A Security Alert Investigation Tool Supporting Tier 1 Analysts in Contextualizing and Understanding Network Security Events

Leon Kersten* *l.kersten.1@tue.nl*
Santiago Darré* *s.darre@student.tue.nl*
Tom Mulders* *t.r.j.mulders@tue.nl*
Emmanuele Zambon* *e.zambon@tue.nl*
Marco Caselli† *marco.caselli@siemens.com*
Chris Snijders* *c.c.p.snijders@tue.nl*
Luca Allodi* *l.allodi@tue.nl*

* *Eindhoven University of Technology* † *Siemens*

***Abstract*—The investigations run by tier 1 (T1) analysts in a Security Operation Center are critical to the SOC operations as they represent the first gateway to alert escalation and incident response. Critically, they demand an accurate and as-complete-as-possible understanding of the events surrounding the investigated alert. This is a complex task inexperienced T1 analysts can easily lose track of. In this work, we collaborate with a commercial SOC to develop an alert investigation support tool to help inexperienced analysts identify and collect all the information relevant to the investigation of an alert. We evaluate the prototype tool with two qualitative studies. The first study employs T1 analysts from the SOC to evaluate the conformity of the tool to the underpinning analysis process. The second study employs 57 students, recruited from the same pool where the SOC acquires its junior analysts from, to evaluate whether it helps inexperienced analysts develop a complete understanding of events surrounding security alert data. Our findings suggest that employing the tool helps inexperienced analysts form a more accurate understanding of attacks, at no time cost. We discuss the wider implications for research and practice.**

***Index Terms*—alert analysis, incident investigation, security operation center, security analysts.**

1. Introduction

Inexperienced tier 1 security analysts are tasked with running complex alert investigations [1] playing a central role in the identification and escalation of potential security incidents in a Security Operation Center (SOC) [2]. These investigations often require complex sense-making of security event data surrounding an investigated alert, whereby the analyst has to identify and evaluate evidence of the events leading to and following the alert under investigation [2], [3]. On the other hand, alert analysis best practices are not fully defined and oftentimes *tacit knowledge* remains a key factor in security decisions made by analysts at all levels [4], [5]. This can be detrimental as it increases communication burdens within the SOC and decreases classification transparency as the evidence the analyst missed cannot be documented [3]. In turn, this lowers confidence on the *quality* of the investigation itself and therefore on the confidence a SOC manager can have on the provided services. Much research has been dedicated to the development of automation tools that can either partially or fully take over the decisions of T1 analysts [6], [7] and dedicated Security Information and Event Management (SIEM) tools are devoted to automate or facilitate the investigation process. Whereas some of these tools can decrease the workload for junior analysts by automatically categorizing previously-seen events or event sequences [6], they do not provide any support for the analysis of the remaining events. Some work developed tools to *transfer knowledge* from senior to junior security analysts [8], or developed cyber-range exercises for specific scenarios related to security forensics [9]. On the other hand, the extent to which these help forming effective and autonomous T1 analysts is not yet clear. For security analysts operating in a SOC, previous research showed that structuring the analysis process can greatly benefit overall quality and accuracy of analysis [3]. An important aspect of an analysis process such as the one described in [3] is that it guides the analyst in collecting and evaluating relevant evidence to contextualize and understand the security alert under investigation. However, this analysis process has only been deployed by requesting SOC analysts to follow the process. In the long term, it is unclear whether analysts will continue following this process, and it may require continuous supervision by senior analysts for junior analysts to follow it.

In this paper, therefore, we build a security alert investigation support system that integrates the analysis process defined [3] in a popular, open source SIEM (Security Onion Console [1]). To do so, we collaborate with a commercial SOC (the Eindhoven Security Hub SOC [2], to evaluate the requirements the tool must satisfy to effectively integrate the SOC processes, and run two studies to evaluate whether the developed tool (1) adheres to said requirements; and (2) aids inexperienced student analysts in developing a better

1. https://securityonionsolutions.com/software
2. https://www.eindhovensecurityhub.nl

understanding of attacks from real security event data. Our findings suggest that the tool helps inexperienced analysts develop a more accurate understanding of investigated attacks, without sacrificing timeliness.

2. Background and related work

2.1. SOCs and Tier 1 analysts

Security operation centers (SOC) monitor the security of networks and infrastructures. To detect incoming attacks, SOCs use solutions ranging from static detection mechanisms, dynamic systems, machine learning and more [3]. Regardless of the employed detection mechanism, SOCs employ SIEMs to aggregate generated security events and aid analysts in analyzing them [2]. Organizationally, most SOCs structure their analysts in a hierarchical fashion with T1 analysts dealing with the mass of alerts, and further up tiers (usually up to T3 or T4) dealing with 'escalated' alerts that may represent severe cybersecurity risks [8], [10], [11].

It is therefore crucial that T1 analysts complete their analysis efficiently and accurately. Yet, security investigations by T1 analysts are relatively error-prone, despite the task's repetitive nature [12]–[15]. The quality of the investigation differs between T1 analysts depending on their background and skill sets [8], which is influenced by their experience and the training they receive [2]. Whereas experience and know-how are intrinsic properties of an analyst, previous research showed that the 'implicit' considerations analysts make should be made explicit to the benefit of analysis quality [3], [4], [16]. Yen et al. [17] and Zhong et al. [18] have focused on the cognitive processes followed by the SOC analyst and mapped those to the AOH ('actions', 'observations' and 'hypotheses') model, but what this information should be and how the analyst should evaluate it remains undefined. Similarly, D'Amico and Whitley [10] considered the actions that analysts should perform (as opposed to the information they should collect). Revisiting this approach, Kersten et al. [3] partially addresses this issue by synthesizing a threat analysis process (TAP) guided by information the analyst should evaluate. However, no practical way of operationalizing this process is provided and evaluated.

2.2. Underpinning alert analysis process

In our work, we developed a tool aiming to nudge analysts to follow a standardized TAP introduced in [3] to collect relevant information to evaluate during analysis. This TAP has shown promising results in improving the accuracy of the alert analysis by junior analysts where the odds of classifying an alert correctly has increased by 167% [3]. The TAP guides the T1 analyst through four different stages, each corresponding to a different type of information that is collected during the analysis. These stages in order are: **Relevance indicators**, **Additional alerts**, **Contextual information** and **Attacker evidence** [3]. Each stage consists of two to four 'steps' that define the detailed information an analyst should collect during that process stage. Steps are not ordered, as none of the information collected within a stage should strictly require other information from the same stage to be made sense of. Table 1 shows the definition and steps corresponding to each stage [3].

2.3. Problem statement and contribution

Research results reported in [3] show the effectiveness of the presented TAP in aiding the analysis of network security alerts. Yet, there is currently no actual support for the analyst to follow the process and keep track of the collected information. This is crucial for analysts to build a complete picture of the security (and non-security) events surrounding the investigated alert, and is key to effective internal alert communication and alert escalation, as well to, incident notification and response. [1], [2] In this paper we develop a prototype tool to support alert investigation aimed at early stage analysts (hereafter on abbreviated as ''AISS", for 'alert investigation support system') by implementing the process in [3] on top of the technical tools used in SOCs. The proposed tool implements the process presented in [3] and is built in collaboration with a professional SOC (hereafter referred to as *'the SOC'*). In addition to the development, we qualitatively evaluate the tool with expert analysts at the SOC, and with inexperienced analysts to evaluate whether their overall understanding of an investigated alert benefits from the AISS.

3. Methodology

3.1. Requirements collection

To determine the requirements of the tool and ensure that these are met during development, we identified the following set of stakeholders: *junior analysts*, *senior analysts*, *SOC manager*, *external UI expert*. To collect requirements we held a series of discussions involving senior staff at the SOC and, separately, the UI expert. We held two meetings, separated by one month, to include feedback from previous meetings and discuss its implementation. Following these meetings, four key requirements were identified for the development of the AISS, summarized in Table 4 in the Appendix. As the devised AISS implements guidelines in [3] to aid analysts in their decision making, it should implement each stage of the model in sequence (`R-1 Adherence to model`). When performing their investigations, T1 analysts must gather information from various different sources to form an evidence-based narrative of the security incident (`R-2 Data collection`). During this research, the analyst may be presented with a deluge of alerts, logs, and other accompanying information. The AISS must therefore allow for the analyst to consolidate their findings in an accessible interface. This interface should be integrated into an environment that the analyst is familiar with and that they will use during their day-to-day activities, to allow ease of use (`R-3`

TABLE 1. STAGES AND STEPS OF THE THREAT ANALYSIS PROCESS DEFINED IN [3]

Stages	Definition	Steps
Relevance indicators	Information to determine whether the signature of the investigated alert is relevant and the targeted host is in the scope of the customer.	signature specificity, signature age, customer scope
Additional alerts	Previous instances of the same alert being triggered and alerts triggered by the same hosts within a relevant timeframe.	alert history, surrounding alerts
Contextual information	Non-alert information that adds context to the investigation, such as network logs and the behavior of the targeted host.	related logs, traffic stream information, target host information, target host behavior
Attack evidence	Evidence about the potential attack, the maliciousness of the attacker and whether the alleged attack has succeeded.	attack/exploit information, attacker information, attack success indicators and relations to use cases

`Seamless integration`). Finally, the AISS must allow for future expansion (`R-4 Extensibility`). In addition to these requirements, we received design-specific suggestions from the UI expert such as the checkbox buttons and the traffic light system described in Section 4.2.

To ensure that these requirements are met, we further held biweekly meetings throughout the design and development processes with all stakeholders and the external UI expert. At these meetings, we would present demos of current iterations of the AISS, after which we would receive feedback which would be evaluated and implemented for the next iteration. We also held separate meetings with the T2 analyst at the SOC in order to ascertain which desired properties or constraints the use of the AISS should consider. Usability related features were considered in meetings with the UI expert who was asked to comment and provide feedback on the first mock-up of the AISS and first implementations to discuss optimal ways to integrate it into the SIEM interface. Finally, we discussed features that are highly contextual to the SOC with the manager of the SOC who provided insights into how different implementations may impact TP and FN rates.

3.2. Empirical evaluation

We ran two evaluation studies to (Evaluation Study 1) collect feedback from expert analysts on the employment of the developed AISS and its fit to the defined requirements, and (Evaluation Study 2) evaluated whether the employment of the AISS can aid inexperienced student analysts in understanding cyber-attacks from data captured in a SOC. The first study employed a think-aloud protocol and asked four expert analysts currently employed at the SOC to verbalize their thoughts on the AISS employment while performing analysis on ten real-world alerts captured by the SOC. The second study recruited 57 students from an advanced cybersecurity course at a mid-size European technical university, divided them into a control (no AISS) and a control (AISS) group and qualitatively evaluated how well students in the two groups understood three security alerts selected from the same pool of ES1. The methodologies for ES1 and ES2 are presented in full detail in Section 5 and Section 6 respectively. Whereas these are two separate studies with different goals, designs, and subjects, they share implementation details on the selected environment and the security alerts subjects were asked to analyze. The remainder of this section details those.

Environment. To maximize the realism of the experiments, we replicated the SIEM environment employed at the collaborating SOC, Security Onion Console. The environment used Suricata and Zeek sensors for the network event analysis and deployed a combination of open source (*Emerging Threat Open ruleset*) and a professional licensed rule set (*Emerging Threat Open PRO ruleset*) to generate alerts. The experimental environment contained alerts and logs collected over a time span of 2.5 weeks from one of the customers of the SOC who possessed more than 1500 unique hosts and multiple DNS and file servers. Additionally, as false positives and alerts not worthy of escalation over-represent real alert data in the SOC, we injected 10 attacks in the environment to represent attack-related alerts. The attacks were injected into the environment using `SAIBERSOC` [19] and all private data such as customer IPs or its location were modified before injection.

Alert selection. The selected alerts are reported in Table 2. To make sure subject analysts did not previously perform an analysis on the selected alerts, alerts were chosen from an environment they do not monitor. We selected 20 alerts from a pool of 50 alerts within the environment which was analyzed by a T2 analyst with over 4 years of experience to generate the ground truth on its classification. We followed the collaborating SOC's own taxonomy and created a dichotomy between '*not interesting*' and '*interesting*' alerts. '*Not interesting*' alerts are alerts that are considered to not be worthy of escalation. More specifically, these are alerts about non-malicious or unsuccessful attack attempts, false positives (benign traffic that happens to match an attack pattern). By contrast, '*interesting*' alerts are related to attacks in which there is evidence that at least one of the attack phases was successful, such as a successful exploitation of network services or a malicious behavior originating from internal hosts. To make this distinction, analysts in the SOC rely on alert data and knowledge within the SOC about the monitored environment, such as the configuration or criticality of the targeted system(s). From the selected alerts, 10 alerts are considered '*not interesting*' alerts while

TABLE 2. ALERTS SELECTED FOR EVALUATION STUDY 1 (ES1) AND EVALUATION STUDY 2 (ES2).

		ES 1				ES 2
Type	Alert name	S1	S2	S3	S4	Sel.
Not intr.	(A10) ET CHAT Skype User-Agent detected	✓	✓			
	(A7) ET DOS Possible SSDP Amplification Scan in Progress	✓	✓			
	(A8) ET SCAN ProxyReconBot CONNECT method to Mail	✓	✓			
	(A3) ET SCAN Suspicious inbound to Oracle SQL port 1521	✓	✓			✓
	(A4) ETPRO SCAN VMware vCenter Chargeback Manager Information Disclosure	✓	✓			
	(B9) ET SCAN Potential VNC Scan 5800-5820			✓	✓	
	(B10) ETPRO HUNTING Generic Inbound URI Directory Traversal			✓	✓	✓
	(B3) ET SCAN Suspicious inbound to Oracle SQL port 1521			✓	✓	
	(B5) ET WEB_SERVER WEB-PHP phpinfo access			✓	✓	
	(B7) ET SCAN Behavioral Unusual Port 445 traffic Potential Scan or Infection			✓	✓	
Intr.	(A1) ET MALWARE EXE Download Request To Wordpress Folder Likely Malicious	✓	✓			
	(A2) ET MALWARE Backdoor family PCRat/Gh0st CnC traffic (OUTBOUND) 103	✓	✓			✓
	(A5) ET JA3 Hash - [Abuse.ch] Possible Dride✓	✓	✓			
	(A6) ET MALWARE JS/TrojanDownloader.Agent.TXV CnC Activity	✓	✓			
	(A9) ET MALWARE Ursnif Variant CnC Beacon - URI Struct M2 (_2F)	✓	✓			
	(B1) ET MALWARE Likely Evil EXE download from MSXMLHTTP non-exe extension			✓	✓	
	(B2) ET JA3 Hash - [Abuse.ch] Possible Quakbot			✓	✓	
	(B4) ETPRO EXPLOIT_KIT RIG EK Landing Apr 04 2017 M4			✓	✓	
	(B6) ET MALWARE Cobalt Strike Beacon Observed			✓	✓	
	(B8) ET MALWARE Trickbot Checkin Response			✓	✓	

10 alerts are considered '*interesting*' alerts. The 10 'not interesting' alerts were randomly selected from a pool of non-interesting alerts at the SOC. From the 10 'not interesting' alerts, 7 alerts pertain to a potential scan, 2 alerts pertain to a possible data exfiltration attempt, and 1 alert warns the SOC about a potential policy violation. Following [3], 'interesting' alerts were generated by injecting 10 malware-based attacks injected into the monitored environment. The list of malware utilized per attack is reported in Appendix A. The 10 selected 'interesting' alerts pertain specifically about either the installation of malware or a malicious connection to a command and control server. ES1 utilized the entire set of 20 alerts (divided among 2 sets of 10 alerts) whereas Evaluation study 2 utilized 3 of the 20 alerts (Alerts A3, A2 and B10 as shown in Table 2), due to time constraints in the educational schedule of the student subjects. The rationale as to why specifically these 3 alerts have been chosen for ES2 is explained in 5.1.

3.3. Ethical considerations

This research was executed under the ethical approval from our institution's ethical review board, with approval number `ERB2022MCS20`. We obtained explicit and informed consent from all subjects. Subjects were assured that participation in the study would in no way affect their daily work conditions (ES1) or study program (ES2). For ES2, if a student did not consent to the experiment, the student would still participate in the in-class exercise, but would not be required to fill in the survey nor provide us with any data.

4. Alert Investigation Support System

4.1. Design principles

To design and implement the AISS, we started from the requirements we collected from the SOC, defined in Table 4. In the following, we explain how each requirement was translated in the design principles of the AISS.

Adherence to model (R-1) The AISS was constructed around the model outlined in Section 2.2. As the model details a sequential series of steps to perform, so too should the AISS be designed such that the user is guided through the process in a sequential manner. On the other hand, the model does not prescribe that a specific analysis order *must* be followed. This requires users to be able to navigate back and forth within the analysis stages in the AISS freely. Therefore, we structured the AISS around four 'analysis stage' tabs, each corresponding to one of the stages. The user is free to switch between tabs at any time during an investigation. The steps corresponding to each stage are listed under their respective tabs along with an accompanying editable text field. This structure provides a unified view on the execution of a specific stage and its steps while allowing the analyst to move freely across stages if necessary.

Data collection (R-2) Throughout their analysis process, analysts must continuously search for information that can be used to construct an evidence-based narrative to describe the security event and the circumstances surrounding it. This information can be acquired from various sources, both

from within and outside of the SIEM environment. It would therefore be prudent to implement some way for the analyst to consolidate any relevant information they come across, regardless of the 'current' model stage being investigated and the current view within the SIEM. To enable this, data from within the SIEM can be 'sent to' any data field within the AISS from any data field with a context menu within the SIEM. The notes and 'send to AISS' features allow the analyst to collect data from various different sources that may not be possible for the AISS to automatically acquire. Furthermore, each alert pinned to the AISS contains a notes field that can be edited at will and is synchronized across tabs, and each field corresponding to the model's steps can be manually edited. Finally, a 'semaphore' tracks the completeness (as reported by the analyst) of the data collection for each stage.

Seamless integration (R-3) The AISS was built into the Security Onion SIEM, the interface which analysts primarily engage with throughout their monitoring shifts. When possible, the AISS uses existing components of the SIEM, such as the escalate or acknowledge buttons, to ensure full integration with the SIEM. For example, the described contextual menus for the 'send-to-AISS' function are integrated in the same contextual menus normally used by analysts in the SIEM.

Extensibility (R-4) The AISS was developed in conjunction with other projects at the SOC, many of which also include modifying the Security Onion SIEM. Future projects are also planned which will build upon this work. The development of the AISS was therefore accompanied by extensive documentation, version control, and collaboration with adjacent projects. The AISS can further be combined with previously developed modifications of the Security Onion. For example, a custom 'claim alert' feature was developed for the SOC whereby analysts can claim an alert for investigation, signaling to other analysts that the alert in question is already under investigation and that they should divert their attention elsewhere. Furthermore, the AISS is programmed such that the back-end functionality conforms to the overall architecture of the Security Onion.

4.2. Interface design and implementation

The implementation of the AISS with its corresponding documentation can be found in: https://gitlab.tue.nl/aiss.

AISS workflow. When beginning their analysis process, the analyst is presented with one or more alerts within a SIEM. If presented with multiple alerts, the analyst may have to perform some preliminary triaging to determine which alert has a higher priority to engage with first. Once they have chosen an alert, the analyst may pin the alert to the AISS and begin their investigation. The analyst will then work through the various stages of the TAP, filling out relevant information for each stage as they go along. When the analyst is satisfied that they have completed a certain step, they may check a checkbox for that step to keep track of their progress. Once the analyst determines that they have completed their investigation, they may 'acknowledge' the alert (i.e. marking it as 'not interesting') or they may 'escalate' the alert to a T2 analyst.

AISS Design. The main interface of the AISS is depicted in Figure 1, filled in with data from a fictional event for illustration purposes. To aid testing, the AISS is implemented directly into the Security Onion Console employed as a SIEM by the collaborating SOC (R-3).[3] The AISS (1) appears at the top of whichever webpage wherein the analyst will encounter a list of alerts. The AISS is initially empty. At the start of the analysis the analyst can pin one or more alerts to the AISS (6), after which they can begin following the TAP through the AISS. Figure 1 shows two pinned alerts: an "ET SCAN" alert (completed) and an expanded "ET MALWARE" alert whose investigation is ongoing. Under each pinned alerts, the AISS interface is divided into four tabs (3), corresponding to each stage of the model (R-1). Under each tab, every step of the of the stage is listed with an accompanying text box (5). In order to help guide the analyst through the TAP, some steps have their text fields populated with automatically generated data or a query (see Figure 5 in the appendix for the latter). This is done when a query can be automatically generated (i.e., with no human interaction) given the data that exists within the SIEM.[4] To determine which steps of the TAP would be eligible for automatic filling, we held meetings with an experienced T2 SOC analyst where for each step in each stage it was determined whether the relevant data collection could be automatized or not. To decide whether automation is possible, we examined the queries necessary to retrieve the relevant information, and evaluated whether the query structure and variables (as opposed to the value of those variables) remain stable or have to change on a case-by-case basis. Outcomes from these discussions were also discussed at biweekly meetings at the SOC and with other two senior analysts as well as the pool of T1 analysts.

In addition to a field for each step in a stage (see Table 1), we added an extra field for taking notes for the analyst to keep track of any specific observation or question that may need to be answered downstream in the analysis process (R-2). To enable this, this field is synchronized across tabs. Any piece of data that has a context menu within the Security Onion can be sent to a specified text box in the AISS, allowing the analyst to record data found in their investigation (see Figure 2) (R-2). Furthermore, a checkbox next to each step allows analysts to mark when they feel confident that they have completed the step in question (R-1). For each pinned alert, there is a traffic light visually indicating the progress of the analysis (see (2) of Figure 1). The traffic lights consist of four circles, one for each model stage, that are initially colored amber but turn green when all checkboxes within the relevant stage are checked by

3. As the integration is done in the training environment of the collaborating SOC, the AISS is displayed as 'Training Support System' in that environment.

4. Automated query generation allows a trainee to focus on understanding the data, rather than focusing on learning the query language

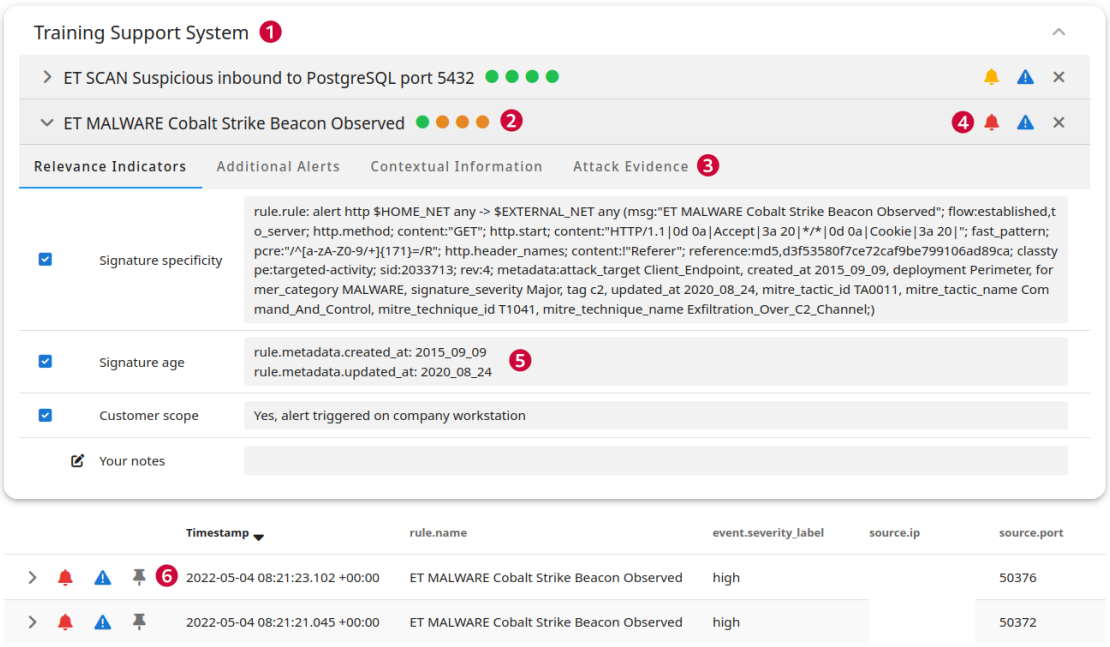


Figure 1. AISS interface as implemented in the Security Onion SIEM (*Security Onion Console*) with labels corresponding to design features detailed in Section 4.2.

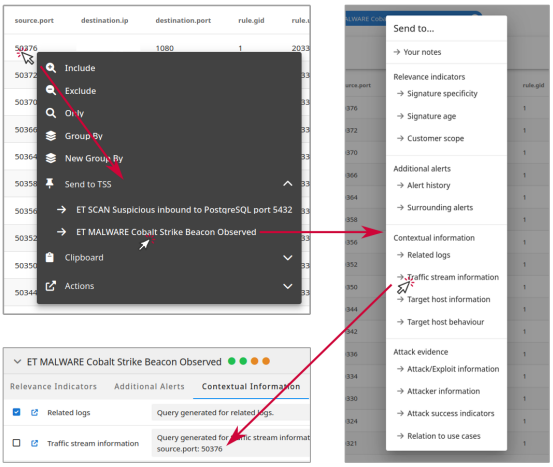


Figure 2. 'Send to AISS' feature

the analyst. The AISS also has buttons for escalating or acknowledging an alert (see (4) in Figure 1) when the analyst believed they have completed the analysis (R-3). For ease of access to the AISS, there is an anchor button that appears in the bottom right corner when the analyst scrolls past the AISS. Clicking the anchor button jumps the page up to the AISS, and clicking it again jumps back down to the previous position on the page. Aesthetically, the AISS conforms to the Security Onion's theme.

Implementation details. The developed AISS is a plugin for the Security Onion. The front-end is written in JavaScript (with HTML components written in the Vue.js framework) and the back-end is written in Go. Wherever possible, HTML components of the AISS are adapted from existing components in the Security Onion Console, such as the escalate and acknowledge buttons. The data for each alert pinned to the AISS is saved in the localStorage object of the analyst's web browser, which allows for investigations to be saved across browser sessions. The stored alert data is updated every time the analyst performs an action with the AISS. When an alert is unpinned, escalated, or acknowledged, the localStorage is emptied.

Automated data collection. The steps *signature specificity* and *signature age* are automatically populated by fetching data from the metadata of the rule generating that alert. The steps that have pre-generated queries are *alert history*, *surrounding alerts*, *related logs*, *traffic stream information*, and *target host behaviour*. These correspond to queries for past instances of the alert, alerts occurring around the same time as the relevant alert, logs relating to the source and target IPs of the alert, logs relating to the relevant alert, and logs relating to only the target IP of the alert, respectively. The query for *alert history* has a default timeframe of 14 days in total, seven days before and after the event. Queries for *surrounding alerts*, *related logs*, and *target host behaviour* have a timeframe of 60 minutes.

5. Evaluation Study 1

5.1. Methodology

Goal of study. The goal of ES1 is to collect qualitative insights on the operation of the AISS from T1 analysts

employing the underpinning TAP at the SOC.

Subjects. The subjects for ES1 are four T1 analysts recruited from the active pool of analysts operating at the SOC at the time of this study (Nov 2023). All four analysts were recruited in the same hiring batch of Sep. 2023 and have therefore received the same training (by a senior T2 analyst) on the TAP employed at the SOC (and implemented in the proposed AISS). Furthermore, all subject analysts have similar academic backgrounds and no experience with previous employment in SOCs. Subjects are chosen in coordination with the SOC senior management on an availability basis.

Study design. Each subject was tasked with analyzing ten alerts using the AISS as an instrument to implement the TAP. Subjects received a list of alerts to start their investigations from, and their task was to classify whether each alert was related to a security incident (and therefore should be classified as 'interesting') or not ('not interesting'). Subjects were explicitly asked to use the provided AISS to guide their analysis. Subjects were not informed if, and how many, alerts were related to known attacks. To capture the analysts thoughts on the employment of the AISS during an investigation, they were tasked to think-aloud during the investigation to verbalize their thoughts on the investigation and the usage of the AISS in particular. The subjects' speech was recorded for later analysis of their cognitive processes. Each subject's screen was also recorded. Subjects were permitted to use any external tools they may normally use during their alert analysis, such as *whois* lookups or external databases such as *VirusTotal*. Prior to the experiment, each subject was given a PDF file with a short description of the TAP.

Alert selection. This experiment employed all 20 selected alerts reported in Table 2. Each alert was independently analyzed by two subjects as reported in Table 2. More specifically, alerts labelled A_1, A_2 and so on were analyzed by subjects 1 and 2, while alerts labelled B_1, B_2 and so on were analyzed by subjects 3 and 4. The order in which the alerts were presented to the analysts was randomized.

Experiment run. The experiment was run with each subject on different days in the SOC office. Subjects were provided with a laptop with the browser opened to the Security Onion Console (the SIEM they use daily during SOC operations) and the auxiliary PDF material. We allotted four hours for the experiment. Actual experiment run times ranged from 2 to 3 hours. Prior to beginning their analyses, subjects were given a brief tutorial for the AISS and they were told that it was not necessary to use every feature present, so that they may use the AISS as they would in a real scenario.

Interviews. After having completed their analyses, each subject was posed a series of questions in a semi-structured interview to gain insight into their perceptions of the AISS. These interviews were recorded. Interview questions can be found in the Appendix.

Analysis. The collected data was used to reconstruct the train of thought of analysts during the AISS operation. Because of the limited number of participants, no coding was employed to analyze the data. The transcripts and video recordings were read and analyzed independently by two of the authors. The two authors then discussed the key points covered by analysts during either the think-aloud exercise and the interviews over the requirements defined in Table 4.

5.2. Results

Overall accuracy. The results of ES1 can be seen in Table 5 in the Appendix. First, we checked whether analyst investigations led to correct classification outcomes. As our subjects were analysts with practical experience, we expected a high accuracy above 80%, as indicated by previous literature [3]. Out of 40 alert classifications across subjects, 36 were correct. All subjects correctly classified 9 out of 10 of their alerts. 3 out of the 4 misclassifications were 'interesting' alerts. This was expected as 'interesting' alerts emerging from attacks are known to be more data-intensive than 'not interesting' alerts. It should be noted that no alert received an incorrect classification by more than 1 analyst.

Time taken for investigations. Table 5 additionally shows the time taken for analysis for each alert investigation. We observed a wide range in alert investigation time from 3 to 44 minutes, with an overall median of 10 minutes. Unsurprisingly 'not interesting' alerts were investigated faster than 'interesting' alerts with median time taken for analysis of 7 and 16.5 minutes, respectively. It is likely that SOC analysts analyze these incidents faster in the SOC than in our experiment as we employed a think-aloud protocol in this experiment. As verbalizing thoughts in real time increases the cognitive workload of the participants [20], experimental subjects may perform tasks more slowly than normally.

Role of the AISS on analyst mistakes. We evaluated whether the mistakes were introduced by the AISS operation or if they were due to other causes. We reviewed the verbalization of the analyst thoughts during the investigation of the misclassified alerts and the related video recording. Overall, we found that all errors but one were due to differences in the experiment setup from the operational setup the analysts are used to. For example, subject S2 misclassified alert A2 as 'not interesting' because "*the [command and control server] is not active anymore*" at the time of investigation. However, the command and control server may have been active when the alert was originally generated in the real-world environment it was taken from. Therefore, the misclassification of this alert may be a result of an imperfect experiment setup, and does not seem related to the operation or data contained in the AISS for this alert. Similarly, subject S3 misclassified alert B8 as 'not interesting' because the command and control server is: "*just checking for a response. It doesn't seem to perform an actual attack*". Whereas the information considered is relevant to the attack and therefore to the analysis, the subject dismisses it because it does not immediately relate it to the infection. We therefore conclude this is an error in judgment from the subject on otherwise relevant and correct information retrieved through the AISS. On the

same line, subject S4 misclassified alert B9 as 'interesting' because they (correctly) deemed the event to be "*malicious but not successful*" and classified it as 'interesting' in the context of this experiment.

The AISS played a role in a mistake due to a mismatch between the pre-defined timeframe for a query and the time of a relevant event. Subject S1 misclassified alert A6 as 'not interesting' because they interpreted a lack of relevant activity in the 30 minutes succeeding the alert, as evidence that the alert did not constitute an interesting event. As the automatically generated queries for *surrounding alerts*, *traffic stream information*, and *target host behaviour* all use 60 minute timeframes (30 minutes before and after the alert), this may have blinded the subject from relevant information to consider. We observed that all analysts have, in the context of other alerts, changed time parameters in the default queries to find relevant information. We therefore conclude that the AISS did facilitate the mistake but did not constitute a barrier *per se* to the identification of the correct information.

Feedback on design requirements. Subjects expressed an appreciation for being guided through the TAP by the AISS (*R-1. Adherence to model*): "*I could just refer back to [the AISS], like, let's just see what steps there are and get myself back on track*" (S3). Subjects also appreciated having their investigation scopes broadened: "*For example, the size of the traffic stream. Normally I quickly overlook that. Reminds you to keep that in mind, which I find useful*" (S1).

Further, subjects found the AISS implementation for *R-2. Data collection* useful as it helped them track information through the AISS: "*Normally I take some notes to my notepad for the handover, so the feature to add notes is nice for reference*" (S4). Subjects also appreciated the ability of the AISS to save investigations across sessions: "*It's nice that it saves automatically*" (S3). These features reduce the frequency at which the analyst has to divert their attention from the SIEM, maintaining focus. They also facilitate the "handover" that occurs between monitoring shifts, where analysts have to write documentation to inform the next analyst of any relevant occurrences during the previous shift.

Finally, subjects found the integration of the AISS into the SOC SIEM effective (*R-3. Seamless integration and R-4. Extensibility*) because of the overall experience (*"it fits with the rest of Security Onion"* (S1)) and the ease of access and use (*"It's very intuitive"* (S2)).

Points of improvement. While the AISS operated smoothly and the interface aligned well with the Security Onion, subjects suggested integrating the supporting document directly into the AISS to further clarify each step of the TAP: "*If you could combine this document with the tool, so I don't have to look at what exactly [the TAP step] means every time, that would be an improvement*" (S1). This reflected the fact that analysts employ the underlying model intuitively, rather than formally, during their investigations. However, this comment remains useful to further support the employment of the TAP during operations, including training. Subjects also highlighted the need for clearer instructions on field inputs and more detailed explanations about what information should be entered: "*Maybe you could make it a little more clear what should be filled into the fields*" (S2). Some subjects described a learning curve in the use of the AISS, stating that alert analysis began slowly, and that it became easier with each successive alert.

5.3. Discussion of results from ES1

Overall, subjects expressed appreciation for the structured guidance offered by the AISS and its capacity to augment their investigative capabilities. They acknowledged the value of features such as automatic data acquisition and query generation, which streamlined the analysis process. However, there was also a shared sentiment among analysts that certain improvements were necessary to optimize the tool's functionality fully. Suggestions included: enhancements to clarify each step of the TAP, further automatic data and query generation, refining of the existing queries, and guidance on what kind of information belongs in each field. Based on the analyst performance and feedback in ES1, and the fact that errors made were almost entirely due to the non-operational setting of the experiment that subject analysts are not used to, we determined that the AISS did not require major revisions for ES2. The issue whereby the AISS included too short of a timeframe for the analysis of a specific alert was not addressed, as after discussion with the T2 analysts it remained within the 'reasonable' timeframe to set as a default for expert analysts, and because the alerts selected for ES2 (see Table 2) did not require any change in the considered timeframe.

6. Evaluation Study 2

6.1. Methodology

Goal of study. The goal of ES2 is to qualitatively evaluate whether the AISS can help inexperienced analysts in forming a correct understanding of events surrounding an alert.

Subjects. The subjects for ES2 were recruited from the same cohort of prospective T1 analysts from which the SOC selected part of its T1 analysts to recruit and train: a cybersecurity master course at a medium-sized European technical university. This course was a mandatory part of the cybersecurity specialization at the university. For ES2 we recruited the totality of students in that class (n=57). The study was integrated as a complementary in-class exercise. As the subjects followed the same MSc cybersecurity education program and were recruited halfway through the first year of that program, we considered the educational background of the students to be comparable. We discussed limitations in Section 7.

Study design. All subjects were tasked with the analysis of three alerts: one 'interesting' alert, i.e., an alert that should

be escalated to higher analyst tiers (according to the collaborating SOC's definition), and two 'not interesting' alerts, i.e., alerts that the T1 analyst should dismiss as they do not indicate credible risks. Subjects were randomly assigned to an AISS or a non-AISS group. All analysts received the same set of alerts to investigate. For each investigation, subjects received a survey to fill in with questions aimed at probing their understanding of the investigated alert and surrounding relevant events.

Alert selection. Considering the time-constraints of an in-class experiment, the subjects analyzed three (as opposed to ten) alerts selected in ES1 (see Table 2). We requested the subjects to analyze alerts A3, A2 and B10 in that order. A3 and B10 are considered 'not interesting' and A2 is considered as 'interesting' by the SOC. To verify that these three alerts are viable to be analyzed by inexperienced analysts, all three alerts were analyzed by a T2 analyst with over 4 years of experience. Additionally, the chosen alerts were presented to the analysts in ascending order of 'complexity', according to the T2 analyst's own assessment. This is to provide analysts with a gradual introduction to security alert analysis during the exercise, as opposed to starting with a complex investigation for their first analysis and risking remaining stuck. The first alert was considered trivial by the SOC as it was a failed scan. When an analyst reaches the contextual information stage of the TAP, the analyst should observe that there is only one connection log with information indicating that no connection has been established between the scanner and the host. Therefore, the analyst can conclude that the attack was not successful. The second alert is more complex as the analyst first needs to find another alert that triggered on the same signature within a short timespan to conclude that the original alert was not a 'lucky hit' (i.e., a false positive alert that triggered due to a large network stream with coincidentally matching bytes as the signature). An analyst must determine that the external IP successfully connected to the internal IP, and that the external IP is malicious. Finally, the third alert is deemed the most complex, as despite having many successful connections between the external and internal IPs, no malicious file is exchanged, thus potentially misleading the analyst.

Experiment preparation. Before the experiment, we randomly assigned each subject to either a control (no AISS) or a treatment (AISS) group [5]. Both groups analyzed an identical set of alerts on the same environment containing the same network logs. Both groups used the same SOC SIEM tool, the only difference being that the SIEM of the treatment group also displays the AISS add-on. A week before the experiment all subjects received a preparatory handout and video which they were instructed to read and watch. The handout contained instructions and visualizations of the employed SIEM. The treatment group received a slightly longer manual where the interface of the AISS was detailed as well. The video contained a demonstration where one of the authors analyzed an alert following the TAP. The video for the AISS group is equivalent to that of the control group and only differs in the usage of the AISS in the analysis.

Experiment run. The experiment was run in two parts. The first session consisted of a baseline training incorporated into a 1.5h long instruction about security alert analysis and the TAP. This instruction was identical for both the control and treatment groups and was given in one sitting to reduce factors beyond the use of AISS influencing the results. The instruction consisted of an hour long training, and half an hour was reserved to demoing an analysis. The instruction was given by an experienced T2 analyst and is equivalent in content and length (demo included) to a training given to T1 analysts in the SOC. The training provides detailed information about each stage and step in the TAP such that all subjects understand the process which they are instructed to follow.

The second part of the experiment was meant to provide the subjects with a hands on experience of analyzing alerts, and was conducted as soon as possible after the first session (two days after) within the limitations imposed by the educational schedule of the university. This was to maximize the retention of information which the students gain in the training instruction. Before the start of the experiment, subjects were asked to analyze one alert as a warm-up exercise to practice with the interface and the analysis process. The warm-up exercise and its solution when following the TAP can be found in the Appendix. Following the warm-up exercise, the experiment took place. Subjects were granted 1 hour and 15 minutes to analyze the aforementioned three alerts. The subjects were instructed to analyze the alerts in the order provided, and to fill in a survey on their investigation immediately after every alert was analyzed (i.e., the subjects have filled in a total of three surveys). Students did the exercise from their own laptops and accessed the analysis interface they were assigned to via the instruction environment employed at the university.

Survey. We created a survey for every alert to collect the classifications of the alerts and the respective rationale. Each survey first asked the subject's classification of the alert and followed with a set of questions reporting information (either true or false) relevant to the alert classification. The section where the subjects fill in their rationales for a given classification was structured over the four stages of the employed TAP (see Table 1). The survey design and the questions can be found in Appendix A. For the data analysis, we only considered surveys that were filled in in the correct order, to prevent the execution of subsequent analyses to confound previous alert classifications and rationales. Timing data is retained to approximate time of analysis of each alert. In total 3 subjects did not complete the survey in the correct order, resulting the sample size for data analysis to be $n = 54$ (as opposed to $n = 57$) from this section onwards.

Analysis. To evaluate whether the AISS leads to better or worse outcomes in terms of alert classification, we evaluated the fractions of correct alert classifications across the treat-

5. As the experiment was prepared multiple weeks in advance, the experiment balance between control and treatment group suffered from, by chance, more students dropping out from our treatment group than from the control group.

TABLE 3. ANALYSIS ACCURACY FOR EVALUATION STUDY 2

	A3 (%)		A2 (%)		B10 (%)	
	Cor.	¬ Cor.	Cor.	¬ Cor.	Cor.	¬ Cor.
AISS (n=24)	87.5	12.5	66.7	33.3	62.5	37.5
¬ AISS (n=30)	76.7	23.3	83.3	16.7	36.7	63.3
Total (n=54)	81.5	18.5	75.9	24.1	48.1	51.9

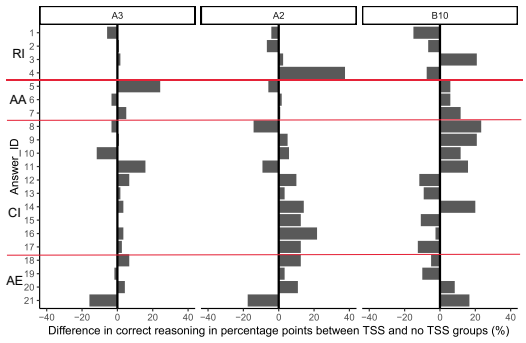


Figure 3. Difference in the correct reason between the AISS and the no AISS group (AISS: n=24, no AISS: n = 30). Positive differences indicate a relative advantage for the AISS group.

ment and control groups. Further, we evaluated how rapidly analysts in the two groups submit their findings, to identify whether the AISS can help analysts reach a conclusion faster than without the AISS. Finally, we evaluated the survey results on the analysts understanding of each alert to qualitatively assess if subjects engaged with the AISS developed an overall more accurate understanding of the security events.

6.2. Results

Accuracy of analysis. Table 3 shows the classification accuracy for each alert for the groups with and without AISS respectively. Both A3 and A2 were classified correctly more than 75% of the times (81.5% and 75.9% respectively). Meanwhile, B10 was classified incorrectly more often than not (48.1%). The subjects who used the AISS classified A3 and B10 ('not interesting') more often correctly than the subjects without the AISS. However, the subjects without the AISS outperformed the group with the AISS in classifying A2 ('interesting') correctly. None of the classification differences across groups are statistically significant[6].

Overall understanding of the attack scenarios. Figure 3 shows the difference, in percentage points of correct answers on an investigated alert, between the group using the AISS and the group not using it. The answer_ID on the y-axis of Figure 3 corresponds with the enumeration of possible responses listed in Appendix A. Positive differences (bars extending to the right) indicate the treatment group (i.e., subjects using the AISS) provided a relatively higher fraction of correct answers for that question. We observe a general trend where the group using the AISS seems to reason more often correctly than the group not using the AISS: most differences are either close to zero, or positive. Overall, however, across the different alerts, there is no specific question for which the AISS consistently improves an analyst's understanding of that alert. This was expected as different alerts entail different analyses, which may uncover different information. In other words, the AISS can only help analysts form a correct idea of an alert for information that *does* exist. For example, if there is no actual attack there is no reason why the AISS should help analysts uncover information related to *Attack Evidence* ('AE' in Figure 3).

This is evident by considering the average of the correctness score across alerts, (+1.6% for A3, +4.5% A2, and +3.5% for B10). The AISS had a minimal impact on the first alert, but was more helpful with the two more complex investigations. Indeed, the first alert was a scan alert (*ET SCAN Suspicious inbound to Oracle SQL port 1521*) that does not lead to any actual attack meaning that there was no information to uncover during the investigation. The second alert, A2, (*ET MALWARE Backdoor family PCRat/Gh0st CnC traffic (OUTBOUND) 103*) was an actual attack. We observed that users of the AISS seemed to be at an advantage in understanding the *Attack Evidence* and *Contextual Information* related to the attack. For example, 5 out of the 8 participants that used the AISS who misclassified A2 reported that the attacker made a successful connection to the victim, which was a key piece of information to classify this alert. By contrast, only 1 out of the 5 participants who misclassified A2 without using the AISS identified this information. This suggested that, while relatively more likely to misclassify this alert, analysts in the AISS group did indeed develop a more accurate understanding of the data surrounding it. Therefore, this suggests that the relative difference in accuracy of classification for this alert (that seems to favour the non-AISS group, see Table 3) may be due to factors other than the subjects' understanding of the alert, such as a lack of clarity on what in the SOC qualifies as an 'interesting' alert. We discussed this in limitations (see Section 7). Finally, the third alert (*ETPRO HUNTING Generic Inbound URI Directory Traversal*) was also 'not interesting', albeit it being a relatively complex alert. We observe that the AISS-group was especially correct in assessments for the stage *Additional Alerts* and *Contextual Information*, and observe a general positive trend in the overall reasoning.

Time taken for alert analysis. Here we evaluate whether the AISS affects the time analysts take in reaching a (correct) conclusion. Figure 4 shows the time taken for each alert analysis for all (red) and correct (green) analyses with its associated medians and confidence intervals. Regardless of the experiment groups, we observe that A3 (first analyzed alert) took on average longer to analyze than A2 and B10

6. This is unsurprising, given the limited sample size. A sample size estimation for the power of a Fisher's exact test indicated that approximately n=180 subjects would have been needed to detect the largest difference we found between the two groups (i.e. for alert 2) with a power of 80% and a confidence of 95%.

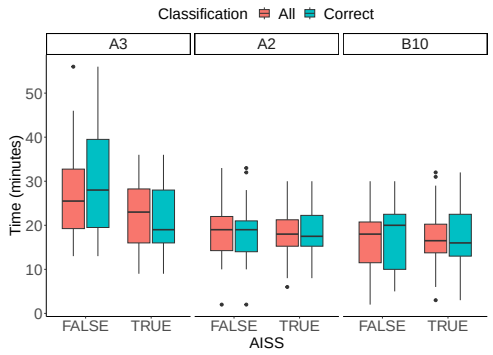


Figure 4. Time taken for analysts to analyze each alert between the experimental conditions

(second and third) with a median analysis time of 25 minutes vs 19 and 17 minutes respectively. Additionally, we also observed a greater spread in the time taken to analyze A3 ($sd = 9.6$) compared to A2 ($sd = 6.3$) and B10 ($sd = 6.7$). This can be explained by A3 being the first alert subjects analyze, and the higher spread resulting from a relatively higher uncertainty (and perhaps the presence of a residual learning curve after the warm-up exercise) in the first analysis than the subsequent ones. However, we observed that subjects employing the AISS are faster in concluding the alert is 'not interesting' for the first alert (i.e., the one that should be trivially dismissed) than analysts not using the AISS ($mean = 25.4, p = 0.03$ for a one-sided Wilcoxon ranked sum test). Additionally, we observed similar results when only considering correctly classified analyses. For A3, we see that the effect of the AISS is even stronger. The median time for correct analyses with the AISS within 21 minutes, as opposed to 25 minutes for all analyses ($mean = 21.7, p < 0.01$ for a one-sided Wilcoxon ranked sum test).

Moreover, we also observed that more experienced analysts (from ES1) spend less time on alert investigations (ES1: 7 and 5 minutes for A3, 18 and 10 minutes for A2 and 4 and 10 minutes for B10), despite the additional cognitive burden imposed by the think-aloud protocol in ES1. Interestingly, the difference in alert investigation time between the subjects in ES1 and ES2 is the largest in A3, a relatively simple scan alert which was investigated within 7 minutes in ES1.

6.3. Discussion of results from ES2

From ES2, we find that varying analysis difficulty does impact analysis outcomes in accuracy, understanding of events, and time. This is evident as Table 3 and Figures 3 and 4 show a wide spread across different alerts. However, in most but not all cases, we saw that there was a positive trend in both the correctness and the time of analysis when inexperienced analysts use the AISS. For the correctness, we observed that the subject group using the AISS classify 2 (A3 and B10) out of the 3 alerts more accurately.

Interestingly, the results suggest that the AISS either decreases or leaves the time taken for analysts for some alerts unchanged, despite the AISS adding additional 'operational overhead' by for example, incentivizing analysts to take notes in the tool, and nudging analysts to double checking whether the relevant information is collected or not. This suggests that the employment of the AISS for inexperienced analysts is not cumbersome and that the integration within the SIEM does not obstruct or interrupt the analyst's workflow. The decreased analysis time for the first alert, for which we observe a higher variance in completion time than for the other two, suggests that the AISS may help inexperienced users remove some uncertainty from their analysis. For example, pre-generated queries in the AISS may have relieved analysts from navigating the query language and focus on the relevant information instead. Moreover, the difference in alert investigation time for subjects in ES1 compared to the subjects in ES2, highlighted the stark differences between analysts with little to no experience and analysts with just few months of experience. We discussed this further in limitations (see Section 7).

7. Discussion and future work

Implications for research. The devised tool directly addresses the issue of alert investigation quality underlined in previous research [1], [2]. However, how to directly integrate analysis support systems in operative environments and for what purposes remains an open research problem. For example, different decision support systems may be integrated in information retrieval tasks, or involved in the sense-making process tying information cues together in a coherent 'story' of unfolding events. The present work shows that technological aid keeping the human in the loop can help analysts in forming a more complete and accurate picture of unfolding events. Future efforts in automating alert investigation [6], may consider adding layers of explainability and transparency to automated classifications to enable analysts in augmenting and integrating those decisions through experience [2]. The proposed tool could also be employed to extend research in the domain of analyst cognition and mental models, to for example, identify how analysts process the obtained information and build an underpinning model, integrating information collection and actions, for example, extending the AOH model [18]. Additionally, we observed that sometimes analysts misclassified alerts because the presented information, although correct, was interpreted wrongfully. This raises a question whether previous approaches focusing on knowledge transfer from senior/experienced analysts to junior analysts [8] is always beneficial, when the tacit knowledge may lead to wrongful conclusions. More research needs to be conducted on the reliability and variability of tacit knowledge [5], as opposed to firstly building training around tacit knowledge from single experienced analysts.

Implications for practice. Subjects appreciated the effective integration with the SIEM tool that is used in the

SOC. Yet, many SOCs do not use Security Onion or other open-source SIEM solutions [14], [21]. This raises the need for additional development of AISSs to be able to be integrated within different SIEMs. Nonetheless, as most SIEMs employ web UIs to interface with analysts, the presented concept can be easily ported to virtually any SIEM on the market, for example, as a browser extension. In addition to this, different SOCs may rely on other logs (e.g., OS logs and SYS logs [5]) as opposed to purely the network logs used by the collaborating SOC. Therefore, to fully realize the potential of the SOC, other types of logs have to be integrated within an AISS.

In addition, this work highlights the potential that a tool such as the AISS may have for alert analysis. Despite previous work [3] showing that following a structured TAP increases accuracy, enforcing a process without a tool requires more (micro-)management and thus more human resources. The proposed AISS reduces this burden by 'nudging' analysts into following this structured process. Furthermore, we provided a scalable alternative of enforcing a structured process as the tool is well-integrated into the SOC's SIEM system.

Moreover, our subjects praised features of the AISS allowing them to gather relevant information more easily with for example, the pre-generated queries. These features are a positive addition compared to simply knowing the structured TAP reported in [3], as knowing what information one needs to acquire does not necessary result in being able to acquire such information efficiently. Importantly, efficient information gathering and presentation to the analyst may potentially reduce the burden on analysts [13], [22], [23]. Crucially, how to effectively visualize alert information to analysts remains an open challenge at the crossroad between research and practice.

Implications at the collaborating SOC. We report considerations received by the SOC on the employment of the AISS . *The AISS has effects on how the training is carried out by senior analysts. The structured data collection allows the trainer to provide direct feedback on the results of the alert investigation, rather then correcting the method in which the findings are captured and the structure of the results. This saves time and removes the analyst's personal interpretation of how evidence should be presented to the trainers. Without the tool, crucial information is often missing when copying such as only including a field value but not the field name, or in screenshots when also irrelevant results are included. This creates large inefficiencies in the training process as the trainer can less easily spot a mistake or get the full picture of how the trainee assessed a certain alert. Further, having the TAP integrated in the SIEM means the analysts need to switch to documentation on the TAP less often, keeping focus on the investigation of the alert itself. Next, improving the AISS according to feedback from experimental subjects, such as adding descriptions of TAP steps, is a logical continuation of this initial research, to maximize the positive impact the AISS may have on the training of new analysts.*

Limitations. Our work contains four main limitations. The first limitation is the nuances of the SOC's taxonomy, which may be hard to be interpreted by student participants. Taxonomies for alert classification may be different across SOCs depending on monitored customers and the type of security alerts triggered. For example, company policy violations may be considered escalation-worthy alerts in one SOC while being considered benign elsewhere. Ideally, subjects ought to be trained fully on decisions related to alert classification, including the context of the customers served by the SOC. This was not feasible in ES2 due to time constraints imposed by educational settings. To mitigate this, we collected rationales behind subject's classifications in addition to the alert's classification in ES2 shown in the Appendix and Figure 3. Through collecting the rationales, we were able to understand if the subject understood the alert (mostly) correctly even if the subject may have misclassified the alert according to the SOC's taxonomy.

A second limitation is inherent to using the think-aloud protocol. Think-aloud protocols add additional cognitive workload on top of the performed task itself [20], potentially resulting in subjects requiring more time to investigate alerts than normal in ES1. Yet, the positive aspects of employing the think-aloud protocol in evaluating the usability of tools such as being able to understand why participants take certain actions in our proposed tool outweigh the limitations.

A third limitation is the recruitment process of subjects in ES2. Due to ethical constraints in an educational setting we opted not to pre-select participants based on specific education or test outcomes. This leaves open the possibility that expertise in security analysis and related skills may be higher for some students than others. To mitigate this, students were randomly assigned to the two experimental groups. Further, experimental outcomes do not reveal the presence of outliers suggesting some students in either group perform significantly better, or worse, than average.

The final limitation is how generalizable our subjects in ES2 are to inexperienced analysts working at a commercial SOC. Despite the educational program the students follow, results in the alert investigation time highlight a gap between students who only received a training on the week of the experiment versus analysts who have been working at a commercial SOC for 3 months. To mitigate this, we introduced two hands-on alert investigation experiences throughout the two educational sessions, one demo at the end of the first instruction lecture and one warm-up exercise before the experiment. Moreover, all T1 analysts must start working at some point and do not have 3 months of work experience. Therefore, ES2 highlights that the AISS is beneficial for the most inexperienced analysts, as well as for those who are in their early training phases of their SOC career.

8. Conclusions

Despite previous work [3] showing that a TAP has positive effects on the alert investigation process, there is a gap on how to implement such process in practice. We filled this gap by presenting a scalable tool to implement a threat analysis process (TAP) for network security alerts in a SOC.

The tool aims to support analysts, and it is integrated in the SIEM interface of the Security Onion. We collaborated with a commercial SOC implementing the TAP to evaluate it with four of their analysts, and ran an experiment with students to evaluate whether the tool aids their understanding of security events around investigated alerts. We found consistent evidence that the tool helps. Following this evaluation, the proposed tool will be integrated in the training procedures of the SOC for further testing.

Acknowledgment

This work is supported by the SeReNity project, Grant No. cs.010, funded by Netherlands Organisation 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 Eindhoven security hub SOC for its collaboration in this work.

References

[1] 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

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] M. Du, F. Li, G. Zheng, and V. Srikumar, "Deeplog: Anomaly detection and diagnosis from system logs through deep learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 1285–1298. [Online]. Available: https://doi.org/10.1145/3133956.3134015

[8] 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

[9] S. Friedl, M. Glas, L. Englbrecht, F. Böhm, and G. Pernul, "Forcyrange: An educational iot cyber range for live digital forensics," in *IFIP World Conference on Information Security Education*. Springer, 2022, pp. 77–91.

[10] 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

[11] R. Sadoddin and A. Ghorbani, "Alert correlation survey: framework and techniques," in *Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services*, ser. PST '06. New York, NY, USA: Association for Computing Machinery, 2006. [Online]. Available: https://doi.org/10.1145/1501434.1501479

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] A. Applebaum, S. Johnson, M. Limiero, and M. Smith, "Playbook oriented cyber response," in *2018 National Cyber Summit (NCS)*, 2018, pp. 8–15.

[17] J. Yen, R. Erbacher, C. Zhong, and P. Liu, "Cognitive process," *Advances in Information Security*, vol. 62, pp. 119–144, 10 2014.

[18] C. Zhong, A. Alnusair, B. Sayger, A. Troxell, and J. Yao, "Aoh-map: A mind mapping system for supporting collaborative cyber security analysis," in *2019 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, 2019, pp. 74–80.

[19] 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

[20] M. F. Pike, H. A. Maior, M. Porcheron, S. C. Sharples, and M. L. Wilson, "Measuring the effect of think aloud protocols on workload using fnirs," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 3807–3816. [Online]. Available: https://doi.org/10.1145/2556288.2556974

[21] A. Szarvák and V. Póser, "Review of using open source software for soc for education purposes – a case study," in *2021 IEEE 25th International Conference on Intelligent Engineering Systems (INES)*, 2021, pp. 000 209–000 214.

[22] 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.

[23] 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.

TABLE 4. DESIGN REQUIREMENTS

ID	Requirement	Description
R-1	Adherence to model	AISS should be constructed around analysis model, following stages and steps in sequence to allow the analyst to internalize the process.
R-2	Data collection	AISS should allow for the analyst to collect all relevant data to a model stage in a single view to help them forming a full view of the evidence and consolidate their findings during the investigation process
R-3	Seamless integration	AISS should be integrated into, and interact with (e.g. to fetch information from the existing interface), an environment that the analyst is familiar with (i.e., Security Onion Console) without being intrusive or interrupting the analyst workflow.
R-4	Extensibility	AISS should be designed such that future extensions are possible and trivial to make, for example to introduce the automation of certain steps or provide suggestions to the analyst.

Appendix

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

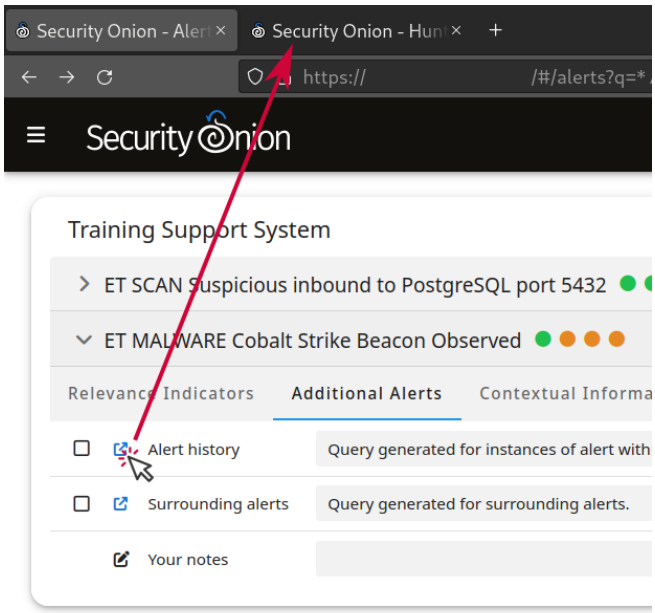


Figure 5. Query generation

Interview Questions.

1) How did you perceive the overall effectiveness of the AISS?
2) What aspects or features of the tool, if any, proved useful for your learning?
3) Were there any notable challenges you faced while using the tool?
4) What are your impressions regarding the tool's user interface and navigability?
5) Where do you see room for improvement in the AISS?
6) Any additional feedback or suggestions regarding the AISS?

Survey questions.

1) Which of the statements below did you consider to determine whether the alert is relevant or not?
2) Which of the statements below did you consider to determine whether any additional alerts indicated additional activity around the alert in question?
3) Which of the statements below did you consider to determine whether any information about or in the logs indicated additional activity around the alert in question?
4) Which of the statements below did you consider to determine whether there was sufficient/insufficient evidence of a successful attack around the alert in question?

Possible responses.

Question 1:

1) The reference URL of the signature indicates that the alert is not relevant
2) The creation date of the signature is not recent enough to be relevant
3) The destination IP is in the scope of the customer
4) The source IP is in the scope of the customer

Question 2:

5) The alert has triggered often in the past
6) There were many alerts generated in the system in the past with the same source or destination IP
7) There were many alerts with the same source or destination IP around the same timeframe as the alert in question

Question 3:

8) The source IP of the alert generated a lot of related logs

9) The destination IP of the alert generated a lot of related logs
10) There were many related conn logs
11) There were many related protocol specific (i.e non-conn) logs
12) There were a lot network connections between the source and destination IP
13) There was a successful connection established between the source and destination IP
14) There were many packet(s) exchanged between the source and destination IP
15) File(s) were exchanged between the source and destination IP
16) The targeted host behaved in an unusual manner
17) The targeted host is known to be not vulnerable to the attack described by the alert

Question 4:

18) The external IP address of the alert was flagged as malicious
19) The potential victim accessed a malicious domain
20) A malicious file has been exchanged between the IPs in the alert
21) A malicious IP established a successful connection with the victim or the victim successfully connected to a malicious IP

TABLE 5. CORRECTNESS AND TIME TAKEN FOR ANALYST CLASSIFICATIONS FOR EVALUATION STUDY 1 (GT: GROUND TRUTH; NI: NOT INTERESTING; I: INTERESTING) TIME IS ROUNDED TO THE NEAREST MINUTE AS THE TIMING ERROR IMPOSED BY THE THINK-ALOUD PROTOCOL MAKES IT IMPOSSIBLE TO COMPARE GROUPS AT THE LEVEL OF SECONDS. MEDIAN FOR THE INTERESTING ALERTS IS 16.5 MINUTES VS 7 MINUTES FOR NOT INTERESTING ALERTS.

GT	Alert	Subject	Correct	Time (min)
NI	A3	S1	✓	7
		S2	✓	5
	A4	S1	✓	13
		S2	✓	8
	A7	S1	✓	10
		S2	✓	5
	A8	S1	✓	16
		S2	✓	7
	A10	S1	✓	7
		S2	✓	4
	B3	S3	✓	12
		S4	✓	15
	B5	S3	✓	4
		S4	✓	7
	B7	S3	✓	3
		S4	✓	9
	B9	S3	✓	4
		S4		4
	B10	S3	✓	4
		S4	✓	10
I	A1	S1	✓	35
		S2	✓	20
	A2	S1	✓	18
		S2		10
	A5	S1	✓	14
		S2	✓	7
	A6	S1		18
		S2	✓	9
	A9	S1	✓	18
		S2	✓	8
	B1	S3	✓	44
		S4	✓	28
	B2	S3	✓	21
		S4	✓	13
	B4	S3	✓	17
		S4	✓	10
	B6	S3	✓	16
		S4	✓	10
	B8	S3		20
		S4	✓	8

Warm up exercise

- 10 minutes; an alert you have seen before
- **Your task for this warmup:**
- Classify the alert as ‘interesting’ or ‘not interesting’
- Follow the process on the right!
- No need to submit anything

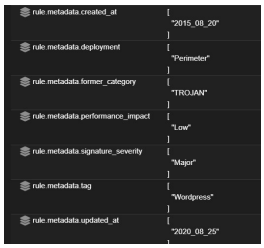
Assets you are protecting:
ANONYMIZED

1

1

Solutions

- Relevance indicators:
  - Generic signature (any .exe file from wordpress website)
  - Recently updates (2020)
  - Internal IP is involved
- Additional alerts:
  - NEW alert in the system
  - Some surrounding alerts

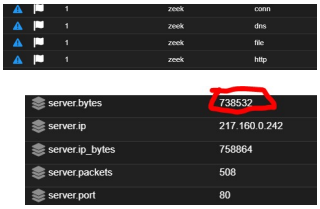


2

2

Solutions

- Contextual information
  - Lots of logs around the alert
  - Lots of bytes in a packet (likely with .exe file)
  - There is a **file** log



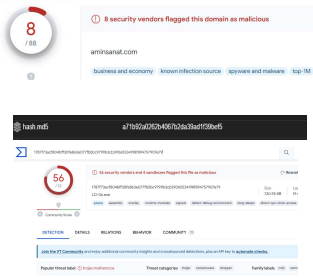
3

3

Solution

- Attack evidence
  - Connection was established
  - Accessed domain has indications of being malicious
  - Downloaded file almost certainly malicious

So definitely an **interesting** alert



4

4

Figure 6. The warm-up exercise and its solution: an example how the SOC classifies an alert as “Interesting”.