A Field Study to Uncover and a Tool to Support the Alert Investigation Process of Tier-1 Analysts

Leon Kersten
Eindhoven University of Technology
l.kersten.1@tue.nl

Kim Beelen
Eindhoven University of Technology
k.i.beelen@student.tue.nl

Emmanuele Zambon
Eindhoven University of Technology
e.zambon@tue.nl

Chris Snijders
Eindhoven University of Technology
c.c.p.snijders@tue.nl

Luca Allodi
Eindhoven University of Technology
l.allodi@tue.nl

***Abstract*—The alert investigation processes junior (Tier-1) analysts follow are critical to attack detection and communication in Security Operation Centers (SOCs). Yet little is known on how analysts conduct alert investigations, which information they consider, and when. In this work, we collaborate with a commercial SOC and employ two think-aloud experiments. The first is to evaluate the alert investigation process followed by professional T1 analysts, and identify criticalities within. For the second experiment, we develop an alert investigation support system (AISS), integrate it into the SOC environment, and evaluate its effect on alert investigations with another cohort of T1 analysts. The experiments observe five and four analysts, respectively, conducting 400 and 36 investigations, respectively. Our results show that the analysts' natural analysis process differs between analysts and types of alerts and that the AISS aids the analyst in gathering more relevant information while performing fewer actions for critical security alerts.**

I. Introduction

Security Operation Centers (SOCs) are tasked with monitoring the security of IT and non-IT infrastructures. Their complexity is rapidly increasing as new detection technologies, logging capabilities, and tooling to conduct incident investigations become available and part of day-to-day operations. Despite a large amount of effort being spent on automation [1]–[4], human analysts are, and are set to remain, at the forefront of security alert analysis [5]. Human analysts are tasked with investigating alerts that cannot be automatically classified [6] or whose automatic classification is too uncertain to be trusted, to identify possible signs of ongoing attacks from alert and related log data. Importantly, junior analysts (generally referred to as tier-1 analysts) play the critical role of quickly and reliably finding attack signals and escalating them to higher tiers in a SOC for in-depth investigations and potentially reporting the detected incident to the monitored network [7]. It is therefore crucial that tier-1 (T1) investigations account for all relevant evidence needed to make an informed decision about a specific attack. Recent related work [8] highlighted the importance of "structuring" the analyst's work around information categories aimed at capturing what information analysts ought to consider. However, it is currently unclear how the T1 investigation process "naturally" unfolds and to what extent it covers all relevant information before leading to a decision. Importantly, understanding the gap between T1 investigations and an "ideal" investigation process would allow the development of better tools supporting analyst investigations and may help SOC managers in devising more effective analysis and training processes within their SOC. Although previous research investigated (cognitive) tasks related to the workflow of SOC analysts [7], [9]–[12], the present study is, to the best of our knowledge, the first taking an information-driven approach to analyze the alert investigation process of security analysts in a real SOC, and integrating related insights in an operational tool supporting their decisions. To do this, in this work we address the following research questions:

1) **RQ1**: What type of information do T1 analysts consider when classifying alerts, what sources do they rely on to collect it, and to what extent does this depend on the type of investigated alert?
2) **RQ2**: To what extent does the analysis process followed by different T1 analysts vary between (a) alerts of different types and (b) analysts?
3) **RQ3**: Can the T1 analyst alert investigation process be improved by the addition of an alert investigation support system, and how does that impact the collected information during an investigation?

To answer RQ1 and R2, we (1) conduct an experiment (ES1) employing a think-a-loud protocol with five T1 analysts employed in a collaborating SOC to investigate what information they consider during the analysis process of 400 security alerts; (2) identify inefficiencies in the analysis process in terms of stability and considered information. To answer RQ3, we (3) develop an Alert Investigation Support System (AISS) tool, integrated in the collaborating SOC's SIEM and (4) run a second study (ES2) with a different cohort of T1 analysts

Symposium on Usable Security and Privacy (USEC) 2025
24 February 2025, San Diego, CA, USA
ISBN 979-8-9919276-5-9
https://dx.doi.org/10.14722/usec.2025.23034
www.ndss-symposium.org, https://www.usablesecurity.net/USEC/

employed at the same SOC to evaluate whether the devised tool effectively addresses the inefficiencies identified in ES1. Our contribution is multifold. Through ES1, we provide novel insights on how the investigation process of T1 analysts unfolds for alerts of different types. ES1 is the first study of its kind conducted in a real analysis environment with professional (junior) T1 analysts. We find that analysts often rely on implicit knowledge or assumptions about the investigated alert without considering all relevant evidence. The investigation process generally unfolds in a disorderly manner, with frequent context changes in the SIEM environment, which may lead to inconsistent reporting on investigation results. To investigate whether technological support can help analysts in their investigations, we develop a dedicated AISS following [8], and conduct a controlled experiment (ES2) to evaluate how the AISS impacts the analysts' investigation process. We find that the AISS helps analysts conduct better informed investigations. Critically, analysts using the AISS collect all relevant information related to actual attacks with significantly fewer actions (i.e., faster) than analysts not employing the tool.

This paper unfolds as follows. Section II introduces the background on T1 analysts in SOCs, and Section III discusses related work. ES1 is presented in Section IV; the AISS tool and ES2 are presented in Section V Finally, Section VI discusses our findings and addresses the limitations of our study, while Section VII provides conclusions.

II. Background

A. SOC and tier-1 analysts

Security Operation Centers (SOCs) provide network monitoring and attack detection services to ensure the security of networks and infrastructures. At first, incoming data such as network traffic or system actions are transformed into security events using technology from intrusion detection systems (IDSs) employing techniques from static detection, dynamic systems, machine learning and others [8]. The large volume of security events is aggregated into logs and alerts and is presented in so-called SIEM (Security Information and Event Management) systems which SOC analysts utilize to investigate potential security incidents [6]. However, effective IDSs and SIEMs do not provide a one-size-fits-all solution for SOC alert investigations. Oftentimes, data-driven problems such as low visibility of devices and networks [13], high volume of alerts [14] and high variety in available data [9] make alert investigations a mentally demanding task [12], which in turn contributes to high rates of burnout for SOC analysts [5], [15]. Furthermore, the operational process used to investigate alerts often depends on the specific SOC, especially as the relevant process becomes more technical and task-oriented [6], [16].

Organizationally, SOCs oftentimes structure their operations through a tiered system of analysts from tier-1 (T1) 'junior' analysts, to higher tiers (T3 or T4) [12], [17], [18]. T1 analysts investigate the high volume of incoming security alerts and discern interesting security events from benign events [8]. Alerts that possibly represent more severe incidents are escalated to higher tiers for analysis. Since T1 analysts control which alerts are considered by higher tiers, correct and efficient alert investigations by T1 analysts are crucial, as wrongfully dismissed attack-related alerts may lead to a significant delay in detecting the attack and thus a delayed response. Despite the criticality of T1 analysts, the process of collecting, connecting and interpreting the information that the analysts acquire is not yet understood in the literature.

B. The "Threat Analysis Process (TAP)"

In this work, we consider the types of information that analysts can observe and employ to explore the natural alert investigation process(es) analysts follow. To define information types, we rely on the framework defined by Kersten et al. [8], who devised a structured threat analysis process (hereafter referred to as the *TAP* for brevity). Table I defines four different types of information (referred to as "Information Category") that analysts can employ for alert investigations. In addition to providing a framework of information types for alert investigations, the TAP describes an order in which the information types should ideally be considered to reach a well-informed conclusion. This order is: *Relevance Indicators*, *Additional Alerts*, *Contextual Information* and *Attacker Evidence* [8]. The TAP proved promising in improving the accuracy of alert investigations conducted by junior analysts. The evaluation carried out in the work of Kersten et al. [8] shows that the odds of correctly classifying an alert increase by 167% when analysts follow the TAP.

III. Related work

The work of SOC employees have been extensively investigated. However, previous research has often focused on the workflow of the SOC as a whole [9]–[12], or adopted an organizational perspective [5], [16], [19], [20], rather than taking the perspective of a SOC analyst. Other research analyzes tasks within a SOC such as triage analysis [7] or rule management [21].

Among the works that conduct cognitive task analyses (CTA) on SOC analysts, the main difference is in the scope of what the task of a SOC analyst is, resulting in differences in how specific the defined actions of SOC analysts are. For example, [12] performs a CTA on the operations of SOC analysts and their decision making process. The authors identify key analyses performed by T1 analysts (such as triage analysis), as well as those performed by higher-tier analysts (such as escalation analysis). By combining the workflow of multiple key analyses, the authors devise a generalized workflow model for SOC analysts.

Differently, Gutzwiller et al. [9] took a more scoped approach and conducted a CTA of SOC analysts with a focus on how analysts develop situational awareness and the sources in which analysts gather their data to conduct their investigations. The authors highlight the complexity of alert investigations in SOCs where analysts handle large volumes of data from many sources. Furthermore, the authors stress that analysts need to

TABLE I
TYPE OF INFORMATION SOC ANALYSTS EMPLOY TO CLASSIFY ALERTS ACCORDING TO KERSTEN ET AL. [8].

Information Type	Definition	Order	References
Relevance Indicators (RI)	Information to classify whether the alert under investigation is even relevant for the SOC, based on the signature and the scope of the customer.	1	[12], [14]
Additional Alerts (AA)	Alerts related to the current alert that the analyst is investigating. This may be previous instances of the same alert triggering or alerts that surround the current alert.	2	[11], [12]
Contextual Information (CI)	Information about the behavior and other observables of the involved internal host.	3	[2], [11], [14], [16]
Attack Evidence (AE)	Any evidence relating to the alleged attack including the type of attack, attacker and any indication of success.	4	[12], [14]

connect the collected data to interpret the alert, which further conveys the complexity of alert investigations.

Zhong et al. [7] captured traces of the triage analysis process with respect to the actions performed and the hypotheses generated by the analyst throughout the task. Their results show that analysts employ different strategies to navigate and interpret the large volume of data considered during the analyses. In later work, Zhong et al. [4], [22] automate the triage analysis process using operation traces collected from expert analysts. They remark that the high performance of their automated tool depends on the quality of the collected traces, which in turn depends on the (however "good" or "bad") judgment of the analyst. Moreover, their approach considers actions such as "searching" or "filtering", yet the rationale why the analyst performs such actions and what information is obtained from these actions remain unclear.

Differently, Cho et al. [20] and Sundaramurthy et al. [19] investigated the process of a SOC from an organizational perspective. Both works stress the importance of tacit knowledge (as opposed to explicit knowledge) on SOC operations. Cho et al. [20] conducted ten interviews with SOC analysts and managers to extract their thought processes in three hypothetical attack scenarios. The authors stress the limitations of conducting interviews with hypothetical scenarios, as opposed to observing analysts in real life. Furthermore, the authors note that it is especially challenging to capture the thought process of new and inexperienced analysts with complex hypothetical attack scenarios. This suggests that junior analysts are not used to complex attack scenarios, possibly because their task often consists of more simple alert investigations. Furthermore, most of the initial alerts they are tasked to investigate are not worth escalating, as also found in other works [14]. However, it is important to capture the process of how junior T1 analysts perform complex alert investigations, as it is part of their mandate to identify "escalation-worthy" alerts that higher-tiered analysts should further investigate. Unlike Cho et al. [20], Sundaramurthy et al. [19] took an anthropological approach with real-life observations to study the SOC ecosystem. The authors find that a SOC is dynamic in nature: as the external world changes and new attacks develop, SOCs adapt their tools and processes to accommodate those changes. In later works [5], [16], [23], the authors highlight that different SOCs utilize different workflows through their organization. In addition, the authors build tools to aid the analysis processes, stressing the importance of introducing technology that supports SOC analysts. Sundaramurthy et al. [5] report that the tools they developed did not remain part of the SOC operations after the study, highlighting that creating tools capable of fitting a SOC workflow is a difficult task. In this paper, we work closely with a SOC to identify issues and inefficiencies in the analysis process followed by their analysts, and build a tool integrated in the SOC's SIEM interface to address those.

IV. EVALUATION STUDY 1 (ES1)

A. ES1 Methodology

Study goals. ES1 aims to evaluate the natural processes of T1 analysts conducting alert investigations for incidents leading to a success attack (Att), or no attack (NAtt). More specifically, our objective is to evaluate the types of information analysts acquire for both Att and NAtt alerts, identify the types of alert investigation processes between different analysts and alert types, and identify the inefficiencies of such processes.

Overview of the method. To answer our first two research questions, we collaborated with a commercial SOC (hereafter referred to as "*the SOC*") which offers network monitoring services for the education, IT services and manufacturing sectors. We conducted a think-aloud experiment with five junior T1 analysts recruited from the SOC. The analysts were tasked with conducting a total of 400 alert investigations on real data from the SOC. As cyber-attacks are relatively rare occurrences in real SOC data [24], we followed [24] and injected ten attacks into the SOC detection environment by replaying attack-related network traffic into the SOC sensors. We modified timestamps and target IP addresses to ensure that the injected attacks are plausible in relation to the monitored infrastructure and data collection. We manually transcribed analysts' verbalization of their security investigations and devised a coding scheme through an iterative process in collaboration with a senior analyst employed at the SOC. We then applied the coding scheme to the 400 investigation transcripts. The coding was performed by the two leading researchers. Given the difficulty and highly technical nature of the transcript data, the coding process was structured in

batches within which each researcher would code a number of the same transcripts to monitor for coding consistency.

The SOC. The collaborating SOC provides network monitoring services to one medium to large European university and numerous SMEs in IT, health, and manufacturing industries. The SOC relies entirely on (heavily customized) open source software based on the Security Onion [25] Linux distribution. Alert data is generated by a mixture of Suricata signatures for network traffic inspection, and Sigma rules for logs (from Windows/Linux hosts systems, network devices such as firewalls, etc.). The SOC permanently employs 7 employees (of which 4 conduct security monitoring) and every educational semester recruits two to nine interns as junior T1 analysts from affiliate MSc-level cybersecurity programs. This procedure is akin to that followed by competing SOCs in the region. The recruitment procedure consists of an initial assessment of the prospect analysts skill set (both technical and communicative) and includes a training program on alert analysis and SOC operations. The training consists of a two-week period in which new analysts attend in total eight theoretical training sessions: three on alert investigation, two shorter sessions on technical skills relating to OSINT and analyzing pcaps, and two short sessions relating to procedures within the SOC (e.g., customer communication and escalation procedures). Additionally, new analysts complete multiple hands-on training sessions, in which they are asked to analyze a plethora of alerts with continuous feedback from an experienced T2 analyst.

After the training phase, T1 analysts are responsible for classifying each incoming alert according to the SOC's alert taxonomy. The alert classification taxonomy for T1 analysts consists of two major categories: `Att` and `NAtt`. `Att` alerts are alerts that must be escalated (with the accompanying information collected during the alert investigation) to the T2 analyst, while `NAtt` alerts are dismissed. The SOC considers `NAtt` alerts any alert related to false positives (such as access to legitimate websites flagged as C&C traffic), benign scans, minor policy violations from network users, and failed attack attempts on internet-facing assets. Alerts related to successful attacks, e.g., a successful exploitation attempt, are considered `Att` and must be escalated for further investigation.

The Subjects. For this experiment, we recruited five junior T1 analysts from the SOC pool of interns.[1] As the SOC, similarly to its competitors, recruits junior analysts from the student pool of the MSc cybersecurity track at a medium-to-large European technical university, the educational background of the subjects is uniform. Furthermore, given that SOC internships overlap with the educational semesters of the university used for recruitment, all subjects have an identical work experience of approximately 3 months. We opted for an approach to recruit subjects with similar backgrounds to reduce confounding factors that can impact the process in which T1 analysts investigate alerts. The limitations of this approach are discussed in Section VI-A.

[1]SOC analysts work in fixed schedules to ensure that a sufficient number of analysts monitors alerts at all times. As such, they are a rare resource. The sample size in this study is comparable to that of related works especially when considering SOC analysts specifically [5], [19]–[21]. Limitations are discussed in Sec. VI-A.

TABLE II
ALERT DISTRIBUTION ACROSS ALERT CATEGORIES.

Category	Amount	Alert Classification
`C&C`	12	`Att`
`Malware` (succ.)	10	`Att`
`Malware` (not succ.)	29	`NAtt`
`Policy`	35	`NAtt`
`Scan`	114	`NAtt`
Total	200	

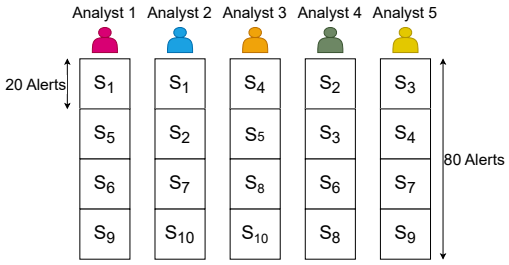

Fig. 1. Distribution of analysis scenarios assigned to the five analysts.

Alerts. As actual attacks are rare within the SOC, it is infeasible to collect enough real escalation-worthy alerts within the experimental time frame. Yet, it is critical to understand how analysts investigate escalation-worthy alerts, as they have the greatest impact if investigated incorrectly. Therefore, we injected ten attacks into the SOC detection environment (i.e., without launching any actual attack on end systems) and generate 22 `Att` alerts. We then sample another 178 `NAtt` alerts from the real-life environment of the SOC, for a total of 200 unique alerts. Details on attack injection and alert sampling are provided in Appendix A. The distribution of the final 200 (`Att, NAtt`) alerts across the different "rule categories" [26] attribute defined in the SOC is shown Table II.

Experimental setup. Figure 1 provides an overview of our experimental setup. To keep the experiment manageable, we split the set of 200 alerts into ten different "experiment runs" (called "scenarios") (S_1, S_2, ..., S_{10}), each featuring 20 alerts. Each scenario contains exactly one attack, and features all alerts pertaining to that attack.[2] All scenarios are completely disjoint (i.e., no alert is shared across multiple scenarios). Each analyst investigates four scenarios (for a total of 80 alerts). To allow comparisons between scenarios, each scenario is investigated by two analysts. This generates a total of 400 alert analyses. Figure 1 provides a graphical summary of the scenario assignment to the five analysts.

To avoid creating overhead on top of the normal analysis work performed by the analysts, the subjects participate in the

[2]This also assures realism of the experiment setup: an attack can take place during one monitoring shift and T1 analysts are expected to escalate all of these alerts to higher tiers.

experiment during their normal working hours. In agreement with the SOC management, the analyst's time spent on the experiment counts as regular service time in the SOC. Each analyst attends four experiment sessions (one per assigned scenario) over a period of one month. These experiment sessions were conducted in-person in a lab environment. To assure independent investigations, each session is attended by only one subject at a time and analysts are instructed not to discuss scenario findings with colleagues in between sessions.

During each experiment session, analysts are asked to investigate the alerts within the assigned scenario in the same way as they would investigate them as part of their daily job at the SOC. Similarly, they are instructed to classify each alert as `Att` or `NAtt` at the end of each investigation. Each scenario consists of a list of alerts, each linked to its corresponding entry in the SIEM environment. Analysts are instructed to 'think-aloud', i.e., to verbalize their thoughts during alert investigations. We strove to be as passive as possible during the alert investigations to not influence the subjects' decision-making process. However, subjects occasionally become silent and stop verbalizing their thoughts. In those cases, we followed best practices [27]–[29] and reminded them to follow the think-aloud protocol. In order to reconstruct the analysis process after the experiment, we recorded screen and audio throughout the experiment sessions (see Section IV-A for ethical considerations on collecting audio-visual data). Finally, following the experiment, we manually transcribed the resulting 400 audio recordings for further analysis.

Coding strategy and coding scheme. *Devising the preliminary coding scheme.* To understand the analysis process and the types of information gathered by the analyst, we devised a coding scheme to apply to the set of investigation transcripts. We code concepts related to information types collected by analysts (RQ1) separately from concepts related to the process they follow (RQ2). Therefore, we divide our coding schema into two main independent categories: `input` (to answer RQ1) and `process` (to answer RQ2). We define `input` as any verbalization indicating an acquisition of *information* pertaining to the security analysis such as observing a network log or a potentially malicious IP address. Furthermore, we also consider any verbalized recollection of information (e.g., remembering that an alert is a common false positive) `input` as well. We define `process` as any verbalized *action* aimed at processing or acquiring relevant information. Examples are checking the existence of specific protocol-related logs or investigating the purpose of a domain. We follow [8] to categorize `input` and `process` codes into four types of information shown in Table I: Relevance Indicators (`RI`), Additional Alerts (`AA`), Contextual Information (`CI`) and Attacker Evidence (`AE`).

To populate the described coding structure with specific codes, we conducted three brainstorming sessions among four of the authors and a senior analyst with more than 4 years of experience at the SOC, who regularly supervises T1 analysts. In these sessions, we discuss the types of information that T1 analysts acquire (**RQ1**), where they acquire it from (**RQ1**) and what specific actions analysts can perform to acquire it (**RQ2**). Importantly, we establish the mappings between the information collected by the subjects and the types of information described in the past literature [8]. Note that at this stage, we opted for a deductive approach relying on senior analyst experience (as opposed to solely derive codes inductively from the data) to first identify the main set of actions and information that would be relevant to look for in the data. This is critical because the T1 analyst's job is highly specialized and contextual in nature. Hence, an inductive approach from the onset may have resulted in illogical categorizations of codes that do not fit the context of that SOC. By contrast, we employ insights from the analysts as guidance for the detailed, data-driven coding procedure.

TABLE III
INTER-RATER RELIABILITY THROUGHOUT THE DEVISING AND APPLICATION STAGES OF THE CODING SCHEME.

	Round	`input` (%)	`process` (%)
Devising	1	-	-
	2	62	75
	3	76	86
Application	1	85	81
	2	91	84
	3	81	83
	4	90	97
	5	93	95
	Total	88	88

Finalizing the coding scheme. After devising the preliminary codes, we refined the coding scheme through an iterative coding process in three rounds, and involving two of the authors of this paper. In each round, we randomly sampled four alerts per subject (one per scenario). Therefore, in each round, we sample $5 \times 4 = 20$ alerts. The two authors leading the coding efforts independently coded the set of 20 alerts and compared the outcomes. For each round, we assessed inter-rater reliability (IRR reported in Table III) by calculating the joint rate of agreement. [3] Disagreements about assigned codes were discussed between the two authors leading the coding efforts and led to an iterative refinement of the coding scheme. Actions taken include merging codes too similar to distinguish, adding previously not considered codes, or tuning the code definitions in the codebook. After 3 rounds of coding, we verified the coding scheme with the T2 analyst to ensure that minor modifications to the coding scheme still accurately reflected the nature of actions performed by a T1 analyst. With the approval of the T2 analyst, the coding scheme and the related codebook were finalized.

[3] We found that many segments were only coded by one of the two coders. This often occurs because subjects verbalize trivial actions with little impact on the alert investigation. Therefore, we only consider codes for segments that both authors find relevant. As this consideration was made after round 1, we miss the IRR for that round.

The final coding scheme. The final coding scheme separates the information that analysts acquire from the actions the analyst conducts (i.e., `input` and `process`) and maps them to the information categories introduced in [8]. Furthermore, our coding scheme contains an NA category for information or actions for which a clear link to the framework described by the aforementioned work [8] is missing. For `input`, we further distinguish between information acquired from the SIEM, external tools (i.e., any tools beyond the SIEM), or recollection of previous experiences or any knowledge acquired in the past. Differently, within `process` each information category contains different possible actions. In total, `input` contains 15 distinct *(category, codes)* pairs, and `process` 29 (`RI`: 4, `AA`: 2, `CI`: 15, `AE`: 6, `NA`: 2). The final coding scheme is reported in Table VII in the Appendix.

Application of the final coding scheme to the transcripts. We applied the final coding scheme in a series of five rounds where in each round each of the two coders were randomly assigned two scenarios (i.e., 40 alerts each). Figure 6 in the Appendix shows an overview of the code application stage for one round. In addition to the two assigned scenarios, every coder coded four randomly selected transcripts of the other coder's batch. Therefore, in each round we coded 88 transcripts (i.e. $(40 + 4) \times 2$ per coder), eight of which (i.e., four per coder) where coded by both coders. Codes for these eight transcripts were cross-compared to monitor the stability of inter-rater reliability scores, and to resolve any remaining conflicts (all of which revealed to be minor). The lower part of Table III shows the inter-rater reliability throughout the aforementioned rounds. We observe a final inter-rater reliability (on codes assigned to relevant segments) of 88% for both `input` and `process`.

Evaluation strategy. To evaluate RQ1, for each code in `input` we count the number of alert investigations in which that code has been observed at least once (as opposed to the number of times the code has been observed). This allows us to evaluate whether an information type played a role in an investigation while avoiding noise from variations in verbalization rates, whereby the same information may be repeated multiple times in one investigation. To aggregate the counts by information category, we check the presence of a code associated with that information category (see Table VII in the Appendix). From these counts, we estimate effect sizes using a set of mixed-effects logistic regression models accounting for multiple measurements for subjects. We consider a threshold $\alpha = 0.05$ for statistical significance.

To evaluate RQ2, for each alert investigation we consider the chronological order of applied `process` codes. To avoid double-counting verbalized information multiple times within the data (i.e., repetitions), we merge subsequent identical codes as one. We evaluate the difference in the process of gathering information across analysts by observing the different information categories traversed throughout the investigation. Additionally, we measure complexity by the number of coded actions (i.e., the length of the aforementioned sequence of chronologically ordered codes) per alert investigation.

Ethical considerations. This research was executed with approval from our institution's ethical review board, with approval number `ERB2022MCS20`. We gained explicit and informed consent from all participants to participate in this experiment and to collect personal audio data from them following the think-aloud protocol. To minimize the risk of leaking any personal information, after transcribing the audio data and coding the transcripts, the audio data has been destroyed from all devices. Additionally, participant's names were anonymized moments after the think-aloud experiment to disassociate their identity from the data (e.g., from any performance evaluations at the SOC). Moreover, participants were assured that participation in the study would in no way affect their daily work conditions or employment.

B. ES1 Results

1) Types of acquired information (RQ1): Table IV provides an overview of the number of investigations in which the analyst gathered information for `Att` and `NAtt` alerts. An overview further dividing the investigated alerts into rule categories is shown in Table VI in the Appendix. In general, analysts consistently acquire information related to `CI` (97%), while only acquiring information related to other information categories (`RI`, `AA` and `AE`) in about half of their investigations (53%, 48% and 58%, respectively). Furthermore, we observe that using the SIEM is the predominant way of acquiring information. However, from the 223 investigations wherein `AE` was acquired, we find that analysts used external tools in 160 of them. Use cases include employing external tools (such as URLscan and VirusTotal) to investigate whether domains or IP addresses are malicious. This is illustrated by Subject 2 remarking: *"So here in the SSL server, I can indeed say that I will use URLscan and virusTotal to investigate if there are any indicators"*. By contrast, for other information types (`RI`, `AA` and `CI`), external tools were used in no more than 30 investigations. We observe that previous knowledge is more commonly used to recall information pertaining to `AA` (22%) than other types of information (`RI`:3%,`CI`:12%,`AE`:9%). Oftentimes, analysts such as Subject 1 would note the regularity of certain `NAtt` alerts appearing in their SOC by remarking: "*We have another SSH brute force, which happens quite often*". In addition, analysts are more often recalling previous knowledge to gather `AE` for `Att` alerts (20%) compared to investigations involving `NAtt` alerts. We notice that this is especially common when analysts notice malicious IPs that they have encountered before. Oftentimes, information from previous investigations aids current investigations. For example, Subject 1 remarks: "*this was not the problematic one the problematic one was [the IP] which sent [the] actual malware*", after observing two IPs that made a connection to the victim. The earlier collected and remembered information enabled the analyst to not repeatedly check the maliciousness of each observed IP.

When considering the SOC's taxonomy of alerts in relation to the acquired information (ref. columns "Total" for each information category in Table IV), we observe that investigations on `Att` alerts are more likely to acquire information related

TABLE IV
COUNT FOR THE EXISTENCE OF EACH CODE IN INPUT FOR ALERT INVESTIGATIONS PER ALERT CATEGORY.
S=SIEM , **P**=PREVIOUS KNOWLEDGE, **E**=EXTERNAL TOOLS, TOTAL DENOTES THE UNION OF THESE THREE SOURCES

	Relevance Ind.				Additional Alerts				Contextual Inf.				Attack Evidence			
Category (n)	**S**	**P**	**E**	Total	**S**	**P**	**E**	Total	**S**	**P**	**E**	Total	**S**	**P**	**E**	Total
Att (44)	35	2	5	**35**	16	10	0	**22**	43	3	7	**43**	29	9	35	**39**
(%)	(80)	(5)	(11)	**(80)**	(36)	(23)	(0)	**(50)**	(98)	(7)	(16)	**(98)**	(66)	(20)	(80)	**(89)**
NAtt (356)	168	11	25	**175**	103	77	0	**170**	344	46	18	**344**	156	26	125	**194**
(%)	(47)	(3)	(7)	**(49)**	(29)	(22)	(0)	**(48)**	(97)	(13)	(5)	**(97)**	(44)	(7)	(35)	**(54)**
Total (400)	**203**	**13**	**30**	**210**	**139**	**87**	**0**	**192**	**387**	**49**	**25**	**387**	**185**	**35**	**160**	**233**
(%)	**(51)**	**(3)**	**(8)**	**(53)**	**(35)**	**(22)**	**(0)**	**(48)**	**(97)**	**(12)**	**(6)**	**(97)**	**(44)**	**(9)**	**(40)**	**(58)**

to RI, as opposed to NAtt alerts (80% vs 49%, Coeff.: 1.8, $p :< 0.01$). We notice that in most cases apart from Subject 4 which consistently checks RI, analysts already know the rule to which most NAtt alerts trigger as such alerts are a common occurrence in SOCs. For example, Subject 3 remarks when double checking how the alert triggers "*Most of the time this alert triggered by some misconfiguration or transmission of SIP packets.*" after realizing that the subject has seen the rule before and already knows how it triggers in the SOC.

Unlike RI, we find no difference in the proportion of investigations where information related to AA is acquired between Att and NAtt alerts (50% vs 48%, Coeff.:0.2, $p : 0.66$). Regarding the collection of AA, we observe (despite their short work experiences) only one case where an analyst tries to understand how often the alert has been triggered in the past from the SIEM interface for both Att and NAtt alerts. Although analysts would often implicitly remark the commonness of certain alerts, only Subject 3 performed a search to see often an alert has been triggered before: "*I correlate other records [of this alert].*". This suggests that to find AA information, analysts exclusively use the SIEM system to find surrounding alerts as opposed to historical investigations of that alert.

Similar to AA, we find no difference in the proportion of investigations where CI is acquired between Att and NAtt alerts (98% vs 97%, Coeff.:0.4, $p : 0.70$). This is to be expected as we observe that CI is collected consistently through almost all investigations. Moreover, despite external tools being used rarely to acquire CI, we find that investigations of Att alerts are more likely to acquire CI via external tools than investigations involving NAtt alerts (Coeff.:1.3,$p :< 0.01$).

Finally, similar to RI but unlike AA and CI, we observe that investigations pertaining to Att alerts are significantly more likely to acquire AE than in NAtt alerts (Coeff:1.9, $p :< 0.01$). Considering the source of information, the use of external tools to acquire AE is especially more prevalent with investigations involving Att alerts compared to NAtt alerts (Coeff:2.0, $p :< 0.01$). As most information relating to AE that subjects acquire in our experiment relates to the (non-)maliciousness of external hosts, it is understandable that analysts collect more AE on Att alerts. Oftentimes, an investigation of a NAtt alert ends before the analyst observers whether the external host is malicious or not as they collect an critical information cue that allows the analyst to conclude that an attempted attack was

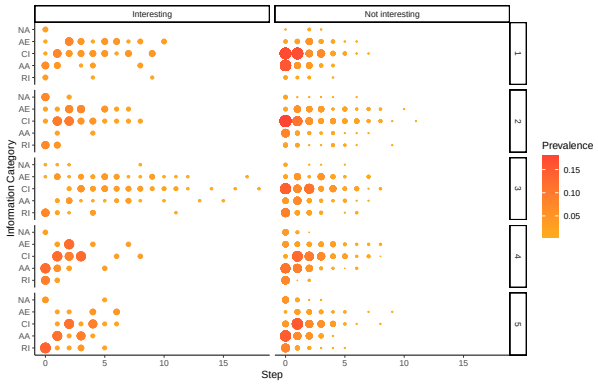

Fig. 2. Prevalence of actions for each information category as performed at different steps of alert investigations.

unsuccessful. For example, when analyzing a false positive where the internal host supposedly accessed many "malicious" domains within a short time frame, Subject 4 said: "*Here we have a get request with the 404 and another 404. And that's [the case for] all of them.*". In investigations such as these, the maliciousness of a domain is irrelevant as it simply did not exist (anymore) at the time the internal host accessed these domains. This phenomenon was commonly verbalized that "*404*" was expressed as a direct reason to dismiss the alert for 24 investigations on NAtt alerts.

2) Alert investigation process (RQ2): Figure 2 visualizes the rate at which analysts seek information of each type at different stages of an investigation. Each row corresponds to an analyst, and columns to alert types. Step counts (x-axis of each plot) indicate at which point in an investigation the analyst takes an action pertaining to a given information type. The size and color of the dots denote the proportion of investigations of that analyst (say, Subject 1) for that alert type (say, Att) during which an action pertaining to a certain information category (say, AE) was taken at that step (say, Step 2). Hence, circles indicate how frequently an action of a certain type will appear at a certain step given an investigation performed by that analyst on that alert type. We observe that emergent patterns differ widely among analysts, suggesting that each follows a different "sense-making" process of alert data. For example, Subject 4 and 5, follow the TAP described in [8] more often (as can be seen from the diagonal patterns

going from bottom left to top right) than Subjects 2 and 3, where the pattern shown in Figure 2 is more chaotic. For example, Subject 5 would always start an investigation of an `Att` alert and some `NAtt` alerts by checking the rule of the alert. Furthermore, we observe through the lengths of the emergent pattern (i.e., the total number of steps) that some analysts, such as Subject 3, often switch their investigation back to `CI` as opposed to acquiring `CI` first and other information categories later. An example that illustrates this is an investigation where Subject 3 alternated between opening logs (i.e., `CI`), observing an IP and checking its maliciousness (i.e., `AE`), realizing that they opened the wrong log, searching for the new log (i.e., `CI`), opening the log and again verifying the IP (i.e., `AE`). This process repeated eight times, making the alert investigation more inefficient than finding the log, verify that it is the relevant log related to this alert and then check the maliciousness of the relevant IPs in them.

Considering the different alert types, investigations on `Att` alerts appear to perform actions pertaining to all information types more often than `NAtt` alerts. Meanwhile, actions to gather more `CI` appear to be more prevalent in `NAtt` alerts as opposed to other actions. Interestingly, we observe that Subject 1 performs more actions relating to `AA` when investigating `NAtt` alerts as opposed to `Att` alerts. Similarly, actions pertaining to `AA` and `CI` appear to be conducted in later phases of an investigation for `Att` alerts while many investigations for `NAtt` alerts (especially for Subjects 1, 2 and 3) oftentimes start with an action to collect `CI`. Furthermore, we find that analysts often expect evidence to dismiss a commonly occurring alert from information pertaining to `CI`. This is best illustrated by an example where Subject 1 was analyzing a simple scan alert. Subject 1 starts the investigation by verbalizing: "*Do we have a response on the con[nection] log?*", indicating that the subject had no interest in checking other relevant information first as the subject was hoping for a failed connection attempt to quickly dismiss the alert.

In terms of the number of executed actions (see also Figure 7 in the Appendix), across all investigations analysts perform a median of 4 actions per investigation. When considering the alert types separately, we find that the median number of actions performed for `Att` alerts is 8 steps, double the number of steps corresponding to `NAtt` alerts, indicating that `NAtt` alerts are less complex to analyze than `Att` alerts ($W = 3184, p =< 0.001$ for a Wilcoxon signed-ranked test).

3) Observations on experimental outcomes: Despite previous works emphasizing that acquiring `RI` and `AA` is crucial for an alert investigation [8], we observe that analysts only acquire this type of information in about half of their investigations. Moreover, we observe that analysts seem to alternate the collection of different types of information relatively often, creating unstable patterns across alerts and analysts. This suggests that analysts struggle to follow an orderly investigation process. Further, the continuous contextual changes needed to switch from external tools (to collect `AE` information) to the internal SIEM can be time-consuming and is oftentimes cognitively taxing [30].

V. Evaluation Study 2 (ES2)

To answer our third research question, we first present the Alert Investigation Support System (AISS) [31] we developed to mitigate the issues identified in ES1: disorderly investigation processes and incomplete collection of information. We then describe the methodology followed for ES2 and its results.

A. Proposed Alert Investigation Support System

We developed an AISS that integrates the structure of the analysis process proposed in [8] in the Security Onion Console UI (i.e., the SIEM interface that the collaborating SOC employs)[4]. The main interface of the AISS is depicted in Figure 3, with data from a real event for illustration purposes. The AISS appears at the top (1) of the list of alerts in the SIEM interface. Analysts can pin alerts to the AISS so that the alert can be analyzed with AISS support (6). For each pinned alert, the AISS displays four tabs (3), each corresponding to an information category of the TAP. Each tab contains text boxes (5) with more specific information related to the information category which the analyst may collect. Information that is trivial to collect (but often not collected as shown in Section IV-B) is automatically retrieved from the alert data and displayed next to the relevant information category. Moreover, the AISS automatically generates SIEM queries to retrieve some information, such as how often an alert has triggered in the past or to show other alerts involving the same IP addresses as the alert under investigation. These queries are fired when the analyst clicks on the name of the specific information category (if and only if there is an automatic query to acquire the corresponding information). When an analyst acquires all the needed information about a certain information category, the analyst can mark the items as complete by checking the corresponding boxes in the interface. Information categories for which the analysts have already acquired all the information are marked by a green circle (2).

In summary, while not forcing analysts to follow the TAP, the AISS serves as (1) a reminder of the different analysis process steps, (2) a repository of information or queries to fetch relevant information for alert investigations, and (3) a notepad to organize observations during the investigation.

B. ES2 Methodology

Study goals. The goal of ES2 is to evaluate whether the AISS presented in Section V-A improves the alert investigation processes observed in ES1. More specifically, we aim to evaluate potential improvements in the investigation processes with respect to the identified inefficiencies in ES1. We consider `Att` and `NAtt` alerts separately when evaluating any potential improvements.

Overview of the method. For ES2 we ran a similar experiment to ES1 (see Section IV-A) approximately one year later (i.e., after evaluating results from ES1 and developing and

[4]The code of the developed AISS is portable to other web-based SIEM interfaces. The AISS code will be released as open source software under CC-BY 4.0 [31].

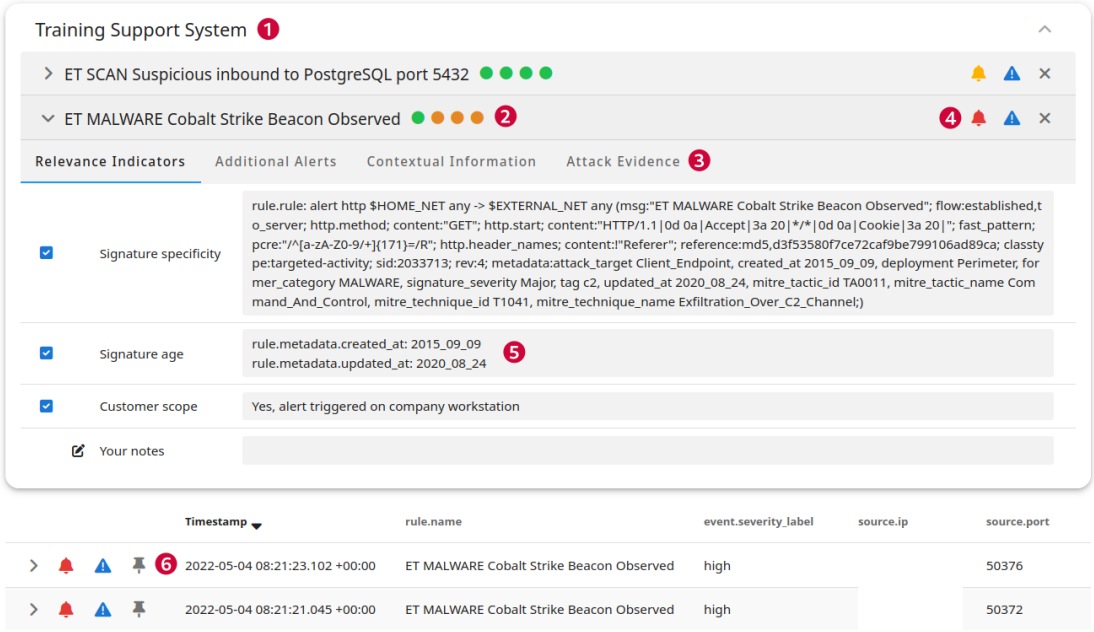

Fig. 3. The Alert Investigation Support System integrated in the SOC SIEM environment.

integrating the AISS in the SOC SIEM). However, due to resource and time constraints at the SOC, ES2 differs from ES1 in the number of analysts participating, and the number of alerts to investigate. We discuss these limitations in Sec. VI-A. For the data analysis we adopt the same code book and coding procedure as in ES1.

The Subjects. We recruited 4 junior T1 analysts from the pool of interns of the SOC. All subjects followed the same official recruitment and training procedure as subjects in ES1 (see Section IV-A) and come from the same recruitment pool. As in ES1, at the time of the experiment, all subjects have a work experience at the SOC of around 3 months. The similar background and identical selection, intake, and training procedure between ES1 and ES2 ensures subjects are comparable between studies.

Alerts. Due to the aforementioned operational time constraints on the side of the SOC, for ES2 we restricted the alert sample to 18 alerts from the set of 200 alerts used in ES1. In order to evaluate the effect of the AISS on the most critical alerts (i.e., the ones classified as `Att` according to the SOC's taxonomy), we opted to randomly select one alert from each injected attack in ES1, resulting in 10 `Att` alerts in our experimental set. The remaining 8 `NAtt` alerts were randomly sampled from the totality of 178 `NAtt` alerts from ES1. This means that *all* alerts investigated in ES2 are also investigated in ES1. This allows us to directly compare the results of ES2 with the corresponding results in ES1, while keeping the experimental setup compatible with the resources available at the collaborating SOC. Furthermore, we used an environment identical to that of ES1 (apart from the added AISS), such that all alert and log evidence available to analysts in ES1 is also available in ES2. This ensures a direct comparison between the two experiments.

Experimental setup and ethical considerations. Similar to ES1, we split the set of 18 alerts into two different "scenarios" (S_A and S_B) of 9 alerts each, each completely disjoint from the other. We assign S_A to Subjects 1 and 2, and S_B to Subjects 3 and 4; the subjects are asked to analyze alerts while thinking aloud. This generates a total of 36 alert investigations over 18 unique alerts. Subjects were asked to use the AISS during alert investigations. They were informed of the AISS features before the experiment was carried out.

Transcript coding and inter-rater reliability. As we want to compare the results of ES2 to those of ES1, we coded the transcripts with the coding scheme we devised for ES1. Given the lower number of transcripts in ES2, we applied the coding scheme in a series of two rounds where in each round, each of the two coders coded an identical set of 18 transcripts. Then, we calculated the inter-rater reliability in an identical manner to ES1 (see Section IV-A). We observe an inter-rater reliability (on codes assigned to relevant segments) of 81% for `input` and 84% for `process`.

Ethical considerations. to To mitigate ethical concerns, identical measures taken in ES1 (as described in Section IV-A) were taken in ES2. Furthermore, ES2 was executed with the same approval from our institution's ethical review board as ES1 (`ERB2022MCS20`).

Evaluation Strategy. To evaluate RQ3, we take an approach similar to that of RQ1 and RQ2. Further, for ES2 we compare the data collected in ES2 (treatment group) with the data collected in ES1 (control group) for those same alerts. We compare the counts of `input` code as shown in Table IV and the chronological ordering of applied `process` codes between the treatment and control group. Finally, we employ a

set of Wilcoxon ranked sum tests to compare the investigation complexities (measured by the amount of information analysts collect before making a decision) of Att and NAtt alerts between the treatment and control groups.

C. ES2 Results

1) Types of information acquired when using the AISS (RQ3): Table V provides an overview of the number of investigations in which an analyst gathered information for Att and NAtt alerts in ES2. We observe that in ES2, compared to ES1, analysts more consistently acquire information related to RI (100%), AA (89%) and AE in addition to CI (94%). We find that with the AISS analysts oftentimes check the rule signature to understand what attacks may trigger the alert but also what normal traffic may cause it to be triggered as a false positive. For example, Subject 1 (of ES2) remarks: "*So is the rule specific to an attack? Not necessarily because it's JA3 hash.*" referring to the fact that alerts triggering on JA3 hashes result oftentimes in false positives (as the same TLS client library can be shared by both malicious and benign software).

Considering the sources from which analysts acquire information, we observe (similarly to ES1) that the SIEM is still the predominant way of acquiring information. Furthermore, external tools are oftentimes used to find AE (81%) and revealing previously acquired knowledge was not uncommon to acquire AA (31%). Overall, we find no compelling evidence suggesting that the use of other sources increase through the use of AISS in addition to subjects simply collecting a wider range of information types for each investigation. For example, 6 out of the 7 times where RI was recalled by an analyst was Subject 1 recalling the usefulness of the availability of such information: "*OK, internal to external. Triggers on some bytes. Simply, yeah. OK, it has a reference URL link. Which are usually very informative.*".

Considering the dichotomy of Att and NAtt alerts, we find that the AISS group collects RI and AA information more than the non-AISS group for both Att (RI: 100% vs 80%, AA: 100% vs 50%) and NAtt alerts (RI: 100% vs 49%, AA: 75% vs 48%). We observe that the increase in acquiring AA for both types of alerts comes from analysts more often checking how often an alert has triggered in the past, while in ES1 this information was rarely checked via the SIEM interface. For example, Subject 3 remarks, while investigating an Att alert: "*Let's look at the alert history. What is it? ... But it triggered 142 times. Well, OK. That's a lot.*". Meanwhile, we find no evidence that the AISS leads to an increase in acquiring CI and AE for both Att and NAtt alerts.

2) Changes to the Alert Investigation process due to AISS usage (RQ3): Figure 4 visualizes the rate at which analysts seek information of each type at what stage of an investigation (similar to Figure 2 in ES1). To aid a direct comparison to ES1, we here aggregate the data of different subjects within the same study to provide an overall view of analysts' investigation processes across the two studies. For Att alerts we observe that subjects using the AISS are more likely to follow the TAP described by Kersten et al. [8] (as it can be

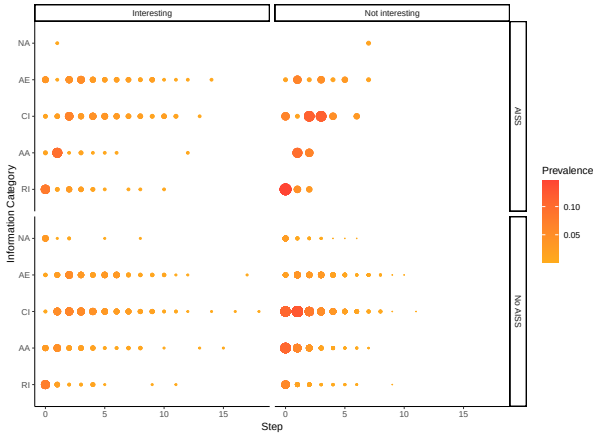

Fig. 4. Prevalence of actions for each information category as performed at different steps of alert investigations for both the treatment and control group.

seen from the diagonal patters from bottom-left to top-right). However, despite the use of AISS, subjects of ES2 commonly alternate between seeking CI and AE similar to the non-AISS group. This is especially evident in alert investigations that take more steps. For example, one investigation conducted by Subject 3 (of ES2) would alternate between the two types of information as they could not find the relevant log to check for "*the malicious domain*" similar to the example of Subject 3 in ES1. This is because the victim IP has accessed many domains (thus creating many logs) during the time frame of the attack. At some point, Subject 3 expresses hopelessness and starts copying every domain the subject encounters in an investigation: "*All [the logs], the same thing. I mean I guess I can copy this [to] VirusTotal*". Moreover, we find that analysts using the AISS conduct fewer actions to seek information outside the scope of the TAP than analysts without the AISS. This is indicated by the lack of actions related to the "NA" information category.

Taking into account investigations related to NAtt alerts, analysts using the AISS are more likely to follow the TAP compared to analysts who do not. Furthermore, we observe that when using the AISS, analysts switch less frequently between different information categories throughout the alert investigation compared to analysts without the AISS, as can be seen from the shorter patterns in Figure 4. Moreover, we find that analysts employing the AISS acquire RI as the first step of their investigation more often than the non-AISS group. Similar to investigations pertaining to Att alerts, we also observe that analysts who use the AISS perform fewer actions to seek information outside the scope of the TAP than analysts without the AISS for NAtt alerts, suggesting that the AISS streamlines the alert investigation process into the information categories described by the TAP.

Figure 5 shows the distribution of the number of actions performed in each investigation for both alert types for the AISS and non-AISS groups respectively. Considering all alert investigations performed in ES2, the median number of (verbalized) actions performed in an alert investigation is 4.5.

TABLE V
INPUT CODE PRESENCE IN INVESTIGATIONS OF DIFFERENT ALERT CATEGORIES USING THE AISS.
S=SIEM , **P**=PREVIOUS KNOWLEDGE, **E**=EXTERNAL TOOLS, TOTAL DENOTES THE UNION OF THESE THREE SOURCES

	Relevance Ind.				Additional Alerts				Contextual Inf.				Attack Evidence			
Category (n)	**S**	**P**	**E**	Total	**S**	**P**	**E**	Total	**S**	**P**	**E**	Total	**S**	**P**	**E**	Total
Att (20)	20	4	6	**20**	20	9	0	**20**	19	0	5	**19**	15	1	19	**20**
(%)	(100)	(20)	(30)	**(100)**	(100)	(45)	(0)	**(100)**	(95)	(0)	(25)	**(95)**	(75)	(5)	(95)	**(100)**
NAtt (16)	16	3	8	**16**	12	2	0	**12**	15	0	3	**15**	6	0	10	**11**
(%)	(100)	(19)	(50)	**(100)**	(75)	(13)	(0)	**(75)**	(94)	(0)	(19)	**(94)**	(38)	(0)	(63)	**(69)**
Total (36)	**36**	**7**	**14**	**36**	**32**	**11**	**0**	**32**	**34**	**0**	**8**	**34**	**21**	**1**	**29**	**31**
(%)	**(100)**	**(19)**	**(39)**	**(100)**	**(89)**	**(31)**	**(0)**	**(89)**	**(94)**	**(0)**	**(22)**	**(94)**	**(58)**	**(3)**	**(81)**	**(86)**

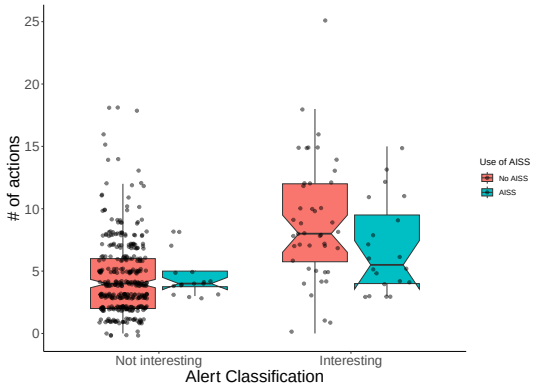

Fig. 5. The distribution of number of actions performed in an investigation aggregated across all analysts per alert classification for treatment and control groups.

Similar to ES1, we find that analysts perform significantly fewer actions when investigating NAtt alerts compared to investigations on Att alerts ($med_{Akk} = 5.5$, $med_{NAkk} = 4.0, W = 103, p = 0.034$). However, the difference in actions required to analyze an Att alert over a NAtt alert is less pronounced (difference of 1.5 actions as opposed to 4.0 actions). A one-sided Wilcoxon rank sum test reveals that analysts perform significantly fewer actions to investigate Att alerts when using the AISS ($W = 311.5, p = 0.031$). Therefore, despite analysts still alternating between actions related to CI and AE, we find that the general efficiency of alert investigations for Att alerts is improved by the AISS.

Meanwhile, we find no evidence that the AISS reduces the number of actions performed when analyzing NAtt alerts ($W = 3241.5, p = 0.828$). Yet, from Figure 5 we observe that the distribution is much less wider when analyzing NAtt alerts using the AISS compared to not using the AISS. By contrast to ES1, in ES2 we observe that analysts rarely dismiss alerts based on information gained from one action. Instead, analysts seek additional evidence to dismiss the alert. For example, Subject 2 (of ES2), when investigating an alert relating to data exfiltration, finds that no successful connection was established between the attacker IP and the host (thus no data was exchanged). Yet, Subject 2 continued investigating whether the attacker IP is malicious and concluded after few failed attempts: "*So I don't really understand the attack, but since I know there were no successful connections, I'm going to label it as NAtt.*", suggesting that not all thorough investigations of NAtt alerts result in more meaningful information.

VI. DISCUSSION

RQ1: Information types and sources Our findings suggest that although analysts acquire a variety of different types of information in an alert investigation, CI is acquired regardless of the analyst or alert type. This highlights the importance of CI for analysts to determine whether an alert should be escalated or not compared to the other information types. This is in line with previous works [2], [12], [14], [15] stressing the criticality of context around an attack and log data (which oftentimes is part of CI). Meanwhile, we found that AA is much less frequently considered across analysts, yet the incidence of AA information seems invariant to the type of alert under investigation. This raises the question whether other factors influence the acquisition of AA information. Regarding RI and AE, our findings show a much stronger effect of alert type in the frequencies at which information of different types is acquired. The lower acquisition frequencies of RI and AE in NAtt alerts as opposed to Att alerts suggest that NAtt alerts require less information for the analyst to conclude an investigation.

Considering the sources that analysts rely on to find information, we find that analysts predominantly use their SIEM system. However, AE information is more often acquired via external tools than other information types, although at similar rates than with SIEMs. This observation could be integrated in future work to provide a more streamlined analysis interface to the analysts. Moreover, our findings show that previous knowledge is recalled mostly in collecting information pertaining to AA. This suggests that information such as the history of an alert is in many cases not information that is truly acquired by an analyst but an already acquired information that resurfaces in the mind of the analyst. The need to recollect information as opposed to acquiring new information shows, in line with previous work [20], the rooted tacit knowledge among analysts. More research is needed to explore the concrete effects of recalling tacit knowledge within the alert investigation process specifically.

RQ2: Analysis process Our findings suggest that the investigation process an analyst follows varies widely among analysts and across Att and NAtt alerts. Some analysts focus on one type of information per step, while others chaotically switch

context between actions aimed at gathering different types of information. Future research can focus on exploring the natural process in different SOCs serving different industries.

Moreover, we find that alerts that ought to be escalated (i.e., `Att` alerts) require more actions on average to investigate than `NAtt` ones. Therefore, analysts are more likely to spend time and cognitive effort when analyzing escalation-worthy alerts. Since more complex tasks are generally harder to automate, SOC employees and researchers aiming to automate tasks within SOCs could focus on `NAtt` alerts, such as scan-related alerts, rather than broader approaches [1], [2]. Similarly, the tools developed for humans to assist in alert investigations should place a stronger emphasis on `Att` alerts, as these alerts require more complex investigations and thus potentially require more aid.

RQ3: Role of an AISS on the investigation process Our findings suggest that the proposed AISS improves the alert investigation process with respect to the inefficiencies identified in ES1. Firstly, analysts systematically and explicitly collect a wider range of information types during alert investigations, in both `Att` and `NAtt` alerts. The existing literature [8] has alluded to gaps between what the T2 analyst expects in a report and what the T1 actually reports. Although the cause of this gap is unclear, enabling the analyst to explicitly collect such information is a start to address this problem. Future research can consider how often T1 analysts do not report collected (yet relevant) information when escalating alerts.

Given from the results of ES1 that external tools are crucial to find attack evidence which is especially relevant for the more serious `Att` alerts, future research can focus on improving the efficiency of the analysts' investigation processes on these external tools. Additionally, as previous research [20] suggests that communication within SOC employees is key to demystifying tacit knowledge, future AISS can integrate features to make this knowledge more explicit.

Interestingly, our findings suggest that the AISS affects the alert investigation processes differently for `Att` and `NAtt` alerts. For investigations pertaining to `Att` alerts, we find that despite analysts often alternating between different contexts of the investigation, the overall investigation requires fewer actions. This suggests that when the AISS is used, analysts need less time to investigate and are more efficient without reducing the quality of the alert analysis. This is critical for `Att` alerts that must be analyzed and escalated as quickly as possible for effective mitigation and response actions. Therefore, aiding the T1 analyst's efficiency for these alerts is crucial, even if it comes at the cost of reduced inefficiency for other types of alert. For investigations pertaining to `NAtt` alerts, our findings suggest that analysts using the AISS alternate less between collecting similar information throughout the alert investigation than analysts without the AISS. Therefore, the AISS helps streamline the process and simplifies the most complex `NAtt` alert investigations. The other side of the coin is that the least complex alerts may become more procedural and lengthier to investigate as analysts look for additional, yet not meaningful, evidence after acquiring the necessary information to dismiss an alert. This suggests that the AISS can help analysts investigate the most complex or previously unseen alerts, whereas trivial ones should be analyzed automatically.

A. Limitations

Subject generalizability. Despite collaborating with a commercial SOC, we recognize that different SOCs operate differently and employ different technologies to generate alerts and conduct alert investigations. Therefore, data collected from analysts from one SOC may not represent another SOC, especially in SOCs that have more experienced T1 analysts compared to students following an internship. Moreover, within the same SOC we recognize that there is some difference in subject's skill sets despite identical work-related contexts. However, these limitations may be mitigated by T1 analysts being, by definition, juniors and often recruited from graduate-level cybersecurity courses, meaning that difference in background may be limited. Additionally, within the same SOC, analysts oftentimes do receive identical trainings and collaborate with other junior analysts.

Multiple observations. This limitation arises from a conscious decision to keep the experimental setup as realistic as possible. T1 analysts at the collaborating SOC are instructed to escalate single alerts, not groups of alerts belonging to the same investigation. Experimentally, this means that analysts in ES1 investigate the same attack (but not the same alert) multiple times. However, only on one occasion did an ES1 analyst explicitly mention noticing a specific alert belonging to an attack they already investigated. This suggests that the effects on the verbalized thought process of the subjects are minimal.

VII. Conclusion

In this work we conducted two experiments with five and four T1 analysts respectively to evaluate the natural analysis process for conducting alert investigations in SOCs and to evaluate how and whether a proposed alert investigation support system (AISS) can improve upon the natural alert investigation process. Our results from the first evaluation study show that the types of information analysts acquire and the actions conducted is different across different alerts. More specifically, we observed that the alerts that ought to be escalated acquire a larger variety of information and actions by the analyst, suggesting a further need to support analysts with such investigations. Based on these identified inefficiencies, we proposed an AISS to attempt to close the gap between the different investigation processes and conducted another evaluation study. Our study shows that an AISS can aid in reducing the complexities for all alerts except the most trivial to analyze.

Acknowledgment

This work is supported by the SeReNity project, Grant No. cs.010, funded by Netherlands Organization for Scientific Research (NWO), by the INTERSECT project, Grant No. NWA.1162.18.301, funded by NWO and by the CATRIN project, Grant No. NWA.1215.18.003. The authors also thank the ESH-SOC for its collaboration in this work.

References

[1] T. van Ede, H. Aghakhani, N. Spahn, R. Bortolameotti, M. Cova, A. Continella, M. v. Steen, A. Peter, C. Kruegel, and G. Vigna, "Deepcase: Semi-supervised contextual analysis of security events," in *2022 IEEE Symposium on Security and Privacy (SP)*, 2022, pp. 522–539.

[2] W. Hassan, S. Guo, D. Li, Z. Chen, K. Jee, Z. Li, and A. Bates, "Nodoze: Combatting threat alert fatigue with automated provenance triage," in *NDSS Symposium*, 01 2019.

[3] M. Ohmori, "On automation and orchestration of an initial computer security incident response by introducing centralized incident tracking system," *Journal of Information Processing*, vol. 27, pp. 564–573, 2019.

[4] C. Zhong, J. Yen, P. Liu, and R. F. Erbacher, "Automate cybersecurity data triage by leveraging human analysts' cognitive process," in *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, 2016, pp. 357–363.

[5] S. C. Sundaramurthy, J. McHugh, X. Ou, M. Wesch, A. G. Bardas, and S. R. Rajagopalan, "Turning contradictions into innovations or: How we learned to stop whining and improve security operations," in *Twelfth Symposium on Usable Privacy and Security (SOUPS 2016)*. Denver, CO: USENIX Association, Jun. 2016, pp. 237–251.

[6] 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.

[7] C. Zhong, J. Yen, P. Liu, R. Erbacher, R. Etoty, and C. Garneau, "An integrated computer-aided cognitive task analysis method for tracing cyber-attack analysis processes," in *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security*, ser. HotSoS '15. New York, NY, USA: Association for Computing Machinery, 2015. [Online]. Available: https://doi.org/10.1145/2746194.2746203

[8] 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)*. Anaheim, CA: USENIX Association, Aug. 2023, pp. 97–111.

[9] R. Gutzwiller, S. Fugate, B. Sawyer, and P. Hancock, "The human factors of cyber network defense," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 59, pp. 322–326, 09 2015.

[10] E. T. Greenlee, G. J. Funke, J. S. Warm, B. D. Sawyer, V. S. Finomore, V. F. Mancuso, M. E. Funke, and G. Matthews, "Stress and workload profiles of network analysis: Not all tasks are created equal," in *Advances in Human Factors in Cybersecurity*, D. Nicholson, Ed. Cham: Springer International Publishing, 2016, pp. 153–166.

[11] A. D'Amico, K. Whitley, D. Tesone, B. O'Brien, and E. Roth, "Achieving cyber defense situational awareness: A cognitive task analysis of information assurance analysts," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, 09 2005, pp. 229–233.

[12] A. D'Amico and K. Whitley, *The Real Work of Computer Network Defense Analysts*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 19–37. [Online]. Available: https://doi.org/10.1007/978-3-540-78243-8_2

[13] F. B. Kokulu, A. Soneji, T. Bao, Y. Shoshitaishvili, Z. Zhao, A. Doupé, and G.-J. Ahn, "Matched and mismatched socs: A qualitative study on security operations center issues," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1955–1970. [Online]. Available: https://doi.org/10.1145/3319535.3354239

[14] B. A. Alahmadi, L. Axon, and I. Martinovic, "99% false positives: A qualitative study of SOC analysts' perspectives on security alarms," in *31st USENIX Security Symposium (USENIX Security 22)*. Boston, MA: USENIX Association, Aug. 2022, pp. 2783–2800.

[15] S. C. Sundaramurthy, A. G. Bardas, J. Case, X. Ou, M. Wesch, J. McHugh, and S. R. Rajagopalan, "A human capital model for mitigating security analyst burnout," in *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*. Ottawa: USENIX Association, Jul. 2015, pp. 347–359.

[16] S. C. Sundaramurthy, J. Case, T. Truong, L. Zomlot, and M. Hoffmann, "A tale of three security operation centers," in *Proceedings of the ACM Conference on Computer and Communications Security*, 11 2014.

[17] R. Sadoddin and A. Ghorbani, "Alert correlation survey: Framework and techniques," ser. PST '06. New York, NY, USA: Association for Computing Machinery, 2006. [Online]. Available: https://doi.org/10.1145/1501434.1501479

[18] C. Zhong, J. Yen, P. Liu, R. F. Erbacher, C. Garneau, and B. Chen, *Studying Analysts' Data Triage Operations in Cyber Defense Situational Analysis*. Cham: Springer International Publishing, 2017, pp. 128–169. [Online]. Available: https://doi.org/10.1007/978-3-319-61152-5_6

[19] S. C. Sundaramurthy, J. McHugh, X. S. Ou, S. R. Rajagopalan, and M. Wesch, "An anthropological approach to studying csirts," *IEEE Security & Privacy*, vol. 12, no. 5, pp. 52–60, 2014.

[20] S. Y. Cho, J. Happa, and S. Creese, "Capturing tacit knowledge in security operation centers," *IEEE Access*, vol. 8, pp. 42 021–42 041, 2020.

[21] M. Vermeer, N. Kadenko, M. van Eeten, C. Gañán, and S. Parkin, "Alert alchemy: Soc workflows and decisions in the management of nids rules," ser. CCS '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 2770–2784. [Online]. Available: https://doi.org/10.1145/3576915.3616581

[22] C. Zhong, J. Yen, P. Liu, and R. Erbacher, "Learning from experts' experience: Toward automated cyber security data triage," *IEEE Systems Journal*, vol. PP, pp. 1–12, 05 2018.

[23] S. C. Sundaramurthy, M. Wesch, X. Ou, J. McHugh, S. Rajagopalan, and A. Bardas, "Humans are dynamic. our tools should be too. innovations from the anthropological study of security operations centers." *IEEE Internet Computing*, vol. PP, pp. 1–1, 06 2017.

[24] M. Rosso, M. Campobasso, G. Gankhuyag, and L. Allodi, "Saibersoc: Synthetic attack injection to benchmark and evaluate the performance of security operation centers," in *Annual Computer Security Applications Conference*, ser. ACSAC '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 141–153. [Online]. Available: https://doi.org/10.1145/3427228.3427233

[25] S. O. Solutions, "Security Onion 2," [Online; last accessed Feb, 2024]. [Online]. Available: https://securityonionsolutions.com/software

[26] @malware_traffic, "My Technical Blog Posts," https://www.malware-traffic-analysis.net/, [Online; accessed October 6, 2024].

[27] M. Haak and M. De Jong, "Exploring two methods of usability testing: Concurrent versus retrospective think-aloud protocols," in *IEEE International Professional Communication Conference*, 10 2003, p. 3 pp.

[28] E. L. Olmsted-Hawala, E. D. Murphy, S. Hawala, and K. T. Ashenfelter, "Think-aloud protocols: Analyzing three different think-aloud protocols with counts of verbalized frustrations in a usability study of an information-rich web site," in *2010 IEEE International Professional Comunication Conference*, 2010, pp. 60–66.

[29] K. A. Ericsson and H. A. Simon, *Protocol Analysis: Verbal Reports as Data*. The MIT Press, 04 1993.

[30] J. C. Chang, N. Hahn, Y. Kim, J. Coupland, B. Breneisen, H. S. Kim, J. Hwong, and A. Kittur, "When the tab comes due:challenges in the cost structure of browser tab usage," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, ser. CHI '21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: https://doi.org/10.1145/3411764.3445585

[31] L. Kersten, S. Darré, T. Mulders, E. Zambon, M. Caselli, C. Snijders, and L. Allodi, "A security alert investigation tool supporting tier 1 analysts in contextualizing and understanding network security events," in *Annual Computer Security Applications Conference (ACSAC)*. United States: Institute of Electrical and Electronics Engineers, Dec. 2024, in the ACM proceedings of .... (ACSAC 25, to appear).

Appendix A
Injected attacks

Injected attacks.

1) Remcos RAT
2) RIG Exploit Kit and Dridex
3) Emotet and Trickbot
4) Qakbot and Cobalt Strike
5) Qakbot and Spambot
6) Hancitor and Cobalt Strike
7) Ghost RAT
8) BazaarLoader and Cobalt Strike

9) MalSpam Brazil
10) Ursnif

To generate `Akk` alerts, we collected PCAP network traffic related to 10 attacks from `malware-traffic-analysis.net` [26], a website containing multi-stage malware attacks. We injected the PCAPs to the NIDS sensors generating the security alerts at the SOC. Further details of the injected attacks are reported in Appendix A. To preserve realism in the experiment, IP ranges of the victims were modified to unassigned IP addressees within the IP range of the monitored networks of the SOC. We utilized the `SAIBERSOC` tool [24] to inject the PCAPs into the SOC's pre-production environment As `NAtt` alerts are plentiful in the SOC we collected them directly from the SOC environment. We collect over 300k security alerts across a period of 17 days. To maintain the experiment setup manageable, we sample 178 `NAtt` alerts from this set. As there is a very strong class imbalance in alert data, we employ a stratified sampling strategy using the "rule category" [26] attribute defined in the SOC. The considered rule categories for `NAtt` alerts are: `Scan`, `Policy` (i.e., potential policy violations) and `Malware` (not succ.). From each of the rule categories, we randomly selected a unique rule and sampled a random alert generated by that rule.

APPENDIX B
APPLICATION OF THE FINAL CODING STAGE

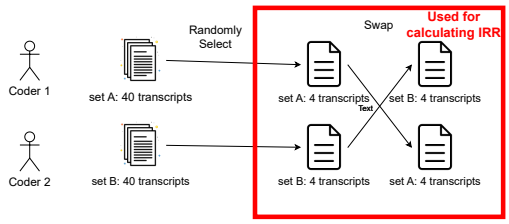

Fig. 6. Overview for one round in the application stage of the final coding scheme.

APPENDIX C
NUMBER OF ACTIONS FOR EACH ALERT TYPE

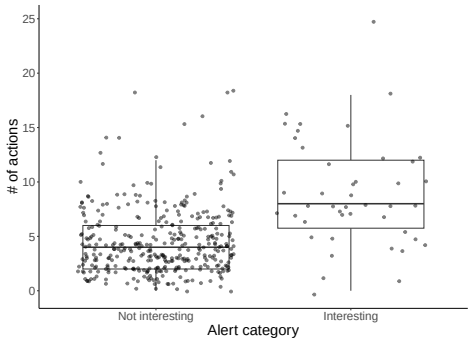

Fig. 7. The distribution of number of actions performed in an investigation aggregated across all analysts per alert classification.

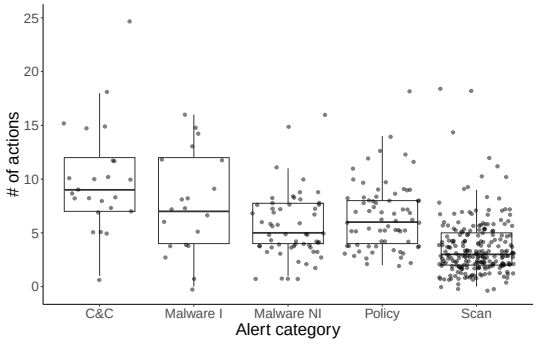

Fig. 8. The distribution of number of actions performed in an investigation aggregated across all analysts per rule category.

TABLE VI

COUNT FOR THE EXISTENCE OF EACH CODE IN INPUT FOR ALERT INVESTIGATIONS PER RULE CATEGORY. C&C AND MALWARE (SUCC.) ALERTS ARE ATT ALERTS WHILE OTHER ALERT SUBCATEGORIES ARE PART OF NATT ALERTS.

	RI				AA				CI				AE			
Category (n)	**S**	**P**	**E**	Total	**S**	**P**	**E**	Total	**S**	**P**	**E**	Total	**S**	**P**	**E**	Total
C&C (24)	21	1	3	**21**	10	5	0	**12**	24	1	4	**24**	15	8	23	**23**
(%)	(88)	(4)	(13)	**(88)**	(42)	(21)	(0)	**(50)**	(100)	(4)	(17)	**(100)**	(63)	(33)	(96)	**(96)**
Malware (succ.) (20)	14	1	2	**14**	6	5	0	**10**	19	2	3	**19**	14	1	12	**16**
(%)	(70)	(5)	(10)	**(70)**	(30)	(25)	(0)	**(50)**	(95)	(10)	(15)	**(95)**	(70)	(5)	(60)	**(80)**
Malware (not succ.) (58)	36	3	6	**39**	19	12	0	**28**	50	4	2	**50**	36	6	34	**44**
(%)	(62)	(5)	(10)	**(67)**	(33)	(21)	(0)	**(48)**	(86)	(7)	(3)	**(86)**	(62)	(10)	(59)	**(76)**
Policy (70)	44	4	9	**46**	22	8	0	**28**	67	18	6	**67**	45	8	53	**59**
(%)	(63)	(6)	(13)	**(66)**	(31)	(11)	(0)	**(40)**	(96)	(26)	(9)	**(96)**	(64)	(11)	(76)	**(84)**
Scan (228)	88	4	10	**90**	62	57	0	**114**	227	24	10	**227**	75	12	38	**91**
(%)	(39)	(2)	(4)	**(39)**	(27)	(25)	(0)	**(50)**	(100)	(11)	(4)	**(100)**	(33)	(5)	(17)	**(40)**
Total (400)	**203**	**13**	**30**	**210**	**139**	**87**	**0**	**192**	**387**	**49**	**25**	**387**	**185**	**35**	**160**	**233**
(%)	**(51)**	**(3)**	**(8)**	**(53)**	**(35)**	**(22)**	**(0)**	**(48)**	**(97)**	**(12)**	**(6)**	**(97)**	**(44)**	**(9)**	**(40)**	**(58)**

TABLE VII

THE FINAL CODING SCHEME.

Information Category	Code: input	Code: process
RI	SIEM External tools Previous knowledge	Checking the signature Checking signature age Checking the relevance of IP addresses Checking threat relevancy
AA	SIEM External tools Previous knowledge	Filter on involved IPs and alerts Filter on the alert name
CI	SIEM External tools Previous knowledge	Checking for logs Inspecting protocol specific logs Investigating response codes Investigating whether SSL established Investigating whether DNS was successfully resolved Investigating whether SSH successfully authenticated Investigating target host information via zeek logs Inspecting conn logs Investigating the number of packets Investigating number of bytes Investigate ports used Investigating the host name Investigating/processing the purpose of the host Check for new behavior from host Check for normal behavior from host
AE	SIEM External tools Previous knowledge	Investigating the hash of a file Investigate whether the domain is malicious or not Investigating how the attack works (e.g., google) Checking the CVE Investigate whether the IP address is flagged as malicious Investigate related threats from IP
NA	SIEM External tools Previous knowledge	Checking network decoded data Checking the security severity level