Alert Alchemy
SOC Workflows and Decisions in the Management of NIDS Rules

Vermeer, Mathew; Kadenko, Natalia; van Eeten, Michel; Gañán, Carlos; Parkin, Simon

DOI
10.1145/3576915.3616581

Publication date
2023

Document Version
Final published version

Published in
CCS 2023 - Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security

Citation (APA)
Vermeer, M., Kadenko, N., van Eeten, M., Gañán, C., & Parkin, S. (2023). Alert Alchemy: SOC Workflows and Decisions in the Management of NIDS Rules. In *CCS 2023 - Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2770–2784). Association for Computing Machinery (ACM). https://doi.org/10.1145/3576915.3616581

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Alert Alchemy: SOC Workflows and Decisions in the Management of NIDS Rules

Mathew Vermeer
m.vermeer@tudelft.nl
Delft University of Technology

Natalia Kadenko
n.i.kadenko@tudelft.nl
Delft University of Technology

Michel van Eeten
m.j.g.vaneeten@tudelft.nl
Delft University of Technology

Carlos Gañán
c.hernandezganan@tudelft.nl
Delft University of Technology

Simon Parkin
s.e.parkin@tudelft.nl
Delft University of Technology

ABSTRACT

Signature-based network intrusion detection systems (NIDSs) and network intrusion prevention systems (NIPSs) remain at the heart of network defense, along with the rules that enable them to detect threats. These rules allow Security Operation Centers (SOCs) to properly defend a network, yet we know almost nothing about how rules are created, evaluated and managed from an organizational standpoint. In this work, we analyze the processes surrounding the creation, management, and acquisition of rules for network intrusion detection. To understand these processes, we conducted interviews with 17 professionals who work at Managed Security Service Providers (MSSPs) or other organizations that provide network monitoring as a service or conduct their own network monitoring internally. We discovered numerous critical factors, such as rule specificity and total number of alerts and false positives, that guide SOCs in their rule management processes. These lower-level aspects of network monitoring processes have generally been regarded as immutable by prior work, which has mainly focused on designing systems that handle the resulting alert flows by dynamically reducing the number of noisy alerts SOC analysts need to sift through. Instead, we present several recommendations that address these lower-level aspects to help improve alert quality and allow SOCs to better optimize workflows and use of available resources. These recommendations include increasing the specificity of rules, explicitly defining feedback loops from detection to rule development, and setting up organizational processes to improve the transfer of tacit knowledge.

CCS CONCEPTS

• **Security and privacy → Usability in security and privacy**.

KEYWORDS

SOC, Security Operation Centers, human factors, NIDS rules, interviews

ACM Reference Format:
Mathew Vermeer, Natalia Kadenko, Michel van Eeten, Carlos Gañán, and Simon Parkin. 2023. Alert Alchemy: SOC Workflows and Decisions in the Management of NIDS Rules. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security (CCS '23), November 26–30, 2023, Copenhagen, Denmark.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3576915.3616581

CCS '23, November 26–30, 2023, Copenhagen, Denmark

ACM ISBN 979-8-4007-0050-7/23/11.
https://doi.org/10.1145/3576915.3616581

1 INTRODUCTION

Security Operations Centers (SOC) are a key part of defending enterprise networks against attacks. Located inside these networks are a variety of security tools and systems – such as Network Intrusion Detection Systems (NIDS) – that generate alerts. Analysts in the SOC are burdened with triaging the deluge of alerts that consist mostly of false positives [3]. For years, academic research and industry reports alike have raised the problem of alert fatigue and analyst burnout [20, 30]

A lot of academic research on improving SOC operations has pursued better automation – most notably using machine learning (ML) to analyze the alerts and generate more informative alarms for the analysts to investigate. Indeed, an "insufficient automation level of SOC components" was considered the top issue by SOC managers, according to a recent study [15]. Prior work includes human-subjects studies with stakeholders, aiming to understand the problems in the SOC workflow to handle alerts [3, 15, 30, 31]. To address workload issues using ML, studies have focused on alert prioritization [12, 19, 37] and false positive detection [28, 37]. While the promise of ML for SOC operations is clear, adoption has been low, and solutions have under-delivered on the promise [11, 13].

The attempts to use ML for improving SOC operations have in common that they are "end of pipe" solutions: they take the flow of alerts as a given and build a system that ingests this flow in order to generate meaningful information that supports analysts in triaging the alerts. Alahmadi et al. [3] aim to improve the development of such ML solutions by identifying properties that the output of the ML, "alarms," should have to be useful to analysts. Ex-post, these systems are designed to take low quality alarm noise as input in order to produce informative and actionable output that SOC analysts can use. Well known within the ML community is that inadequate training data will have a detrimental effect on the quality of the resulting model, as is exemplified by the familiar saying "garbage in, garbage out" [23]. And even though their work focuses on alleviating the limitations of alarms through the use of ML-based tools, all of the mentioned limitations are also closely related to the features that determine the quality of a rule. By improving the quality of the rules, the quality of triggered alerts will also improve, thereby also bettering the quality of the input for whatever ML system uses this data.

Thus, we propose a different and complementary approach: rather than end-of-pipe, we focus on the source of the problem and want to understand how to improve the quality of the alerts that go into the pipe. Where do the alerts come from? Typically, they are generated by systems that compare the network traffic against a set of rules or signatures. As early as 2010, experts had forecast the demise of rule-based detection [7, 36] and focused their research on statistical and machine-learning approaches – e.g., [17, 25, 29]. So far, however, the predicted death of rule-based approaches has not materialized. More than a decade later, the industry still predominantly relies on rules for generating the alerts that are the basis for SOC detection of attacks.

The craft of developing effective rules is, thus, critical to producing meaningful alerts and enabling detection, also for those solutions that rely on "end-of-pipe" ML. Yet, we lack understanding as to how practitioners in SOCs design rules for processing network activity and how an alert is established and evaluated as being effective in practice. Rule development and management take place with limited knowledge, such that alerts require investigation to determine if a threat exists on the network. Most rules never trigger any alert [35]. Of those that do, the overwhelming majority of the alerts are false positives. True positives are scarce and false negatives might remain hidden. Under these constraints, how do rule developers figure out what is a good rule versus a bad rule? How do they evaluate the quality of the overall ruleset? What practices do they follow in producing rulesets that are effective in detecting attacks?

In this paper, we present the first interview study focused on professionals who develop or revise rules used in network threat detection – as the input for SOC incident reporting systems, rather than how the analysts manage the outputs of these systems. We aim to answer the following questions: (1) What does the organizational ruleset management process look like?; (2) What are the main factors and success criteria in managing and evaluating NIDS rules and rulesets?; (3) How can security professionals improve rule management and network incident monitoring workflows to optimize SOC processes?

Between June 2020 – March 2022, we interviewed 17 professionals who have developed NIDS rules, and who either work at Managed Security Service Providers (MSSPs) providing network monitoring as a service, or at government agencies and firms that conduct their own network monitoring. Some of the rules they work with were manually developed in-house, some were commercially acquired, and some were shared within their community. While our interviewees have different roles, inside and outside SOCs, they all are mandated to develop or change the detection rules running in their production environment.

While rules are designed to various degrees of quality, we found that the true test of quality only emerges after the rules are deployed in production. Such evaluation is mostly based on total number of generated alerts and false positives. Most quality criteria are optimizing for the SOC analysts' workload, rather than for detection – in other words, for reducing false positives rather than false negatives. The aspects of NIDS rules that determine their quality are balanced against each other to match the organizational processes and resources. In fact, SOC teams appear to work in a delicate equilibrium, operating on a knife's edge, where their resources are in balance with the workload and the services they offer to clients. They provide 'enough' security to keep clients content, and, given their specific organizational processes, more resources would not necessarily result in a 'better' product. In terms of necessary improvements, we were unable to identify common issues, as there was no consensus between respondents regarding any critical challenges; most conversations revolved around specific aspects of the employer's organizational processes. This emphasized the importance of understanding analysts' workflows. Contrary to recent work on SOCs, no interviewee mentioned automation to replace rules – that is, using AI or machine-learning approaches like anomaly detection – as an important direction that they believed would improve their work.

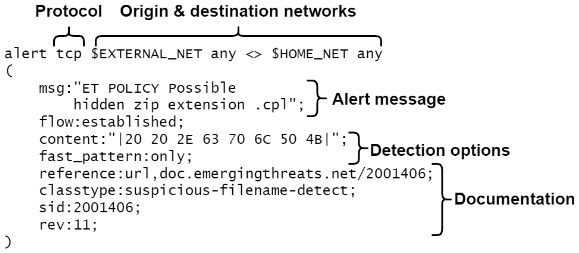
In sum, our main contributions are:

- We present the first interview study focused on professionals who make active use of – i.e., develop or revise – rules for network detection, and how they experience and perform their specific rule creation and management processes within their organizations;
- We find that there is no one trivial way in which to manage a SOC; many critical factors in the function of a SOC (such as rule specificity and false positive thresholds) must be balanced against other equally critical factors. This speaks to the value in understanding how these decisions are made in context, and providing evidence to inform decisions such as rule management and SOC resource allocation;
- SOCs balance resourcing and capabilities against customer expectations, wherein we evidence alternatives to a traditional approach to security of 'more is better'. Examples include security tailored to specific threats and environments instead of blanket coverage of the global threat landscape;
- External (e.g., commercial) rulesets create a negative externality for its users: while they are designed to provide broad threat coverage, users incur the costs of deactivating or fine-tuning individual rules, or manage the noise they often generate by them if they opt not to do so;
- There is barely any feedback loop in place for handling false negatives. Since SOCs cannot know when and how an incident will be missed, and proactive 'horizon scanning' of threats is an effort to compensate for this shortcoming;
- We present a number of recommendations for internal SOCs processes that can help with improving the overall effectiveness of SOC teams and the services that they provide, with a focus on making improvements at an earlier stage of the network monitoring and incident response process.

2 BACKGROUND

2.1 NIDS, NIPS, rules, and rulesets

Intrusion detection systems (IDSs) and **intrusion prevention systems** (IPSs) are categorized based on where they are placed and their methods of detection. There are network-based, host-based, and application-based systems. Furthermore, they can be either signature-based or anomaly-based depending on how they detect threats [16].



```
alert tcp $EXTERNAL_NET any <> $HOME_NET any
(
    msg:"ET POLICY Possible
        hidden zip extension .cpl";
    flow:established;
    content:"|20 20 2E 63 70 6C 50 4B|";
    fast_pattern:only;
    reference:url,doc.emergingthreats.net/2001406;
    classtype:suspicious-filename-detect;
    sid:2001406;
    rev:11;
)
```

Figure 1: Example of a rule used by Snort and Suricata NIDSs. Rule developers document the purpose of the rule and background information related to the threat in the alert message and “documentation” blocks, respectively. The “detection options” block is used by the system to detect the actual threat.

A network-based IDS (NIDS) is positioned at a key point in the network and scans all incoming network packets for signs of attacks. By comparing the content of packets with a predefined collection of patterns or events that are typical of known attacks, signature-based IDSs identify malicious activity. **Network-based IPSs** (NIPSs) are installed inline with a network, which gives it the ability to actively block traffic in case of malicious activity detection. Examples include Snort [8], Suricata [32], and Zeek [21].

These systems detect threats in network traffic using **rules** that tell the system precisely what to look for in network traffic that might be indicative of malicious activity. Using these rules, analysts can specify characteristics of network traffic such as origin and destination IP addresses, protocol used, packet payload content, among others. Additional documentation and metadata may be added to the rule to improve manageability. See Figure 1 for an example of such a rule. A collection of rules are grouped together into a **ruleset** and deployed on an IDS/IPS.

When traffic meets the conditions specified in a rule, it triggers an **alert**. This alert is then typically sent to an analyst to be validated as an actual attack or threat and of what kind. These analysts can be in different locations, but they typically are in SOCs. Some organizations operate their own SOC, some have it outsourced to an MSSP. A SOC is confronted with an alert flow that can quickly go into the thousands of alerts per day.

Rule developers are professionals with the mandate to write, change or remove rules. They might be analysts in the SOC who, based on the alerts they see, change the rule to provide more useful signals – i.e., make the rule more precise to avoid legitimate traffic triggering the alert. Other rule developers work outside the SOC. At MSSPs, for example, they might be located close to the threat intelligence department, to craft new rules in light of attacks observed elsewhere. And some rule developers work at companies that sell a whole set of detection rules as a service, e.g., Proofpoint’s ET ruleset [22] and Cisco Talos’ ruleset [9]. Often, these commercial rulesets consist of tens of thousands of rules [35]. Organizations, including MSSPs, subscribe to these rulesets in order to complement the rules they develop themselves. They can then mandate people in their own organization to assess, deploy, change or remove rules from these commercial sets, depending on what alerts they trigger.

2.2 Related Work

Shutock and Dietrich consider people, process, and technology in SOC management [26]. They enumerate current challenges in operating a SOC, including staffing issues, and outsourcing of SOC capabilities from within organizations to external MSSPs. The authors discuss platform consolidation against the converse view of SOCs maintaining their own internally-developed tools – we find that there are factors relating to business offerings and client needs which inform these choices, beyond the amount of effort involved.

Kokulu et al. [15] interviewed 18 SOC analysts and managers, with a view to both technical and non-technical challenges. An argument is that current SOC arrangements are insufficient to counter current threat levels and that they are failing to operate sufficiently well; further, the authors presume that there are practical problems within SOCs that can be identified and addressed to improve their capabilities. Among their findings are that false positives in malicious activity detection do not majorly impact SOC operations. The authors included budget-related questions for managers, Identified SOC issues were framed according to whether they were perceived by one of the analysts or managers, or both (mismatched or matched). Among matched issues were ‘poor quality reports and logs’, and ‘high false positive rate’. Of note is that managers in the Kokulu et al. study stated that they had ‘sufficient’ budget to operate their SOC, where the authors identify ‘insufficient budget’ for SOCs in terms of managers being very aware of having limited budget which can impact their capacity to enact change programs, training or travel.

Sundaramurthy et al. [31] apply Activity Theory in a long-term (3.5 year) observation of how SOC professionals work together to satisfy goals and objectives. This uncovered tensions and contradictions, specifically issues with tools and operating rules.

Alahmadi et al. [3] focus on false positive alert generation in SOCs, examining this persistent challenge through interviews with 20 practitioners. The authors arrive at five indicators of quality for alerts: reliable, explainable, analytical, contextual, and transferable. An approach to improving alert quality is taken as opposed to a general focus elsewhere to find tools to reduce the volume of alarms.

We agree with Alahmadi et al. [3] that improving alert quality is urgent and critical to better detection and SOC performance. However, their call also underlines that so far, the prior work has focused on how to handle the flow: helping analysts to triage and validate the voluminous influx of alerts into false positives, true positives, and more fine-grained distinctions. It has not investigated how that flow is generated, namely via detection rules. To improve the quality of alerts means improving the quality of the rules that generate these alerts. How professionals aim to do that is the focus of our study.

Basyurt et al. [4] conduct interviews with nine SOC practitioners to uncover challenges that SOCs face when collecting and analyzing data, as well as communication cyber situations. The challenges that the authors identify are of a technical nature, such as the collection and compilation of data sources and trustworthiness assessments of information, and suggest the development of a tool that is able to automate those tasks. As opposed to Basyurt et al., we lay the focus more on organizational challenges.

3 METHODOLOGY

Here we detail the structure of our interview study, along with our recruitment activities and approach to analysis.

3.1 Study design

Our overarching research question is, "How do security professionals manage network incident monitoring processes to achieve security?" We addressed this through interviews with professionals. Interviews have been used in prior studies to understand challenges around management of provisioned security (e.g., [15]).

Our interviews were structured to capture the following information, as also detailed in the question set itself (see Appendix 2):

(1) Participant details, including their role, organization, and qualities of the services they provide;
(2) Analyst's organization services and workflows, focusing on how detection rules are constructed and managed on a regular, day-to-day basis;
(3) The specific processes analysts use for ruleset evaluation;
(4) The management of rules, including collaboration with others in the organization, and related responsibilities for assessment and corroboration of evidence around rule-related decisions;
(5) How analysts meet objectives in practice, including views on improvement.

At the beginning of the study, we performed pilot interviews in order to test our interview protocol. Minor adaptations were made to the phrasing of some questions. An additional sample rule for evaluation was also added.

3.2 Recruitment and participants

We recruited professionals to participate in interviews from a range of different organizations providing managed security services (similar to [3]). This allowed for the comparison of working practices and decision-making around rulesets and their management, etc.

The initial seven participants from Org1 (see Table 1) were recruited through snowballing—the process of asking from each participant a short list of names of other people within the company's NIDS rule development and management teams whom they believe to be relevant to this study. This process was halted when the only names we received were professionals whom we had already interviewed before and two participants who did not respond to our invitations. After the initial batch of participants, we encountered similar recruitment challenges as Alahmadi et al. [3], with it being challenging to recruit from a profession wherein the occupation requires individuals to almost always be active or available for work-related activities.

Due to the hour to 1.5-hour duration of this interview, and the time constraints and daily task requirements that analysts work with, we were unable to find many more participants at the analyst level. Instead, the majority of the participants in this study, after the initial seven, are security professionals at a more senior level.

We leveraged our research team's institutional relationships to contact the remaining 10 participants, either through direct emails or through open invitations on LinkedIn. The prerequisite for participants was that they must be involved in the processes surrounding the creation or management of NIDS/NIPS rules, be they analysts or managers.

The participants in this study come from nine different organizations across five different sectors. Not only do they differ in sector, but also in size. See Table 1 for an overview of the participants. Although we cannot disclose specific characteristics of these organizations for reasons of confidentiality, their sector may provide an indication of the scale of the networks that they manage.

3.3 Ethics

Before starting our research, we followed the ethics approval and research data management procedure outlined by our institution. Data Management Plan has been approved for this study to ensure accountability, transparency, and compliance. We also followed the principles of the Menlo Report of ethics for ICT Research [14] during the study. This included respondents explicitly consenting to the recording, transcription of the interview, and to the usage of quotes, as well as being informed about their options as to their participation in the study. We minimized the risks of data leaks by pseudonymizing all data gathered during the interviews. The quotes have been assessed by the team members regarding the risk of reverse identification and de-pseudonimization of research participants. The recordings were stored for the duration of this research on an encrypted hard drive and destroyed when it was no longer necessary to keep them. All answers were confidential and only available to the researchers involved in this project.

3.4 Data analysis

Interviews were transcribed, after which thematic analysis [5, 6] was conducted with the transcripts. This involved a process of qualitative coding – three coders of diverse backgrounds were involved in the codebook development, and regular codebook review meetings to arrive at a final codebook of cross-cutting themes that emerged from the interviews (and interviewees) themselves. For the quotations used, reverse identification checks were conducted by the researchers to safeguard the anonymity of research participants.

The thematic analysis resulted in the following themes emerging: (1) services offered by the organizations and associated workflows; (2) rule evaluation; (3) ruleset evaluation; (4) internal and external collaboration processes; and (5) desired points of improvement. The subsections in the next section (Results) are arranged according to these themes, elaborating on the views that emerged from the interviews according to these cross-cutting areas. We present the prominent codes under each code theme, representing the strongest points of discussion across our participant group.

4 RESULTS

The interviews revolved around four main topics: (1) the different workflows within the organization that make up its incident monitoring and response services, (2) the different processes and objectives that surround the management of NIDS rules and rulesets, (3) how the participants collaborate with peers and clients to carry out said processes and objectives, and (4) points of (dis)satisfaction regarding work processes expressed by the participants.

In the sections below, we specifically use the term "alert" to refer to the notifications generated by NIDS/NIPS rules. This is in

Table 1: Overview of all participants, their corresponding organization and business sector, and their role and experience within that organization.

ID	Org. ID	Sector	Role	Experience	ID	Org. ID	Sector	Role	Experience
P1	Org1	Cybersecurity	Analyst	4 years	P10	Org4	Cybersecurity	Senior security researcher	16 years
P2	Org1	Cybersecurity	Forensics	3 years	P11	Org5	Shipping	SOC manager	7 years
P3	Org1	Cybersecurity	Analyst	3 years	P12	Org6	Telecom	Senior analyst	8 years
P4	Org1	Cybersecurity	Senior security expert	14 years	P13	Org7	Consultancy	SOC manager	4 years
P5	Org1	Cybersecurity	Incident response	6 years	P14	Org8	Government	SOC manager	15 years
P6	Org1	Cybersecurity	Incident response	6 years	P15	Org8	Government	Project architect	19 years
P7	Org1	Cybersecurity	Analyst	6 months	P16	Org9	Consultancy	Senior analyst	15 years
P8	Org2	Cybersecurity	Analyst	17 years	P17	Org9	Consultancy	Analyst	1.5 years
P9	Org3	Government	Analyst	5 years					

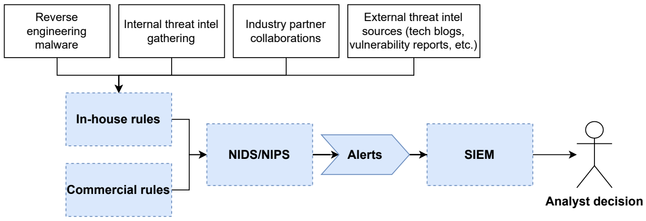

Reverse engineering malware
Internal threat intel gathering
Industry partner collaborations
External threat intel sources (tech blogs, vulnerability reports, etc.)
In-house rules
Commercial rules
NIDS/NIPS
Alerts
SIEM
Analyst decision

Figure 2: Simplified version of the NIDS/NIPS pipeline employed by organizations. In-house and/or commercial rules are installed on an NIDS or NIPS, which produces alerts that are used as input by SIEM (tools). Analysts then make decisions regarding detected threats based on the output of the SIEM tools. Illustrated are also the different data sources that organizations use when creating their in-house rulesets.

contrast to the term "alarm", which is used by related literature to refer to the notifications generated by SIEM (security information and event management) tools "as a result of the correlation of multiple alerts" [3]. This is illustrated in Figure 2: the NIDS/NIPS generates alerts, which are used by the SIEM as input.

4.1 Organization services and workflow

To understand the culture and workflows that govern the SOC, we first set out to identify the different aspects that make up the security services each SOC offers. This section will discuss the results that describe initial client interactions and on-boarding, ruleset management and development practices, and type of network monitoring offered by the SOC (i.e. NIDS or NIPS).

4.1.1 Client on-boarding. All organizations but one mention client on-boarding procedures, whereby the organization gathers information about a client's network before active network monitoring begins. This allows SOCs to calibrate the network sensor output to the resources available to the SOC and familiarize themselves with the client network. It can consist of in-person meetings and preliminary alert processing, which includes setting up the network sensor infrastructure, and analyzing alerts without performing direct incident response. Gathered information can include (sub)network descriptions, most valuable assets, and non-standard software running within the network. The exception to this is the shipping organization Org5, which does not monitor its clients' networks; instead, it monitors its own network, which it makes available to all of its clients. However, Org5 does ensure that the NIDS/NIPS rules that monitor their network do not interfere with the functionality of client software interacting with the network.

Participants mentioned the lack of a formal time limit set for this on-boarding procedure, although they all indicate a usual duration of several weeks. Seeing as the time limit for this phase is not formally defined, it likely varies from client to client, and it is up to the analysts to determine when the network sensors and rules are properly configured to the client network.

Notable on-boarding procedures exist at Org3 and Org9. Instead of having a single ruleset that is used for all client environments, these two organizations create a custom ruleset in conjunction with their clients that are completely tailored to the networks of said clients. Similarly, Org7 creates custom rulesets for each client, although with much less client involvement.

4.1.2 Ruleset management and development. Ruleset types can be split into the following: free external (i.e., community), paid external (i.e., commercial), and in-house rulesets. Both types of external rulesets can be obtained from threat intelligence vendors, and can contain tens of thousands of rules [35]. In-house rulesets vary from a few tens to a few thousands, depending on the size and resources of the organization, and its business model.

All organizations except Org3, Org4, and Org9, use some sort of external ruleset. Although there is considerable diversity regarding the usage and management of external rules, there certainly is a consensus around their quality. All organizations that use such rulesets criticize the poor quality of rules, which leads to a large amount of noise and many false positives (Org1, Org2, Org5, Org6, Org8). However, participants also admit the added value that these external rulesets provide: *"I do think you need to cover all types of malware, also like the new ones that are coming out. So that is what I think we have covered by [the external ruleset]" (P14).* Though the value of external rulesets is recognized, many participants also noted the noisiness that these rulesets tend to produce: *"We tend to find that they're [...] extremely noisy in our environment. Just little things can set them off. (P11)"*

P10 explains that it is due to this noisiness and tendency to produce many false positives that Org4 refrains from using such external rulesets. Since Org4 operates an inline NIPS, packets are dropped when they trigger a rule. Having a rule erroneously trigger on legitimate traffic then becomes a more significant problem. Floods of false positives will cause much legitimate traffic to be dropped, negatively affecting their clients' business continuity. Org5 operates a hybrid system, where rules can be set to either just alert when triggered or drop the potentially malicious packet. While this system is also set up inline to be able to block packets, this allows them to switch blocking rules to alerting rules in case of sub-optimal detection performance (e.g., too many false positives) so as to not affect business continuity in the same manner as Org4.

P8 mentions that free NIDS rulesets are not used at all or disabled immediately due to the large number of false positives it produces. P8 also claims that these rulesets *"generally look at non-conformity towards RFCs (Request for Comments),"* and since many applications on the Internet *"bend the RFCs"* to properly work, using such rulesets will easily flood a SOC with alerts. P8 is also disillusioned by the detection performance of commercial rulesets. Many rules contain IP addresses and domains of known C&C servers, for instance, that are out of date and are no longer used for malicious activities, and can therefore trigger many false positives. *"Those are generally, for 99% out of date, and they will trigger way too much false positives"* (P8). P8 continues: *"I will come across indicators of compromise easily three or four years old still in a rule set so, and from a paid service you'd expect more."* P11 shares a similar opinion: *"we preach about it a lot in our environment that the [IP and domain] indicators are worthless."* Thus, an organization that decides to employ external rulesets must design its operations such that dealing with that amount of noise and false positives is feasible for the technologies and the manpower available to it.

Due to this lack of quality, many organizations build their own in-house ruleset. All organizations except Org8 indicate the use of a custom ruleset crafted in-house. Such in-house rulesets can be built and tuned to the specific requirements and workflow practices of each organization.

In-house rulesets are used by Org1, Org5, and Org6, in a fashion complementary to external rulesets. As P4 states, the external rulesets—specifically the commercial rulesets—are made by a bigger team and thus generate more general coverage, allowing Org1 to focus their own rule development efforts on more severe and potentially targeted threats that are not covered by the external rulesets (which may focus more on, e.g., botnet traffic, untargeted commodity attacks). Org3, Org4, Org7, and Org9 forgo external rulesets completely to avoid the noise and false positives it can potentially produce. Instead, they rely solely on their own in-house ruleset, opting to delegate the security provided by the external rulesets to another layer in their defenses. Even though an in-house ruleset is currently not employed at Org8, both participants from Org8 (P14 and P15) acknowledge the importance of such a custom ruleset and share the plans of their organization to incorporate this aspect into their workflow in the near future: *"we found out that it's really important to separate things, separate the rulesets from the engines to be able to always [...] use your own rules"* (P15).

Org3 and Org9 create a custom-made ruleset for every client. The rulesets created by these two organizations are particularly small (under 100), especially compared to the rulesets employed by the rest of the organizations, and are developed to detect threats specific to the network and threat landscape of a client. The rest of the organizations employ custom rulesets with a volume that can run up into the thousands of rules (tens of thousands if external rulesets are counted). Different data sources are employed by organizations to create their in-house rulesets, as illustrated in Figure 2.

4.1.3 Network incident monitoring. All but one of the organizations uses an NIDS for network monitoring. Org4 is the only organization that employs an NIPS. P14 mentioned that Org8 had recently made the switch from running NIPSs to NIDSs due to issues with latency (as the system is set up inline with the network) and false positives dropping legitimate traffic. As a result, their rule management procedures and evaluation criteria needed to be adapted to this new form of network monitoring. This is described in Section 4.2.

Table 2: Criteria used by participants to determine the quality of rules.

Rule quality criteria	Participants	Percentage
Rule design		
Rule specificity	17	100%
Syntax and structure	6	35%
Dislike of domain and IP blocklists within a rule	4	24%
Usage of the fast_pattern keyword in rules whenever possible	4	24%
Specifying origin and destination networks	3	18%
Rule performance		
Number of total alerts triggered by a rule	17	100%
Number of false positives triggered by a rule	16	94%
Danger of resource intensive rule	4	24%
False negatives	3	18%
Ruleset evaluation		
Volume and resource intensity	17	100%
Legacy rules	17	100%
Coverage, of which	15	88%
- threat coverage for a specific environment	7	41%
- overall threat coverage	2	12%

4.2 Rule evaluation

Rules that are developed and put into operation require a level of quality that makes them effective at securing the network. As different organizations have different processes for securing networks, there cannot be a universal definition of what a quality rule is. Still, analysts from distinct organizations look for these attributes in much the same ways in order to assess the quality of rules. This quality control is done in two stages: pre-deployment, during rule design, and post-deployment, when the detection performance of a rule is evaluated after adding it to the production environment. This section will elaborate on the different factors that the participating organizations take into account when determining the quality of a rule. An overview of the different criteria can be found in Table 2. Our findings identify contributing factors ahead of the impacts that analysts have raised as challenges in the work of Agyepong et al. [1], specifically volume of alerts, false positives, manual and repetitive processes, and workloads.

4.2.1 Rule design. Much of what a SOC "sees" originates from the rules that are installed on NIDS systems. How these rules are designed largely determines the information that a SOC receives about potential threats. Organizations have no say in how external rules are designed; that is entirely within the purview of the rule vendors. As a result, assessment of external rules is limited to post-deployment evaluation of detection performance – options are limited to switching off rules rather than modifying external rules (as ruleset updates undo those modifications). For pre-deployment assessment, organizations can only focus on how they design their own in-house rules. Analysts and rule writers often go through research, experimentation, and multiple rounds of testing for every rule that they design. After conducting the interviews, we identified characteristics of a rule that influence its potential quality.

Specificity. The single most important aspect of rule design is to make the rule specific to the threat for which it is being designed; all participants make mention of the importance of this quality criterion during their respective interview (see Table 2). A rule that is specific to a particular threat will, for instance, try to detect the threat using some very unique characteristic that make it easy to distinguish from other types of traffic. By contrast, a rule that is more generic will use indicators that are less tied to a particular instance of a threat, in order to capture more variations of said threat. As such, a generic rule will likely trigger more often than a more specific one.

As there can be many exploits that target the same vulnerability, and malware families using these exploits can become quite large, creating a rule for every single instance quickly becomes infeasible: *"I'm going to write as few rules as possible to detect as many web shells as possible. Rather than creating [...] static checks for each and every one of them, [...], you try to find a pattern between all these web shells and see if you can make one rule to cover maybe 80% of all the web shells out there, and then see if you can make it work"* (P6). Therefore, these two aspects must be balanced when developing rules. P10 explains: *"you want to be as specific as you can to prevent false positives, but you don't want to be so specific that you're going to miss different variants of a piece of malware or something other than the POC (proof of concept)."* Create a rule that is too specific, and it might not trigger on different versions of the same threat; create a rule that is too generic, and you might overload your analysts and SOC: *"in practice [it can be] really difficult [...] to actually get really good detection rules that apply to the threats but do not generate false positives, and it's this balance that we need to keep up every day. Sometimes we get too strict and you don't see things, sometimes we write and update the rules, then we could get called by the 24/7 team that is getting a lot of noise in the monitoring system because the rules apply to a lot of traffic"* (P12).

Indeed, the aspect of specificity does not exist in a vacuum. It not only has an effect on the potential false positive alerts, but also on the number of alerts in general. This will be discussed in Section 4.2.2.

Rule syntax and structure. Six participants mention a rule's syntax and structure as an aspect that influences quality. These aspects determine the readability of a rule because it makes maintenance of the rules easier (P4, P10, and P11). Three participants (P2, P4, and P6) mention the importance of using variables to specify the origin and destination networks of the traffic being analyzed by the rule. Configuring a rule to be executed for any origin and destination IP will increase the resource intensity of the rule, since the rule will be applied to every packet entering and leaving the network. Specifying origin and destination networks or addresses for a rule ensures that the rule is applied only to relevant packets, thereby reducing its resource consumption. Additionally, four participants (P1, P4, P6, and P10) mention the usage of the `fast_pattern` keyword in rules whenever possible. With this keyword, an analyst can tell the system how to perform more efficient pattern matching for the potentially malicious content of the packet, which can also greatly increase the speed at which the rule is executed.

Blocking IPs and domains. Finally, four of the participants also displayed some animosity towards the use of domain and IP blocklists within the rule, not only due to the resource intensity of these types of blocklist rules, but also because they are difficult to maintain and quickly become out of date, making them easily prone to false positives.

Although looking at the rule itself can provide an indication of its quality, it by itself does not provide enough information for a definitive verdict: *"it's very hard to classify a rule as good or bad based on only the rule itself, so you really need to [...] have the rule tested on real network traffic"* (P1).

4.2.2 Rule performance. All participants share a similar opinion regarding characteristics that determine rule quality. Interestingly, the only element on which all participants unanimously agree is the significance of the number of alerts that a rule triggers. The number of false positives a rule triggers is a close second, with all but P15 making a mention of this second aspect. The degree to which the remaining characteristics are deemed important differs depending on the workflow practices of the organizations.

A rule that triggers an abundance of alerts, and thus potentially floods the SOC, was stated by all participants to be undesirable. Analysts need to process every alert that arrives at the SOC, and an alert flood interferes with the analysts' ability to perform their work effectively. Furthermore, a rule that starts producing a very large amount of alerts might also overload the SOC systems, causing downtime, leading to the SOC not being able to monitor their clients' networks at all. Indeed, most participants deem flooding of the SOC the most severe failure that can occur within their operation.

An example of when this can go wrong comes from a few participants from Org1: *"So [our rule vendor] created a rule. They found out that SMB1 traffic was now deprecated, and should be considered malicious because it can be exploited. So, basically, they said 'if SMB1 traffic occurs, then it should create an alert, because that is a really bad sign.' And in some way it is. However, there are a lot of companies who still use that legitimately. So basically what happened when they introduced that rule and we synced it into our production environment, [is that] it broke everything"* (P5). P4 elaborates: *"The flooding of the SOC is not only bad because you cannot handle the logs anymore [...], but also it could take the backend down because of the database not responding [...]"* Ways to safeguard against such failures include mandatory testing of rules on real network traffic before being placed in production, as well as syntax checks to ensure that rules only trigger a set amount of times within a certain period.

P3 describes the Org1's heuristic process of creating "good" rules that takes the amount of triggered alerts into account. Apart from triggering on true positives, with that being the ideal scenario, another gauge of rule quality is the *lack* of triggers while being tested on customer network traffic. It is difficult to test a rule's true positive rate, since there is no guarantee that it will detect specific malicious traffic during its testing period. And depending on the rarity of a threat, many of the alerts will turn out to be false positives. So instead of going through the difficulties of measuring a rule's true positive rate, rule developers can look at the *absence* of false positives to estimate the quality of a rule. This heuristic is clearly not a one-size-fits-all and includes implicit assumptions about the severity of the threat in question, since many (potentially less severe) threats are much more accessible and present in the wild, and will inevitably trigger these less severe rules when exposed to real network traffic (e.g., port scans). *"Another good indicator [of* *rule quality is the following.] We [push] the [rules in our] testing repository [to several of] our biggest customers [for] a reason: that is because if it doesn't trigger there, then the rule is probably well written. [...] So then if it doesn't trigger [there,] then we [...] make the assumption that, [though] it's not triggering, it's also not flooding. So the traffic is unique enough — or at least the indicators are — to put in production. And that's when we put it in production"* (P3).

It is not to say, however, that "good" rules will remain that way indefinitely. Due to the constant evolution of the Internet and the types of traffic that you can find in it, there remains the chance that even the most specific of rules will start detecting a new type of traffic as potentially malicious. It is then up to the SOC to handle such situations quickly and effectively to prevent further incidents.

The number of false positives triggered by a rule is closely tied to the previous aspect. A rule that triggers many false positives will inevitably tie up SOC resources unnecessarily, as these cases will still need to be evaluated individually by an analyst. There is a trade-off here, though, that was mentioned by several participants from Org1, namely that the acceptable proportion of false positives a rule triggers depends on the severity of the threat that the rule is trying to detect: *"we as writing experts can say, well, this is truly malicious behavior and we are willing to risk, let's say, a 50% false positive rate so we can catch these as soon as possible, because the weight of it will balance the false positives"* (P5). P13 also mentions such a trade-off that is sometimes made in cases of new discoveries of severe threats. In such instances, the severity of a threat and the necessity of being able to detect it can sometimes trump potential standards for noise and false positives: *"Generally, these rules are a little more cowboy fire from the hip, because we're trying to be fast, and we accept some additional noise"* (P13).

Org4 operates an NIPS that drops traffic when it triggers a rule, as opposed to an NIDS that raises an alert, and Org8 also did up until their recent switch to an NIDS. The case of Org8 gives us the opportunity to examine the changes in rule evaluation criteria caused by the shift in network monitoring approach. When operating an NIPS, P14, along with P10, who also operates an NIPS, describe preventing false positives as the most crucial aspect of rule development and management, since dropping legitimate traffic impacts business continuity directly. After switching to an NIDS, P14 states that they are still concerned about false positives, although for a different reason, namely the burden placed on the SOC and analyst fatigue.

Participants from Org1, Org4, Org6, and Org9 mention the danger of resource-intensive rules, which can potentially lead to sensor downtime. Thus, evaluating the resource intensity of a rule also plays a role in determining the quality of the rule, although not as significant a role as the previous aspects.

Only three participants (P5, P7, P14) mention false negatives as a main concern in regards to NIDSs. Given that the participating organizations are tasked with securing client networks, it is curious why failing to detect an intrusion is not considered more critical, especially since such SOC failures could be directly caused by an inadequate rule for which analysts and rule developers are responsible. P5 regards false negatives as most severe, while viewing false positives as "not that bad", in contrast to most of the other participants. This could be due to P5's role within the organization's incident response team: false negatives can quickly lead to security incidents, meaning that P5 would be directly influenced by such occurrences. This is in contrast to the majority of the other participants, who are responsible for SOC operations instead of incident response. P14 views both false negatives and false positives as equally severe, although depending on the point of view of the assessor: from a risk point of view, false negatives are deemed most severe, while from the point of view of security analysis workload, false positives can be seen as most severe. Finally, P7 is an analyst with six months of experience in the field. Ahmad et al. state that junior analysts have "less developed mental models," [2], which offers an explanation as to why their opinion does not match with the statements of their more senior peers. The issue of false negatives is also identified by Agyepong et al. as one of the challenges that analysts face, although to a significantly lesser degree [1].

4.3 Ruleset evaluation

Through the interviews, we identified three primary drivers of ruleset quality, namely threat landscape coverage, ruleset volume and resource intensity, and management of legacy rules (rules designed to detect older threats, software, or systems).

Coverage. The topic of ruleset evaluation that featured the most is the aspect of coverage. Out of all participants, 15 agree that coverage is a primary factor of ruleset quality. The type of coverage different organizations strive for, however, does differ, and can be generally split up into two broad categories: "overall threat coverage" and "threat coverage for a specific environment". The majority of the organizations adhere to the latter category, while only Org1 and Org5 explicitly strive for larger overall threat coverage. An interesting case is Org3, where P9 states that, ideally, a client's entire environment threat landscape be covered, but clients of Org3 themselves find such coverage unnecessary and are satisfied with more moderate monitoring.

Legacy rules. There also seems to be some overlap between this philosophy on threat coverage and the way organizations deal with "legacy" rules. These are rules that, for instance, detect older threats, old versions of software, or rules that simply do not trigger any alerts anymore. This is not to say, however, that such rules are considered "bad". These are simply rules that are potentially no longer relevant for a specific detection environment.

Five of the seven organizations that aim for threat coverage of a specific environment remove these legacy rules from operation if the client network no longer expressly requires it (e.g., all instances of a certain software are updated to a newer version). Management of legacy rules brings with it different costs that the organization can incur, one of which is the traffic processing costs in case legacy rules remain in the ruleset indefinitely, which will be touched upon later in this section. On the other hand, removing legacy rules may incur higher costs to the organization. These rules would have to be individually and manually assessed before deciding on whether to keep or remove them. This means that the larger a ruleset grows, the larger the cost will be that an organization will incur when performing this manual work. This could explain why the aforementioned five organizations are able to easily remove superfluous legacy rules: the environment-specific rulesets that they create are smaller and, therefore, more manageable.

The rest of the organizations choose to keep these legacy rules in place if the threat is deemed severe enough, lest they miss a security incident in the future due to the removal of that rule. Org4 presents an interesting case here, where they balance between providing environment-specific threat coverage and coverage of more severe threats that might not be as relevant to a network anymore. In general, it also seems that the size of the ruleset employed by organizations is not a determining factor when it comes to ruleset quality. Each organization operates with rulesets of different sizes, and their network monitoring services have been calibrated to work efficiently with the rulesets of that size; none have indicated desire for a different type of ruleset than they already use. The SOC teams in the organizations have calibrated their processes to the types of services that they offer. For instance, P8 says of Org3 that the analysts are capable of processing around 300 alerts per day, which is almost exactly what the SOC receives from the client networks using only a commercial ruleset for threat coverage. Adding more rules to increase coverage will likely not add any benefit for clients, and may in fact be detrimental to overall security, since the analysts will then become overburdened.

Volume and resource intensity. On a ruleset-wide level, resource intensity is primarily influenced by the number of rules contained within the ruleset. As was the case for the resource intensity of individual rules, a ruleset's resource intensity does not seem to be a major issue for most of the organizations. *"There's a correlation, obviously, between a hundred thousand rules vs. one thousand rules in terms of performance. Now, I can have a thousand poorly performing rules that perform worse than a hundred thousand good rules, so you can't say there's a causation there [...]. But if you start talking about engine specifics and how it's doing everything from the multi-pattern matching to the actual rule inspection, [...] all that makes a difference, so volume doesn't really factor into it as much,"* explains P10. However, Org6 and Org8 do explicitly acknowledge the point, the reason being that both organizations monitor networks where the data rate can reach 100 Gbps. At these high rates, the network sensors used by organizations can easily be overwhelmed if every packet has to be tested against a ruleset that is much too large. Other organizations do not monitor networks or network segments that generate that much traffic, or simply operate with a single ruleset that is maintained small enough that throughput degradation due to volume will never be an issue.

4.4 Collaboration

When asked about the way they worked together with immediate peers to achieve their workflow objectives, participants spoke mostly about two main elements: clients and peers. Client input/feedback was a theme that featured most often and consistently across all the interviews. Some respondents came up with informal examples of interactions, such as *"it's a good change that you've done"* (P16) and *"yeah, why didn't you put these rules into production?"* (P17), while the others described a detailed setup communication mechanism on case-by-case basis: *"For instance, we have an entire email inbox set up for end users [...] to email us directly and say, hey, there's something weird either going on my computer or something I received in an email. Do you mind checking it?"* (P11).

In terms of processes, client input is seen as necessary early on, during on-boarding (as reported in [18]) to familiarize with client systems, and also to fine-tune specific rules. Responding to needs extends to when clients are reacting to specific developments reported by news sources, e.g., *"you'll see the news reports, whatever hits the news, all clients come in [and say] 'oh, do you have coverage for SolarWinds?"'* (P10), and changes in the business environment, e.g., *"the government decides what we have to do"* (P14). Although there may be these inputs to the on-boarding and management of rules, generally engaging with clients constructively, it can nonetheless swing between imploring the client to *"trust us and let us do our thing. We'll show you the summaries of what we've done"* (P13), and having client representatives equally versed in the technology who might be *"a really up-to-date security officer, or maybe a network engineer who is curious if we cover [a particular threat] too"* (P12). Although Onwubiko et al. [18] also mention the importance of the client perspective, their work makes no specific mention of client input regarding the desired threat coverage.

The second element, collaboration with peers during rule design, development, and evaluation, is essential to safeguard the flexible, tailored approach that we have mentioned in the previous sections. *"We usually don't have an issue working together. To be honest, it's the opposite - it's actually working together has got us through situations [where] we had no idea what to do"* (P16). Respondents from different organizations describe having agile protocols in place to execute the "four eyes" principle[10] and to resolve issues within the team, as well as possibilities to solicit external expertise. Trust and freedom to both ask for clarification and to implement the solutions deemed necessary are commonplace. *"There's a mutual trust between us as coworkers that you know what you're doing; and if you didn't, then you should have asked"* (P3). Specialist expertise can be solicited through various means: workshops, chat groups, collaboration with colleagues from the other teams, industry-specific circles of trust, as well as through conducting own research. Time and capacity restrictions influence collaboration in different ways: prioritizing threats (*"Yes, so our first line does the initial filtering [...] So that goes from 300 alerts to five that go to my line and my colleague"* (P8)), and prioritizing analyst work-life balance (*"I want to say we will turn off the rule if it goes haywire, doesn't let us sleep at night"* (P17)). A notable improvement point is calling for even broader collaboration within the industry and security community: *"so if you have the second, third, thousandth opinion from a number of researchers and security professionals around the world working on different rules, that is very very useful. So I'm a big fan of these standardized languages for rules. And I think more people should use them in the industry"* (P16).

4.5 Improvement points

In general, we could not identify a common issue for the respondents. When specifically asked about the possible points for improvement, participants related mainly to different organizational aspects, in turn reflecting the diversity of their work processes. The improvement mentioned most often was better documentation of the different aspects of the rules (6 out of 17 interviews), such as actions that have been taken in the past or history of changes. Platforms or tools to systematize and access this information were mentioned as another desired improvement point: *"You don't have, like, one dashboard that says: this is what's happening right now, and these people should be cleaning up their rules"* (P6).

Other suggestions varied greatly and included better-organized client feedback loop, introducing an asynchronous aspect into peer review, better flow between deployment and testing, better quality of the rulesets, better and more testing, and more integrity checks. Automation was mentioned two times as an improvement component: automation of some "boring" processes, as well as extra quality assurance in an automated fashion to be able to employ more part-timers (i.e., a solution to a possible problem rather than a current one). This is mirrored in existing work [31]. As for the areas that could become better, one participant mentioned broader research to maintain detection quality, and another stressed the importance of keeping up with the current trends: *"[...] you start basing it on techniques and tactics and procedures that attackers use, [...], so it goes more from an artifact-based approach to behavior-based approach. And from that, I guess, you could move into even more, say, non-standard territory: statistical models and baselining; and you start going into data science and artificial intelligence stuff, right. So that's probably what, I would say, is an improvement that needs to happen across the board"* (P16).

Perhaps indicative of the optimized nature of SOCs is the diversity in the desired improvement points within SOCs. Participants seem generally satisfied with internal processes; there is no single shortcoming that is severe enough for all participants to point out and that needs immediate fixing. As systems become more optimized to specific work processes, they also introduce fragility into the system, as any change will upset the delicate balance necessary for all workflow optimizations to function properly. This is exemplified by an occurrence at Org1, where the addition of a SIEM tool to their existing pipeline that was meant to lighten SOC workload actually overloaded analysts by tasking them with using a tool they were unfamiliar with.

5 DISCUSSION

Many factors are considered in decisions around the use of rules in NIDS and NIPS systems. These are weighed against each other on a continuous basis in order to optimize the management of potential intrusion events. This phenomenon has been explored elsewhere [24] but only through construction of a conceptual SOC model representing analyst expertise and alert triage times, rather than direct empirical data gathered from practice (though the requirement to 'trade off' aspects against each other is highlighted). Many proposed solutions aim to optimize network monitoring and response workflows at a higher level (e.g., after the SIEM layer [3]), taking the inefficiencies of lower-level components (e.g., NIDS rulesets) as a given and immutable. Our results demonstrate that such inefficiencies are not immutable at all, as it is the proper balancing of the aforementioned factors that determines the quality of threat detection and SOC effectiveness. Through the balancing of these factors and trade-offs at a lower level, SOCs can significantly reduce the impact of such low-quality information on the higher levels.

5.1 Creating quality rules and rulesets

A recurring theme during our interviews with practitioners was the importance of using good rules, since that is what provides the SOC with security information and allows them to respond accordingly to threats. Creation, acquisition, and management of rules allow a SOC to both detect incidents and have the capacity to meaningfully respond to the incidents, supporting the *detecting-responding alignment* [27].

Kokulu et al. [15] investigate challenges surrounding the scalability of SOC operations, referring to *"complexity and the difficulty of integrating new technology"* into a SOC as the singular issue regarding scalability. Furthermore, while prior work has identified trade-offs made by SOCs during rule management, network monitoring, and incident response, they are limited to balancing false positives and false negatives [15], and trade-offs revolving around monetary costs of SOC analysts [24]. In this work we identify numerous additional aspects within SOC operations that are weighed and balanced against each other to effectively operate a SOC. Additionally, we also find that the structure and content of rules are determined not only by the rule design itself, but also by the externalities created by the characteristics and workflows of SOC teams. By examining these characteristics at the lower level of NIDS rules, we are adding to the understanding of how management practices and security practices influence each other.

In order to write good rules and create effective rulesets, a combination of experience and understanding is needed in order to find an appropriate balance between the multiple critical factors that influence rule and ruleset quality. The critical factors that were mentioned by participants are (1) specificity, (2) number of alerts and false positives, and (3) coverage. These separate factors are often interlinked, and so by altering one factor, another factor will be affected by that change, perhaps detrimentally. Choices about how to balance these factors are reflected in the security processes and approach of each organization. This indicates that balancing these factors is a non-trivial task that varies per environment. Such tasks thus require a significant degree of familiarity with the respective environment and its (human and computerized) components to make an appropriate decision. A particular configuration of these factors might work for one SOC, but possibly overwhelm another. This can also influence how new technologies are implemented into existing workflows, and their potential effectiveness within its respective environment. Slight differences in resource and workload capacity between SOCs can result in wildly distinct outcomes.

Specificity. Specificity in a rule interacts with all other rule management factors; increasing a rule's specificity to a threat makes it less prone to false positives, but also inherently decreases the threat coverage provided by the rule. In turn, this requires a larger volume of distinct rules to provide broad coverage. This is balanced against having more general rules, which, although they may increase the threat coverage to detect, e.g., variations of the same exploit or malware, it will also potentially cause the rule to produce more false positives.

The general consensus among our participants is that it is more desirable to have specific rules than generic rules, in order not to overwhelm the SOC – and the analyst team – with false positives. A study analyzing SOC and rule update logs found the same pattern: a general trend towards making rules more specific [35]. This may also explain why commercial rules are regarded as less than ideal by our interviewees, and why multiple organizations complement commercial rulesets with their own in-house ruleset, or choose not to use such rulesets at all.

Alerts and false positives. All participants remarked that minimizing alert floods and false positives is an objective to strive for. This ties into the previous point on specificity, which can reduce the risk of both. There are instances, however, when a larger amount of false positives is tolerated, namely when the severity of a threat is deemed high enough, or after the emergence of a new threat for which no coverage or understanding yet exists. Due to their time-sensitive nature, such pressing circumstances thus force detection to rely on underspecified rules that also trigger on benign events.

In the case of Org2, for instance, work processes are calibrated to instances where the used commercial ruleset works as intended (i.e., no floods or excessive false positives). During instances where it does not, the SOC disables the infringing rules, thereby reducing false positives and workload on the SOC, sacrificing threat coverage in the process.

Coverage. In Section 4 we explained how organizations deal with threat coverage in distinct ways, which can depend on the type of service that they provide to clients. One of the decisions an organization needs to make is whether to optimize for overall threat coverage, or to optimize threat coverage for a specific environment.

Among our participants, two particular themes emerged: firstly, retaining rules that rarely triggered but were seen as important to keep just in case, because the impacts associated with a true positive or incident absolutely had to be avoided. This relates to a concern of potential 'Black Swan'-style risks [33] (low probability but high impact). This also relates to non-trivial decisions about the retention of legacy rules.

Second was a phenomenon similar to the 'benign triggers' observed by Alahmadi et al. [3], where in essence, a rule was triggered which analysts had already determined a response for, i.e., action had already been taken prior to the alert and there was seen to be no need to be told about it again. A simple example is network policy rules that are tolerated due to client-specific use cases. Persistent triggering of such a rule was then because the rule had not been refined, seemingly because it was not seen as worthwhile to do so, and the fact that the rule still works as intended and, therefore, still provides the threat coverage it was designed for. While tolerated, such benign triggers can potentially provide additional information about a potential security incident in the future.

5.1.1 *Internal and external feedback loops.*

The interviews tell us that there is no strictly-defined feedback loop for quality in terms of threat detection on an organizational level, and that rule management relies in part on the intuition of the analyst. Analysts rely on their intuition to make appropriate decisions. Since experienced analysts perform better than more novice peers [24], this indicates that a feedback loop exists for the fine-tuning of this intuition. Decisions regarding network incident monitoring and response are very context-dependent, and analysts state that the process for developing this intuition consists of investigating and analyzing alerts and security incidents within the contexts of different networks and organizations. This iterative fine-tuning of analyst intuition is critical for the proper functioning of a SOC, since determining the value and validity of a rule is not a straightforward task.

In many cases, the only indications of rule quality are the number of alerts a rule generates (i.e., whether it floods the SOC with alerts), and the number of false positives it produces. This must be considered alongside the analysts' understanding of the threat(s) related to the events, and to what extent it is believed that those threats relate to the organization; this is reflected in analysts' proactive searching for emerging threats, which are then considered against the knowledge of the organization/client network and vulnerable systems.

All organizations that create their own rules test these rules on real network traffic before pushing them to the production environment in their clients' networks. Testing yields an indication of rule quality. Only when true positive detections arrive at the SOC and analysts compare that to the number of false positives, can they truly determine whether the rule is of high quality. The proactive 'horizon scanning' for new threats is an effort to compensate for this, i.e., to avoid a false negative – a successful attack – being the moment when a threat is discovered. Among our participants, there was a perception of being successful at catching all possible threats, which indicates that they are working effectively and protecting their organizations/clients. This does mean, however, that there rarely is a feedback loop of false negatives (breaches) back into the detection process, and in turn, a lack of experience with false negatives. Detection capabilities then evolve based on a mixture of incidents 'elsewhere' that are communicated out in the professional community or via news, as well as the aforementioned 'horizon scanning' for new threats, and problem-solving as to how those threats may relate to managed environments. The feedback loop consists of targeted rule edits and additions to provide coverage for the threat(s) that contributed to the breach. Since there is no way for the SOC to know when an incident will be missed, this is the only action SOCs can take for rule management.

Prior work [34] has studied this 'horizon scanning' and the use of publicly available data in the creation of rules. While this falls out of scope for this work, future work could further examine how reliance on this data affects SOC workflows and eventual network security, especially since analysts have stated that such open-source threat research is integral to the creation of in-house rules.

5.1.2 *External rulesets.*

Our participants regarded commercial rulesets as akin to 'starter packs'. Whether these rulesets are useful for the organization or not depends on the manner in which they provide their security services. Akin to the concerns about rare, high-impact events, external rulesets were regarded as useful for standard threat landscape coverage, containing the low-hanging fruit of threats that smaller teams of rule developers and analysts cannot create themselves due to time and resource constraints. Additionally, since they were unanimously regarded as "noisy," organizations need to have the technology and resources to be able to manage all the noise that they create. There is then a negative externality created by these external rulesets – their intention is to provide broad and comprehensive coverage, as a signal to customers of the quality of their rulesets as a product. However, this does not consider the amount of 'noise' they generate for those using the rulesets, who must then choose whether to selectively switch off or abandon the rules, as the cost of fine-tuning each rule to be less noisy was regarded as simply too great by our participants.

5.1.3 Intrusion detection vs prevention. Through the interviews, we also discovered that rule quality assessment differs depending on the type of system that is being used. Three of the nine organizations considered in this study operate an NIPS, or have done so in the recent past, and all state false positives as the greatest concern in this context, since traffic is immediately dropped. Consequently, the balance between making a rule specific or generic tilts more towards making rules specific, in order to reduce the risk of false positives. The fact that false positives remain a major concern is a strong indicator of why most organizations opt for an NIDS. Client business continuity is a top priority for all organizations, which is why even Org5 uses a hybrid NIDS/NIPS where most rules drop traffic, while some rules that are prone to false positives are modified to only produce alerts when triggered, instead of dropping the infringing packet.

The case of NIPSs dropping packets directly after triggering a rule highlights the importance of start-of-pipe improvements (see Figure 2). Taking such drastic action beforehand, and attempting to correct the issue in the SOC afterwards is effective for neither the SOC nor the client, since the consequences of such misconfigurations have already carried out their effect on the client network, potentially, business continuity.

5.2 Wider implications

From the interviews, we can ascertain that all organizations seem to be satisfied with how their network monitoring processes are set up. No participant suggests that additional resources are necessary to improve internal processes.

Where Kokulu et al. [15] mention that SOCs have insufficient budget to accommodate, e.g., one-off costs such as travel, here we find our participants in general agreement that they have sufficient budget to operate their SOC and, crucially, to meet the needs of their clients. This, however, may be a consequence of action to limit 'floods' of false positives or 'noise' created by general (external) rules, so that alerts are manageable within the workload of the analysts. It is possible that budget issues are not reported because SOCs are matching their workload to their work capacity.

Current literature claims that signature-based NIDSs are becoming more antiquated by the day [7, 36], and should therefore be replaced by ML-based systems. However, ML-based are often positioned as being a solution much later in the process of protecting a network; here we see that the rules used to first detect events must be mastered, and that ML-based systems cannot necessarily build a picture of the system in retrospect so late in the process. ML-based systems are not unaffected by the problem of false positives [3].

It is also often claimed that signature-based NIDSs cannot keep up with the fast evolution of the threat landscape, are unsuitable for the detection of new threats, and can therefore create security risks for the organization that employs such systems. Not a single participant mentions this as a potential drawback of their signature-based systems. Even Org7, which is actively and explicitly investing and focusing on machine learning-based approach, nevertheless finds most clients subscribing to the signature-based service, and tends to use their ML approach in environments that are more standardized, such as point of sale systems.

However, ML was seen as having a potential role in the protection of systems, and it holds the promise of reducing the alert volume to be investigated. Moreover, explainability of alerts is mentioned by Alahmadi et al. as crucial for the efficient functioning of a SOC, and current implementation of AI and ML models largely remain opaque [3], meaning that issues regarding explainability will only get amplified in case of the implementation of these systems.

5.3 Recommendations

From this study we have identified a preliminary set of recommendations:

Rule specificity and reduction of false positives. Consistent with findings from Agyepong et al. [1], yet contrary to the work by Kokulu et al. [15], we find that false positives are a major concern for SOC teams, primarily due to workload issues. SOCs often take false positives as an indication that a rule needs to be made more specific to the threat it is trying to detect. Therefore, a straightforward recommendation is to make rules more targeted and specific. Ideally, though, this adjustment should be assessed on an individual basis. From our interviews, we learned that false positives are often tolerated to some extent if the threat being detected is novel or severe enough. This indicates that driving down false positives, while an important aspect of SOC management, is not a cure-all. As stated in our Results, making a rule more specific will also reduce threat coverage, and thereby also losing information about any potential malicious traffic inside a network. And as explained by P6, generic rules are more useful in cases where there is a preponderance of similar malware that can be easily covered by a single rule.

Still, reducing the amount of false positives a rule generates will make the rule more reliable, and subsequent alerts generated by that rule more trustworthy and actionable, thereby also reducing the potential workload for SOC analysts. Proposed solutions to current workload issues include ML systems that are designed to provide analysts with actionable tasks by filtering out false positive noise from already inherently noisy alerts [3]. Even though addressing the issues of false positives at the end-of-pipe stage may improve workload difficulties for a SOC, this is treating the symptom and not the cause. Furthermore, attempting to improve workflows by adding additional technologies to the network monitoring pipeline may very well upset the delicate balance within a SOC, since our participants indicated that they were generally satisfied with their workflows, and comfortable with using the tools that they are currently familiar with.

Feedback loops. Just as it is labor-intensive to create a large collection of specific rules to catch all variations in a malware family, so, too, is it to build every single exception into rules to filter out every potential false positive. This is in contrast to our finding regarding making rules more specific and is an important counter to an unquestioning implementation of such a policy: specificity is a goal to strive for, but it is too effortful to write specific exceptions for every corner case. This all depends on the rule, the threat, and the situation itself, meaning that there is no magic bullet that will work for every case. What is clear is that if a SOC wants to increase the quality of the rules, one way to accomplish that goal is to have much more feedback from the detection process in the rule development process. In practice, this feedback is very limited, often only occurring in cases of (false positive) floods. Including more feedback from additional steps in the rule evaluation processes, such as a rule's testing phase, or when rules stop producing alerts, can yield analysts with valuable information regarding rule quality. Additionally, proactively implementing discoveries within the wider security community is a process that needs to be exploited further.

Creation of quality rules mainly relies on analyst intuition, which, in turn, can be developed by means of these feedback loops. Through the inclusion of novice analysts to a greater extent within these feedback loops, organizations can help cultivate such necessary level of experience.

Document intuition and tacit knowledge. Finally, another factor closely related to aspects of internal collaboration is the aspect of tacit knowledge, something also identified in work by Agyepong et al. [1], where they recommend the creation of playbooks and distribution of documentation about SOC processes that less experienced analysts can draw knowledge from. In addition to this, we recommend the setting up of well-defined systems of collaboration within teams using rulesets (whether they are within a SOC or elsewhere), such as developing rules in a pair programming fashion, or under the four eyes principle [10], and peer review sessions of developed rules. This will allow for a more fluent exchange of tacit knowledge from more experienced analysts to the more junior ones, leading to more effective use of rulesets.

6 CONCLUSION

In this paper we aimed to shed light upon the processes that surround the development and acquisition of rules for network detection. We found that there are a number of critical factors, such as rule specificity and total number of alerts and false positives, that dictate the manner in which rules are managed, and that there was significant consensus regarding the importance of these factors when they are used to determine the quality of rules and rulesets. These factors are weighed against each other in different ways by different organizations and carefully calibrated to the organization's network monitoring practices. Previous work has aimed at improving SOC effectiveness through different means, many of which fail to take the aforementioned factors into account, instead opting for solutions that make SOC data easier to deal with by automatically filtering out noise. We argue that such solutions are sub-optimal, and we presented a number of concrete recommendations that address these factors at the earlier stages of the network monitoring and incident response pipeline. With this, we propose a path forward that aims not to treat the symptoms, but to address a root cause of potential SOC ineffectiveness while leveraging a SOC's current resources and technologies.

ACKNOWLEDGMENTS

This work was supported by the Air Force Research Laboratory (AFRL) under agreement number FA8750-19-1-0152 and by the European Commission through the H2020 program in project CyberSec4Europe (Grant No.830929). The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government. We would like to express our gratitude to the seventeen participants of this study, without whom this research would not have been possible. Their invaluable insights, time, and willingness to share their experiences have greatly enriched our understanding and have been instrumental in shaping the outcomes of this work. Lastly, we thank B. Dacier for his contributions to the design of the interview protocol and substantial assistance conducting a large number of interviews.

REFERENCES

[1] Enoch Agyepong, Yulia Cherdantseva, Philipp Reinecke, and Pete Burnap. 2020. Challenges and performance metrics for security operations center analysts: a systematic review. *Journal of Cyber Security Technology* 4, 3 (2020), 125–152.

[2] Atif Ahmad, Sean B Maynard, Kevin C Desouza, James Kotsias, Monica T Whitty, and Richard L Baskerville. 2021. How can organizations develop situation awareness for incident response: A case study of management practice. *Computers & Security* 101 (2021), 102122.

[3] Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. 2022. 99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms. In *Proceedings of the 31st USENIX Security Symposium (USENIX Security)*. USENIX Association, Boston, MA, 2783–2800. https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi

[4] Ali Sercan Basyurt, Jennifer Fromm, Philipp Kuehn, Marc-André Kaufhold, and Milad Mirbabaie. 2022. Help Wanted-Challenges in Data Collection, Analysis and Communication of Cyber Threats in Security Operation Centers. (2022).

[5] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[6] Virginia Braun and Victoria Clarke. 2021. One size fits all? What counts as quality practice in (reflexive) thematic analysis? *Qualitative research in psychology* 18, 3 (2021), 328–352.

[7] Bricata. 2021. IDS is Dead! Long Live IDS! An Analyst Prediction from 2003 Remains Relevant. https://bricata.com/blog/ids-is-dead/

[8] Cisco. 2021. Snort - Network Intrusion Detection & Prevention System. https://www.snort.org/

[9] Cisco. 2021. Talos - Author of the Official Snort Rule Sets. https://www.snort.org/talos

[10] European Commission. 2021. Four eyes principle | CROS. https://ec.europa.eu/eurostat/cros/content/four-eyes-principle_en

[11] Chris Crowley and John Pescatore. 2019. Common and best practices for security operations centers: Results of the 2019 SOC survey. *SANS, Bethesda, MD, USA, Tech. Rep* (2019).

[12] Nitika Gupta, Issa Traore, and Paulo Magella Faria de Quinan. 2019. Automated Event Prioritization for Security Operation Center using Deep Learning. In *2019 IEEE International Conference on Big Data (Big Data)*. 5864–5872. https://doi.org/10.1109/BigData47090.2019.9006073

[13] Arthur S. Jacobs, Roman Beltiukov, Walter Willinger, Ronaldo A. Ferreira, Arpit Gupta, and Lisandro Z. Granville. 2022. AI/ML for Network Security: The Emperor Has No Clothes. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 1537–1551. https://doi.org/10.1145/3548606.3560609

[14] Erin Kenneally and David Dittrich. 2012. The Menlo Report: Ethical principles guiding information and communication technology research. *Available at SSRN 2445102* (2012).

[15] Faris Bugra Kokulu, Ananta Soneji, Tiffany Bao, Yan Shoshitaishvili, Ziming Zhao, Adam Doupé, and Gail-Joon Ahn. 2019. 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)*. 1955–1970.

[16] P. Mell. 2003. Understanding Intrusion Detection Systems. In *IS Management Handbook*. Auerbach Publications, 409–418.

[17] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai. 2018. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. (2018).

[18] Cyril Onwubiko and Karim Ouazzane. 2019. Cyber onboarding is 'broken'. In *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*. IEEE, 1–13.

[19] Alina Oprea, Zhou Li, Robin Norris, and Kevin Bowers. 2018. MADE: Security Analytics for Enterprise Threat Detection. In *Proceedings of the 34th Annual Computer Security Applications Conference* (San Juan, PR, USA) *(ACSAC '18)*. Association for Computing Machinery, New York, NY, USA, 124–136. https://doi.org/10.1145/3274694.3274710

[20] Ponemon Institute LLC. 2019. Improving the Effectiveness of the Security Operations Center. http://www.surfline.com/surf-news/maldives-surf-access-controversy-update_75296/

[21] The Zeek Project. 2020. The Zeek Network Security Monitor. https://zeek.org/

[22] Proofpoint. 2021. Emerging Threats Pro Ruleset | Proofpoint. https://www.proofpoint.com/us/threat-insight/et-pro-ruleset

[23] Hillary Sanders and Joshua Saxe. 2017. Garbage in, garbage out: how purportedly great ML models can be screwed up by bad data. *Proceedings of Blackhat* 2017 (2017).

[24] Ankit Shah, Rajesh Ganesan, Sushil Jajodia, and Hasan Cam. 2018. Understanding tradeoffs between throughput, quality, and cost of alert analysis in a CSOC. *IEEE Transactions on Information Forensics and Security* 14, 5 (2018), 1155–1170.

[25] N Shone, T.N. Ngoc, V.D. Phai, and Q. Shi. 2018. A Deep Learning Approach to Network Intrusion Detection. *IEEE Trans. on Emerging Topics in Computational Intelligence (TETCI)* 2, 1 (2018), 41–50.

[26] Matthew Shutock and Glenn Dietrich. 2022. Security Operations Centers: A Holistic View on Problems and Solutions. In *Proceedings of the 55th Hawaii International Conference on System Sciences*.

[27] Sessika Siregar and Kuo-Chung Chang. 2019. Cybersecurity agility: antecedents and effects on security incident management effectiveness. In *23rd Pacific Asia Conference on Information Systems (PACIS 2019)*. 8–12.

[28] Awalin Sopan, Matthew Berninger, Murali Mulakaluri, and Raj Katakam. 2018. Building a Machine Learning Model for the SOC, by the Input from the SOC, and Analyzing it for the SOC. In *2018 IEEE Symposium on Visualization for Cyber Security (VizSec)*. 1–8. https://doi.org/10.1109/VIZSEC.2018.8709231

[29] N. Srivastav and R.K. Challa. 2013. Novel Intrusion Detection System Integrating Layered Framework with Neural Network. In *Proceedings of the 2013 IEEE Advance Computing Conference (IACC)*. IEEE, IEEE, 682–689.

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

[31] Sathya Chandran Sundaramurthy, John McHugh, Xinming Ou, Michael Wesch, Alexandru G Bardas, and S Raj Rajagopalan. 2016. 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)*. 237–251.

[32] Suricata. 2021. Suricata | Open Source IDS / IPS / NSM engine. https://suricata-ids.org/

[33] Nassim Nicholas Taleb. 2007. *The black swan: The impact of the highly improbable.* Vol. 2. Random house.

[34] Ivo Vacas, Ibéria Medeiros, and Nuno Neves. 2018. Detecting Network Threats using OSINT Knowledge-Based IDS. In *2018 14th European Dependable Computing Conference (EDCC)*. 128–135. https://doi.org/10.1109/EDCC.2018.00031

[35] Mathew Vermeer, Michel van Eeten, and Carlos Gañán. 2022. Ruling the Rules: Quantifying the Evolution of Rulesets, Alerts and Incidents in Network Intrusion Detection. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security (AsiaCCS '22)*. 799–814.

[36] Giovanni Vigna. 2010. Network Intrusion Detection: Dead or Alive?. In *Proceedings of the 26th Annual Computer Security Applications Conference* (Austin, Texas, USA) *(ACSAC '10)*. Association for Computing Machinery, New York, NY, USA, 117–126. https://doi.org/10.1145/1920261.1920279

[37] Tian Wang, Chen Zhang, Zhigang Lu, Dan Du, and Yaopeng Han. 2019. Identifying Truly Suspicious Events and False Alarms Based on Alert Graph. In *2019 IEEE International Conference on Big Data (Big Data)*. 5929–5936. https://doi.org/10.1109/BigData47090.2019.9006555

APPENDIX 1: INTERVIEW STUDY PROTOCOL

1) Welcome
2) Short overview of the study
3) Explanation of the interview
4) Informed consent
5) Start interview
6) Debriefing

APPENDIX 2: INTERVIEW QUESTIONS

Workflows

1) Could you describe to me your workflow? a. Probe about routine and non-routine tasks
2) What do you see as the main objectives of your work? a. Probe on incentives that they have to reach this objective
3) Can you walk me through the process of the acquiring, creating, changing, and deactivating rules? a. Probe on when a rule is added into the sensor b. Probe on all the testing that is done before rules are added into the sensor
4) How many rules do you investigate every day? a. Probe on how they perceive this task (creative / procedure / workload) b. Do you have any tasks that you don't have enough time for? c. Probe rule evaluation

Management

5) How do you work together with other colleagues on making or changing rules? a. Probe on working together or separation of tasks in specific client's rulesets? b. Probe if they ever had a disagreement on certain rules
6) How do you seek additional information in order to assess a rule? a. Probe on who they asked, what advice they received. b. What kind data they were looking for.
7) Did someone ever follow up with you after you made or changed a rule?
8) In your experience, what is the most severe thing that could go wrong with the rulesets you are using? a. Probe on fear of FN b. What are potential consequences of a ruleset that isn't functioning properly c. Probe on the amount of risk that they perceive on missing TP d. Probe on the amount of FP and their perception and definition of a FP e. Probe on how likely they think consequences might happen
9) Have there been made any mistakes while adapting rulesets? a. Probe on how this came to light.
10) What procedures does the organization have on making or changing rules?
11) Who is responsible for the quality of the rules? a. Probe on differences between the responsibility of individual, senior, manager.
12) How is a client involved in the creation of rules? a. Probe on relationship with clients
13) Can you give an example of feedback that you received from clients?
14) In your opinion, what could be improved in the management of rules?

Objectives

15) In your opinion, what is a good ruleset? a. Probe on the influence of the volume of a ruleset
16) How do you optimize a ruleset as a whole?
17) What is the best ruleset that is achievable in practice?

Evaluating Rules

18) Can you walk me through the process on how you determine whether a rule is good or bad?
19) Can you give an example of a good rule? a. Probe on why this is a good rule
20) Can you give an example of a bad rule? a. Probe on why this is a bad rule
21) Which data do you use for evaluating rules? a. Probe on what they think is the most important data
22) Is there additional data that you would like to have?
23) What do you do when you have doubts on a rule?
24) How do you deal with rules that do not or no longer generate any alerts?
25) Is there anything else you would like to tell me that could benefit our research?
26) Live evaluation of sample rule #1.
27) How would you evaluate the previous rule if it is known to have generated 4000 false positive alerts in the last month?
28) Live evaluation of sample rule #2.
29) How would you evaluate the previous rule if it is known to have generated 20 true positives in last month?
30) Live evaluation of sample rule #3.
31) How would you evaluate the previous rule, taking its performance impact into account, if it generates a single true positive in a year?

Closing Demographics

32) What is your name?
33) What is your job title?
34) How old are you?
35) What is your educational level?
36) How many years have you been doing this work?
37) Do you know anyone else who we could interview for this research?

APPENDIX 3: INTERVIEW SAMPLE RULES

Sample rule 1

```
alert http $HOME_NET any -> $EXTERNAL_NET any (
  msg:"ET POLICY Vulnerable Java Version 1.8.x Detected";
  flow:established,to_server;
  content:" Java/1.8.0_"; http_user_agent;
  content:!"251"; within:3; http_user_agent;
  flowbits:set,ET.http.javaclient.vulnerable;
  threshold: type limit, count 2, seconds 300, track by_src;
  metadata: former_category POLICY;
  reference:url,www.oracle.com/technetwork/java/javase/8u-relnotes
        ↪ -2225394.html;
  classtype:bad-unknown;
  sid:2019401;
  rev:30;
  metadata:affected_product Java, attack_target Client_Endpoint,
        ↪ deployment Perimeter, deployment Internal,
        ↪ signature_severity Informational, created_at 2014_10_15,
        ↪  performance_impact Low, updated_at 2020_04_27;
)
```

Sample rule 2

```
alert tcp $HOME_NET any -> any any (
  msg:"ET EXPLOIT Possible OpenSSL? HeartBleed? Large HeartBeat?
        ↪ Response (Client Init Vuln Server)";
  flow:established,to_client;
  content:"|18 03|";depth:2;byte_test:1,<,4,2;
  flowbits:isset,ET.HB.Request.CI;
  flowbits:isnotset,ET.HB.Response.CI;
  flowbits:set,ET.HB.Response.CI;
  flowbits:unset,ET.HB.Request.CI;
  byte_test:2,>,150,3;
  threshold:type limit,track by_src,count 1,seconds 120;
  metadata: former_category CURRENT_EVENTS;
  reference:cve,2014-0160;
  reference:url,blog.inliniac.net/2014/04/08/detecting-openssl-
        ↪ heartbleed-with-suricata/;
  reference:url,heartbleed.com/;
  reference:url,blog.fox-it.com/2014/04/08/openssl-heartbleed-bug-
        ↪ live-blog/;
  classtype:bad-unknown;
  sid:2018377;
  rev:4;
  metadata:created_at 2014_04_09, updated_at 2014_04_09;
)
```

Sample rule 3

```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg: "ET TROJAN [
      ↪ PTsecurity] Tinba (Banking Trojan) Check-in";
  flow: established,to_server;
  content:!"Referer|3a|";
  http_header;
  content: "|0d0a0d0a|";
  depth: 2000;
  byte_extract: 2, 0, byte0, relative;
  byte_extract: 2, 0, byte1, relative;
  byte_test: 2, =, byte1, 6, relative;
  byte_test: 2, !=, byte1, 7, relative;
  byte_test: 2, =, byte1, 10, relative;
  byte_test: 2, !=, byte1, 11, relative;
  byte_test: 2, !=, byte1, 23, relative;
  byte_test: 2, !=, byte0, 25, relative;
  byte_test: 2, !=, byte1, 27, relative;
  byte_test: 2, =, byte0, 40, relative;
  byte_test: 2, =, byte1, 42, relative;
  byte_test: 2, =, byte0, 44, relative;
  byte_test: 2, =, byte1, 46, relative;
  byte_test: 2, =, byte0, 48, relative;
  byte_test: 2, =, byte1, 50, relative;
  content:!"|0000|";depth:30; http_client_body;
  content: "|0000|";offset:34;depth:2; http_client_body;
        ↪ fast_pattern;
  content: "|0000|";distance:2;within:2; http_client_body;
  content: "|0000|";distance:2;within:2; http_client_body;
  metadata: former_category TROJAN;
  reference:md5,be312fdb94f3a3c783332ea91ef00ebd;
  classtype:trojan-activity;
  sid:10003433;
  rev:1;
  metadata:affected_product
        ↪ Windows_XP_Vista_7_8_10_Server_32_64_Bit, attack_target
        ↪ Client_Endpoint, deployment Perimeter, tag Banker,
        ↪ signature_severity Major, created_at 2018_08_07,
        ↪ malware_family Tinba, performance_impact High;
)
```