'Give Me Structure'

‘Give Me Structure’: Synthesis and Evaluation of a (Network) Threat Analysis Process Supporting Tier 1 Investigations in a Security Operation Center

Leon Kersten, Tom Mulders, Emmanuele Zambon, Chris Snijders, and Luca Allodi, *Eindhoven University of Technology*

https://www.usenix.org/conference/soups2023/presentation/kersten

This paper is included in the Proceedings of the Nineteenth Symposium on Usable Privacy and Security.

August 7–8, 2023 • Anaheim, CA, USA

978-1-939133-36-6

Open access to the Proceedings of the Nineteenth Symposium on Usable Privacy and Security is sponsored by USENIX.

‘Give Me Structure’: Synthesis and Evaluation of a (Network) Threat Analysis Process Supporting Tier 1 Investigations in a Security Operation Center

Leon Kersten
Eindhoven University of Technology

Tom Mulders
Eindhoven University of Technology

Emmanuele Zambon
Eindhoven University of Technology

Chris Snijders
Eindhoven University of Technology

Luca Allodi
Eindhoven University of Technology

Abstract

Current threat analysis processes followed by tier-1 (T1) analysts in a Security Operation Center (SOC) rely mainly on tacit knowledge, and can differ greatly across analysts. The lack of structure and clear objectives to T1 analyses makes operative inefficiencies hard to spot, SOC performance hard to measure (and therefore improve), results in overall lower security for the monitored environment(s), and contributes to analyst burnout. In this work we collaborate with a commercial SOC to devise a 4-stage (network) threat analysis process to support the collection and analysis of relevant information for threat analysis. We conduct an experiment with ten T1 analysts employed in the SOC and show that analysts following the proposed process are 2.5 times more likely to produce an accurate assessment than analysts who do not. We evaluate qualitatively the effects of the process on analysts decisions, and discuss implications for practice and research.

1 Introduction

As the volume and sophistication of cyber-attacks increase, the security of networks and systems is of key societal and economic importance. *Security Operation Centers* (SOCs) are business units (within a larger organizational setting), or services (that typically sell managed security services such as security monitoring to third party organizations) whose purpose is to detect cyber-attacks within the monitored environments. Their effectiveness is of primary importance both operationally (to maintain security) and strategically (to deter attacks) [6, 31]. A typical SOC is structured around a tiered system of analysts whereby incoming security events in the form of alerts are first analyzed by tier 1 (T1) analysts, who are typically junior and relatively inexperienced [20] and only escalated to higher tiers (typically through T2 and up to T3 analysts) when the T1 believes the event to be a potential threat to an organization [14]. This tiered system generates a procedure whereby T1 analysts analyze plentiful of false positive alerts (i.e., that are ‘not interesting’ for escalation) [2] and pass on relevant information to higher tiers for more in-depth investigation on alerts for which T1s cannot rule out evidence of attack [14].

Thus, the timeliness, accuracy, and relevancy of the information T1 analysts pass on to T2/T3 analysts is crucial to effective and efficient SOC operations, and by extension to the security of the monitored environments. Despite this, in many SOCs, the actions that SOC analysts take and the information they seek to inform their decisions depends mainly on tacit knowledge and their own background [5, 23], as opposed to a clear structure or framework to identify relevant evidence leading to well-informed decisions. T1 analysts typically do receive training, but generally in the form of internal procedures, systems, and new vulnerabilities [22], rather than in the form of an evidence-based decision-making process for effective threat analysis. This can lead to large differences in accuracy across T1 analysts [26, 28]. An unstructured analysis process can also be problematic in terms of the information passed over to a T2 analyst when an alert is escalated. T2 analysts then need to process reports that are less standardized, coherent and actionable.

Partially mitigating this issue, most SOCs implement so-called ‘playbooks’ or ‘runbooks’, documenting the procedures T1 analysts should follow when analyzing alerts related to a certain use case (e.g. a use case for ‘ransomware’). However, playbook documents are known to have update and maintenance issues, or only cover generic use cases that may induce analysts to use them less than originally intended [5]. In other extreme cases, some managed SOCs employ playbooks that are, in essence, automation rules to report information back to their customers. In those SOCs the T1 analyst (if present

USENIX Symposium on Usable Privacy and Security (SOUPS) 2023.
August 6–8, 2023, Anaheim, CA, USA

at all) is essentially reduced to an automaton executing a specific algorithm for each type of alert [22]. This is problematic considering the dynamic nature of cyber-attacks, and the high rate of false positives generated by detectors [2]. Indeed, the relevant literature suggests that analysts perform better when trained on a large variety of threats whose investigation require high cognitive engagement, as opposed to executing pre-determined tasks [8]. In addition, automaton execution possibly leads to a higher likelihood of burnout [22]. For these combined reasons, numerous previous studies have stressed the importance of humans and their decision making abilities in a SOC [10, 22, 31]. In this work we focus on supporting T1 analysts' cognitive engagement by proposing a general framework ('structure') to guide the T1 through the threat analysis process. More specifically, we collaborate with a commercial SOC (the Eindhoven Security Hub SOC [1]) to devise, together with three senior analysts, a structure for the T1 threat analysis process. We then evaluate the effects of the proposed process via a controlled experiment with 10 T1 analysts recruited at the SOC jointly analyzing 200 alerts from a real monitored environment.

Scope and contribution. The aim of this paper is to evaluate whether the creative cognitive process behind threat analysis can be aided by providing analysts with guidance on what information to collect to address specific 'stages' of the threat analysis process. This is different from building an 'algorithm' or set of heuristics to automate analyst decisions, which is not within the scope of this paper. In terms of scope of the proposed process, we focus on network event analysis.

The remainder of the paper is structured as follows. Section 2 introduces the background on the role of T1 analysts in SOCs, their security analysis process, and discusses related work. Section 3 presents the research questions addressed in this work, and Section 4 describes the methodology to answer them. Section 5 presents our baseline threat analysis process in detail and Section 6 provides the results to validate our process. Finally, Section 7 discusses our findings and Section 8 provides conclusions.

2 Background and Related Work

2.1 SOCs and tier 1 analysts

A security operation center (SOC) is a provider of security services to organizations. SOCs can be internal, where they provide security services to their own (often large) organization, or external by providing security services to third parties. The security services provided by SOCs can range from network intrusion detection systems (NIDS), to endpoint detection and firewall monitoring. The tools providing these services may utilize static detection mechanisms, dynamic systems, machine learning, artificial intelligence and more. However, most of these technical security solutions in an operational SOC generate security events which are in the first instance evaluated by a T1 analyst in the SOC. The analysis performed by T1 analysts aims to discern interesting security events from not interesting events. In other words, identify 'interesting' security events to escalate to T2 and T3 security analysts for further investigation, communication to customers and possible mitigation actions.

Since T1 analysts are the first to analyze and classify security events, the accuracy and timeliness of their analysis is fundamental to a SOC. Indeed, an accurate and timely identification of a cyber-attack minimizes the time available for the attack to complete [8], or may prevent its impact to be fully realized. Unfortunately, investigations performed by T1 analysts are known to be, in general, error prone and time-consuming, despite being repetitive [10, 12, 22, 31]. Naturally, the quality of the analysis and thus the accuracy of the classification is dependent on the individual skill of the analyst. However, external factors can impact this significantly, such as the arrival of external vulnerability information, or the addition of a new network segment in scope of an analyst's monitoring. Given that this occurs regularly, yet not necessarily predictably, the quality of security event analyses can be quite variable [26].

2.2 The security analysis process

The analysis process of a T1 analyst has as input a security event, and as output a classification of this security event. Regardless of internal taxonomies, analysts can in general assign a security event to one of two groups: alerts worthy of escalation to a higher tier for further investigation (further referred to as 'interesting' alerts), and those who should not be escalated (further referred to as 'not interesting' alerts) [10].[2] To arrive at this conclusion, the analyst's job is to look for evidence relating to the security event under analysis with other (security or network) events from any available and relevant source, with the goal of performing a triage for a possible escalation to higher tiers [6, 19, 29].

The T1 analysts' workflow takes as input a large volume of information that a SOC analyst has to account for in order to classify a security alert. They may look at the source and destination IP addresses [7, 23], at an increase in activities on a certain network port [7], or at the packet size and the content of a payload [7]. From the literature we identify four main categories of information that is considered by an analyst: *Relevance indicators*, evaluating whether an alert is relevant to the scope of the analysis [6]; *Additional alerts*, considering whether other evidence exists that an attack may be ongoing [6, 7]; *Contextual information*, evaluating whether some evidence of an attack is present at or around the affected hosts or systems [7, 21]; *Attack Evidence*, evaluating whether

[1] https://www.eindhovensecurityhub.nl/

[2] More fine-grained evaluations (e.g., Command & Control traffic, suspicious/benign scanning activity, ..) are always possible and commonly employed in SOCs as 'metadata' attached to the categorization above.

there is evidence that the events generating the alert led to a (successful) attack [6]. Table 1 provides a summary.

On the other hand, different analysts are known to employ different strategies to analyse a specific alert [26, 28]. Indeed, there is no clear framework of reference on what information to collect and what evidence is relevant to which phase of the investigation [2]. This suggests that there is no one clear predefined process for T1 analysts to follow, and that analysis results are entirely left to an analyst's own background, knowledge, and skills [23]. This may lead to increased analyst burnout [14], and is particularly undesirable given the high-turnover nature of T1 analysts within SOCs (that are regularly substituted by more junior and inexperienced analysts).

2.3 Related Work

Most of the previous research on SOC analysts focuses on the work process of the SOC as a whole rather than specific roles within the SOC. Of the papers mentioned in this section, three are qualitative studies [6, 14, 23] and two include some quantitative results [28, 30]. Furthermore, previous work can be divided in those that specifically consider the alert analysis process of SOC analysts [6, 23, 28] (although with the abstraction level of SOCs as a whole) or those who consider the work of the analysts from an organizational perspective [14, 22].

D'Amico and Whitley [6] conducted a cognitive task analysis (CTA) on the general workflow of a SOC, the security analysis process and the decision making process of an analyst. The authors identified a tiered system where a large volume of data enters the SOC, and that in each tier data is either discarded or retained to transfer to the next tier. Their work identifies several pieces of information that SOC analysts utilize to analyze security events and based on the CTA the authors draw conclusions on how visualization tools can and should integrate in the SOC enviroment. Although their work does not conduct a CTA for specific roles within the SOC (e.g T1 analyst), it provides an overview of how the SOC as a whole analyze and handle incoming security events. Similarly, Zhong et al. [28], captured analysis operations performed by analysts in a SOC and the hypotheses they generate and utilize in this process. The authors conduct CTA to capture fine-grained processes performed as part of the alert analysis. Interestingly, the authors highlight the observation that analysts employ different strategies and processes to explore the data and generate hypotheses to investigate. In later work Zhong et al. [30] propose a tool that automates the data triage aspect of aT1 analyst's work. They observed high-performance and satisfactory false-positive rates. They do note, however, that the quality of the system depends on the quality of the triage traces, which in turn depends on the quality of the analyst. Notably, this approach utilizes the operations of the analysts, such as "searching","selecting" and "filtering" [30], and does not capture why an analyst performs this action, nor what evidence is obtained from this operation.

Kokulu et al. conducted a qualitative study on issues within the SOC [14]. One of the primary findings is the current metrics for SOC performance are not effective. Moreover, this is a point of contention between security analysts and their managers. They also found the speed of response and the level of automation to be similarly (and very) important for effective SOC operations. Additionally, they noted that poor analyst training and high false-positive rates are issues within the SOCs in their research. This all culminates into poor quality of analyses, if left unaddressed.

Another interesting observation noted by Sunderamurthy et al [23] is the problem of tacit knowledge within the SOC; decisions made by security analysts are based on intuition and not documented. Often, the security analysts cannot clearly communicate their knowledge related to the incident and the reason for their classification of this incident [23]. This is a key component of the services provided by SOCs, as the contact point for the monitored environment must be provided with accurate and actionable evidence of a security incident. The contact person must be convinced that mitigation is necessary and warrant a potential interruption of business processes. T1 security analysts must do this as well when escalating to T2 or T3 analysts. Good communication about what a T1 analyst has observed, supporting evidence and their decision process is therefore a must in an effective SOC. On this line, [23] reports that "*SOC jobs such as incident response and forensic analysis have become so sophisticated and expertise driven that understanding the process is nearly impossible without doing the job.*"

3 Problem Statement and Research Questions

The process of alert investigation is repetitive, time-consuming and error prone [10, 12, 22, 31]. Much research has been done on the automation of individual steps or parts of the investigation, such as correlation and alert reduction, often relying on automated learning techniques [24, 31]. Past research also provided an high level overview of the workflow of an analyst [6, 7, 28]. However, to our knowledge a clear structure of the investigative process, and an evaluation of the extent to which it would aid in accurate decision making by T1 analysts, is currently missing [25]. The problem statement above gives rise to the following two research questions:

RQ1: Which sequence of tasks and information gathering should a tier 1 analyst perform when executing a threat analysis process to analyze network security alerts?

RQ2: To what extent can the derived threat analysis process increase the accuracy of classifying network security alerts for tier 1 analysts?

Table 1: Categories of information SOC analysts employ to classify alerts

Information Category	Definition	References
Relevance indicators	Information to classify whether the alert under investigation is even relevant for the SOC, based on the signature and the scope of the customer.	[2, 6]
Additional alerts	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.	[6, 7]
Contextual information	Information about the behavior and other observables of the involved internal host.	[2, 7, 12, 21]
Attack evidence	Any evidence relating to the alleged attack including the type of attack, attacker and any indication of success.	[2, 6]

4 Methodology

Overview of method. To answer our research questions we rely on an ongoing collaboration with a commercial (managed) SOC, the Eindhoven Security Hub SOC (for brevity referred to as 'the SOC'), providing network monitoring services for small and medium-size organizations active in education, IT-services, and manufacturing.

RQ1. To derive the threat analysis process we worked closely with a T2 security analysis expert with 4+ years of experience who is currently active in the SOC to identify which information a T1 analyst should consider for the 'escalation' of the alert to be useful for the higher-tier analysts. The derived information was then mapped to the categories presented in Table 1 and used to build a step-wise process for the analysis. This process was then iteratively and independently evaluated by two senior analysts (with respectively 15+ and 10+ years of experience) active in the SOC, until all three experts were in agreement on the resulting threat analysis process. Implementation details and results are given in Section 5.

RQ2. To validate the identified threat analysis process we designed an experiment to compare the performance of SOC analysts who employ the process to conduct analysis of alert data in the SOC, against that of analysts who do not. We employ sensor data from one of the organizations monitored by the SOC to sample alerts from a real-life environment. This ensures that baseline information (such as the IP space of that organization) is already known to the analysts. In addition to using alerts from our sensor, we generated additional alerts by injecting attacks into the virtual SOC environment to validate our process on alerts relating to successful attacks. To not affect SOC operations, we reproduce a near identical virtual environment that T1 analysts employ in the SOC in their day to day work (details of the environment can be found in Appendix B), and recruit our subjects from the pool of analysts employed at the time at the SOC.

4.1 Synthesis of the threat analysis process

Figure 1 provides an overview of the iterative process we employ to construct our proposed threat analysis process.

Preliminary alert analysis. In order to establish a set of information a T1 analyst should collect we adopted a bottom-up approach and sampled a set of 10 security alerts from a prototype sensor of the SOC. This relatively low number of alerts was chosen in first instance under the observation (and in consultation with the involved SOC experts) that the T1 analysis process is very repetitive and does not vary significantly across (network) security alerts. To make sure saturation in the collected information steps is reached, we adopted a step-wise process whereby additional information of relevance to the process is added to the set as each new alert is analyzed.

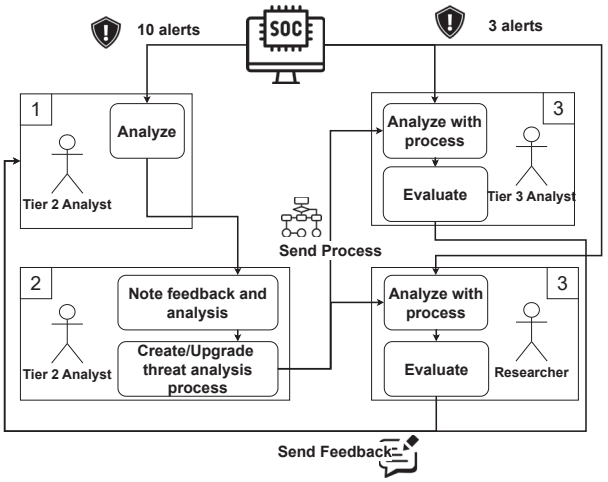


Figure 1: Synthesis of the baseline threat analysis process

The ten sampled alerts consist of malware, exploits, command & control, policy violations and scan alerts. The alerts were randomly selected from one of the monitored environments in the SOC. These alerts were completely analyzed by the T2 analyst. The result is an information set with all information utilized by the analyst during their analysis.

Process construction. Having identified the steps of the analysis process, the T2 analyst mapped each step to the stages reported in Table 1. The T2 analyst then employed the obtained mapping to reconstruct the process they employed during their analyses.

Process verification. The obtained threat analysis process is then given in input to two separate senior analysts (one T3 analyst with 15+ years of experience and a security researcher with 10+ years of experience in threat analysis). Each expert is

asked to independently analyze three security alerts randomly obtained from the SOC environment (distinct from the ten employed for the process derivation) using the provided threat analysis process. Each senior analyst independently provided the T2 analyst with feedback on the process and considered information, and the process is updated accordingly. This process verification and update loop was repeated until all three experts agreed on the devised threat analysis process.

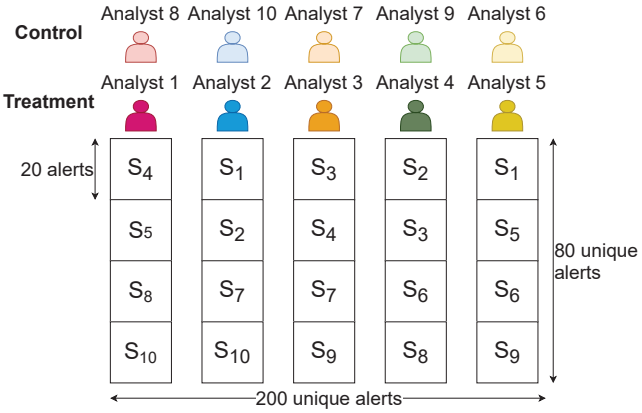
4.2 Experimental evaluation

To evaluate the effect of our threat analysis process on analysts' accuracy (**RQ2**), we ran an experiment involving T1 analysts and real alert data from one of the SOC sensors.

Experimental design. Figure 2 provides an overview of the experimental design. From the SOC environment we sampled 200 alerts and divided these in ten batches ('scenarios') of 20 alerts each. We then recruited ten T1 analysts and asked them to analyse four batches of alerts each (for a total 80 alert analyses per analyst);[3] each analyst was assigned to either the treatment group (i.e., following the proposed process for the analysis) or the control group, and asked to classify each alert as either 'interesting' or 'not interesting'. These assessments are then compared against a ground truth of assessments (defined by the SOC's T2 analyst) to evaluate differences in accuracy between the treatment and control groups. In the following, we describe our choice of subjects, how we designed the sets of alerts per subject, how we derived the ground truth and the details of the experimental setup.

Subjects. We recruited as subjects of our experiment ten junior analysts over a period of six months, in two batches of five analysts each. To maintain comparable experience levels across recruited analysts, we recruited them immediately after they joined the SOC. T1 analysts in the SOC are structurally hired as interns from the security program at a technical university in Europe. Their turnover rate varies between four and six months of employment.[4] All subjects are assessed before joining the team for their background, and are given the same technical on-the-job training. Analysts in the treatment group were given an additional training on the devised threat analysis process they will employ during the experiment.

Designing alert sets. *Collecting 'baseline' alert data.* To maintain realism of the experimental setup, we collected alerts from the network environment of a customer of the SOC over the course of 2.5 weeks. To make sure the collected alerts were not already investigated by our subjects as part of their normal job activities, we selected a network whose



Each analyst evaluates overall 80 alerts over four of ten scenarios. Each of the ten scenarios contains 20 unique alerts, at least one of which related to an injected attack, for a total of 200 unique alerts across scenarios. Each scenario is assigned to a single analyst at most once, and is analysed by two different analysts per experiment condition. For example, S_4 is analysed by Analyst 1 and Analyst 3, who are assigned to the treatment group, and Analyst 8 and Analyst 7, who are assigned to the control group.

Figure 2: Overview of experiment design

data is only captured for technical testing purposes by the SOC (as opposed to for security monitoring). The environment from which the alert data is collected comprises over 1500 unique hosts and multiple DNS and file servers. We logged approximately 100M connections attempts and 48M DNS requests. These connections generated 350k security events distributed across 150 unique security alerts. As SOC data are over-represented by alerts of certain kinds (e.g. alerts related to scan activities), we employed a stratified random sampling method over the collected alerts. We employ the 'rule category' [17] attribute that comes with alerts to define the type of each alert. The considered alert categories are: `Scan`, `Malware`, `CnC` and `Policy`. From each of the rule categories, we randomly selected a unique rule, and from that we sampled a random alert generated by that rule.

Generating 'successful' attacks. To generate 'interesting' alerts we could not rely on existing data in the SOC, as actual attacks are rare. We therefore employed PCAP network traffic from malware-traffic-analysis.net [15], which provides records of malicious network traffic of (multi-stage) malware attacks, to inject simulated attacks to the SOC sensors generating security alerts. Details of the attacks are reported in Appendix A. To assure the realism of the attacks, IP addresses in the obtained PCAPS are adapted to those expected within the range of the monitored network from which the 'baseline data' is derived. To avoid conflicts in the data, only unassigned IP addresses are used to rewrite the PCAPs; only internal IPs to the infected network were changed (i.e. IPs of the malware infrastructure remained unaltered). DNS servers used in the attacks are set to be the actual internal DNS servers in that sub-net, reflecting the corporate policy for the sub-net of the monitored environment. To inject the PCAPS in the SOC

[3] In consultation with the T2 and T3 analysts involved in this research, we estimated an average assessment time of 10 minutes per alert. Therefore, expected that no scenario would take more than 4 hours of a T1 analyst's time.

[4] The high turnover rate is due to the SOC's contractual policy (that follows the study program followed by the student analysts at the time of their recruitment), rather than to a high 'drop out' rate.

Table 2: Alert distribution across alert categories

Category	Ground truth	Overall
Command and Control	12	12
Malware	13	39
Policy	5	35
Scan	20	114
Total	50	200

network sensor to generate security alerts, we employed the SAIBERSOC tool [18].

Overall, our scenarios consist of 178 alerts sampled from the baseline monitored environment and 22 alerts generated by the injected attacks, for a total of 200 alerts. The 'Scenarios' ($S_1, S_2, \ldots, S_{10}$ in Figure 2) were then created by constructing 10 non-overlapping sets of 20 alerts. Each scenario contains alerts generated by exactly one injected attack; each attack generates at least one (and at most four) alerts.

Ground truth derivation. Once we derived our scenarios and related alerts, the T2 analyst ran a blind analysis over 5 alerts per scenario (for a total of $10 \times 5 = 50$ alerts) to label them as 'interesting' or 'not interesting'. All alerts related to an attack in a scenario were included in the set given to the T2 analyst. The remaining alert(s) for the ground truth were chosen randomly.[5] The distribution of alerts per category is reported in Table 2.

Experimental setup. The first batch of five analysts was assigned to the treatment condition. This choice was motivated by the need to empirically verify the internal consistency of the baseline threat analysis process before booking analysts' time away from the SOC. Details are reported in Appendix D. This batch received an in-depth training on the devised threat analysis process; the training was delivered by a T2 analyst, during the T1 analysts' intake at the SOC. The second batch was assigned to the control condition and only received generic training that was in place at the SOC before the introduction of the devised process. Analysts from both batches were asked to record their classification for each of the twenty alerts in a scenario as 'interesting' or 'not interesting', and to motivate their decision in plain English. Additionally, analysts from the first (treatment) batch were asked to record their evaluation for each of the steps identified in the threat analysis model for each of the analyzed alerts, in a separate worksheet.

4.3 Ethical considerations

This research was executed under ethical approval from our institution's ethical review board under approval number ERB2022MCS20. We gained explicit and informed consent from all subjects to participate in this experiment. Subject's names were anonymized to disassociate their identity from any performance evaluations. Furthermore, to alleviate the workload of our subjects, subjects participated in the experiment during working hours, as opposed to participating on top of their regular commitment with the SOC.

[5] Classifying the entire set of 200 alerts was not feasible due to the required time during which the T2 analyst would have been unavailable to the regular SOC operations.

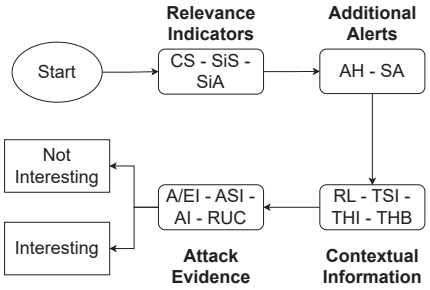


Figure 3: Overview of the baseline threat analysis process

5 A threat analysis process for network events

Following the iterative process described in Section 4.1, agreement on the details of the threat analysis process was reached at the fourth feedback iteration (at which point none of the three experts had any further remark). The result is 13 information steps mapped into the 4 stages reported in Table 1. Table 3 provides a summary of the final mapping between the process steps and stages. The final analysis process is visualized in Figure 3. The process guides the analyst in collecting evidence of an attack through the four identified stages; at the end of the process, the analyst decides whether there is enough evidence to classify the alert as 'interesting', or not.[6] The rest of this section details each step for every stage within the proposed threat analysis process.

5.1 Relevance Indicators

The first process stage consists of three steps; signature specificity (**SiS**), signature age (**SiA**) and customer scope (**CS**).

Signature specificity. To determine 'specificity', the analyst first determines whether the triggered signature is specific to a certain attack or service, or whether it is only a generic indicator of an attack. This step allows one to establish the initial priority of the alert analysis (e.g. specific indicators may be prioritised over generic indicators), as well as determine what to investigate in future steps. Generally, identifying that

[6] Importantly, we note that this process is not meant to be prescriptive, in that it does not provide instructions or thresholds to make specific decisions on the classification. Differently, it provides a framework of reference for the analyst to collect relevant information to make well-informed decisions on what action to take (i.e., ultimately, escalate or not escalate). Whether this decision can be at least partially automated or scripted away (on the basis of the collected information), or safely taken at a specific stage of the process, is out of the scope of this contribution.

Table 3: Mapping between the stages and the steps

Stage	Step	Description
Relevance indicators	signature specificity(**SiS**)	Indication of how specific the trigger condition is for the signature of the alert. (i.e Whether the signature is easily triggered).
	signature age(**SiA**)	The creation date of the signature, as new signatures are often more trustworthy and up-to-date than older ones.
	customer scope(**CS**)	Whether the alert is within the agreed scope of monitoring.
Additional alerts	alert history(**AH**)	The history of the same alert in the past. i.e. how commonly the alert triggerd in the past and for what reasons. This step is useful to detect common false positives.
	surrounding alerts(**SA**)	Other alerts relating to the specific alert under investigation. This step is useful for identifying alerts related to the same attack.
Contextual information	related logs(**RL**)	Logs related to the alert under investigation.
	traffic stream information(**TSI**)	The volume and content of the packets involved between the attacker and defender, compared to the expected volume and content for the protocol used.
	target host information(**THI**)	Any available information about the possibly affected host, such as whether it is a server or desktop, its OS etc.
	target hosts behaviour (**THB**)	The change in behavior of the host after the persumed attack
Attack evidence	attack/exploit information(**A/EI**)	The exact attack, objectives of the attack and the tools involved. The analyst can estimate the impact of the attack to its customer using this information.
	attacker information (**AI**)	Information about the attackers behavior, and whether the attack is from an unknown source.
	attack success indicators (**ASI**)	Information regarding whether the presumed attack was successful, such that the analyst may decide to not escalate unsuccessful attacks.
	relation to the use cases (**RUC**)	Indication of how much the possible attack overlaps with the use cases of the affected environment. The analyst considers the impact of the attack to the affected environment in this step.

the signature is specific here increases confidence in the event being interesting. Identifying that the signature is generic may decrease the confidence, depending on the level of generality.

Signature age. Analysts can check the signature creation date and last updated data of an alert to estimate if the behaviour triggering the alert is a recent or an old threat. This signals the age of associated threat, or indicates a potentially not interesting security event if the trigger conditions of the indicator are time-dependent.

Customer scope. The analyst determines if the alert is within the monitored scope by reviewing if necessary the customer security policy, as well as the service level agreements about sub-nets and reporting. The alerts with no to low impact, for example a guest network of the customer, can in this way be evaluated early on in the process as lower priority.

5.2 Additional alerts.

The second stage consists of two steps; alert history (**AH**) and surrounding alerts (**SA**).

Alert history. The analyst investigates the history of the alert under investigation. Namely, how often the alert has been triggered in the past, how often it was considered interesting, and whether it triggered for the same internal host before. Using this information, an analyst can verify quickly whether the observed alert is a common false positive or not. If past occurrences have been flagged as 'not interesting' due to them being false positives, the analyst can consider that the alert under investigation may be a false positive as well.

Surrounding alerts. The analyst investigates additional alerts similar to the one under investigation that were triggered by one or multiple of the involved hosts, around the time of the potential attack. When looking at these surrounding alerts, analysts may observe different alerts with similar names, indicating the same potential attack. This adds evidence that the event underlying these alerts may be interesting. Additionally the analyst may observe alerts for different phases of an attack, further strengthening the case for continuing to investigate the alert. For example, investigating a malware alert, a surrounding alert may be CnC activity. Identifying these surrounding alerts allows the analyst to get a more encompassing picture of an ongoing attack, if present.

5.3 Contextual information.

The third stage consists of four steps; related logs (**RL**), traffic stream information (**TSI**), target host information (**THI**) and target hosts behaviour (**THB**). At this stage, the analyst collects concrete evidence generated by the security systems, and information provided by the owner of the monitored environment. If the analyst finds no evidence of a potential attack reaching a vulnerable host, the analyst may consider it as evidence to classify the alert as 'not interesting'.

Related logs. This step focuses on identifying logs useful

to evaluate the cause or the outcome of the attack under investigation. This selection is largely dependent on the type of alert, and the type of traffic it is triggered on. In general, **RL** consists of at least a connection log, a protocol specific log (such as HTTP, or SSH), in addition to the alert log. Furthermore, any other logs generated by the receiving host of the protocol in the alert, or DNS logs, are typically related.

Traffic stream information. The analyst considers more detailed information about the packets sent to and from the host. This information can include the total number of bytes and packets sent by the attacker and defender, and the data contained in those packets. The protocol used between the communication of the attacker and defender is an important consideration in this step as it determines whether the number of packets and the data contained in them are abnormal or not in the specific context. For example, this step allows analysts to identify successful port scans by verifying if a response packet was sent back to the source. This also allows analysts to determine whether any 'lucky hits', were generated. A 'lucky hit' occurs when the trigger conditions of a signature (typically a non-specific one, as assessed in the **SiS** step) are met by pure chance on a random sequence of bytes, and thus trigger on benign traffic, producing false-positives.

Target host information. The analyst can utilize information about the host, such as whether it is a desktop or a server, its purpose (for example DNS server), its OS, its associated sub-net, host name, open ports and so on, to reason about whether the attack under investigation can ever lead to successful violation of corporate policies. This step can vary greatly from SOC to SOC and even from monitored environment to monitored environment, as corporate policies differ between organizations.

Target host behaviour. Next to utilizing known information about the targeted host, the analyst reviews the current behavior of the targeted host. For this, the logs produced by the IDS and network sniffer are utilized to review the behaviour of the host before and after the attack. If the host behaves abnormally compared to how the hosts normally would behave, it may indicate that the host was impacted by the attack.

5.4 Attack evidence

This stage consists of four steps; attack/exploit information (**A/EI**), attack success indicators (**ASI**), attacker information (**AI**) and relation to the use cases (**RUC**).

Attack/Exploit information. In this step, the analyst determines the exact attack and tools involved. The information required to determine this originates from the signature which triggered the alert, and open source information about the corresponding attack. From this step, it should be clear whether there is an attack, and if so what attack specifically. Using this information, in relation to that collected in the previous steps (e.g. **RL**, **THI**) the analyst estimates how this specific attack can have impact on the customer.

Attacker information. The analyst investigates the behaviour of the attacker (or at least of the attacking system). Using the logs generated as a result of the attacker behaviour, as well as using public sources, the analyst can determine whether the attacker is an actual attacker. This step is needed to rule out known and trusted sources such as (vulnerability) scanners, as well as help identifying false positive alerts generating 'hits' on backup streams, software updates, and benign network downloads.

Attack success indicators. The analyst investigates whether the attack was successful. Analysts use information obtained from former stages and open sources that identify clear indicators of successful attacks. Generally, the attack success indicators are highly dependent on the specific attack, however, generic indicators such as DNS requests for unusual top-level domains or internal scanning can be used as well.

Relation to use cases. The analyst consults the use cases for the affected environment. This step helps them to correctly identify the full impact for the environment, and thus the final classification of the alert. It also eliminates any alerts which are not important to the environment. For example, investigating a generic malware alert, having determined that it is actually adware on a desktop, the use cases may call for no action at all, depending on the environment. Finally, the use cases may provide useful information and guidance on what to report to higher tier analysts or the affected customer.

6 Experiment results

Table 4 provides an overview of the alert analyses performed by our subjects. Collectively, analysts classified an alert as 'interesting' 114 times, and 686 times as 'not interesting'. Furthermore, we observe that analysts who followed our process classify alerts more often as 'interesting' (67 times) than the analysts who did not follow the process (47 times, $\chi = 3.69, p = 0.055$). Whereas only borderline significant, this suggests that following the proposed process may increase the likelihood of escalating an alert to a higher tier. Meanwhile, we do not observe any within-group difference across analysts in terms of their classification outputs in either group (treatment: $\chi = 1.36, p = 0.85$; control: $\chi = 2.07, p = 0.72$). This suggests that the likelihood of an analyst's classification for a given alert may depend on the treatment group the analyst is assigned to, rather than on the analyst themselves.

Focusing on analysts accuracy, we observe that overall analysts not following our process show a accuracy of 82% in the classification; by contrast, analysts following the proposed process show an overall accuracy of 92%. Interestingly, this difference disappears when only considering alerts whose ground truth classification is 'not interesting'. By contrast, 'interesting' alerts were classified correctly only 65.9% (29 our of 44 possible assessments on 'interesting' alerts') of the times by analysts not employing our process, while the group

Table 4: Overview of analysts' classifications

		All alerts		Alerts included in ground truth					
				Interesting		Not Interesting		Total	
Analyst	Process	Int.	Not Int.	Correct	Wrong	Correct	Wrong	Correct	Wrong
1	Yes	14	66	8 (88.9%)	1 (11.1%)	11 (100.0%)	0 (0.0%)	19 (95.0%)	1 (5.0%)
2	Yes	10	70	7 (87.5%)	1 (12.5%)	11 (91.7%)	1 (8.3%)	18 (90.0%)	2 (10.0%)
3	Yes	14	66	7 (87.5%)	1 (12.5%)	11 (91.7%)	1 (8.3%)	18 (90.0%)	2 (10.0%)
4	Yes	15	65	8 (100.0%)	0 (0.0%)	11 (91.7%)	1 (8.3%)	19 (95.0%)	1 (5.0%)
5	Yes	14	66	9 (81.8%)	2 (18.2%)	9 (100.0%)	0 (0.0%)	18 (90.0%)	2 (10.0%)
6	No	9	71	9 (81.8%)	2 (18.2%)	9 (100.0%)	0 (0.0%)	18 (90.0%)	2 (10.0%)
7	No	8	72	4 (50.0%)	4 (50.0%)	11 (91.7%)	1 (8.3%)	15 (75.0%)	5 (25.0%)
8	No	11	69	8 (88.9%)	1 (11.1%)	10 (90.9%)	1 (9.1%)	18 (90.0%)	2 (10.0%)
9	No	7	73	5 (62.5%)	3 (37.5%)	12 (100.0%)	0 (0.0%)	17 (85.0%)	3 (15.0%)
10	No	12	68	3 (37.5%)	5 (62.5%)	11 (91.7%)	1 (8.3%)	14 (70.0%)	6 (30.0%)
				Overall					
With process		67	333	39 (88.6%)	5 (11.4%)	53 (94.6%)	3 (5.4%)	92 (92%)	8 (8%)
Without process		47	353	29 (65.9%)	15 (34.1%)	53 (94.6%)	3 (5.4%)	82 (82%)	18 (18%)
Total		114	686	68 (77.2%)	20 (22.8%)	106 (94.6%)	6 (5.4%)	174 (87%)	26 (13%)

who did follow the process classified the same set correctly 88.6% (39/44) of the times. This suggests that the proposed process is particularly useful for alerts related to attacks, reducing the classification inaccuracy by more than 20%. Generally, we find T1 analysts to perform better at classifying 'not interesting' alerts as opposed to 'interesting' alerts with a classification accuracy of 94.6% and 77.2% respectively.

To evaluate the effects of the proposed threat analysis process on assessment accuracy, we perform a logistic regression on the dependent variable *Correct*, which is a dummy variable set to 1 if an analyst correctly classifies the security alert, and 0 otherwise. The explanatory variables in the regression model are *Process* and *Category*. *Process* is a dummy variable set to 1 if the analyst followed our threat analysis process; *Category* is a categorical variable representing the category of an alert among the categories `Scan`, `Malware`, `CnC` and `Policy`. In addition, we run checks to evaluate whether a mixed effect model is required to account for the fact that multiple observations are assessed per subject, and checks to account for additional effects caused by the specific scenarios. To do this, we consider whether `Analysts` or the `Scenarios` play a role in the outcome. We run two separate logistic regression models: one with analyst dummy-variables as predictors, and one with scenario dummy-variables as predictors. Table 5 provides an overview of the results. For both models, we find no significant effect of any analyst or scenario on the predicted outcome. A joint ANOVA test confirms this as the null-hypothesis of all coefficients being equal to zero is not rejected, which is consistent with `Analyst` and `Scenario` not playing a role in differentiating assessments ($p = 0.365$ and $p = 0.323$ respectively). We therefore do not include either variable in the final model presented here, and use logistic regression to fit the model $Correct = c + \beta_1 Process + \beta \mathbf{Category}$.[7]

Table 5: Logistic regression on the correctness of evaluations: once with subject dummies and once with scenarios dummies.

Variable	Coeff.	p	Variable	Coeff.	p
(Intercept)	2.94	0.004	(Intercept)	2.20	0.003
Analyst 2	-0.75	0.556	*Scenario* 2	-0.46	0.635
Analyst 3	-0.75	0.556	*Scenario* 3	-0.81	0.384
Analyst 4	<0.01	1.000	*Scenario* 4	-0.81	0.384
Analyst 5	-0.75	0.556	*Scenario* 5	<0.01	1.000
Analyst 6	-0.75	0.556	*Scenario* 6	0.75	0.556
Analyst 7	-1.85	0.108	*Scenario* 7	-1.35	0.130
Analyst 8	-0.75	0.556	*Scenario* 8	0.75	0.556
Analyst 9	-1.21	0.314	*Scenario* 9	0.75	0.556
Analyst 10	-2.10	0.065	*Scenario* 10	<0.01	1.000

Table 6: Logistic regression on the correctness of evaluations, as dependent on the used process and the alert category

Variable	Coeff.	OR change (%)	p-value
(Intercept)	2.56	NA	<0.001
Process	**0.98**	**167.0**	**0.035**
Reference category: `Scan`			
Category : `CnC`	-1.20	-69.8	0.070
Category : `Malware`	**-1.89**	**-84.9**	**0.002**
Category : `Policy`	-0.76	-53.1	0.406

Table 6 shows the effect sizes alongside associated p-values from the fixed effects logistic regression model. Coefficients shown in bold denote an associated p-value of 0.05 or less which we consider statistically significant. As `Scan` is the most common alert category in the SOC, we choose it as the

[7] For completeness, we also estimated the mixed effects model. The coefficients are qualitatively identical both in magnitude and direction to those reported here.

baseline category for the variable `Category`; coefficients for other categories should therefore be interpreted relative to it. The coefficient of *Process* is 0.98 with a p-value of 0.035, showing that analysts following the proposed threat analysis process were significantly more likely to classify alerts correctly than analysts in the control group. This corresponds to a change in the odds of correct classification of 167%, i.e. a shift in probability of generating a correct assessment from approximately 82% in the control group to 92% in the treatment group. We also observe that there is a significant difference between `Malware` and `Scan` ($p = 0.002$) alerts. Malware related alerts were more often incorrectly assessed than scan alerts, indicating that Malware alerts are significantly more difficult to analyze correctly. Other differences across categories are smaller and not statistically significant.

6.1 Qualitative evaluation

We now try to qualitatively characterize the differences between the two groups by looking at specific classification tasks in the two groups. To reconstruct this, we look at the data annotated by the analysts with the motivations of their decisions for a classification for each specific alert.

Firstly, from the data we observed that subjects who do not follow our process typically based their decision to discard an alert (i.e classifying it as 'not interesting') after a single 'step' in the decision process. For example, a `CnC` alert (`ThreatFox BazarBackdoor botnet C2 traffic`, whose instance in our data is classified as 'interesting' by the T2 analyst) in scenario no. 3 was erroneously classified by an analyst as 'not interesting' as the network communication related to this alert "only" contained 10 packets. One of the analysts remarks:"*Its [the count is] below 50. So, this alert can also be dismissed.*" Whereas '50' is not a limit specified anywhere in the SOC for this type of alert, we later learned that this cut-off number is considered relevant for SSH brute force attacks. This suggests that the analyst erroneously considered this a universal threshold when deciding whether a communication is large enough to be considered potentially interesting, despite the alert in question being completely unrelated to SSH brute forcing. This seems in line with the generally accepted notion of 'implicit knowledge' being employed by analysts [5, 22]. Although our process does not prevent these mistakes from happening, following it may at least aid analysts in considering other steps as well to potentially classify alerts in a more informed manner.

In another investigation on alert `ET JA3 Hash - [Abuse.ch] Possible Dridex`, two analysts who erroneously classified it as 'not interesting' had previously observed high false positive rates with alerts associated to 'JA3' hashes. Therefore, analysts investigating JA3 alerts often mumbled that this is most likely going to be a false positive. Further, this alert was associated to a limited (9) number of packets, leading analysts not following the process to classify it as 'not interesting' despite the presence of concrete evidence of a connection from a suspicious IP being established with the host. By contrast, analysts in the treatment group identified that this alert was related to another 'interesting' alert at the **SA** step. Whereas the T1 analysts could not observe much of the data relating to the network communication of this alert, they identified sufficient evidence to escalate it, considering that if the alert was related to an attack it would have a high impact to the organization. One of the analysts following our process remarked the following related to this alert: "*Could not tell that the decoded message would have had a relation to this traffic. It is still malware-related, making it more significant for the customer and this same IP was also involved with the Threatfox backdoor alert.*" Interestingly, another analyst following the process and correctly classifying this alert as 'interesting' commented "*It's weird. JA3 is never interesting for us.*". This suggests they made similar considerations to the analysts not employing the process, but corrected their belief on the basis of the additional evidence collected.

We find three cases where following the process lead to analysts classifying a 'not interesting' alert as 'interesting', i.e. generating a false positive classification. One scan alert (`ET SCAN MS Terminal Server Traffic on Non-standard Port`) was classified as 'interesting' because there was insufficient evidence in one 'step' of the investigation. Almost all 'steps' in this investigation were leading to the conclusion that the alert was indeed not interesting. However, as the subject could not observe the behavior of the host, the subject decided to classify it as 'interesting' nonetheless to verify the alert with a T2 analyst. Another interesting case was when an analyst over-relied on **AI** instead of other steps in the process when investigating `ET SCAN ProxyReconBot CONNECT method to Mail`. This alert was raised despite the attempted scan receiving no response packets from the host. Yet, as the IP which was scanning the network corresponded to an untrusted domain, the T1 analyst decided to classify the alert as 'interesting'. Overall, these errors seem to be caused by a mistaken interpretation of the evidence (or lack thereof) by the analyst, rather than being induced by an incorrect evaluation strategy imposed by the process.

7 Discussion

Our findings show that analysts are significantly more likely to classify alerts correctly (odds increase by around 2.5 times) when following our baseline threat analysis process. This suggests that a structured process that T1 analysts can follow can when compared to sole reliance on tacit knowledge [5], aid in the correctness of security alert classification. Interestingly, in our experiment this increase in accuracy can be mainly attributed to 'interesting' alerts. In our experiment, we observe that for our analysts the rate of correct classifi-

cations of a 'not interesting' alert is higher than 90%; this suggests that a 'not interesting' alert may be easier to analyse and thus may benefit to a lesser extent from a structured way of processing information. For example, 'not interesting' alerts raised by attempted (but failed) port scans can often be dismissed by simply observing that the host system did not communicate back to the attacker. Therefore, most analysts would classify such alerts correctly, regardless of how rigorously they analyse the evidence. By contrast, T1 analysts in our experiment struggled more with analysing 'interesting' alerts correctly. This is unsurprising as these alerts are in general more complex and require the analysis of more information. Considering a delta of 20% in correct assessments for 'interesting' alerts between the two experimental conditions, our results suggest that structuring the analysis process may improve the classification accuracy specifically for the hardest alerts to analyze. Our example in Section 6.1 illustrates that this may be the case as analysts who do not follow our process may over-rely on one information point and simultaneously not consider other relevant information required for the analysis, whereas analysts who do eventually find the relevant information.

7.1 Implications for practice

Training. SOCs conduct training for their T1 analysts to for example, update analysts on the latest threats and how to analyze them [8, 11, 14]. As the training directly improves the effectiveness of analysts, it is considered a crucial aspect of a SOC [2, 11, 20]. However, T1 analysts need to have a baseline level in their work, such that analysts are able to perform adequately even if they have not been trained on that specific set of alerts. By structuring the workflow of a T1 analyst, it streamlines the baseline knowledge a T1 analyst should have in a SOC. The specific information T1 analysts should collect is explicitly defined, and thus SOCs can tailor their training towards how to collect the required information.

Measuring analyst performance. It is important for SOCs to measure the performance of their analysts such that they know where different detection tools or more training are required. For example, if a SOC realizes that analysts are having significant difficulties interpreting relevant logs, it may consider training their analysts on logs specifically. However, current quantitative metrics for SOC analysts often fail to measure the actual performance of the analyst [14, 22]. Our proposed threat analysis process can standardize the workflow of T1 analysts in terms of what information they should collect during their analyses. This gives SOC managers more concrete directions to measure the performance of their analysts, and of the processes they oversee. Although, it is out of the scope of this paper to present better metrics for T1 analyst performance, measuring performance of analysts at specific steps gives a more accurate overview of their analysis performance as opposed to only considering the number of escalated alerts [14], handled alerts [22] and time needed to analyse an alert [14]. Importantly, this may reveal 'weak' spots in the detection and escalation processes in place at a SOC, giving managers accurate metrics on which to base future adjustments.

Escalation. When an alert is being escalated by a T1 analyst, the analyst escalates the alert itself with supporting evidence why the alert has been escalated [13]. However, what may constitute as supporting evidence may differ for each individual analyst. SOCs may have their own standards and expectations on what T1 analysts include in their 'ticket'. On the other hand, the proposed threat analysis process (or any structured process analysts can follow) can be used to provide a ticketing standard that is in tune with the process that the T1 analyst follow. Furthermore, by removing uncertainty on the expectations of a T2 analyst on what information they will receive from a T1 analyst, the time needed to interpret each ticket by a T2 analyst may be reduced.

7.2 Implications for research and future work

There have been numerous previous studies presenting proposed tools pertaining to issues in the threat analysis process of SOC analysts [3, 4, 9, 31]. In line with this, future work could integrate a threat analysis process into an operational SOC. In our work, a subset of analysts were required to follow our process, however this is hard to enforce outside the controlled experiment. Observing how analysts would classify real security events using an integrated system that guides them into a desired process would potentially yield interesting insights into how such a process would function in practice.

Similarly, future work may evaluate 'how much information' is 'enough information' to collect to take an accurate decision on a specific alert. This may aid the navigation of an analysis process for analysts to 'quit' the process early on when enough evidence has been collected to take a negative decision. Similarly, the proposed process may be extended to other types of data, e.g. considering host or cloud log data rather than network (event) data.

In our work we focused on the accuracy of the classification of an analyst as the sole metric of a T1 analyst. However, previous studies have shown that timeliness of analysis is important as well in an operational SOC [8, 22]. Even if our threat analysis process leads to a better outcome in classification, it would be problematic if it added a significant time overhead for T1 analysts. Future work could investigate how following such a process influences the timeliness (and not only the accuracy) of the classification of security events.

7.3 Threats to validity

Construct validity. All injected attacks in our experiment consisted of malware related attacks, where a malware is installed, the host is controlled via a command and control server and where possibly some lateral movement took place

within the network. Considering that SOCs encounter other forms of attacks, a set of alerts generated by malware related attacks may not fully reflect the concept of 'interesting' alerts.

Internal validity. We assume in our experiment that all our subjects are equally skilled in analyzing security events and do not influence the accuracy of the classifications. However, in reality some analysts may be more skilled than others despite similar job experiences and educational background. To mitigate this, we checked whether concrete evidence exists that analysts influence the classification of the alerts or not. Additionally, when collecting data regarding the internal consistency of our process we used the response options in Table 7 in the Appendix. However, we did not test whether the interpretations of the response options differ among our subjects. In other words, subjects may give different responses to a step with the same observed data. Meanwhile, subjects may give identical responses to a step even though they have interpreted the data completely differently.

External validity. The main threat to external validity is the extent to which the employed SOC represents data and operations adopted by other SOCs. SOCs vary widely over both dimensions as they employ different technologies and sensors, SOC operations are not standardized, and by monitoring different networks/infrastructures they may evaluate different alerts over different environments [25]. However, virtually all SOCs at least monitor network traffic [6, 16, 27] and typically employ junior T1 analysts as a first line of defense to decide whether to escalate or ignore incoming alerts [25]. As the SOC under analysis performs only network analysis, and employs T1 analysts from the same 'pool' as most SOCs (i.e., junior staff in need of specialized cybersecurity training to operate well within a SOC [25]) we consider it to be representative of SOCs in general, over these two dimensions. Further, as the SOC under analysis *only* performs network monitoring, we can evaluate model effects without additional confounding factors caused by multiple data sources (e.g. system host logs). Whereas this suggests that our finding that a structured analysis process can help analysts in making accurate evaluations over network alerts, effect sizes may vary significantly across SOCs. Further research is needed to derive and evaluate analysis processes across different SOCs, monitored environments, and monitoring technologies, and their interactions. Additionally, whereas the collaborating SOC only allows two possible classifications for T1 analysts, past research [21] show that other SOCs may have more options to classify an alert. Our threat analysis process does not incorporate such frameworks for classifying alerts and thus, our process requires modification to accommodate different classification systems.

8 Conclusions

In this work we devised a threat analysis process to attempt to structure the work process of T1 SOC analysts. Our threat analysis process consists of four stages where it guides the analyst into collecting information relevant for their analysis. Furthermore, we conducted an experiment using real alert data with ten T1 analysts working in a commercial SOC to investigate the effect of structuring the threat analysis process. Our results show that our process increases the odds of our subjects correctly classifying an alert by 167%. More specifically, we observed that alerts correlated with a cyber attack are the alerts who significantly benefit from using our threat analysis process. Overall, our study suggests that structuring the analysis process of a T1 analyst aid in the correct classification of security alerts.

Acknowledgments

This work is supported by the SeReNity project, Grant No. cs.010, funded by Netherlands Organisation for Scientific Research (NWO) and by the INTERSECT project, Grant No. NWA.1162.18.301, funded by NWO. The authors also thank the Eindhoven security hub SOC for its collaboration in this work.

References

[1] Alan Agresti. *Categorical data analysis*, volume 792. John Wiley & Sons, 2012.

[2] Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 2783–2800, Boston, MA, August 2022. USENIX Association.

[3] Kenneth B. Alperin, Allan B. Wollaber, and Steven R. Gomez. Improving interpretability for cyber vulnerability assessment using focus and context visualizations. In *2020 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pages 30–39, 2020.

[4] Louise M. Axon, Bushra A. AlAhmadi, Jason R. C. Nurse, Michael Goldsmith, and Sadie Creese. Sonification in security operations centres: what do security practitioners think? *CoRR*, abs/1807.06706, 2018.

[5] Selina Y. Cho, Jassim Happa, and Sadie Creese. Capturing tacit knowledge in security operation centers. *IEEE Access*, 8:42021–42041, 2020.

[6] A. D'Amico and K. Whitley. *The Real Work of Computer Network Defense Analysts*, pages 19–37. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[7] Anita D'Amico, Kirsten Whitley, Daniel Tesone, Brianne O'Brien, and Emilie 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*, volume 49, pages 229–233, 09 2005.

[8] Varun Dutt, Young-Suk Ahn, and Cleotilde Gonzalez. Cyber situation awareness: Modeling the security analyst in a cyber-attack scenario through instance-based learning. In *20th Annual Conference on Behavior Representation in Modeling and Simulation 2011, BRiMS 2011*, pages 280–292, 07 2011.

[9] Roman Graf, Florian Skopik, and Kenny Whitebloom. A decision support model for situational awareness in national cyber operations centers. In *2016 International Conference On Cyber Situational Awareness, Data Analytics And Assessment (CyberSA)*, pages 1–6, 2016.

[10] Eric T. Greenlee, Gregory J. Funke, Joel S. Warm, Ben D. Sawyer, Victor S. Finomore, Vince F. Mancuso, Matthew E. Funke, and Gerald Matthews. Stress and workload profiles of network analysis: Not all tasks are created equal. In Denise Nicholson, editor, *Advances in Human Factors in Cybersecurity*, pages 153–166, Cham, 2016. Springer International Publishing.

[11] Robert Gutzwiller, Sunny Fugate, Ben Sawyer, and Peter Hancock. The human factors of cyber network defense. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 59:322–326, 09 2015.

[12] Wajih Hassan, Shengjian Guo, Ding Li, Zhengzhang Chen, Kangkook Jee, Zhichun Li, and Adam Bates. Nodoze: Combatting threat alert fatigue with automated provenance triage. In *NDSS Symposium*, 01 2019.

[13] Christopher Healey, Lihua Hao, and Steve Hutchinson. Visualizations and analysts. *Advances in Information Security*, 62:145–165, 10 2014.

[14] Faris Bugra Kokulu, Ananta Soneji, Tiffany Bao, Yan Shoshitaishvili, Ziming Zhao, Adam Doupé, and Gail-Joon 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*, CCS '19, page 1955–1970, New York, NY, USA, 2019. Association for Computing Machinery.

[15] @malware_traffic. My Technical Blog Posts. https://www.malware-traffic-analysis.net/. [Online; accessed May 5, 2022].

[16] Joseph Muniz, Gary McIntyre, and Nadhem AlFardan. *Security operations center: Building, operating, and maintaining your SOC*. Cisco Press, 2015.

[17] Proofpoint. TECH BRIEFET Category Descriptions. https://tools.emergingthreats.net. [Online; accessed August 3, 2022].

[18] Martin Rosso, Michele Campobasso, Ganduulga Gankhuyag, and Luca Allodi. Saibersoc: Synthetic attack injection to benchmark and evaluate the performance of security operation centers. In *Annual Computer Security Applications Conference*, ACSAC '20, page 141–153, New York, NY, USA, 2020. Association for Computing Machinery.

[19] Reza Sadoddin and Ali Ghorbani. Alert correlation survey: Framework and techniques. PST '06, New York, NY, USA, 2006. Association for Computing Machinery.

[20] Sathya Chandran Sundaramurthy, Alexandru G. Bardas, Jacob Case, Xinming Ou, Michael Wesch, John McHugh, and S. Raj Rajagopalan. A human capital model for mitigating security analyst burnout. In *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, pages 347–359, Ottawa, July 2015. USENIX Association.

[21] Sathya Chandran Sundaramurthy, Jacob Case, Tony Truong, Loai Zomlot, and Marcel Hoffmann. A tale of three security operation centers. In *Proceedings of the ACM Conference on Computer and Communications Security*, 11 2014.

[22] Sathya Chandran Sundaramurthy, John McHugh, Xinming Ou, Michael Wesch, Alexandru G. Bardas, and S. Raj 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)*, pages 237–251, Denver, CO, June 2016. USENIX Association.

[23] Sathya Chandran Sundaramurthy, John McHugh, Xinming Simon Ou, S. Raj Rajagopalan, and Michael Wesch. An anthropological approach to studying csirts. *IEEE Security & Privacy*, 12(5):52–60, 2014.

[24] Thijs van Ede, Hojjat Aghakhani, Noah Spahn, Riccardo Bortolameotti, Marco Cova, Andrea Continella, Maarten van Steen, Andreas Peter, Christopher Kruegel, and Giovanni Vigna. Deepcase: Semi-supervised contextual analysis of security events. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 522–539, 2022.

[25] Manfred Vielberth, Fabian Böhm, Ines Fichtinger, and Günther Pernul. Security operations center: A systematic study and open challenges. *IEEE Access*, 8:227756–227779, 2020.

[26] John Yen, Robert Erbacher, Chen Zhong, and Peng Liu. Cognitive process. *Advances in Information Security*, 62:119–144, 10 2014.

[27] Chen Zhong, Awny Alnusair, Brandon Sayger, Aaron Troxell, and Jun 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)*, pages 74–80, 2019.

[28] Chen Zhong, John Yen, Peng Liu, Rob Erbacher, Renee Etoty, and Christopher 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*, HotSoS '15, New York, NY, USA, 2015. Association for Computing Machinery.

[29] Chen Zhong, John Yen, Peng Liu, Rob F. Erbacher, Christopher Garneau, and Bo Chen. *Studying Analysts' Data Triage Operations in Cyber Defense Situational Analysis*, pages 128–169. Springer International Publishing, Cham, 2017.

[30] Chen Zhong, John Yen, Peng Liu, and Robert Erbacher. Learning from experts' experience: Toward automated cyber security data triage. *IEEE Systems Journal*, PP:1–12, 05 2018.

[31] Chen Zhong, John Yen, Peng Liu, and Robert 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)*, pages 357–363, 2016.

A Injected attacks

The list below identifies the attacks that were injected as part of the experiment, and the general behaviour that could be determined from the logs and alerts the attacks generated in the experiment environment. All attacks involve a malware(s). Alerts are generated from installations of such malware, command and control traffic or lateral movements. The table below shows which of the three aforementioned components of an attack generated an alert. **I** stands for installation, **CnC** for command and control and **LM** for lateral movements.

ID	Attack	I	CnC	LM
1	Remcos RAT	X	X	X
2	RIG Exploit Kit and Dridex	X	X	
3	Emotet and Trickbot		X	X
4	Qakbot and Cobalt Strike	X	X	
5	Qakbot and Spambot	X	X	
6	Hancitor and Cobalt Strike	X	X	X
7	Ghost RAT		X	
8	BazaarLoader and Cobalt Strike	X	X	X
9	MalSpam Brazil	X	X	
10	Ursnif	X	X	

B Environment

To ensure that the only additional training required for the experiment is the training related to our threat analysis process, we replicated the SOC environment on which our subjects work, and received their generic intake training. The environment is based on the Elastic Stack (ELK) and employs instrumented Suricata and Zeek sensors for the network event analysis (Suricata for attack detection and Zeek for logging network traffic). In our experiment Suricata was deployed with the open source *Emerging Threat Open ruleset*, as well as the licensed *Emerging Threat PRO ruleset* employed at the SOC. Replicating the configuration used in the production environment of the SOC, a subset of rules was configured to not trigger alerts (i.e., starting points for analyst investigations), but rather to generate logs stored in the SIEM. These logs can be used by analysts to further enrich the context of the events that triggered the investigated alert. These 'muted' signatures include *hunting*, *policy*, and *info signatures*. On top of the alerts and network logs generated by the sensors, the analysts were allowed to seek additional information from online sources (e.g. to check file hashes, IP address reputation, perform `whois` queries, ..) as normally performed during real operations. Because of storage limitations in the experiment environment, analysts did not have access to raw network traffic in PCAP files.

C Analysis sheet

Table 7 provides a summary of the options given to analysts for each of the process stages and steps identified in Fig. 3.

D Alert assessment consistency

We first evaluate whether the proposed threat analysis process produces consistent evaluations by the analysts. To evaluate this, we compare the evaluations made by analysts in the first batch across all steps of the proposed process. To do this, we compute the agreement score between analyses within the same scenario. The agreement score is calculated by counting the frequency per step where the two analysts outputted identical answers and then dividing it by the total number of alert instances (i.e 200).

We first consider the extent to which analysts agree on their assessments for each step of the process delineated in Table 3. Agreement scores for each step in the process are calculated across 200 pairwise comparisons of assessments performed by two separate analysts. Figure 4 shows the calculated agreement rates across each process stage and step. We find that, overall, analysts agree on the evaluation of the information

Table 7: Response option for each step

Stage	Step	No. Response options	Response options
Relevance indicators	signature specificity(**SiS**)	2	Old, New
	signature age(**SiA**)	2	Generic, Specific
	customer scope(**CS**)	2	Yes, No
Additional alerts	alert history(**AH**)	4	First occurence, Typically NI, Typically FP, Inconclusive
	surrounding alerts(**SA**)	2	Adds Evidence, Does not add evidence
Contextual information	related logs(**RL**)	4	Logs which indicate the result/impact of the event causing the alert, Logs which indicate the event which is the cause of alert, Both, None
	traffic stream information(**TSI**)	3	Small, Normal, Large
	target host information(**THI**)	2	Vulnerable, Not vulnerable
	target hosts behaviour (**THB**)	2	Normal behavior, Unusual behavior
Attack evidence	attack/exploit information(**A/EI**)	2	Attack, No attack
	attacker information (**AI**)	2	Trusted external host, Unknown external host
	attack success indicators (**ASI**)	2	Definitely unsuccessful, Successful, Unknown
	relation to the use cases (**RUC**)	2	Unrelated, Related

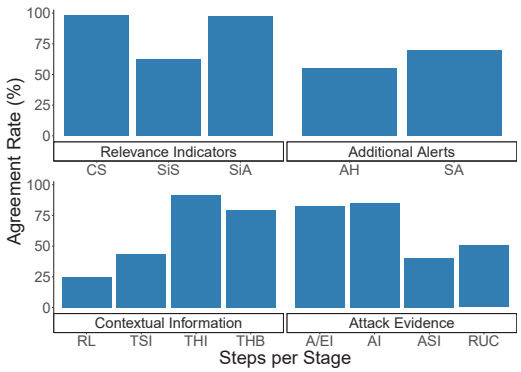


Figure 4: The internal consistency of our baseline threat analysis process.

collected in each step. We stress that the process does not mandate or instruct the analysts in how to find relevant information to make an assessment on that specific step (e.g. no query template is provided to identify 'related logs', `RL`, in the `Contextual information` stage). It is therefore to be expected that, the wider the information space associated to the assessment of a specific step is, the lower the expected agreement of analysts is. Our findings suggest that this holds also for our pool of analysts who have similar background and level of experience, and who received the same professional training. On the other hand, we observe that for each stage one or more steps consistently achieve relatively high agreement levels of 70% or more. To evaluate analysts agreement on the outcome of the process, we calculate Cohen's Kappa on analysts' final classification of an alert as 'interesting' or 'not interesting' for those analysts who employed the proposed process ($\kappa = 0.52$, $CI : [0.36, 0.68]$), and those who did not ($\kappa = 0.45,$, $CI : [0.28, 0.64]$). Whereas a straightforward interpretation of κ is not possible, both scores indicates a 'moderately strong agreement' [1, Ch.11.5.4] within the two groups. However, we do not find significant differences in the agreement levels between the two groups. This suggests that analysts in either group take similar decisions when compared to analysts in the same group.

E Changes to the process as a result of expert feedback

The initial sequence steps identified was extended with **CS** after the first round of verification, and **RUC** was moved from the start of the sequence to the end. **CS** was previously covered by **RUC**, but was found to be atomic and impactful enough to justify its own step. Additionally, **CS** can be determined more easily and thus earlier, than **RUC**.

As discovered during the verification by the experts, **RUC** requires details about the attack, the affected system and the impact, which are not available early on in the analysis process. For this reason as well, **RUC** was moved to the end of the sequence of steps. The adjustments detailed above were implemented before the experiment design and execution.

The question corresponding to the stage `Contextual information` was changed from "2-way communication established between attacker and attacked host" to "Vulnerable host reached by potential attack", to capture cases were attacks or exploits do not result in a two-way communication between the attacker and the attacked host.

Finally, a step named "signature quality" (now omitted) was split up into "signature specificity" and "signature age", the step "target host information" was split up into "target host information" and "target host behaviour" and the step "attack/exploit information" was split into "attack/exploit information" and "attacker information". These changes were made to make the steps more atomic.