.5% Understanding ML concept 24, 7.0% |
| 4 Tool/Software/System Understanding (62 codes (163 samples)) | Focused on gaining insight into how specific tools, software platforms, or system configurations work | Tool understanding 54, 33.1% Software understanding 14, 8.6% Registry understanding 5, 3.1% |
| 5 LLM Chat: Greeting, Thanking, Version, Memory, Auditing (25 codes (156 samples)) | Centred around greetings, expressions of thanks, comments on versioning, or audit-related feedback | Interaction (thanking) 75, 48.1% Casual chat / conversation 15, 9.6% Interaction (greeting) 20, 12.8% |
| 6 File & Malware Analysis (30 codes (153 samples)) | Related to understanding file attributes, locations, and content, often in the context of detecting malicious behaviour | File understanding 94, 61.4% File location analysis 12, 7.8% File threat analysis 6, 3.9% |
| 7 Attack, Security & Threat Intelligence (75 codes (141 samples)) | Related to understanding attacks, malwares, and security threats for contextualisation or reporting | Attack understanding 24, 17.0% Attack type understanding 6, 4.3% Trojan understanding 7, 5.0% |
| 8 Query, Search, & Data Processing (23 codes (120 samples)) | Involves creation, refinement, or execution of search queries, often for use in log analysis, data filtering, or threat hunting | Query writing 72, 60.0% Answering multiple choice questions 8, 6.7% Query Understanding 6, 5.0% |
| 9 Network & Traffic Analysis (42 codes (92 samples)) | Focused on analysing IP addresses, URLs, & network traffic to understand communication patterns, identify threats, or interpret network behaviour | IP address analysis 14, 15.2% Network traffic understanding 18, 19.6% URL analysis 8, 8.7% |
| 10 Access Control & Authentication (45 codes (92 samples)) | Concerns remote access protocols, authentication methods, and policy or permission management | Remote management protocol understanding 10, 10.9% Policy & access control 12, 13.0% Authentication method understanding 9, 9.8% |

Figure 4: Summary of Task Themes (top 10)

Across all scenarios, the rank order of the top ten and the top two themes remained unchanged.

The findings from the exploratory analysis reveal an interesting landscape of LLM use in our dataset. Despite wide variation in query volume, engagement style, and query lengths, **usage trends point to a transition from periodic exploration to routine integration, with increased adoption over time and sustained interaction by a subset of analysts, with a stronger adoption among 11 analysts.** These observations offer grounded insights into how LLMs are appropriated in practice, setting the stage for the next phase of our study: a qualitative analysis of the tasks and interaction strategies.

4.3.1. SOC Analyst Task Types. Task themes reflect the actual work that analysts aim to accomplish when interacting with an LLM. **We find that analysts used LLMs for a range of tasks central to day-to-day SOC operations [1],**

[3]. These include interpreting and analysing commands; editing and improving the clarity of incident descriptions, documentation, and alert text; and understanding or troubleshooting code, scripts, and regular expressions. Analysts also used LLMs to analyse malware, explore system behaviours, build queries, and clarify email rules, highlighting the breadth of operational tasks. We highlight three common task themes below (see Figure 4 for the top 10), showing how LLMs integrate into workflows. For task adoption over time, see Figure 9 in the appendix.

Command Understanding, Analysis & Generation.

This was the most frequent task theme (31% of queries). Queries typically involved analysts submitting command-line instructions and requesting clarification on functionality, syntax, or behavioural purpose. Common prompts such as “*what does [x] do*” or “*explain this [x]*” ([x] is the actual command(s)) suggest investigative or verification-oriented goals, with the LLM supporting the analyst through interpretive scaffolding. While we did not perform a thorough thematic coding of the LLM responses, we **did** analyse the responses it provided for the command-related questions to understand why this theme was likely prominent.

The LLM often began with an overview of the command, explaining the components and their relationships, inferring the intent behind complex operations, and highlighting suspicious behaviours. For example: “*In summary, the [anonymised command] is trying to login as [anonymised user] without password and ...*”.

Many responses also included parameter breakdowns, example outcomes, risk implications, and recommended follow-up actions. Based on their sustained usage, we **hypothesise that by decomposing command components and synthesising likely intent and behaviour, the LLM has shown the potential of enhancing the analysts’ situational understanding**. This aligns with prior work (e.g., [17]), which finds that SOC analysts prefer explanations that contextualise evidence and inform action.

Text Generation & Editing. This theme, comprising 22% of queries, included requests to rephrase incident descriptions or alert summaries or correct grammar. Analysts frequently used queries such as “*make this better*”, “*can you correct this*”, or “*rephrase*”. **This theme underscores the increasing expectation that LLMs will assist SOC analysts in producing clear, coherent written documentation for SOC systems and peer or client communication.**

Code, Script, & Regex Analysis. Representing 11% of queries, this theme included submissions of code blocks (e.g., Python, PowerShell, JavaScript) and regex expressions. Analysts requested help understanding their behaviour, decoding strings, or generating new logic. Example queries included “*what does this script do*”, or “*write a regex to detect [x]*”. This theme suggests that **analysts and the LLM engaged in complex reasoning around syntax, logic, and security behaviour of scripts and code**. Regex generation was also notable, typically used in filtering logs.

Other Technical Tasks. In addition to the top three themes, analysts used LLMs for a wide range of other tasks including *Tool/ Software/ System Understanding*; *Attack, Security & Threat Intelligence*; *Access Control & Authentication*; *File & Malware Analysis*; *Network & Traffic Analysis*; and *Query, Search & Data Processing*. These tasks highlight specialised needs where LLMs offer targeted support. Notably, *LLM Chat* interactions reveal a growing rapport, with five analysts explicitly expressing thanks.

The results reveal a **growing reliance on LLMs across multiple functions**: from interpreting low-level artifacts, to generating or refining work products such as incident notes, queries, and email, as well as conducting on-demand look-ups, including definitions. **Findings suggest that LLMs are emerging as flexible cognitive aids** (e.g., as explainer or interpreter, drafting assistant, coding helper, and on-demand reference tool) rather than just static chatbots. The diversity of tasks calls for a unified integration.

4.3.2. SOC Analyst Query Subjects. We encoded query subjects to preserve anonymity (to not disclose proprietary information), resulting in a strong one-to-one mapping between subjects and task themes. For example, the task *Command Understanding* aligns with the subject *command*.

Across the 3,090 coded queries, subject matter clustered into a few dominant types: *command*-related artifacts (e.g., *powershell* and *curl* commands) led by a wide margin ($\approx 30\%$), followed by requests about phrasing or formatting snippets of written content ($\approx 17\%$) and code or scripting issues such as regex or PowerShell ($\approx 9\%$). Tool-specific questions ($\approx 7\%$) and file- or directory-path look-ups ($\approx 5\%$) rounded out the top five. The remainder was dispersed across LLM meta-chat, malware or threat look-ups, MDR/EDR query tuning, alert narration, network indicators, email-rule checks, and assorted ‘how-to’ prompts.

Telemetry tells us what was observed; analyst queries reveal which aspects they seek to interpret using LLMs. While security teams collect rich telemetry via tools like SIEMs and EDRs, LLMs are increasingly used to interpret parts of that raw input, transforming it into potentially actionable insights or contextual understanding. Analysts appear to have relied on LLMs to interpret telemetry-derived artifacts (e.g., commands, log fragments), to likely improve situational awareness and support timely decision-making.

The MITRE ATT&CK data sources⁵ describe telemetry collected by sensors (e.g., process creation, file access, authentication logs). Most **analyst queries focused on interpreting artifacts that these sensors collect, such as commands, scripts, OS processes or file paths**. While subject themes do not directly align with MITRE ATT&CK

5. <https://attack.mitre.org/datasources/>

| Themes | Theme Definition | Top Code Patterns (count,%) |
|--|--|---|
| Functional Understanding 129 codes (666 samples) | Aimed at understanding the behaviour, role, or functionality of code, commands, etc | What does [x] do 311, 46.7% What is [x] doing 68, 10.2% What is this [x] doing 32, 4.8% |
| Text Processing, Editing, Summarising 214 codes (494 samples) | Prompts ask the LLM to improve, rephrase, or refine given text | Make this better [x] 68, 13.8% Reword [x] 25, 5.1% Make it better [x] 21, 4.3% |
| Definition, Clarification or Categorisation Requests 137 codes (454 samples) | Questions seek definitions, descriptions, or classifications of a concept, term, or entity | What is [x] 186, 41.0% What is this [x] 38, 8.4% What are [x] 16, 3.5% |
| Explanation, Justification, Causal 181 codes (408 samples) | Requests for reasoning or explanations about what something does or why it behaves a certain way | Explain this [x] 97, 23.8% Tell me about [x] 26, 6.4% Explain [x] 21, 5.1% |
| Content Generation & Expansion 105 codes (211 samples) | Prompts instruct the LLM to create, add to, or expand content such as code or text | Create [x] 24, 11.4% Can you give me [x] 19, 9.0% Write [x] 12, 5.7% |
| LLM Interaction & System Queries 58 codes (161 samples) | Queries & responses directed at the system, including gratitude or system-related questions | Thank you [x] 40, 24.8% Thanks [x] 25, 15.5% What [x] of gpt are you 7, 4.3% |
| Confirmation or Open-ended 114 codes (138 samples) | used to either confirm a known or suspected fact (yes/no, true/false) or to explore a topic | is [x] [x] 9, 6.4% What are your thoughts on this 3, 2.2% True or False 3, 2.2% |
| How-To, Problem-Solving & Procedural 99 codes (124 samples) | Queries about performing a task, solving a problem, or carrying out a procedure | How can i [x] 6, 4.8% Follow the same procedure as [x] 4, 3.3% How to [x] 5, 4.0% |
| Feasibility & Capability Checking 80 codes (91 samples) | Queries testing whether the LLM can perform a certain action or process a specific input | Can you [x] 5, 5.5% Can you [x] the following [x] 3, 3.3% Does [x] support [x] 2, 2.2% |
| Correction & Error Fixing 19 codes (89 samples) | To identify, explain, or resolve errors in outputs, responses, or code | Can you correct the [x] 62, 69.7% Correct [x] 5, 5.6% Can you correct [x] 4, 4.5% |

Figure 5: Summary of Pattern Themes (top 10)

data sources, they reflect reasoning shaped by telemetry. For example, a PowerShell command launching a scheduled task maps to *Command Execution*.

4.3.3. SOC Analyst Query Patterns. Figure 5 shows how analysts linguistically framed their prompts. The most frequent pattern, **Functional Understanding** (21%), included queries like “*what does [x] do*”. These were used to interpret unfamiliar commands, scripts, or processes. **Text Processing and Editing** (16%) reflected efforts to refine reports or alerts (“*reword this*”), while **Definition** and **Explanation** requests (15% and 14%) were used to close knowledge gaps. Less common but operationally meaningful patterns included **Content Generation** (e.g., writing rules or code) and **Validation/Correction** (e.g., “*is this regex valid?*”). Importantly, these patterns reflect not just style but **intent**. For example, “what does” and “explain” queries almost always mapped to command interpretation tasks, while “make this better” appeared almost exclusively in text editing.

Query phrasing acts as a proxy for analyst reasoning goals, reflecting their underlying intentions and priorities, as well as their problem-solving approach and intended outcomes..

4.3.4. Alignment with Professional Cybersecurity Frameworks. Building on the task and query subject, we explored whether analyst queries aligned with established cybersecurity roles and competencies. We reviewed all queries against the NICE Framework for Cybersecurity⁶ by matching each task and subject to the NICE categories. This comparison revealed that 93% of analyst queries could be reasonably associated with at least one NICE Task, Knowledge, or Skill element; the other 7% mostly included task themes like *LLM Chat* or *Other*. For instance, command interpretation queries align with K0805 (Knowledge of command-line tools and techniques), while email crafting aligns with S0610 (effective communication skills).

These findings suggest that **LLM use was far from arbitrary**; it **clusters around** competencies already codified by **industry standards, such as the NICE framework**, suggesting that LLMs could be tailored to support NICE-specific work roles rather than providing generic assistance.

4.4. Conversation Level Analysis

One-off interactions comprised 41% of all interactions; the remaining 59% formed 532 multi-step conversations. The task themes ordering was stable across interaction types: *Command Understanding/Analysis* was most common, followed by *Text Processing*, then *Code/Script* tasks.

While the previous section focused on individual queries, this phase examines complete analyst-LLM *conversations*, each defined as a sequence of (temporally) linked queries. This broader lens allows us to examine how analysts pursue and refine their goals over multiple interaction steps. Our analysis examines (1) the distribution and structure of analyst-LLM conversations, highlighting common short and iterative interaction patterns; (2) the ways analysts structure queries and refine their outputs within multi-step sessions; and (3) the task transitions and sequences that characterise real-world analyst workflows when engaging with LLMs. This descriptive lens provides insight into how analysts employ LLMs as flexible, context-sensitive tools for reasoning, drafting, and refinement in SOC operations.

Our thematic coding centred on analyst queries, aligning with our objective of understanding how SOC analysts engage with LLMs in practice. Although we reviewed LLM responses to contextualise conversation-level analyst intent, we did not formally code them. This decision was based on the following considerations: (1) LLM outputs can vary in content or structure even when prompted identically, depending on deployment settings, making systematic analysis less reliable; (2) our primary research objective was to analyse analyst behaviour rather than audit or benchmark the LLM itself; (3) defining response utility is highly analyst- and context-dependent and often lacks objective ground

6. <https://www.nist.gov/itl/applied-cybersecurity/nice/nice-framework-resource-center>

truth; and (4) LLM responses were de-identified, making it difficult to judge the full task context. This analytic focus allowed us to prioritise the human perspective, capturing analyst reasoning patterns and interaction strategies, without over-attributing meaning to the LLM’s behaviour, which may vary depending on deployment or model settings.

4.4.1. Conversation Lengths. Table 2 summarises conversation length and the associated mean and median durations (elapsed time between first and last analyst message; LLM response time is not measured). Most conversations were brief: 57% involved just two analyst steps, and 75% contained only 2–3 queries. Only a small fraction (just over 4%) exceeded 10 steps. Durations increase with length, but longer conversations show much higher variance, with mean durations often far exceeding the median, suggesting that many multi-turn sessions are prolonged by idle periods or interruptions rather than sustained engagement. The following sections unpack how tasks and reasoning patterns unfold across these conversation lengths.

TABLE 2: Summary of Conversation Lengths and Durations

| Length | # Conversations | % of Total | # Analysts | Mean / Median Duration (minutes) |
|--------|-----------------|------------|------------|----------------------------------|
| 2 | 303 | 57.0% | 35 | 15.8 / 2 |
| 3 | 96 | 18.0% | 22 | 10.2 / 5 |
| 4 | 39 | 7.3% | 18 | 28.9 / 13 |
| 5 | 27 | 5.1% | 10 | 33.1 / 13 |
| 6 | 22 | 4.1% | 13 | 56.4 / 22 |
| 7 | 13 | 2.4% | 8 | 143.4 / 27.5 |
| 8 | 8 | 1.5% | 5 | 145.0 / 89.5 |
| 9 | 7 | 1.3% | 4 | 99.0 / 45.5 |
| 10 | 6 | 1.1% | 5 | 806.2 / 146 |
| 11 | 3 | 0.6% | 3 | 66.0 / 53 |
| 12 | 1 | 0.2% | 1 | 21.0 / 21 |
| 13 | 1 | 0.2% | 1 | 31.0 / 31 |
| 14 | 1 | 0.2% | 1 | 1151.0 / 1151 |
| 19 | 1 | 0.2% | 1 | 49.0 / 49 |
| 21 | 1 | 0.2% | 1 | 298.0 / 298 |
| 33 | 1 | 0.2% | 1 | 5770.0 / 5770 |
| 36 | 1 | 0.2% | 1 | 211.0 / 211 |
| 37 | 1 | 0.2% | 1 | 17728.0 / 17728 |

4.4.2. Most Common Two-Step Conversation Sequences. Analysis of frequent two-step sequences shows distinct patterns, with the most common being consecutive *Command Understanding/Analysis* (16.2% of conversations). This suggests that analysts often explore multiple related commands in succession or seek clarification on different aspects of the same command. Analysts often asked variations of “What does [x] do?” in both steps. Another common sequence is *Command Understanding/Analysis* followed by *Summarising Command* (3%), reflecting a workflow of interpreting and then summarising commands for reporting.

Several editing-related loops include repeated *Editing/Rewriting Text* (11%) and iterative revisions for tone or clarity. The third most common loop (4.7%) is repeated *Code, Script & Regex Analysis*. Other two-step patterns (1–

3%) involve *Attack & Threat Intel*, *Tool/Software Understanding*, *Query/Search Related*, and *Email Analysis*.

The recurrence of short, structured sequences suggests that analysts are not simply issuing isolated queries. Instead, they use the LLM to fill in gaps as they piece together the context of an event. For example, repeated command understanding shows how analysts incrementally contextualise related commands, while rewriting or summarising sequences often reflect the need to refine language or tone for reporting or documentation. **This behaviour reinforces the LLM’s role as a flexible tool: analysts use it to interpret telemetry or polish content in small, focused iterations that support the analysts’ larger investigative or communicative goals.**

4.4.3. Three-Step Conversation Sequences. In three-step conversations, analysts often stayed on one theme, e.g., [*Editing/Rewriting Text*, *Editing/Rewriting Text*, *Editing/Rewriting Text*] (5%), making iterative refinements like “Make this better”, then “Make it shorter”, and “Make it sound better”, showing that the LLM was used as a drafting and refinement tool for alerts, notes, or emails. Triplets of command understanding queries (4%) often repeated prompts like “What does [x] do?” or “Explain [x]”, showing iterative probing to interpret commands or components.

A small number of sequences involved transitions between different task themes (e.g., from command analysis to text editing, or alternating between command and tool understanding), highlighting cases where the analyst first interprets a command and then translates that understanding into an incident narrative. **These three-step interactions reveal how analysts engage the LLM to incrementally make sense of complex information, refine their articulation, and support analytical workflows.** The sequences reflect real-world cognitive processes, particularly when dealing with uncertainty or unfamiliar commands or scenarios.

4.4.4. Longer Conversations. Most interactions were short (2–3 turns), but longer ones, though rarer, show how analysts build on prior queries to deepen insight and refine results.

These longer conversations spanned a wide range of tasks. Some involved exploratory learning or upskilling, such as understanding machine learning fundamentals or understanding authentication mechanisms. Others reflected hands-on problem-solving, such as writing or debugging scripts, analysing remote shell behaviour, analysing registry modifications, or investigating potentially malicious behaviour (e.g., detecting potential ransomware). A distinct subset of longer conversations centred on professional development, likely related to CVs or performance reviews. These cases highlight the LLM’s use beyond core SOC functions, supporting analysts in broader professional contexts.

Conversations with four or more consecutive *Command Understanding/Analysis* steps suggest thorough exploration and careful interpretation of related commands in high-stakes environments. Other sequences chain distinct but related subtasks, combinations such as *File Understanding* → *Command Understanding/Analysis* → *Threat Analysis*

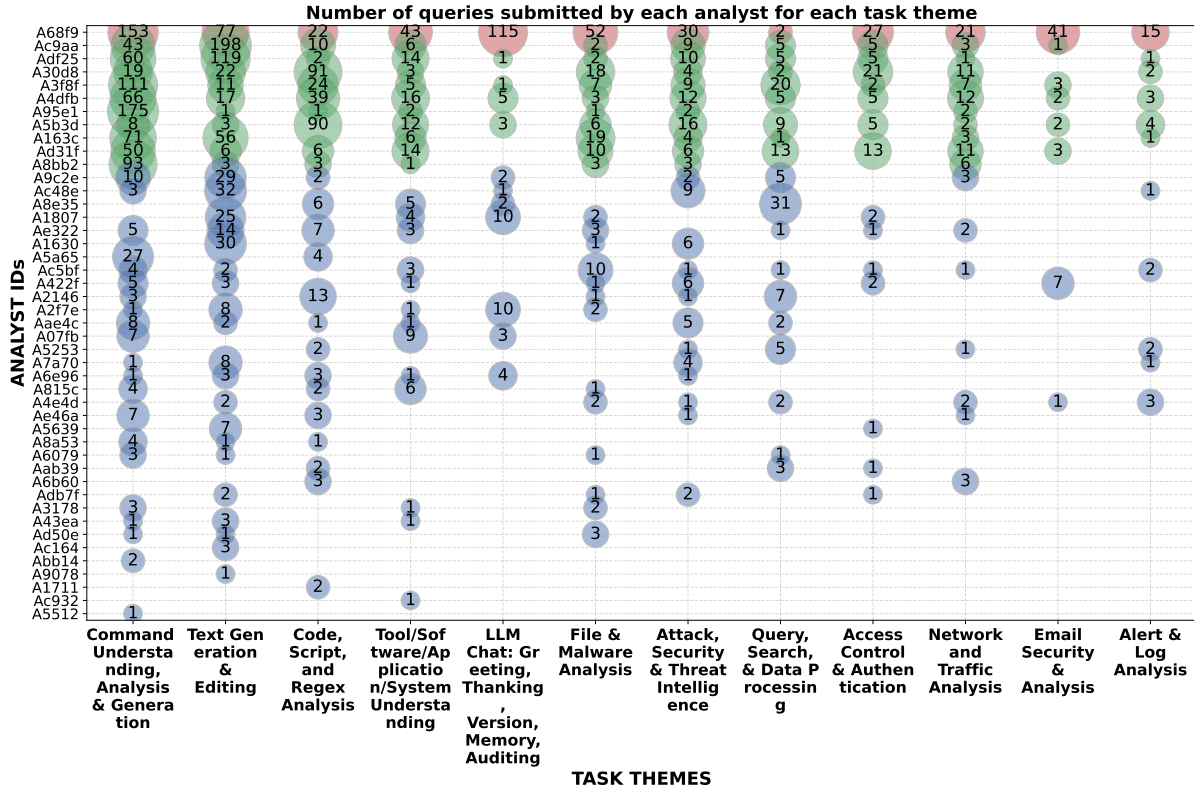


Figure 6: Number of queries per analyst (ordered by activity) and task theme (ordered by frequency). Analyst clusters are colour-coded: red for most active, green for moderate, and blue for low-usage analysts (same as Figure 3)

→ *Summarise LLM Output* indicate layered workflows: first clarifying data or command behaviour, then assessing threats, and finally seeking help with incident reporting. We also observed iterative development conversations with repeated cycles of *Code Writing and Understanding*, *Understanding Errors*, and *Query Writing*, suggesting that LLM supports implementation, error checking, refinement, and formatting. Several multi-step interactions focused on communication tasks—*editing text*, *improving clarity*, and *crafting empathetic emails*, combining structural edits with tone adjustments to produce polished, context-appropriate messages for diverse stakeholders.

Across these conversations, **patterns suggest that SOC analysts often use LLMs as iterative, context-sensitive tools for reasoning, drafting, and refinement**. Most conversations are short and focused, typically 2–3 steps, centred on tasks such as command understanding, text editing or coding tasks. While longer sessions are rare, they reveal more complex workflows, indicating that LLMs can support layered analytical and communicative goals. These findings suggest emerging usage patterns, though they should be interpreted with caution, given the predominance of brief exchanges and the absence of direct cognitive measures capturing analysts’ reasoning processes.

4.5. Analyst Level Analysis

Engagement with the LLM varied widely across analysts in query volume and thematic breadth. Analysts clustered into three groups: *low-usage* (Cluster 0, $n = 34$), *moderate* (Cluster 1, $n = 10$), and a single *high-volume outlier* (Cluster 2, $n = 1$).

We examined daily LLM query activity across all analysts (Figure 6) and observed distinct engagement patterns. One analyst (A68f) consistently used the system at a high volume across multiple months. A group of moderate users (e.g., Ac9a, A5b3) engaged in concentrated bursts of activity, likely linked to specific investigations. In contrast, most analysts issued only a few queries on scattered days, indicating sporadic or task-specific use. These behavioural differences support cluster-based segmentation and highlight varied LLM integration styles.

Analysts varied markedly in the task domains they engaged with through the LLM. The most prolific user, A68f, demonstrated broad usage across multiple domains, particularly *Command Understanding, Analysis & Generation*, *LLM Chat/System Queries*, and *Text Generation & Editing*. This suggests deep LLM integration into technical investigations and communication-related tasks. In contrast, Ac9a concentrated overwhelmingly on *Text Generation & Editing*, with limited engagement in command analysis, indicating a primarily communicative or documentation-focused use.

A30d and A5b3 mainly focused on *Code, Script, and Regex Analysis*, with moderate exploration of other tasks. Adf2 showed a balanced profile across *Text Generation*, *Command Understanding*, and *Tool/System Understanding*, using the LLM for both exploratory and functional support. A4df combined command and code tasks with moderate *Query/Data Processing* and *File Analysis*, indicating technically grounded use. Among lower-volume users, such as A180, Ac48, and A95e, task engagement was more narrowly focused, typically confined to one or two functional categories such as *Text Editing* or *Command Understanding*. Notably, the recurring appearance of *Command Understanding* and *Text Generation* across many analysts underscores their central role in LLM support within SOC workflows.

These task distinctions are mirrored by varying cognitive strategies at the pattern level. Figure 7 shows the top 5 task themes (left) and top 5 pattern themes (right) used by the 20 most active analysts. On the left side, usage is dominated by *Command Understanding* and *Text Generation & Editing*, though the mix varies by analyst. On the right, *Text Processing, Editing, & Summarising* is the most prevalent interaction pattern, but analysts also use a mix of patterns.

A68f, the most prolific user, led in *Functional Understanding* and *Clarification Requests*, with heavy use of *LLM Interaction & System Queries*. This pattern aligns with their diverse technical workload and diagnostic dialogue style. In contrast, Ac9a showed a documentation-focused usage profile via *Text Processing, Editing & Summarising*, alongside frequent clarification queries. Adf2 exhibited an iterative refinement strategy, issuing *Correction & Error Fixing* and *Content Generation & Expansion* queries; a cycle of generating, testing, and refining outputs.

Distinct pattern signatures also appear to reflect role-based tendencies. A30d and A5b3, for example, consistently used the LLM for *Feasibility Checking*, *How-To Reasoning*, and *Causal Explanation*, aligning with their hands-on technical work and exploratory debugging. Conversely, low-volume users such as A07f, A2f7, and Aae4 issued queries primarily related to *Clarification* or *Functional Understanding*, suggesting that for many, the LLM functioned as a reference assistant and a reasoning partner. However, it is important to note that a comprehensive understanding of analysts' specific tasks and context is necessary to accurately interpret such trends; variability in queries may simply reflect the evolving nature of investigative work.

We observed that **LLMs are not used uniformly** across analysts. Instead, **analysts appear to adapt the tool to their task demands, specific needs, and investigative styles**, ranging from quick clarification and script debugging to collaborative drafting and iterative refinement. The **LLM successfully supported this diversity without requiring extensive prompt engineering**. This highlights that a well-configured, domain-aligned LLM can support a broad range of task types and prompt styles, enabling quick and effective interaction, even without deep prompting expertise.

4.5.1. Discussion on Potential Disengagement Patterns.

The following analysis examines behavioural patterns among SOC analysts with lower engagement levels to understand where, why and how disengagements occur. We observed two distinct groups of low-engagement analysts.

Group 1: Low-Engagement, Early Drop-off Analysts (n = 13) These analysts interacted for only 1-2 months. Most (9 out of 13) showed no signs of frustration, implying exploratory or episodic use rather than dissatisfaction. However, three analysts exhibited potential breakdowns that coincided with their last few queries.

A2f7 repeatedly queried the model version (perhaps unmet transparency need), A5a6 received two consecutive erroneous answers, with a misclassification ending the session, and A6e9 disengaged immediately after the LLM refused a request. Another analyst (Ae46) ceased engagement after several command analysis queries, although no clear failure pattern was identified. Further analysis revealed that task theme rankings for this group mirrored the broader pool of analysts, indicating that early drop-off was not task-specific. Collectively, this group suggests that **early disengagement occurred for multiple reasons: in some cases without clear frustration, and in others following specific model failures**.

Group 2: Persistent but Mixed-Experience Analysts (n = 6) These analysts remained active for four to eight months. Two analysts (Ae32, Aae4) experienced apparent errors yet persisted, indicating that perceived utility can offset isolated failures. Other patterns suggest diverse usage motives: A9c2 moved past an early debugging issue; A214 relied on the LLM for repeated Sumo Logic refinements; A8bb may have engaged only when necessary; and A525 mixed non-SOC queries with SOC ones. **This group suggests that analysts tolerate occasional mistakes, perhaps if using the LLM for specific tasks is beneficial**. While these patterns are informative, future work should observe SOC analysts and conduct interviews to validate disengagement reasons.

4.5.2. AI-Assisted Decision-Making Perspective. Only a small proportion of analysts explicitly sought *recommendations* (we use recommendations here to mean classification requests, such as, “*is it malicious*” or “*is this a threat*”) from the LLM. Across the dataset, approximately 4% of all queries involved requests for such binary judgements. For example, if we only consider *Command Understanding, Analysis & Generation*, in total there were 950 queries (31%), and only 3% involved explicit recommendation requests, and these were issued by just seven analysts.

This suggests that **most analysts do not seek recommendations** from the LLM. Instead, they appear to **prefer evidence or context** that allows them to understand what is happening and **make decisions independently**.

To effectively support SOC analysts, AI systems must align with this preference for evidentiary support over prescriptive output, for example, by using *machine-in-the-loop*

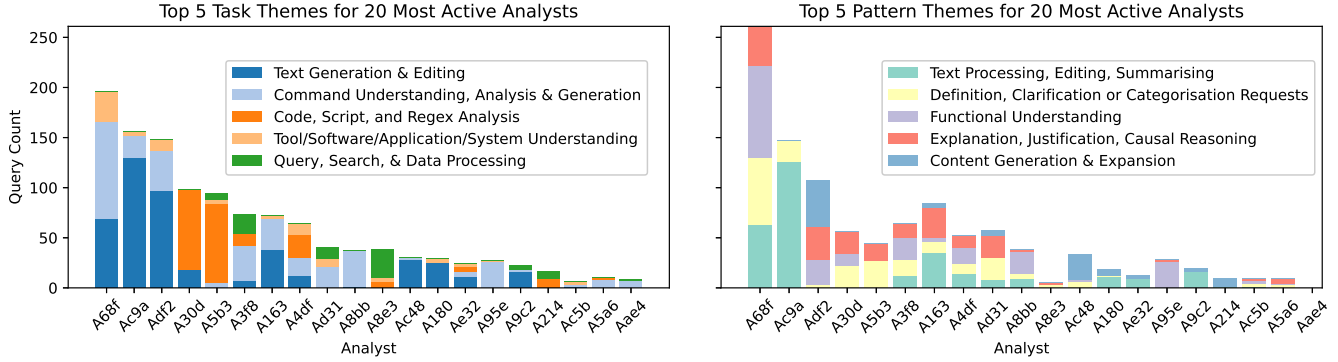


Figure 7: Top 5 task themes (left) and pattern themes (right) used by the 20 most active analysts.

approach [43], which presents evidence for and against outcomes rather than definitive answers to support informed decisions and preserve human agency [44], [45]. However, because some analysts prefer explicit recommendations, a one-size-fits-all approach risks leaving needs unmet. Therefore, investigating the need for adaptive systems that support both interpretive and directive workflows is essential.

5. Discussion

This study set out to investigate a core question (restated briefly): *How do SOC analysts utilise LLMs in their daily workflows, including the specific tasks they apply them to, how these uses align with established cybersecurity frameworks (e.g., NICE), and the conversational patterns that characterise their interactions?* Below, we address this question through a multi-level analysis and conclude with a discussion of the study’s limitations and the broader implications for designing LLM-based assistance in SOCs.

5.1. Query Level: Analyst Needs and Task Focus

The progressive rise in LLM use across tasks (Figure 6) and sustained engagement by more analysts (Figure 3) show that **analysts gradually integrated the LLM into daily workflows**, applying it to more tasks over time (see Figure 9 in Appendix). These usage patterns suggest that for some analysts, the **LLM delivered sufficient value to merit repeated use during investigative sessions**.

The LLM fulfilled a key, hidden interpretive layer missing from current SOC tooling, particularly in interpreting low-level telemetry, such as commands, files, system processes and scripts. This highlights a key function of the LLM: acting as a real-time interpreter or explainer of low-level telemetry. During investigations, Tier-1 analysts need to make sense of raw telemetry [5]. Instead of relying solely on documentation or prior knowledge, analysts used LLMs as on-demand aids to translate such artifacts into actionable context, potentially helping analysts discern whether observed activity was benign or malicious [5]. In this way, LLMs appear to reduce time-consuming lookups, ease cognitive load, and improve situational awareness.

Text processing use cases suggest a role in enhancing operational efficiency. Rather than spending time on repetitive communication or documentation tasks, analysts utilised the LLM to generate more polished and comprehensive materials for reporting and sharing. This would have offloaded non-investigative cognitive effort, supporting productivity without disrupting core investigative flow.

LLM usage is overwhelmingly concentrated in command interpretation and text processing, with most other task types showing signs of increasing adoption, but there is only preliminary evidence in our dataset to claim widespread use. Command and text processing tasks are used continuously and with growing frequency across analysts and months (see Figure 9 in the Appendix).

Query patterns reflected clear cognitive intent. Analysts’ queries were not just syntactic variations but reflected clear cognitive intent. Functional queries like “*what does [x] do*” mapped to interpretive tasks such as command understanding, while requests like “*reword this*” reflected text refinement goals. This alignment between linguistic patterns and task types suggests that query structure can serve as a proxy for the analyst’s reasoning objective.

Design Implication: Surface-level telemetry is not the same as true comprehension. Embedding plain-language, *just-in-time* explanations into analyst workflows could close this gap. For example, **embedding analyst role- and history-adaptive LLM explanations in SOC dashboards could accelerate triage, reduce cognitive load, and support evidence-based decisions**. Prior research highlights the need for actionable, context-rich explanations for SOC analysts [17], and that user-tailored interfaces improve decision accuracy and satisfaction [54]. Agentic RAG [55] could be explored to enable such explanations by dynamically tailoring retrieval and generation based on the analyst’s role and recent interactions, ensuring that each explanation is contextually relevant and personalised to the analyst’s expertise and workflow.

5.2. Conversation-level: Reasoning Sequences and LLM Roles

Our analysis reveals that SOC analysts engage with LLMs through brief, focused, and task-driven conversations. The results indicate that these 2-3 step conversations are not exploratory; rather, they are concise, goal-directed exchanges grounded in real-time investigative needs. Such behaviour aligns with the notion of *assisted intelligence* [34], in which LLMs are used as on-demand tools for clarification, explanation, or transformation.

In operational SOC, LLMs act as lightweight, on-demand aides rather than persistent copilots, helping analysts reach objectives without elaborate prompts. This real-world perspective contrasts with prior work focused on technical capabilities or prototypes [7], [8], [28], highlighting the need to study LLMs in live environments.

LLM use in our dataset aligns closely with established cognitive and collaborative AI role frameworks. For instance, command understanding aligns with the *clarifier* [36], the *Assisted Intelligence* role [34], and the *consultant/search engine* archetypes [35]. Text processing tasks correspond to the design phase in [36], and the *Augmented Intelligence* category [34]. Meanwhile, iterative reasoning and hypothesis testing interactions resemble the *deliberative partner* model [38], *ExtendAI* role [37] and align with *Co-operative Intelligence* [34]. Together, these roles underscore the varied, situational ways LLMs support cognition within the SOC environment.

Design Implication: Embed LLMs directly into SOC dashboards **so analysts can summon help for specific microtasks** (e.g., interpreting a suspicious command, summarising related logs, or drafting an incident note) without leaving their current workflow to **reduce context switching**, and suggest logical next steps (for example, offering a summary after several explanation requests).

Research Opportunity: While most analyst-LLM interactions focused on single tasks, some involved task transitions (e.g., command analysis to summarisation), raising questions about the value of task-specific LLMs and how to support seamless switching. Future studies could assess if task-specific LLMs enhance speed, transparency, and trust in SOC decision-making. Furthermore, *Agentic AI* could be explored to reduce context switching by assigning specialised agents to key micro-tasks within the analyst’s workflow [56], and even dynamically incorporate expert analysts [55], while carefully managing the security risks and operational complexity [57].

5.3. Analyst-Level: Usage Patterns and Integration Approaches

Our analyst-level analysis reveals key differences in LLM usage styles, cognitive strategies, and decision preferences, offering critical implications for the design, deployment, and evaluation of AI-powered SOC tooling.

LLM use varies. Analysts differed not only in the frequency of LLM engagement but also in the types of tasks they delegated and how they framed their requests. Some used LLMs for command parsing and debugging; others focused on editing or summarising. These distinctions suggest an opportunity for investigating the effectiveness of role-aware, adaptive integration strategies that personalise tooling based on emerging usage patterns. Our findings build on early claims of LLMs as cognitive aides and validate emerging theories of diverse user strategies and interaction modes [7], [8], [13], [39]. While prior work explored LLMs for structured extraction and task acceleration, we show that usage patterns naturally align with analysts’ task orientations, without prompt engineering or adaptation.

LLMs function primarily as evidence interpreters, not decision-makers. Only 4% of queries sought explicit recommendations (e.g., “is this malicious?”). Instead, most analysts requested explanation, interpretation, or contextualisation, highlighting a strong preference for maintaining decision authority. Our findings support designing LLMs as evaluative aides that surface evidence without prescribing actions, aligning with concerns about over-reliance on automation and reinforcing the need for expert-centred, machine-in-the-loop systems [41], [43], [44], [45]. Analysts’ preference for interpretation over recommendations warrants further study.

5.4. Other factors that modulate routine engagement: isolated errors and trust.

Disengagement is often episodic, not systemic. Among low-usage analysts, disengagement typically followed isolated errors or unmet expectations (e.g., refusal). Others continued using the LLM despite early failures, indicating that value outweighs occasional breakdowns. Clarifying outputs or allowing retries may reduce drop-off.

Sustained engagement hints at emerging trust. Trust underpins effective human-AI teaming, shaping user engagement over time. Prior work distinguishes *situational trust* (based on immediate interactions) from *learned trust* (built from experience) [58], [59], [60]. In our dataset, repeated use of the LLM for complex interpretive tasks, especially by some analysts, suggests the latter, where users expect reliable support in specific contexts. However, without direct feedback, we cannot confirm trust as the driving factor. Future research should combine usage data with trust metrics to clarify the nature of trust in human-AI collaboration.

Design Implication: Surface *evidence, not recommendations for investigative tasks* [37], [43], [44]. Because only 4% of analyst queries requested explicit malicious/benign judgements, for tasks that request such information, interfaces should default to an *evidence-first* mode, highlighting log excerpts, command traces, and ATT&CK linkages while omitting prescriptive labels.

Research Opportunity: Future studies could investigate whether the preference for evidence-oriented outputs (4% in our study) generalises to different settings, and also compare the impact of evidence-oriented versus recommendation-oriented outputs on accuracy, response time, and calibrated trust [40], [41] to determine which support style minimises misuse yet maintains efficiency during high-pressure triage.

Answering the RQ: SOC analysts primarily use LLMs as cognitive aids to interpret technical artifacts and streamline communication using brief, goal-oriented queries that are closely aligned with the NICE Workforce Framework and MITRE ATT&CK Framework. Most analysts view the LLM as a cognitive aid rather than a decision-maker. **Our findings indicate that LLMs serve as flexible, on-demand aids that enhance rather than replace analyst expertise.** Analysts appear to optimise both time and LLM utility through short, high-value interactions, quickly interpreting obfuscated telemetry for situational awareness [5] and efficiently completing writing tasks, while retaining full decision-making authority.

5.5. Limitations and Future Work

This study has several limitations. First, all data were collected from a single SOC using a specific LLM deployment. SOC vary in tools, workflows, and policies, so our findings may not generalise to other operational contexts or different LLMs. Second, we lack objective performance metrics, such as triage speed, false-positive rates, LLM response accuracy, or overall investigation accuracy. However, it is important to note that the primary goal of this study was not to evaluate performance outcomes but to explore how analysts interacted with the LLM in practice. High engagement levels suggest analysts found the LLM useful, but we cannot conclude whether these interactions translated into better or faster decisions. Third, there may be a novelty effect, with some analysts engaging with the LLM simply because it was new or conveniently accessible, rather than due to demonstrated utility, and some may have avoided it because of the perceived hallucination issues. As such, it is hard to separate genuine productivity gains from curiosity-driven use without further investigation.

Future work can build on our findings in several ways. First, replicating this study across multiple SOC with different toolchains, team sizes, and operating procedures will help assess the generalisability of observed interaction patterns and support the use of objective measures, such as time-to-triage. Second, integrating LLM features directly into SOC dashboards could reduce context-switching and offer deeper insight into how interface design shapes analyst-LLM interaction. Finally, longitudinal studies that track new and experienced analysts over time can reveal how trust, adoption, and productivity evolve as the novelty of the LLM wears off. Future longitudinal studies should include

interviews to capture analysts' experiences and perspectives, integrating these insights into design guidelines for effective LLM-analyst collaboration. By pursuing these directions, future work can move from descriptive usage patterns and provide more comprehensive insights into the role of LLMs in supporting analyst decision-making and collaboration within operational SOC.

Ethical Considerations: Ethical approval for this study was granted by the researchers' host organisation's Ethics and Privacy Office (Approval #: 025/25). Analysts were informed that their interactions would be anonymously logged to understand how they engaged with the new LLM tools, and consented to this. Informed consent was obtained by the industry partner, and participation was completely voluntary. Prior to data sharing, the industry partner de-identified the data and removed sensitive information. All procedures were reviewed and approved by both the industry partner and the host organisation's ethics and privacy teams.

6. Conclusion

This paper explored how SOC analysts interact with LLMs in an operational setting. By analysing thousands of analyst-generated queries, we found that analysts use LLMs as on-demand, task-focused cognitive aids for a variety of tasks, including explaining commands, writing scripts, or improving documentation, rather than as full-time copilots. Future tools could embed LLMs directly into SOC systems to deliver adaptive, context-sensitive assistance. From a research perspective, our work provides a grounded foundation for understanding analyst-LLM collaboration and underscores the need for outcome-based evaluations of their real-world impact on SOC decision-making.

7. Acknowledgments

This work is supported by the Commonwealth Scientific and Industrial Research Organisation (CSIRO) Collaborative Intelligence (CINTEL) Future Science Platform (FSP) and eSentire, Inc. Figures are designed using icons from Flaticon.com.

References

- [1] M. Vielberth, F. Böhm, I. Fichtinger, and G. Pernul, "Security operations center: A systematic study and open challenges," *IEEE Access*, vol. 8, pp. 227 756–227 779, 2020.
- [2] M. Baruwat Chhetri, S. Tariq, R. Singh, F. Jalalvand, C. Paris, and S. Nepal, "Towards human-AI teaming to mitigate alert fatigue in security operations centres," *ACM Trans. Internet Technol.*, vol. 24, no. 3, pp. 1–22, Aug. 2024.
- [3] K. Knerler, I. Parker, and C. Zimmerman, "Eleven strategies of a world-class cybersecurity operations centre," The Mitre Corporation, 2022.
- [4] F. Jalalvand, M. Baruwat Chhetri, S. Nepal, and C. Paris, "Alert prioritisation in security operations centres: A systematic survey on criteria and methods," *ACM Computing Surveys*, vol. 57, no. 2, pp. 1–36, 2024.

- [5] L. Kersten, T. Mulders, E. Zambon, C. Snijders, and L. Allodi, “‘give me structure’: Synthesis and evaluation of a (network) threat analysis process supporting tier 1 investigations in a security operation center,” in *Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*, 2023, pp. 97–111.
- [6] S. Tariq, M. Baruwat Chhetri, S. Nepal, and C. Paris, “Alert fatigue in security operations centres: Research challenges and opportunities,” *ACM Computing Surveys*, vol. 57, no. 9, pp. 1–38, 2025.
- [7] S. Freitas, J. Kalajdjieski, A. Gharib, and R. McCann, “AI-driven guided response for security operation centers with microsoft copilot for security,” *arXiv [cs.LG]*, Jul. 2024.
- [8] G. Siracusano, D. Sanvito, R. Gonzalez, M. Srinivasan, S. Kamatchi, W. Takahashi, M. Kawakita, T. Kakumaru, and R. Bifulco, “Time for aCTIon: Automated analysis of cyber threat intelligence in the wild,” *arXiv [cs.CR]*, Jul. 2023.
- [9] F. Sufi, “An innovative GPT-based open-source intelligence using historical cyber incident reports,” *Natural Language Processing Journal*, vol. 7, no. 100074, p. 100074, Jun. 2024.
- [10] M. Feffer, A. Sinha, W. H. Deng, Z. C. Lipton, and H. Heidari, “Red-teaming for generative AI: Silver bullet or security theater?” *AIES*, vol. 7, pp. 421–437, Oct. 2024.
- [11] J. Gao, S. A. Gebreegziabher, K. T. W. Choo, T. J.-J. Li, S. T. Perrault, and T. W. Malone, “A taxonomy for human-llm interaction modes: An initial exploration,” in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–11.
- [12] Y. Kim, B. Chin, K. Son, S. Kim, and J. Kim, “Applying the gricean maxims to a human-llm interaction cycle: Design insights from a participatory approach,” *arXiv preprint arXiv:2503.00858*, 2025.
- [13] J. Guo, V. Mohanty, J. H. Piazzentin Ono, H. Hao, L. Gou, and L. Ren, “Investigating interaction modes and user agency in human-llm collaboration for domain-specific data analysis,” in *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–9.
- [14] S. Ullah, M. Han, S. Pujar, H. Pearce, A. Coskun, and G. Stringhini, “LLMs cannot reliably identify and reason about security vulnerabilities (yet?): A comprehensive evaluation, framework, and benchmarks,” in *2024 IEEE Symposium on Security and Privacy (SP)*. IEEE, May 2024, pp. 862–880.
- [15] M. Lochner and K. Keplinger, “Collaborative intelligence: Topic modelling of large language model use in live cybersecurity operations,” *arXiv [cs.CR]*, Aug. 2025.
- [16] L. S. Nowell, J. M. Norris, D. E. White, and N. J. Moules, “Thematic analysis: Striving to meet the trustworthiness criteria,” *Int. J. Qual. Methods*, vol. 16, no. 1, p. 160940691773384, Dec. 2017.
- [17] J. Mink, H. Benkraouda, L. Yang, A. Ciptadi, A. Ahmadzadeh, D. Votipka, and G. Wang, “Everybody’s got ML, tell me what else you have: Practitioners’ perception of ML-based security tools and explanations,” in *2023 IEEE Symposium on Security and Privacy (SP)*. liminyang.web.illinois.edu, 2023, pp. 2068–2085.
- [18] F. Binbeshir, M. Imam, M. Ghaleb, M. Hamdan, M. A. Rahim, and M. Hammoudeh, “The rise of cognitive SOC: A systematic literature review on AI approaches,” *IEEE Open J. Comput. Soc.*, vol. 6, pp. 360–379, 2025.
- [19] M. Khayat, E. Barka, M. Adel Serhani, F. Sallabi, K. Shuaib, and H. M. Khater, “Empowering security operation center with artificial intelligence and machine learning—a systematic literature review,” *IEEE Access*, vol. 13, pp. 19 162–19 197, 2025.
- [20] T. Yang, Y. Qiao, and B. Lee, “Towards trustworthy cybersecurity operations using bayesian deep learning to improve uncertainty quantification of anomaly detection,” *Comput. Secur.*, vol. 144, p. 103909, May 2024.
- [21] L. Chavali, A. Krishnan, P. Saxena, B. Mitra, and A. Sreevalabh Chivukula, “Off-policy actor-critic deep reinforcement learning methods for alert prioritization in intrusion detection systems,” *Comput. Secur.*, vol. 142, no. 103854, p. 103854, Jul. 2024.
- [22] Z. T. Sworna, M. Ali Babar, and A. Sreekumar, “IRP2API: Automated mapping of cyber security incident response plan to security tools’ APIs,” in *2023 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, Mar. 2023, pp. 546–557.
- [23] I. Hasanov, S. Virtanen, A. Hakkala, and J. Isoaho, “Application of large language models in cybersecurity: A systematic literature review,” *IEEE Access*, vol. 12, pp. 176 751–176 778, 2024.
- [24] G. Deng, Y. Liu, V. Vilches, P. Liu, Y. Li, Y. Xu, M. Pinzger, S. Rass, T. Zhang, and Y. Liu, “PentestGPT: Evaluating and harnessing large language models for automated penetration testing,” *USENIX Secur. Symp.*, pp. 847–864, 2024.
- [25] M. A. Ferrag, A. Battah, N. Tihanyi, R. Jain, D. Maimuř, F. Alwahedi, T. Lestable, N. S. Thandi, A. Mechri, M. Debbah, and L. C. Cordeiro, “SecureFalcon: Are we there yet in automated software vulnerability detection with LLMs?” *IEEE Trans. Softw. Eng.*, vol. 51, no. 4, pp. 1248–1265, Apr. 2025.
- [26] K. Thomas, P. G. Kelley, D. Tao, S. Meiklejohn, O. Vallis, S. Tan, B. Bratanić, F. T. Ferreira, V. K. Eranti, and E. Bursztin, “Supporting human raters with the detection of harmful content using large language models,” *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 2772–2789, 2025.
- [27] L. D. Manocchio, S. Layeghy, W. W. Lo, G. K. Kulatilleke, M. Sarhan, and M. Portmann, “FlowTransformer: A transformer framework for flow-based network intrusion detection systems,” *Expert Syst. Appl.*, vol. 241, no. 122564, p. 122564, May 2024.
- [28] M. Hassanin, M. Keshk, S. Salim, M. Alsubaie, and D. Sharma, “PLLM-CS: Pre-trained large language model (LLM) for cyber threat detection in satellite networks,” *Ad Hoc Netw.*, vol. 166, no. 103645, p. 103645, Jan. 2025.
- [29] S. Tariq, M. B. Chhetri, S. Nepal, and C. Paris, “A2c: A modular multi-stage collaborative decision framework for human-ai teams,” *Expert Systems with Applications*, vol. 282, p. 127318, 2025.
- [30] T. Koide, N. Fukushi, H. Nakano, and D. Chiba, “ChatSpamDetector: Leveraging large language models for effective phishing email detection,” *arXiv [cs.CR]*, Feb. 2024.
- [31] S. Tariq, R. Singh, M. B. Chhetri, S. Nepal, and C. Paris, “Bridging expertise gaps: The role of llms in human-ai collaboration for cybersecurity,” *arXiv preprint arXiv:2505.03179*, 2025.
- [32] H. Cuong Nguyen, S. Tariq, M. Baruwat Chhetri, and B. Quoc Vo, “Towards effective identification of attack techniques in cyber threat intelligence reports using large language models,” in *Companion Proceedings of the ACM on Web Conference 2025*, 2025, pp. 942–946.
- [33] S. Mitra, S. Neupane, T. Chakraborty, S. Mittal, A. Piplai, M. Gaur, and S. Rahimi, “LOCALINTEL: Generating organizational threat intelligence from global and local cyber knowledge,” *arXiv [cs.CR]*, Jan. 2024.
- [34] T. Jiang, Z. Sun, S. Fu, and Y. Lv, “Human-AI interaction research agenda: A user-centered perspective,” *Data Inf. Manag.*, vol. 8, no. 4, p. 100078, Dec. 2024.
- [35] B. Song, Q. Zhu, and J. Luo, “Human-AI collaboration by design,” *Proc. Des. Soc.*, vol. 4, pp. 2247–2256, May 2024.
- [36] E. Eigner and T. Händler, “Determinants of LLM-assisted decision-making,” *arXiv [cs.AI]*, Feb. 2024.
- [37] L. Reicherts, Z. T. Zhang, E. von Oswald, Y. Liu, Y. Rogers, and M. Hassib, “AI, help me think—but for myself: Assisting people in complex decision-making by providing different kinds of cognitive support,” in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, Apr. 2025, pp. 1–19.
- [38] S. Ma, Q. Chen, X. Wang, C. Zheng, Z. Peng, M. Yin, and X. Ma, “Towards human-AI deliberation: Design and evaluation of LLM-empowered deliberative AI for AI-assisted decision-making,” in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, Apr. 2025, pp. 1–23.

- [39] R. Y. Pang, H. Schroeder, K. S. Smith, S. Barocas, Z. Xiao, E. Tseng, and D. Bragg, "Understanding the LLM-ification of CHI: Unpacking the impact of LLMs at CHI through a systematic literature review," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, Apr. 2025, pp. 1–20.
- [40] R. Fogliato, S. Chappidi, M. Lungren, P. Fisher, D. Wilson, M. Fitzke, M. Parkinson, E. Horvitz, K. Inkpen, and B. Nushi, "Who goes first? influences of human-AI workflow on decision making in clinical imaging," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '22. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 1362–1374.
- [41] Z. Bućinca, M. B. Malaya, and K. Z. Gajos, "To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, no. CSCW1, pp. 1–21, Apr. 2021.
- [42] K. Z. Gajos and L. Mamykina, "Do people engage cognitively with AI? impact of AI assistance on incidental learning," in *27th International Conference on Intelligent User Interfaces*, ser. IUI '22. New York, NY, USA: Association for Computing Machinery, Mar. 2022, pp. 794–806.
- [43] T. Miller, "Explainable AI is dead, long live explainable AI! hypothesis-driven decision support using evaluative AI," in *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '23. New York, NY, USA: Association for Computing Machinery, Jun. 2023, pp. 333–342.
- [44] T. Le, T. Miller, R. Singh, and L. Sonenberg, "Towards the new XAI: A Hypothesis-Driven approach to decision support using evidence," in *ECAI 2024*. IOS Press, 2024, pp. 850–857.
- [45] B. Shneiderman, *Human-Centered AI*. Oxford University Press, Jan. 2022.
- [46] R. Singh, M. B. Chhetri, S. Nepal, and C. Paris, "ContextBuddy: AI-enhanced contextual insights for security alert investigation (applied to intrusion detection)," *arXiv [cs.CR]*, Jun. 2025.
- [47] J. Li, J. Li, and Y. Su, "A map of exploring human interaction patterns with llm: Insights into collaboration and creativity," in *International Conference on Human-Computer Interaction*. Springer, 2024, pp. 60–85.
- [48] S. Chan, S. Fu, J. Li, B. Yao, S. Desai, M. Prpa, and D. Wang, "Human and llm-based voice assistant interaction: An analytical framework for user verbal and nonverbal behaviors," *arXiv preprint arXiv:2408.16465*, 2024.
- [49] J. Schneider, A. C. Flores, and A.-C. Kranz, "Exploring human-llm conversations: Mental models and the originator of toxicity," *arXiv preprint arXiv:2407.05977*, 2024.
- [50] V. Clarke and V. Braun, "Thematic analysis," *J. Posit. Psychol.*, vol. 12, no. 3, pp. 297–298, May 2017.
- [51] K. Abdalgader, A. A. Matrouf, and K. Hossin, "Experimental study on short-text clustering using transformer-based semantic similarity measure," *PeerJ Comput. Sci.*, vol. 10, no. e2078, p. e2078, May 2024.
- [52] G. Louwers, S. Pont, D. Gommers, E. van der Heide, and E. Özcan, "Sonic ambiances through fundamental needs: An approach on sound-scape interventions for intensive care patientsa)," *J. Acoust. Soc. Am.*, vol. 156, no. 4, pp. 2376–2394, Oct. 2024.
- [53] J. Chen, A. Lotsos, L. Zhao, C. Wang, J. Hullman, B. Sherin, U. Wilensky, and M. Horn, "A computational method for measuring "open codes" in qualitative analysis," *arXiv [cs.CL]*, Nov. 2024.
- [54] V. Lai, Y. Zhang, C. Chen, Q. V. Liao, and C. Tan, "Selective explanations: Leveraging human input to align explainable AI," *Proc. ACM Hum. Comput. Interact.*, vol. 7, no. CSCW2, pp. 1–35, Sep. 2023.
- [55] X. Xu, D. Zhang, Q. Liu, Q. Lu, and L. Zhu, "Agentic RAG with human-in-the-retrieval," in *2025 IEEE 22nd International Conference on Software Architecture Companion (ICSAC-C)*. IEEE, Mar. 2025, pp. 498–502.
- [56] R. Sapkota, K. I. Roumeliotis, and M. Karkee, "AI agents vs. agentic AI: A conceptual taxonomy, applications and challenges," *arXiv [cs.AI]*, May 2025.
- [57] N. Kshetri, "Transforming cybersecurity with agentic AI to combat emerging cyber threats," *Telecomm. Policy*, no. 102976, p. 102976, Apr. 2025.
- [58] W. Duan, S. Zhou, M. J. Scalia, X. Yin, N. Weng, R. Zhang, G. Freeman, N. McNeese, J. Gorman, and M. Tolston, "Understanding the evolvement of trust over time within human-AI teams," *Proc. ACM Hum. Comput. Interact.*, vol. 8, no. CSCW2, pp. 1–31, Nov. 2024.
- [59] M. J. McGrath, A. Duenser, J. Lacey, and C. Paris, "Collaborative human-AI trust (CHAI-T): A process framework for active management of trust in human-AI collaboration," *Computers in Human Behavior: Artificial Humans*, vol. 6, no. 100200, p. 100200, Dec. 2025.
- [60] S. Marsh and M. R. Dibben, "The role of trust in information science and technology," *Annu. Rev. Inf. Sci. Technol.*, vol. 37, no. 1, pp. 465–498, Jan. 2005.

Appendix

LLM Usage Summary: Monthly Patterns. Figure 8 summarises SOC analyst engagement with the LLM system, showing monthly active users (top) and new vs. returning users (bottom), highlighting adoption trends and increasing integration into daily workflows.

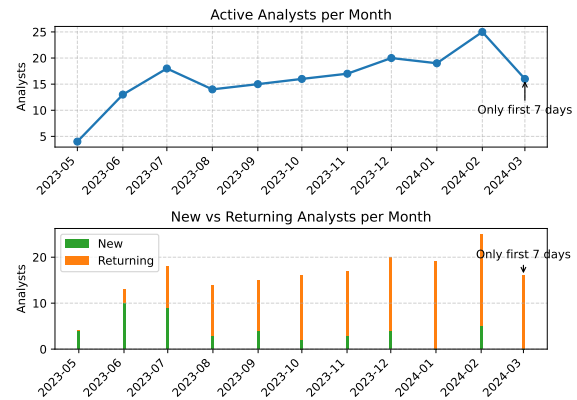


Figure 8: Analysts' engagement over months.

Monthly Task-Theme Usage. Figure 9 shows monthly query volumes by Task Theme, highlighting broad growth that suggests LLMs have become multi-role assistants in SOC investigations, reporting, and automation.

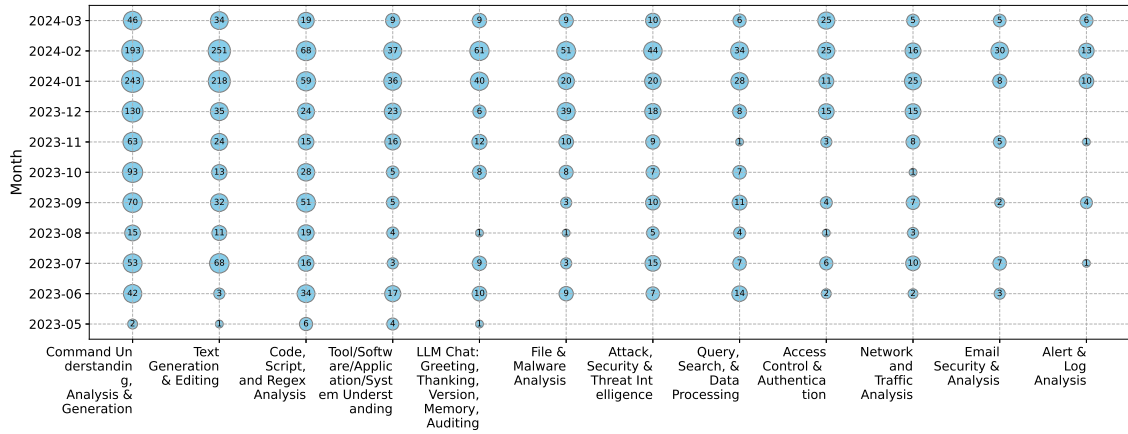


Figure 9: Monthly query volume by task theme.

TABLE 3: Summary of Pattern Theme with Top 3 Codes

| Theme | Theme Definition | Top 3 Code Patterns (Count, %) | # Codes (Total) |
|--|--|---|-----------------|
| Functional Understanding | Queries aimed at understanding the behaviour, role, or functionality of code, commands, etc | what does [x] do (311, 46.7%), what is [x] doing (68, 10.2%), what is this [x] doing (32, 4.8%) | 129 (666) |
| Text Processing, Editing, Summarising | Prompts ask the LLM to improve, rephrase, or refine given text | make this better [x] (68, 13.8%), reword [x] (25, 5.1%), make it better [x] (21, 4.3%) | 214 (494) |
| Definition, Clarification or Categorisation Requests | Questions seek definitions, descriptions, or classifications of a concept, term, or entity | what is [x] (186, 41.0%), what is this [x] (38, 8.4%), what are [x] (16, 3.5%) | 137 (454) |
| Explanation, Justification, Causal Reasoning | Requests for reasoning or explanations about what something does or why it behaves a certain way. | explain this [x] (97, 23.8%), tell me about [x] (26, 6.4%), explain [x] (21, 5.1%) | 181 (408) |
| Content Generation & Expansion | Prompts instruct the LLM to create, add to, or expand content such as code or text | create [x] (24, 11.4%), can you give me [x] (19, 9.0%), write [x] (12, 5.7%) | 105 (211) |
| LLM Interaction & System Queries | Queries and responses directed at the system itself, including gratitude or system-related questions | thank you [x] (40, 24.8%), thanks [x] (25, 15.5%), What [x] of gpt are you (7, 4.3%) | 58 (161) |
| Confirmation or Open-ended | used to either confirm a known or suspected fact (yes/no, true/false) or to explore a topic | is [x] [x] (9, 6.5%), what are your thoughts on [x] (3, 2.2%), [x] True or False (3, 2.2%) | 114 (138) |
| How-To, Problem-Solving & Procedural | Queries about performing a task, solving a problem, or carrying out a procedure | how can i [x] (6, 4.8%), how to [x] (5, 4.0%), in [x] pick [x] (4, 3.2%) | 99 (124) |
| Feasibility & Capability Checking | Queries testing whether the LLM can perform a certain action or process a specific input | can you [x] (5, 5.5%), can you [x] the following [x] (3, 3.3%), does [x] support [x] (2, 2.2%) | 80 (91) |
| Correction & Error Fixing | To identify, explain, or resolve errors in outputs, responses, or code | can you correct the [x] (62, 69.7%), correct [x] (5, 5.6%), can you correct [x] (4, 4.5%) | 19 (89) |
| Data Analysis and Interpretation | Prompts focused on decoding, interpreting, or drawing insights from data or patterns | decode [x] (15, 22.7%), how is [x] (3, 4.5%), are these [x] same (2, 3.0%) | 48 (66) |
| Information Gathering & Retrieval | Requests for factual information, examples, or references related to a topic or concept | tell me about [x] (2, 3.4%), Is there [x] (2, 3.4%), give me [x] (2, 3.4%) | 52 (58) |
| Explore Options & Alternatives | Questions that explore multiple possibilities, compare options, or consider different approaches to a problem, decision, or scenario. Often involves weighing alternatives or identifying the most appropriate course of action. | which of the following [x] (2, 3.8%), i need [x] for the [x] (1, 1.9%), which [x] patched [x] (1, 1.9%) | 52 (53) |
| Event & Outcome Analysis | Aimed at interpreting security events, determining what occurred, assessing impact, or understanding post-activity implications (e.g., after a login, installation, or process execution) | what is the outcome of [x] (6, 15.8%), whats happening here [x] (4, 10.5%), what is the result of [x] (3, 7.9%) | 27 (39) |
| Observation and Analyst Feedback | Statements or follow-up queries where analysts reflect on or react to prior LLM responses, sometimes providing their own assessment, confirming correctness, or probing further clarification | im trying to [x] could [x] help with that (1, 2.6%), but the previous version of the [x] worked (1, 2.6%), I can't find any information to verify [x] (1, 2.6%) | 38 (38) |

TABLE 4: Summary of Task Theme with Top 3 Codes

| Theme | Theme Definition | Top 3 Code Patterns (Count, %) | # Codes (Total) |
|---|---|---|-----------------|
| Command Understanding, Analysis & Generation | Centred on interpreting, analysing, or generating command-line inputs, including their structure, function, and potential threat relevance | Command Understanding/Analysis (861, 90.8%), Command Threat Analysis (25, 2.6%), Command Writing (17, 1.8%) | 21 (948) |
| Text Generation & Editing | Involves the generation, rewriting, or refinement of text, including grammar correction | Editing/Rewriting Text (130, 19.2%), Grammar Correction (76, 11.2%), Improving Incident Description (31, 4.6%) | 116 (676) |
| Code, Script, and Regex Analysis | Involves writing, reading, or interpreting code, scripts, and regular expressions across various programming or automation contexts | Code Writing and Understanding (128, 37.5%), Script Understanding (44, 12.9%), Understanding ML concept (24, 7.0%) | 41 (341) |
| Tool / Software / Application / System Understanding | Focused on gaining insight into how specific tools, software platforms, or system configurations work | Tool Understanding (54, 33.1%), Software Understanding (14, 8.6%), Registry Understanding (5, 3.1%) | 62 (163) |
| LLM Chat: Greeting, Thanking, Version, Memory, Auditing | Centred around greetings, expressions of thanks, comments on versioning, or audit-related feedback | LLM Interaction (Thanking) (75, 48.1%), LLM Interaction (Greeting) (20, 12.8%), Casual Chat/Conversation (15, 9.6%) | 25 (156) |
| File & Malware Analysis | Related to understanding file attributes, locations, and content, often in the context of detecting malicious behaviour | File Understanding (94, 61.4%), File Location Analysis (12, 7.8%), File-Related Threat Analysis (6, 3.9%) | 30 (153) |
| Attack, Security & Threat Intelligence | Related to understanding attacks, malware types, vulnerabilities, and broader security threats for contextualisation or reporting | Attack Understanding (24, 17.0%), Trojan Understanding (7, 5.0%), Attack Type Understanding (6, 4.3%) | 75 (141) |
| Query, Search, & Data Processing | Involves the creation, refinement, or execution of search queries, often for use in log analysis, data filtering, or threat hunting | Query Writing (72, 60.0%), Answering Multiple Choice Questions (Query) (8, 6.7%), Query Understanding (6, 5.0%) | 23 (120) |
| Network and Traffic Analysis | Focused on analysing IP addresses, URLs, and network traffic to understand communication patterns, identify threats, or interpret network behaviour | Network Traffic Understanding (18, 19.6%), IP Address Analysis (14, 15.2%), URL Analysis (8, 8.7%) | 42 (92) |
| Access Control & Authentication | Concerns remote access protocols, authentication methods, and policy or permission management | Policy & Access Management (12, 13.0%), Remote Management Protocol Understanding (10, 10.9%), Authentication Method Understanding (9, 9.8%) | 45 (92) |
| Email Security & Analysis | Involving assessment of email rules, phishing detection, and analysis of potentially malicious email behaviours like forwarding, auto-deletion | Email Rule Understanding (21, 34.4%), Email Rule Threat Assessment (13, 21.3%), Email Rule Analysis (5, 8.2%) | 16 (61) |
| Alert & Log Analysis | Tasks focused on interpreting security alerts, analysing logs and event messages | Log Analysis (4, 11.8%), Log Event Analysis (3, 8.8%), Windows Event Management (3, 8.8%) | 24 (34) |
| Others | Miscellaneous | Finding Movie Word Counts in a Range (5, 14.7%), General Knowledge Related Task (4, 11.8%), Keyword Research (SEO) (2, 5.9%) | 21 (34) |
| Concept Definition | To define, clarify, or understand security-related concepts, systems, or identity mechanisms in depth | Concept Definition (7, 23.3%), Term Understanding (6, 20.0%), Concept Understanding (4, 13.3%) | 15 (30) |
| Document/Evidence Search | Aimed at retrieving or summarising reports, templates, or locating relevant intelligence sources or documentation to support investigative work | Identifying Supporting Reports/Papers (13, 61.9%), Obtaining a Report (4, 19.0%), Fact Checking (1, 4.8%) | 6 (21) |
| System Process Understanding & Analysis | Involving analysis of system-level behaviours, understanding inter-process communication or OS-level services | OS Process Understanding (3, 21.4%), Process Activity Investigation (3, 21.4%), Process & IPC Analysis (2, 14.3%) | 8 (14) |
| Numerical Analysis | Involving math-related problem solving, percentage calculations, or interpreting numerically expressed patterns | Math Problem (6, 42.9%), Numerical Analysis (3, 21.4%), Math Problem Notation (1, 7.1%) | 7 (14) |

TABLE 5: Summary of Subject Theme with Top 3 Codes

| Theme | Theme Definition | Top 3 Code Patterns (Count, %) | # Codes (Total) |
|---|--|--|-----------------|
| command related (specific command, command line interfaces, multiple commands) | About command-line syntax, utilities, or individual commands used in operating systems or security-related tasks | command (831, 88.9%), command - malicious (9, 1.0%), command line utility (8, 0.9%) | 60 (935) |
| message / paragraph / phrase / sentence / term / article / report / templates / formatting instructions | About written content such as phrases, sentences, messages, or templates, including wording, formatting, terminology, and stylistic adjustments | message/phrase/paragraph/sentence (277, 53.4%), previous llm response (40, 7.7%), term (21, 4.0%) | 105 (519) |
| coding and scripting related (specific language, function, formula, code snippet, implementations, libraries, packages, development environment, errors, encoding types, regex) | About code, scripts, or regular expressions, including syntax, structure, or functionality across various programming or scripting languages | script (41, 14.3%), code (28, 9.8%), coding error (10, 3.5%) | 155 (286) |
| software & tool related (different types, OS, OS-level software, apps, applications, device drivers, development environments, installations, processes, OS-level processes, tools) | About specific software tools and applications, including features, setup options, or security-related configurations | tool (14, 6.2%), software (10, 4.4%), specific bitlocker configuration - bitlocker policy (9, 4.0%) | 150 (225) |
| file and directory related (including naming conventions, types, file actions, permissions, file behaviours, sharing mechanism, directory, directories related) | About file names, file paths, directories, and file system locations relevant to detection or investigation | file (file name) (82, 52.9%), file (file name) - specific location/folder (17, 11.0%), directory (3, 1.9%) | 51 (155) |
| LLM/GPT related (including privacy, chat auditing, version, history) | Referencing gpt or large language models, including their versions, capabilities, or role-based configurations | llm_gpt (119, 79.9%), gpt version (9, 6.0%), persona description (6, 4.0%) | 15 (149) |
| attack & security related (attack type, indicators, attacker motivations, concepts, solutions, scam, phishing, incidents, APT groups, threat groups, malware, malicious activity) | About malicious activity, including different types of malware (e.g., trojans, ransomware), attack techniques (e.g., phishing, brute force), and associated vulnerabilities that enable exploitation | malware type (12, 8.7%), attack type (12, 8.7%), network scanning method (4, 2.9%) | 94 (138) |
| MDR & Query related (including edr, edr query, rules, ids, rules, configuration, log related) | About queries, filters, or rules related to managed detection and response (MDR) systems or log analysis platforms | query (20, 16.5%), sumologic query (9, 7.4%), Sumo logic (6, 5.0%) | 70 (121) |
| alert, investigation, incident, security event related | About security alerts, incident details, investigation steps, or descriptions of suspicious or confirmed events | incident description (33, 30.8%), alert description (29, 27.1%), investigation description (13, 12.1%) | 22 (107) |
| network communication mechanisms related (including sockets, http, ftp, ports, API, rdp, packet data, protocol, ip address, id address range, cidr) | About network-level elements such as IP addresses, port numbers, and remote access protocols like RDP | port number (7, 6.7%), ip address (6, 5.7%), regex - ip address (3, 2.9%) | 77 (105) |
| email and mail flow management related (email rules, forwarding, mail flow config, client communication: emails, alerts, incidents, replies) | Different types of email rules and their interpretation | email (22, 26.8%), email rule (11, 13.4%), email rules (8, 9.8%) | 38 (82) |
| Education and Interview related (training materials, Q&A, interview questions) | Content focused on training, self-assessment, and preparation for interviews, such as scenario-based questions | scenario based interview questions (15, 20.5%), mitre-related questions and answer (13, 17.8%), different questions and detailed answers (10, 13.7%) | 22 (73) |
| access management related (users, accounts, groups, roles, privileges, active directory, authentication) | Topics related to user identities, roles, groups, privileges, authentication methods, and account activity | authentication methods (2, 3.1%), authentication method - specific os (1, 1.5%), organisation units - gpo - hosts (1, 1.5%) | 64 (65) |
| general data types (numbers, strings, data, time, hash, hex, base64, table, lists, json) | Numbers, strings, data, time, hash, hex, base64, table, lists, JSON | decoding data (10, 17.2%), math problem (6, 10.3%), number division (3, 5.2%) | 30 (58) |
| online sources and corroborating evidence or source related (research, papers, reports, online sources) | Evidence that could backup LLM or analyst | url (9, 30.0%), corroborating evidence or source - research (2, 6.7%), url click event (2, 6.7%) | 19 (30) |
| general terms | Miscellaneous or off-topic queries | movie word counts (5, 23.8%), world event (4, 19.0%), joke (4, 19.0%) | 10 (21) |
| device and hardware identify related (host, hostnames, domain names, storage device information) | Identifiers and information about hardware, hosts, domains, USB devices, or endpoints | domains (3, 25.0%), usb details (2, 16.7%), device (1, 8.3%) | 9 (12) |
| organisation and agreement related (companies, competitors, organisational policies and agreements) | Organisational context, including companies, competitors, compliance requirements, policies | compliance/agreement (2, 22.2%), company competitors (2, 22.2%), internal organisation process (1, 11.1%) | 7 (9) |