# POSTER: SuriCap – A Measurement Platform to Study and Evaluate Intrusion Detection Rule Engineering

Koen T. W. Teuwen
k.t.w.teuwen@tue.nl
Eindhoven University of Technology
Eindhoven, The Netherlands

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

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

## Abstract

Organizations deploy Intrusion Detection Systems (IDSs) like Suricata to defend against threats. Although rulesets, rules, and the resulting alerts have been studied previously, little is known about the process by which rules are engineered thus far. We aim to address the previously mentioned gaps by studying how network intrusion detection rules are derived from incidents. To this end, we propose the SuriCap measurement platform and organize Jeopardy-style workshops in which participants compete to engineer Suricata rules. We collect a rich dataset consisting of over 364 rules from 28 participants. Preliminary results suggest our experimental design is viable and, together with the SuriCap measurement platform, can enable us to answer several research questions surrounding the engineering process of network intrusion detection rules.

## CCS Concepts

• **Security and privacy → Usability in security and privacy**; **Intrusion detection systems**; *Network security*.

## Keywords

Security Operations Center (SOC), Network Intrusion Detection System (NIDS), Network Intrusion Detection Rules

**ACM Reference Format:**

Koen T. W. Teuwen, Emmanuele Zambon, and Luca Allodi. 2025. POSTER: SuriCap – A Measurement Platform to Study and Evaluate Intrusion Detection Rule Engineering. In *ACM Asia Conference on Computer and Communications Security (ASIA CCS '25), August 25–29, 2025, Hanoi, Vietnam.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3708821.3735349

## 1 Introduction

Organizations may deploy Intrusion Detection Systems (IDSs) to protect against threats. Network-based IDS (NIDS) are commonly deployed to monitor network traffic. Suricata is a prevalent NIDS that has been the topic of previous research on Security Operations Centers (SOCs) [5–8] and is also leveraged in this work.

Modern SOCs still face various challenges, among which on a technical level, concerns about the specificity and coverage of the ruleset are frequently discussed [7]. Previous work has already investigated high false positive rates within SOCs [1], in relation to rulesets [8], and in relation to rules [6]. However, no understanding of how rules are engineered so far has been presented. The closest work to this investigated the evolution of rules in rulesets, but found that they rarely receive updates and provide no understanding of the process that led to the initial rule being included in the ruleset [8]. Previous work established designed principles that have the potential to increase the coverage and specificity of rules, but their application has thus far been limited to improving pre-existing rules [6].

We aim to address the aforementioned gaps by studying how NIDS rules are derived from incidents. To this end and inspired by previous work in related fields [2], we develop SuriCap and organize several Jeopardy-style workshops in which participants compete to engineer Suricata rules. We collect data on the steps taken in the workshop to acquire a rich dataset describing the process. Using this dataset, we characterize network rule engineering processes and relate these processes, the design principles for NIDS rules, the experience of rule engineers, coverage, and specificity, to gain a better understanding of rule engineering and improve the efficiency and effectiveness thereof. Concretely, we devise a methodology and platform to answer the following research questions (RQs):

- RQ1 How are network intrusion detection rules engineered given concrete samples of malicious and benign behavior?
- RQ2 What is the effect of the rule design principles on the rule engineering process?
- RQ3 How does the experience of rule engineers affect rule engineering?

## 2 Experiment Design

We devise a methodology supported by a measurement platform to answer the RQs presented in Section 1. The measurement platform collects data on Suricata rules while they are being engineered, providing us with the possibility of studying this process (RQ1). To measure the effect of the design principles [6] (RQ2), we randomly split the group into a treatment and a control group. The treatment group receives an additional instruction on the design principles prior to the activity, whereas the control group is only offered this instruction afterward. Moreover, to gain insight into how prior experience may aid rule engineers, we employ a questionnaire to measure relevant experiences of participants prior to providing them any instruction. During the activity, participants will be asked to engineer a Suricata rule for a specific scenario with the goal of detecting malicious traffic once whilst impervious to benign traffic. They can submit infinitely many rules to receive feedback to further refine their rules. During and after the activity, additional tests are run to gauge coverage and specificity beyond the network traffic captures (PCAPs) visible to the participants. Inspired by Capture The Flags (CTFs), we organize the activity in a Jeopardy-like [2]


*ASIA CCS '25, August 25–29, 2025, Hanoi, Vietnam*

ACM ISBN 979-8-4007-1410-8/2025/08
https://doi.org/10.1145/3708821.3735349

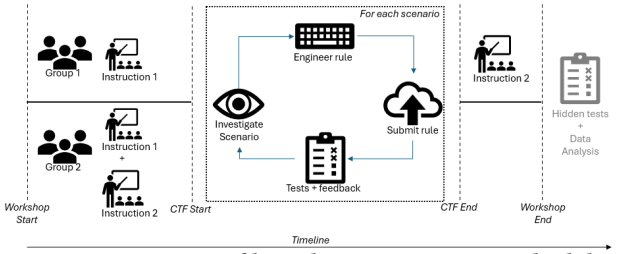


**Figure 1: An overview of how the measurement methodology is set up to answer the research questions.**

CTF where participants compete with each other fast-paced for the best coverage and specificity on and beyond the provided PCAPs. An overview of the methodology is also depicted in Figure 1.

*Scenario.* We focus on a single scenario: *PHP Information Disclosure*, which covers the unintended leakage of information describing web servers through the built-in `phpinfo` function from which adversaries can derive installed software, as well as configuration settings. This knowledge can be used to determine whether exploitable software versions or configurations are used. To this end, the SOC may choose to detect this disclosure of information, which is why the participants of the CTF were instructed to detect this behavior.

We manually collected various PCAPs from servers running different PHP versions where the disclosure was successful with and without the use of evasive methods, such as the inclusion of irrelevant query parameters. Similarly, we capture traffic from different servers where no information was disclosed to test for False Positives (FPs). Furthermore, we collect PCAPs by accessing web pages that can trigger FPs, such as those documenting the `phpinfo` function. Lastly, we include several PCAPs that contain only benign traffic from [4]. We only present participants with the successful disclosure corresponding to the oldest PHP version for which we collected data, and a PCAP containing benign traffic originating from a Windows 7 PC. The remaining 23 PCAPs are used as hidden tests.

*Ethical Considerations.* This research was carried out with ethical approval from our institution's ethical review board under the approval number ERB2024MCS34. We obtained explicit and informed consent from all subjects whose data are used in this study, and assured subjects that the study would not affect their study/work conditions in any way. Participation in the activity is non-obligatory and solely encouraged through the educational and entertaining aspects. Moreover, for participants of the activity, participation in the research is non-obligatory, and consent thereto can be withdrawn anytime. The experiment design is explained prior to the activity, and sample solutions are provided afterward. Measurement of user input occurs in a non-intrusive manner, whereby no collection takes place without explicit user input. After data collection took place and the data points were correlated, personal information was removed from the data so that researchers could not use those data to identify the individual people to whom the data are relating.

*Experience Questionnaire.* The intake questionnaire is designed to measure relevant experiences of participants that may affect rule engineering or experimental outcomes. Concretely, we devise questions to measure experience with topics relevant to the workshop to avoid reliance on self-assessment. The topics include: network protocols, offensive computer security, Wireshark, IDS, and Suricata. The questions regarding offensive computer security specifically relate to tactics like reconnaissance, initial access, and command and control. In addition, we collect demographics such as age and education level to describe the population and inquire about participants' experience with CTFs. A control question was included to ensure that the answers to the questionnaire are genuine and reliable.

*Preparatory Instructions.* To prepare participants for the activity, we provide them with an instruction that contains the essential information required, which is considerably general to prevent bias in experimental results. The instruction familiarizes participants with the required tools and the platform they will interact with. To this end, practical aspects of Wireshark [3] are covered whilst discussing commonly used features such as protocol recognition, display filters and following streams. Subsequently, the concept of IDS are generally introduced and Suricata is covered more in-depth by explaining the rule syntax and highlighting common functional keywords including `content`, `file.data`, `flow`, `flowbits`, and `threshold` as well as transformations, modifiers, and protocol-specific keywords for DNS, HTTP, and TLS. The existence of other options is emphasized with reference to the Suricata documentation. Moreover, we exemplify how a minimal rule containing all mandatory keywords can be constructed. This instruction also explains the scenario that users will tackle during the activity, highlighting why one would want to detect such a scenario and potentially relevant information without prescribing a characteristic that they should focus on. The general instruction ends with some practical details on the employed platform, the leaderboard scoring method, and the User Interface (UI) through a live demonstration. This general instruction takes approximately one hour. The additional instruction on the design principles discusses each principle similarly to [6] and comprises approximately 5 minutes. The instructions are made available to the participants in the form of a video lecture prior to the activity.

## 2.1 The SuriCap Measurement Platform

A measurement setup is required to observe and collect data about the rule engineering process to enable answering the RQs from Section 1. To this end, we propose SuriCap: a measurement platform to study and evaluate Suricata rules. SuriCap acts as a facade and decorator for Suricata to capture user input in the form of rules and automatically provides feedback to the user while hiding its CLI and configuration and simplifying and interpreting its output. Throughout the platform, various incentives encourage frequent rule submission to maximize data collection non-intrusively.

To bootstrap the rule engineering process, SuriCap uses the notion of a *scenario*, which represents a self-contained challenge to be solved by each user. Each scenario has at least one associated PCAP containing malicious behavior that the user should detect using a Suricata rule. Scenarios can contain additional instructions in the form of a textual description or a file (i.e., PDF) to provide relevant information. Furthermore, each scenario has the notion of one or more *tests*, which are automated to run after a submission to assert whether and how many alerts are raised on a PCAP and whether this is in line with the expected outcome of the test. Tests are used to evaluate the specificity and coverage of rules and need not be visible to users, offering a means to evaluate the generalizability of rules to previously unseen traffic during and after engineering of a rule.

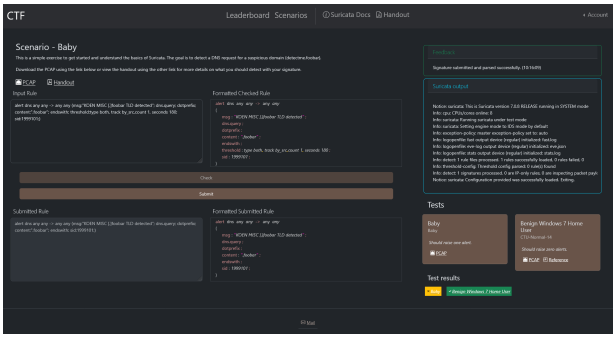


**Figure 2: A screenshot of the User Interface of SuriCap**

To incentivize frequent submissions of rules to enable data collection, the user receives swift feedback on every rule submission. Considering a single Suricata rule must syntactically consist of a single line, which easily exceeds 120 characters, obscuring the readability, we show a formatted rule splitting options to separate lines and coloring keywords and special characters. The platform also indicates whether Suricata was able to successfully load the rule. If a submitted rule is invalid, the user can view the detailed Suricata output to fix the syntax. When a user submits a rule, they will receive feedback indicating whether the malicious behavior is correctly detected. Additionally, they receive similar feedback indicating whether any FPs were triggered on benign network traffic. In addition to feedback, we also stimulate frequent submission by rewarding individuals who obtain high scores early in the case that the same score was obtained by others. Through the provision of these various types of feedback and the incorporation of time into the scoring method, we stimulate frequent submissions to collect more data points.

All elements aforementioned are combined in the UI, which has been designed such that it is possible to work on a scenario from a single page containing most relevant information. The UI for a scenario in SuriCap is visualized in Figure 2. Initially, many elements such as the formatted rules and Suricata output are hidden to not overwhelm users. Relevant aids, such as instruction materials and Suricata documentation, are easily accessible from the menu bar. As a user submits a rule, feedback appears. Notifications are given to notify the user of updated feedback, which may be otherwise difficult to notice in the case of incremental changes or similar feedback.

## 3 Preliminary Results and Discussion

We organized several CTF sessions to gather data from a total of 28 participants who together submitted 364 valid rules. From these valid rules, we consolidate 564 individual changes (e.g., adding or modifying a Suricata option) that describe the engineering process followed by the participants (RQ1). Combined with the 4450 test results for all valid rules, we can relate specific steps in the engineering process to performance implications, indicating rule quality.

Using the data collected, we can compare the control and treatment group as shown in Figure 3 (RQ2). The figure shows that participants in the treatment group, who received additional instruction in design principles, have a more consistent performance and typically outperform participants of the control group.

Using the intake questionnaire, we establish the experience of the participants in various areas as described in Section 2. We can regress over the aggregated experience score of the participants, as shown in Figure 4. Although the regression does not suggest a

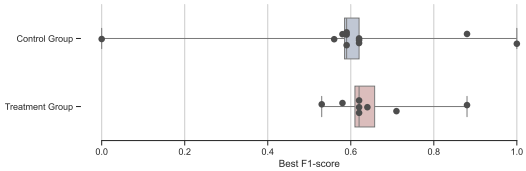


**Figure 3: Boxplot comparing between performance of participants with and without instruction on design principles**

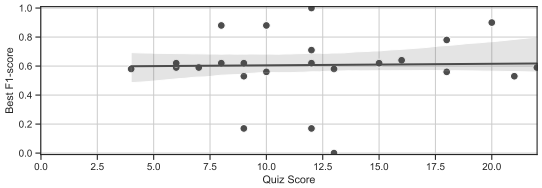


**Figure 4: Regression of a participant's performance over their experience according to the intake questionnaire.**

significant effect on participant performance, it exemplifies the potential of the data collection methodology (RQ3). Since experience in different areas may be relevant and experience in some areas need not necessarily be advantageous, we intend to separate the effects to discover in which ways experience is relevant to rule engineering.

The highlighted results above suggest that the experimental setup from Section 2 is viable and, together with the SuriCap measurement platform, can enable us to answer the RQs of Section 1. We intend to further extend our study by including additional scenarios covering different tactics and recruiting additional participants. Together with a more thorough analysis of the many collected data points, describing the rule engineering process, we aim at uncovering the rule engineering process in line with our RQs.

## Acknowledgments

The authors thank all CTF participants and organizers. This publication is part of the CATRIN, INTERSECT, and SeReNity projects (numbers NWA.1215.18.003, NWA.1160.18.301, and CS.010) which are financed by the Dutch Research Council (NWO).

## References

[1] Bushra A. Alahmadi, Louise Axon, and Ivan Martinovic. 2022. 99% False Positives: A Qualitative Study of SOC Analysts' Perspectives on Security Alarms. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 2783–2800. https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi

[2] Kevin Burk, Fabio Pagani, Christopher Kruegel, and Giovanni Vigna. 2022. Decomperson: How Humans Decompile and What We Can Learn From It. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 2765–2782. https://www.usenix.org/conference/usenixsecurity22/presentation/burk

[3] Wireshark Foundation. 2025. Wireshark. https://www.wireshark.org/

[4] Sebastian Garcia. 2025. Malware Capture Facility. https://stratosphereips.org

[5] Open Information Security Foundation (OISF). 2025. Suricata. https://suricata.io/

[6] Koen T.W. Teuwen, Tom Mulders, Emmanuele Zambon, and Luca Allodi. 2025. Ruling the Unruly: Designing Effective, Low-Noise Network Intrusion Detection Rules for Security Operations Centers. In *Proceedings of the 2025 ACM on Asia Conference on Computer and Communications Security (ASIA CCS '25)*. Association for Computing Machinery, New York, NY, USA, 14 pages.

[7] 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)*. Association for Computing Machinery, New York, NY, USA, 2770–2784.

[8] 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 (ASIA CCS '22)*. Association for Computing Machinery, New York, NY, USA, 799–814.