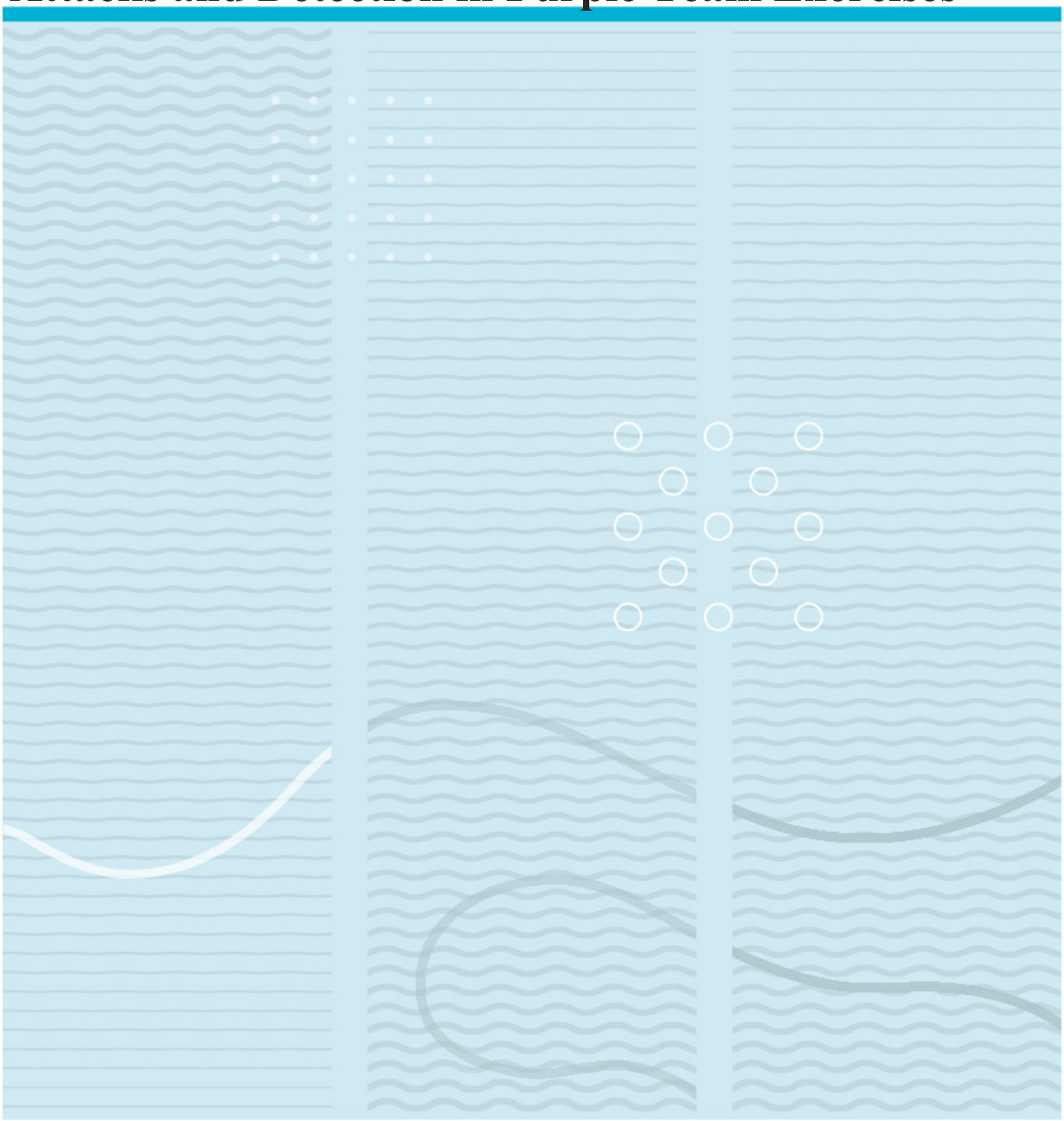
University of South-Eastern Norway
Faculty of Faculty of Technology, Natural Sciences, and Maritime Sciences
Department of Science and Industry Systems
Master's Thesis in Computer Science
Study programme: MACS5000
2022-2023

Alina Steshenko

Automating Correlation Between Attacks and Detection in Purple Team Exercises



University of South-Eastern Norway
Faculty of Faculty of Technology, Natural Sciences, and Maritime Sciences
Department of Science and Industry Systems
PO Box 235
NO-3603 Kongsberg, Norway

http://www.usn.no

© 2023 Alina Steshenko

USN Supervisor: Geir Myrdahl Køien

This thesis is worth 60 study points

University of South-Eastern Norway

Automating Correlation Between Attacks and Detection in Purple Team Exercises

Master's Thesis in Computer Science

Alina Steshenko

Supervisor(s): Geir Myrdahl Køien

Department of Science and Industry Systems

University of South-Eastern Norway

Campus Kongsberg

May 2023

Acknowledgements

I would like to express my gratitude towards my USN supervisor, Geir Myrdahl Køien, for providing constructive and valuable advice in our weekly meetings and for the support. A special acknowledgment and thanks are extended to my external supervisor from mnemonic, Martin Eian. I am deeply grateful for your wisdom, guidance, and support throughout this thesis.

My sincere gratitude goes to mnemonic for facilitating this opportunity. Thanks to all the interview participants whose valuable insights have greatly enriched this study. A distinctive thank you goes to my remarkable boss, Siri Bromander, and the rest of the R&D department. Your support has been invaluable, and the camaraderie has made you feel like family.

I would also like to thank the USN program coordinator, Jose Manuel Martins Ferreira. Your commitment to upholding the quality of education and your generous support to all students is truly remarkable.

Last but certainly not least, a special thanks to Erik Simonsen for always being there for me. Your encouragement has been a source of motivation and has propelled me to complete my thesis.

"If we knew what we were doing, it would not be called research, would it?"

— Albert Einstein

Abstract

This thesis examines the possibility of automating the correlation between attacks and detection in purple team exercises, aiming to enhance the efficiency of threat detection engineering. The study utilizes a hybrid research approach consisting of interviews with cybersecurity professionals and experimental investigations. The interviews reveal common challenges associated with manual correlation, especially that it is a time-consuming and tedious task, stressing the need for further exploration and innovative tooling in purple teaming and threat detection. Through experiments, it is demonstrated that the correlation process can be successfully automated. Several correlation methods are proposed based on the most common parameters identified through the interviews. Furthermore, a proof-of-concept tool is developed, and the proposed correlation methods are rigorously tested and compared in a controlled cloud environment. Notably, one approach stands out, exhibiting exceptional results in accuracy and efficiency. This promising outcome demonstrates the potential for automating the manual correlation process. It also sets a compelling path for future research and development in purple team exercises and detection engineering. The thesis highlights the significance of purple teaming as a collaborative approach in cybersecurity, promoting effective communication and cooperation between Red and Blue teams. While purple teaming has gained popularity, the limited research and tooling available emphasize the need for further exploration. This study contributes to the field by addressing the challenges of manual correlation and presenting an automated approach that shows promise for enhancing the overall efficiency of purple team exercises and threat detection engineering.

Contents

List of Figures

List of Tables

1 Introduction

This chapter provides a foundation for the study by establishing research context and relevance, articulating the need for research on purple teaming and automated correlation in purple team exercises. The problem statement and the primary motivation for the study are presented. Subsequently, the research objective is stated, constraints and boundaries inherent in the study are clarified, and the anticipated contributions of the research are highlighted. The chapter is concluded with an outline of the remaining thesis.

1.1 Motivation and Problem Statement

The continually evolving cyber threat landscape poses a general need for innovative and proactive cybersecurity defensive strategies to stay ahead of adversaries. The concept of "purple teaming" has become increasingly popular in cybersecurity as organizations look for ways to improve their security posture. One of the primary benefits of purple teaming is that it enables organizations to identify and address security risks more quickly. By combining the offensive (red team) and defensive (blue team) approaches, organizations are able to gain a better understanding of their security posture and identify potential areas of improvement. Additionally, purple teaming allows organizations to gain insights into their security practices that might not be possible when a single team. Close cross-team collaboration enables organizations to identify threats faster and to develop more comprehensive defense strategies.

One of the challenges associated with purple teaming is the need for coordination between the red and blue teams. Additionally, the need for resources to effectively manage a purple teaming program and the cost of commercial tools and purple teaming platforms can be prohibitive for many organizations. Despite these challenges, there are potential solutions that organizations can utilize to make purple teaming more practical, such as automation and open-source tools.

Currently, there is a limited number of specialized software tools and platforms to support purple teaming activities, and only one is open-sourced. Some commercial platforms offer automated testing, reporting, and analysis tools, either as native features or as supplementary services. However, publicly available platforms have not yet integrated automated correlation between attacks and detection. Automation is essential in purple teaming because it enables efficient execution of security testing and detection and helps reduce the time and resources required to conduct them, allow-

ing security teams to focus on other important tasks. Furthermore, automation can help improve the accuracy and reliability of the results by reducing the potential for human error. To date, the blue team has to manually report and connect the detection observables to the attacks executed by the red team in publicly available purple team platforms, which is time-consuming and tedious. Purple teaming and detection engineering will benefit significantly if this task is automated and open-sourced.

1.2 Research Objective

This study aims to determine whether the correlation between attacks and detection can be automated and consider the benefits and limitations of the automation in purple teaming, particularly for the blue team and threat detection engineering. The questions we aim to answer are as follows:

- (RQ1) What is the current state of the art in purple team exercises?
- (RQ2) What is the current state of the art in automating the correlation between attacks and detection in purple team exercises?
- (RQ3) Can the correlation between attacks and detection in purple team exercises be automated?
- (RQ4) Can automation of the correlation process in purple team exercises enhance the efficiency of detection engineering?

1.3 Limitations and Delimitations

Scientific research, by its nature, operates within certain boundaries. These limitations and delimitations, or boundaries of the study, can be of various kinds, including methodological constraints, theoretical frameworks, or practical considerations. Limitations typically stem from factors beyond the researcher's control, affecting the scope of the study, while delimitations are choices made by the researcher which set the study's boundaries. Understanding these helps interpret the findings and conclusions of the study in the correct context. The following list is a summary of the limitations and delimitations (detailed in Section 6.3) inherent to this study:

- **Novelty of the Purple Teaming Concept:** Purple Teaming's nascent status in cybersecurity limits the breadth of scholarly work available, potentially affecting the results, particularly RQ1 and

RQ2. Despite this, the popularity and novelty of the concept emphasize the need for further research.

- **Interview Sample Size and Profile:** This study draws from a limited set of interviews. However, the industry status of the participants and the diversity of their years of experience add depth to the findings.
- **Third-Party Tooling:** Utilizing the existing purple teaming platform, Vectr, and the Stratus Red Team for attack simulations, this study is subject to their inherent limitations, which could affect its implementation and reproducibility.
- **Atomic Test Cases:** Using atomic test cases in the experimental design restricts the simulation of a complex real-world sequence of events during a cyber attack. This constraint is due primarily to the atomic structure of the Vectr test cases and the Stratus Red Team library.
- **Testing Environment:** The controlled, isolated laboratory setting in which research experiments were conducted may limit the applicability of findings to real-world production environments and non-cloud attack vectors. This poses the need for further testing in a live environment with different settings.
- **Format of Correlation Parameters:** The effectiveness of automated correlation methods is contingent on the uniformity of correlation parameters, a common challenge in detection and log systems that are not extensively delved into in this study.

1.4 Contributions

This section outlines contributions and advancements brought forth by this thesis to both the theoretical understanding and practical application of purple teaming in the cybersecurity landscape.

Theoretical Contributions:

- Firstly, this study contributes to the existing literature by investigating the relatively uncharted cybersecurity domain of purple teaming and extending the understanding of how the synergistic approach can improve an organization's defensive capabilities — contributing to the enrichment of the current body of knowledge on proactive cybersecurity practices.
- Secondly, this study contributes to the existing literature by assessing existing open-sourced purple teaming tools, illuminating their potential and limitations. This provides valuable insights for improving upon the tooling for future work in this field.

- Additionally, this thesis contributes to the existing literature by offering an academic investigation of the industry-relevant issue, and insights from interviews with cybersecurity professionals, bridging the gap between theoretical research and the practical application of purple teaming.

Practical Contributions:

- The thesis implements and experimentally validates a proof-of-concept method for automated correlation between attacks and detection in purple team exercises, which enhances the efficiency of the purple team exercises and detection engineering by minimizing manual correlation efforts.
- Additionally, the contribution is bolstered by the implementation and testing of the method in a cutting-edge cloud environment. This closely aligns research outcomes with real-world issues and modern technological environments, enhancing the value and applicability of the findings, further bridging the gap between academia and practice.
- Lastly, by publishing the open-source prototype tool that implements the method, this thesis contributes to publicly accessible resources, promoting collective progress in cybersecurity. The prototype provides a foundation for future research, improvements, and advancements in purple teaming tooling.

1.5 Thesis Outline

This thesis is organized into seven chapters:

1. **Introduction** – This chapter lays the foundation for the thesis by presenting the motivation and problem statement, research objectives, limitations and delimitations, contributions, and the thesis outline.

2. **Background** – This chapter addresses RQ1 and RQ2. Relevant literature is reviewed in this section, and essential background knowledge is provided about red, blue, and purple teaming, threat intelligence and adversary emulation, purple teaming frameworks, platforms, threat detection, and log analysis. The chapter is concluded with a summary of the current state of the art of purple teaming, which addresses RQ1 and RQ2.

3. **Research Approach** – This chapter justifies the chosen research approach and describes the combination of the research designs, including data collection procedures and analysis plans.

4. Implementation – This chapter details the implementation process, addressing the technical elements of the conducted experiments.

5. Results – This chapter presents the aggregated results from interviews and experiments conducted during the research process.

6. Discussion and Future Work – This chapter contains a discussion of research findings and avenues for future research.

7. Conclusion – The final chapter summarizes the research and its contributions to the field of cybersecurity, highlighting the main findings and their implications for the industry.

2 Background

In this chapter, our primary objective is to address two research questions:

- RQ1: *"What is the existing state of the art in purple team exercises?"*
- RQ2: *"What is the existing state of the art in automating the correlation between attacks and detection in purple team exercises?"*

The chapter is designed to provide the reader with a solid foundation of the research area being studied and set the stage for the rest of the project. The "Previous Work" section overviews existing research on purple teaming and outlines the literature search strategy for the rest of the chapter. The following sections introduce the key concepts and theories that form the basis of the presented research. By the end of this chapter, the reader should have a clear understanding of the current state of the field and the context for the rest of the thesis.

2.1 Related Work

The initial literature review indicated limited academic purple teaming research. Academic publications explicitly referencing *purple teaming* are no older than five years old (Olsen, 2022; Chowdhury, 2019; Chaplinska et al., 2022; Ilca & Balan, 2021). Purple teaming has been around for some time, but it has been more recently adopted and popularized in the cybersecurity space. The term "purple teaming" was first introduced in 2013 by the SANS Institute, which defined it as "the practice of using a combined red and blue team to achieve better results in defending and protecting systems and networks"(Dale, 2019). Since then, the concept has gained traction among security professionals, with many organizations leveraging the approach to identify and mitigate threats.

Unlike academic research, which is often theoretical, practitioner-sourced resources provide real-world, practical insights and knowledge about the latest trends and developments in the field. By leveraging data and insights from the industry, academia can better understand the challenges and opportunities in the field of cybersecurity and develop effective solutions to address them. Because of that and the scarcity of academic research on purple teaming, non-academic practitioner-sourced materials were consulted as well (Reiber, Opel, & Wright, 2021; Routin, Thoores, & Rossier, 2022; Dale, 2019; Paper, 2021; Orchilles, 2021; Booz, 2020).

Olsen (Olsen, 2022) reviewed the most recent academic literature on purple teaming, revealing its

evolution from red teaming as an emerging trend. Through inductive reasoning, literature search was extended to *offensive and defensive security* (Ajmal, Shah, Maple, Asghar, & Islam, 2021; Diogenes & Ozkaya, 2022; Oakley, 2018), *proactive security strategies* and *red teaming* (Oakley, 2019; Mansfield-Devine, 2018; Rehberger, 2020), *blue teaming* (Carey & Jin, 2020), and *cyber security exercises* (Seker & Ozbenli, 2018; Vykopal, Vizváry, Oslejsek, Celeda, & Tovarnak, 2017; Babayeva, Maennel, & Maennel, 2022; Brilingaitė, Bukauskas, Juozapavičius, & Kutka, 2022, 2020). Furthermore, to understand the problem domain and provide background, key search words such as *threat intelligence* and *adversary emulation* (Ajmal et al., 2021; Saarainen, 2021), *threat detection*, *log analysis* and *event correlation* (Svacina et al., 2020) were added to the list.

2.2 Red, Blue and Purple Teaming

According to the Oxford English Dictionary (OED), "*Purple is having the color of blue and red mixed together*" ("Definition of Purple", 2022). As the color definition indicates, purple teaming in cyber security involves red and blue teams. The mix of red and blue to create purple and the need for more consensus on the relatively new concept of purple teaming can foster a common misconception that purple teaming entails just red and blue teams. The purple teaming concept is more complex than merging the two teams as it may require knowledge from other areas of expertise, such as cyber threat intelligence and management (Olsen, 2022).

In her Ph.D. dissertation, Olsen sought to answer what Purple Teaming is. She proposed a high-level definition as *"cyber threat intelligence-driven, full-knowledge red team engagements that emulate TTPs with blue team collaboration to produce threat detection or additional knowledge about an organization's defensive posture"* (Olsen, 2022).

The academic research and some practitioner-sourced publications built on previous red team literature and argue that purple teaming evolved from red teaming as a strategy (Olsen, 2022; Chowdhury, 2019; Routin et al., 2022; Orchilles, n.d.), while others describe it as an equal merge of offensive and defensive strategies due to the need of proactive defense in the continually evolving threat landscape (Booz, 2020; Chaplinska et al., 2022; Chowdhury, 2019). Nevertheless, this study builds on previous red and blue teaming literature to provide context and a high-level understanding of both.

Red team versus blue team exercises have a military origin, and the idea is that the blue team defends against red team attacks (Oakley, 2019). The Center of Advanced Red Teaming at the University at Albany proposed the following definition for Red Teaming: *"Any activities involving the simulation of*

adversary decisions or behaviors, where outputs are measured and utilized for the purpose of informing or improving defensive capabilities" ("Towards a Definition of Red Teaming", 2019). Although it is correct, it might be outdated as the same definition can be applied to Purple Teaming and still be valid. In contrast to purple teaming, the adversary simulation is executed by a red team with no direct involvement from a blue team. Consecutively, the Red Team is a group of offensive security experts authorized and organized to conduct red teaming ("Red Team Definition", n.d.).

The traditional definition of a Blue Team is the group of individuals responsible for defending an enterprise against the red team ("Blue Team Definition", n.d.). However, defensive security experts describe the blue team as any team that is not red (Carey & Jin, 2020). It is individuals directly responsible for monitoring, defending, and responding to incidents, but also those indirectly responsible for the organization's security posture, such as engineers and architects designing systems to be proactively more secure. Dependent on the size of the organization, the blue team may be the general term to reference a single department or multiple teams that make up defensive operations. It may consist of Security Operations Center (SOC), Digital Forensics and Incident Response (DFIR), Managed Security Service Providers (MSSP), detection engineers, threat hunters, and more (Carey & Jin, 2020; Orchilles, n.d.).

One of the biggest issues with the traditional Red versus Blue Team approach is the *success of one means failure of the other* mindset. Purple teaming aims to solve the competitive approach and optimize offensive and defensive security efforts in a common direction by enhancing collaboration between the red and blue teams (Routin et al., 2022; Dale, 2019).

2.3 Threat Intelligence and Adversary Emulation

The continually evolving threat landscape poses a general need for innovative and proactive cybersecurity defensive strategies to avoid future attacks. Understanding the target organization and identifying the adversary to emulate is essential in purple teaming, and it requires threat intelligence (Routin et al., 2022; Orchilles, 2021; Reiber et al., 2021; Olsen, 2022). National Institute of Standards and Technology (NIST) defines Threat Intelligence as *"threat information that has been aggregated, transformed, analyzed, interpreted, or enriched to provide the necessary context for decision-making processes"* ("Threat Intelligence Definition", n.d.).

Although there is a growing trend in leveraging internal data for threat intelligence generation (Lee, 2020), little academic literature is available. Olsen (Olsen, 2022) provided the most recent aca-

demic research on how cyber security professionals prioritize and leverage threat intelligence in purple teaming exercises. Saarainen (Saarainen, 2021) reviewed the most common frameworks for red team adversary simulations, including Threat Intelligence-Based Ethical Red-teaming European Union (TIBER-EU); MITRE Adversarial Tactics, Techniques, and Common Knowledge (MITRE ATT&CK); Lockheed Martin Cyber Kill Chain; Threat Intelligence-led assessment - CBEST; Adversarial Attack Simulation Exercise (AASE). However, in Olsen's survey (Olsen, 2022), 90 % of respondents use MITRE ATT&CK ("MITRE ATT&CK", 2015-2022) for communication between teams or to track purple team exercises.

MITRE ATT&CK uses Tactics, Techniques, and Procedures (TTPs) as a taxonomy for threats leveraged to create adversary emulation plans ("MITRE ATT&CK", 2015-2022). While Threat Modelling is out of the scope of this study, it is an essential part of the purple teaming exercise framework (discussed in Section 2.4). It is important to have a high-level understanding of the key concept of TTPs, which is defined by NIST as follows: "The behavior of an actor. A tactic is the highest-level description of this behavior, while techniques give a more detailed description of behavior in the context of a tactic, and procedures an even lower-level, highly detailed description in the context of a technique" ("TTP Definition", n.d.).

2.4 Purple Teaming Frameworks

While MITRE ATT&CK provides a common taxonomy for threats in adversary emulation and for tracking coverage of TTPs, it does not provide a framework for the actual process of the purple teaming exercise. Unfortunately, there is no industry standard for purple teaming yet, and little academic literature is publicly available detailing the phases and procedures of the exercise. However, there is a sufficient amount of practitioner-sourced literature to provide a high-level understanding of the practice. Practitioners suggest the following frameworks:

- Prepare, Execute, Identify, and Remediate (PEIR) framework (Routin et al., 2022)
- Atomic Purple Team Framework (Ickler & Orchilles, 2020)
- Purple Team Exercise Framework 2.0 (PTEFv2) (Orchilles, 2021)

It can be argued that PTEFv2 is the most mature of the reviewed frameworks. PIER is very similar to regular red vs. blue team exercises. It mimics red teaming, and it needs to be clarified how it addresses the competition issue of the traditional red vs. blue exercise. Atomic Purple Team Framework lacks proper documentation and details explaining each framework phase. PTEFv2, on the other

hand, is well-documented and details stages of the purple teaming lifecycle as illustrated in Figure 1.

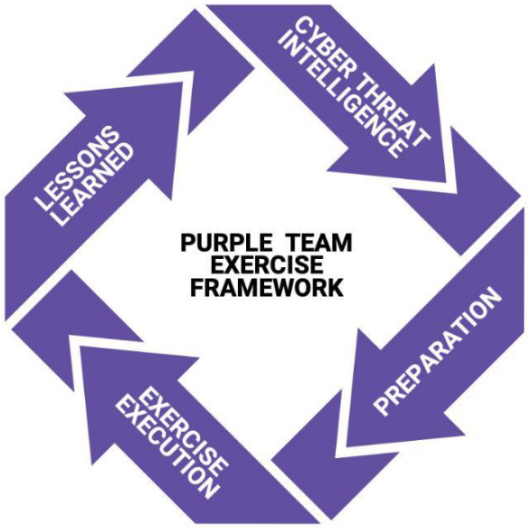


Figure 1: Purple Team Exercise Framework (Orchilles, 2021)

2.5 Purple Teaming Platform

Effective purple teaming exercises necessitate robust software tools to streamline cross-team planning, collaboration, documentation, and activity tracking. A search for open-source purple teaming platforms revealed only one viable option: Vectr ("VECTR - purple teaming tool", n.d.). Vectr is a collaboration and reporting tool that facilitates manual coordination between blue and red team activities. Commercial automated attack platform Scythe ("Scythe - purple teaming tool", n.d.) and offensive security reporting tool Plextrac ("Plextrac - purple teaming tool", n.d.) integrate with Vectr. However, an in-depth academic review of Scythe, Plextrac, and other commercial platforms that claim to offer purple teaming capabilities is beyond the scope of this study due to the prohibitive licensing cost. Regardless, no open-source purple teaming platforms automatically connect detection observables to attacks.

Vectr provides limited automation capabilities for threat simulation. It supports creation or import of libraries of automated tests, such as Atomic Red Team ("Atomic Red Team", n.d.). However, we did not identify any native, out-of-the-box attack simulation automation that could be executed directly from the platform.

Atomic tests are organized as *Test Cases* within a *Campaign*, with campaigns grouped into *Assessments*, and Assessments further organized into *Databases*. Figure 2, taken from Vectr documentation

("VECTR - purple teaming tool", n.d.), illustrates this structure.

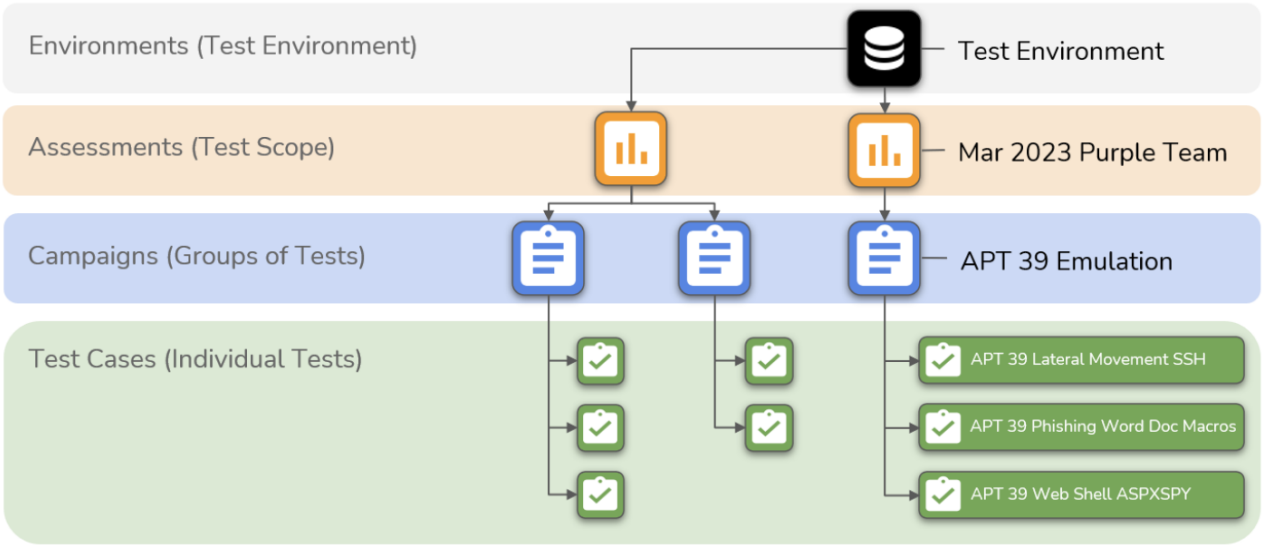


Figure 2: Vectr Structure ("VECTR - purple teaming tool", n.d.)

Despite Vectr documentation ("VECTR - purple teaming tool", n.d.) claiming that the platform enables users to capture whether defense tools detected an attack, testing of the tool did not reveal any automation accommodating this functionality. The blue team can manually add outcome notes, evidence files, and notes regarding detection and prevention for each test case. However, Vectr has a GraphQL API endpoint that can potentially be used to query data needed for integration with other automated tools.

2.6 Threat Detection and Log Analysis

One of the aims of purple teaming is to improve an organization's defensive security, which heavily relays on threat detection. Threat detection generally involves the following steps:

1. **Log Collection**: System, network, security, and application logs are collected from various data sources. This step may also include normalization of the logs to convert raw log data into a consistent and standardized format to make it easier to analyze, search, and correlate log data from different sources.

2. **Log Analysis and Threat Identification**: The collected log data is analyzed to identify potential security threats. This can be done using various techniques such as regular expressions, rule-based systems, statistical analysis, machine learning algorithms, correlation, and human review.

3. **Alert generation**: Once a potential threat has been identified, an alert is generated. This can be done through various means, such as SOC or a security information and event management (SIEM) system.

Svacina et al. (Svacina et al., 2020) provided a literature review on recent trends in security log analysis. The research showed that automation is necessary regardless of the detection technique due to large amounts of various log data making manual analysis infeasible. This conclusion can be extended through inductive reasoning to detection engineering in purple team exercises, which involves designing accurate rules and algorithms to detect threats with a low false positive and false negative rate based on the generated log data.

In threat detection, log analysis plays a crucial role in identifying potential security incidents. However, the effectiveness of log analysis relies on the consistent formatting of log data across various systems, applications, and devices. The absence of a uniform log format can significantly impede the correlation of log data, making it challenging to search for and identify potential threats.

The inconsistent representation of the same log data across different systems can further complicate the log analysis process. For instance, timestamps may differ in format, with some applications utilizing the ISO 8601 standard while others use Unix timestamps. Moreover, the interpretation of timestamps can vary, with some logs indicating the time when an event occurred, and others may indicate the time when the system logged the event. Knowing the semantics of timestamps and other log data is essential to ensure accurate and meaningful analysis.

Today, various tools and techniques are available for log analysis, such as log management and analysis platforms, security information and event management (SIEM) systems, and open-source tools. Most such systems can handle large amounts of data and normalize log data with a common schema that can accommodate diverse log types, facilitating the correlation of log data from multiple sources and providing a basis for analyzing log events. One of the widely adopted and prevalent combinations of open-source tools for the analysis and visualization of monitoring data is the Elasticsearch, Logstash, and Kibana (ELK) stack (Chhajed, 2015). *Elasticsearch* is a search and analytics engine, *Logstash* is a data collection and log-parsing pipeline, and *Kibana* is a visualization tool for data stored in Elasticsearch ("What is ELK Stack?", n.d.). Log data in ELK is normalized and structured with Elastic Common Schema (ESC), enabling uniform examination of log data. More details about ELK utilization can be found in the implementation Chapter 4.

While the theory behind threat detection engineering is relevant to the correlation of attacks and detection in purple team exercises, it is essential to highlight the differences. A threat detection rule typically aims to identify a single, a set, or a threshold of specific log entries while filtering out "harmless" activities. This implies that even if a malicious actor executes an activity, triggering an alert is not necessarily desirable. For instance, malicious actors systematically scan systems, networks, and services across the internet for potential vulnerabilities or weaknesses. This scanning activity can be considered an Indicator of Attack (IOA), a piece of data or occurrence that points to an ongoing, but not necessarily successful, attack. However, it is a continuous and automated process that generates noise in detection systems. As a result, a "harmless" port scan, although an IOA, might not generate an alert without additional activity. This additional activity could serve as an Indicator of Compromise (IOC), a piece of data that signifies a potential security breach. Unlike IOAs, IOCs indicate that a security breach has already occurred. Therefore, in this example, an alert would be more desirable if there is an IOC, such as evidence of successful unauthorized access or a change in system settings, following a port scan.

In contrast to detection rules, the desired outcome of correlation in purple team exercises is to detect *all* log instances of an attack. Unlike red teaming, the focus of purple teaming is not solely to discover weaknesses in an organization's defense but also to improve it during the exercise rather than as a post-activity. The advantage is that the attacks are fully "transparent," enabling the blue team to utilize all available data to engineer detection rules.

2.7 Summary

This chapter has delivered an overview of the key concepts and terminologies related to purple teaming, including it's evolution, and present adoption state, thereby addressing RQ1 and RQ2. It is evident that purple teaming has evolved from the traditional red team vs. blue team approach and is now acknowledged as a collaborative process that fosters communication, cooperation, and knowledge-sharing between the two teams. This collaborative paradigm shift has enabled organizations to identify vulnerabilities and weaknesses in their cybersecurity defenses more efficiently, improving their overall security posture.

Despite its growing recognition, purple teaming is still in it's infancy, with limited research and innovation. Review of currently available software tools facilitating purple teaming activities reveals the a lack of open-source tools that can automatically correlate detection observables to attacks. Consequently, a well-established state-of-the-art cannot be decisively stated, emphasizing the necessity

for further exploration into the underlying principles, methodologies, and tools associated with the concept of purple teaming.

In conclusion, a review of the RQ1 and RQ2 highlights the pressing need for additional research and innovation in both purple teaming strategy and tooling. This need is driven not only by the novelty of purple teaming as a concept but also by the general need for innovative and proactive cybersecurity defensive strategies to stay ahead of adversaries. As the practice of purple teaming continues to expand across both governmental and private sectors, efforts should be directed towards enhancing its efficacy, potentially through automation and other innovative methodologies.

3 Research Approach

This chapter presents the research model, methods, and techniques for addressing RQ3 and RQ4. It also elaborates on the rationale behind the approach choices and discusses the limitations, assumptions, and constraints associated with the selected approach.

3.1 Rationale for the Research Approach

In light of the limited availability of research data on purple teaming and the automation in purple team exercises, a mixed research approach was adopted to address RQ3 and RQ4. The rationale behind this decision was to obtain a more holistic understanding of the research problem, leading to more robust findings and comprehensive answers to the research questions.

The qualitative research design utilized data-driven content analysis to identify themes and patterns from the interviews with cyber security professionals. This analysis helped identify common parameters employed in the manual correlation process and the associated challenges. The findings from the interviews informed the development of an automated correlation method, which was tested for accuracy and efficiency in the subsequent quantitative research phase.

The qualitative approach provided a contextual foundation and guidance for the quantitative aspect of the study. Furthermore, it generated findings that offer insights for future practices and research endeavors related to purple teaming and threat detection engineering.

3.2 Interviews

This chapter describes the procedures used to select participants, conduct short structured interviews, and analyze data for the qualitative portion of the study. The complete interview script can be found in Appendix A with the following interview questions:

- (Q1) How many years of experience do you have in cybersecurity?
- (Q2) In your experience, how long does it take to correlate detection data to an attack technique manually?
- (Q3) In your experience, what are the three most common parameters used to correlate detection data to an attack technique?

- (Q4) In your experience, what are the most common challenges with correlating detection data to an attack technique?

The goal of RQ4 is to investigate whether automating the correlation between attacks and detection in purple team exercises can enhance the efficiency of detection engineering. To answer this, (Q2) was designed to collect data about time used on manual correlation, so it could be compared with time data used on automated correlation, which was collected in the quantitative part of the study described in Section 3.3. Further, (Q3) was used to gather information about correlation parameters commonly used in detection engineering, which were then implemented in the automation design described in Section 3.3. Finally, (Q4) was used to gain insights into detection correlation challenges and inform potential solutions to improve the process.

3.2.1 Participation and Data Collection Procedures

Eight participants were selected based on their role as detection engineers at one of the leading cybersecurity companies in Norway: mnemonic AS. It should be noted that mnemonic AS is the researcher's employer and that the participants were selected based on convenience sampling, given the time and resource constraints. This could introduce a potential bias in the findings, as discussed further in Section 6.1.1.

The individual interviews were in-person and via video conferencing, depending on the participant's availability and preference. Each interview lasted approximately 30 minutes and was audio-recorded with the participant's consent to ensure accurate data collection. The recordings were transcribed, cleaned for filler words, and approved by the participants before the analysis. The transcriptions can be found in Appendix B.

3.2.2 Data Analysis Plan

Response categories from Q1 and Q2 were represented using a numerical scale, the average time frames were calculated, and findings were summarized using descriptive statistics in Section 5.1. The transcribed data collected from Q3 and Q4 was analyzed using data-driven content analysis, which involved coding the data and identifying themes and patterns that emerged from the participant's responses.

3.2.3 Ethical Considerations

The interviews were conducted in accordance with ethical principles, including informed consent, confidentiality, and anonymity. The participants were informed of the study's purpose, the nature of their participation, and the data collection and analysis procedures. The participants were assured of the confidentiality of their information and that their names would not be disclosed in any publications or reports. The participants were also given the right to withdraw from the study at any time without any consequences.

3.2.4 Trustworthiness

The interviews were audio recorded to accurately represent the participant's responses, ensuring the research's trustworthiness. Participants reviewed the transcribed interviews to ensure their accuracy and validity. A detailed description of the data collection and analysis procedures was provided in this chapter to increase the transparency of the research process.

By using the same set of questions for all participants, the researcher aimed to reduce potential interviewer bias, where the interviewer might unintentionally influence the participants' responses. Further, the researcher was careful not to express personal opinions or judgments during the interviews. Instead, the researcher practiced active listening to create a respectful environment and ensure that the participants felt comfortable and willing to share their experiences and opinions, leading to more informative and insightful data.

3.3 Experiments

To answer RQ3 and RQ4 the following research hypotheses was tested:

- Hypothesis 1 (from RQ3): The correlation between attacks and detection in purple team exercises can be automated with the method described in Section 3.3.1.
- Hypothesis 2 (from RQ4): Automating the correlation process in purple team exercises will significantly enhance the time efficiency of the correlation process.

3.3.1 Correlation Method

The following correlation parameters are suggested based on the evaluated results (Section 5.1) from the interview question 3: "What are the most common parameters used to correlate detection data

to an attack technique?".

1. **Timestamps**: start and stop time of the attack.
2. **Log sources**: log types relevant to the attack.
3. **Identity-based data**: IP address, host, user, etc.
4. **Indicators of Attack/Compromise (IOA/IOC**): metadata relevant to the attack.

Based on those parameters, a correlation method was proposed that searches for events meeting all the conditions below in order to identify those relevant to the attack effectively:

- The event occurred within the time interval of the attack (between the start and stop times).
- The event has one or more of the specified log types.
- The event has one or more of the specified ID parameters.
- The event one or more of the specified IOAs

The aim is to demonstrate the method as a proof-of-concept by automating querying log data based on suggested parameters for each attack case and posting the query results to the purple teaming platform Vectr. The implementation is described in Section 4.3.

3.3.2 Testing Environment

A cloud-based environment was chosen for testing the proposed correlation method for several reasons:

- As more and more organizations transition to the cloud, conducting experiments in a cloud environment closely aligns research outcomes with real-world issues and modern technological environments. This enhances the practical value and applicability of the findings, making the research more relevant and valuable for current organizational challenges.
- A cloud-based environment provides a controlled testing environment tailored to specific needs. This ensures that the test environment is consistent and accurate, providing reliable results.
- Using a cloud-based environment eliminates the need for physical infrastructure, which can be costly and time-consuming to set up and maintain. This provided a cost-effective alternative that allowed the researcher to experiment without incurring significant expenses.

- A cloud-based environment provided a scalable solution that could be easily adjusted to meet changing testing needs if they evolved as the research progressed.
- Cloud-based environments provide flexibility by allowing remote access.

Despite the numerous advantages of using a cloud-based environment, there are certain drawbacks to be considered:

- Limited control over the underlying infrastructure is a challenge since cloud service providers manage the cloud infrastructure. This restriction can impact system configurations and log access, potentially limiting the ability to simulate specific scenarios or customize the environment similar to a locally hosted solution.
- The method's effectiveness might not extend to all possible attacks, particularly those not represented in the cloud attack vector, due to the infrastructural constraints of the cloud environment.
- Testing in a controlled and isolated lab environment could pose potential risks to method validity when compared to a live environment. This can affect the generalizability of the results to real-world scenarios.

These disadvantages suggest that although a cloud-based testing environment provides relevance, convenience and scalability, it also poses challenges that must be considered when designing and conducting the experiments. These issues are discussed further in Section 6.

3.3.3 Selection of Test Cases

This study utilized the open-source project Stratus Red Team ("Stratus Red Team", n.d.) for attack emulation. Stratus Red Team is a self-contained binary for emulation of offensive attack techniques against cloud environments. It was chosen based on the testing environment and the need for diverse and representative cloud attack scenarios. The following criteria were considered when selecting the Stratus Red Team:

- Publicly available threat intelligence
- Relevance to the cloud attack surface
- A diverse range of attack scenarios, including various techniques and tactics
- Open-source availability and mapping to the MITRE ATT&CK framework

Utilizing the Stratus Red Team library introduces potential threats to external validity and the risk of bias due to its inherent limitations, mainly its focus on the specific cases selected by the Stratus Red Team, which may not encompass all possible cloud attack scenarios. Moreover, while highly valuable in their application, automated attack libraries such as Stratus Red Team do come with the caveat of limited granularity and insight into the technical details of the implementation and execution of the attacks. Additionally, the library offers atomic attack cases that may not represent a chain of events during a real-world attack. This limitation is not unique to the Stratus Red Team library but is also a known constraint of the purple teaming platform, Vectr, as discussed in Section 2.5. Chapter 6 provides a more detailed discussion of this issue.

Nevertheless, given the selection criteria and the considerations mentioned above, the Stratus Red Team library was an appropriate choice for the objectives of this study. Table 11 in Appendix D presents a complete overview of the 27 attacks selected for the experiments. Attacks 2, 10, 12, 13, and 16 were excluded from the test due to technical reasons detailed in Section 4.2. The remanding set includes 22 techniques targeting different components of the cloud environment, such as compute instances, storage services, and networking infrastructure. Furthermore, the library encompasses Initial Access, Lateral Movement, Persistence, and Exfiltration techniques to represent various threat scenarios.

3.3.4 Experiments Design Strategy

A series of experiments were conducted to evaluate Hypothesis 1 and assess the correlation method proposed in Section 3.3.1. The list of experiments is presented in Table 1. The experiments were performed in a controlled cloud environment, isolated from external factors, and utilized consistent attack simulations.

All methods utilized time and log-type correlation. The specific parameters for methods incorporating IOA-based correlation (Table 11, Appendix D) were derived from Stratus detection recommendations for each respective attack technique. The identity parameter for the methods utilizing ID-based correlation was the name of the IAM user designated for the attack executions.

Table 1: Experiments Plan

Experiment	Correlation Method	Expected Outcome
1	Automated (ID ∩ IOA)	Data for efficiency and accuracy evaluation
2	Manual	Data for Ground Truth
3	Automated (ID)	Data for efficiency and accuracy evaluation
4	Automated (IOA)	Data for efficiency and accuracy evaluation
5	Automated (ID ∪ IOA)	Data for efficiency and accuracy evaluation

The initial experiment tested the automated correlation method based on all four originally suggested parameters as detailed in Section 3.3.1. Subsequently, a manual correlation was carried out to establish a ground truth, serving as a benchmark for evaluating the accuracy of the automated correlation method. The manual correlation experiment was conducted after the automated correlation experiment to prevent bias. It ensured that the researcher could not influence the test results by adjusting the correlation parameters, such as IOAs and ID parameters, based on the findings from the manual correlation.

Initially, the plan was to repeat the experiments in an identical environment but with the introduction of noise to simulate real-world traffic and conditions of a production environment. However, after conducting the first two experiments, it became evident that this approach would contribute little value to the research at this point. Due to the overly specific nature of the method, the results would be identical to those from the first experiment unless the noise perfectly matched the activities executed by the same user during the attack simulations. This issue is further explored in Section 6.2. Consequently, subsequent experiments were designed to test other variations of the correlation method to determine if any of these approaches would yield a more suitable fit. The method in follow-up experiments was modified in three different ways:

- by excluding identity-based parameters (experiment 3)
- by excluding IOA-based parameters (experiment 4)
- by including results matching the identity-based *or* IOA-based parameters (experiment 5)

3.3.5 Metrics and Evaluation

As mentioned in Section 2.6, the desired outcome of the correlation in purple team exercises is to detect *all* log instances of an attack. The following metrics were used to assess the effectiveness and accuracy of the correlation method:

- **True Positives (TP)**: The number of correctly identified attack instances in the correlation process.
- **False Positives (FP)**: The number of non-attack instances incorrectly identified as attacks.
- **True Negatives (TN)**: The number of correctly identified non-attack instances in the correlation process.
- **False Negatives (FN)**: The number of attack instances incorrectly identified as non-attacks.

These values contribute to a confusion matrix, which is widely employed for assessing classification algorithms' accuracy and effectiveness, such as intrusion detection systems, malware classifiers, and anomaly detection models. By evaluating the correct detection (TP and TN) and the incorrect detection (FP and FN), the confusion matrix facilitates a better understanding of a solution's strengths and weaknesses. This knowledge allows for fine-tuning systems to optimize performance and make informed decisions regarding implementation and configuration.

Additionally, the following metrics were calculated to provide a more comprehensive assessment of the method's performance:

- **Precision** = $\frac{TP}{TP+FP}$
  This metric measures the proportion of correctly identified instances of an attack (True Positives) among all instances identified as attacks. A high precision score indicates that the correlation method has a low rate of false positives (instances incorrectly identified as attacks).
- **Recall** = $\frac{TP}{TP+FN}$
  This metric measures the proportion of correctly identified attack instances (True Positives) among all actual attack instances. A high recall score represents a low rate of false negatives (instances incorrectly identified as non-attacks).
- **F1 Score** = $2 * \frac{Precision*Recall}{Precision+Recall}$
  This metric combines precision and recall into a single value that balances both aspects of the correlation method's performance. It is particularly useful when there is an uneven distribution

of attack and non-attack instances. A high F1 score indicates that the method is effective in identifying attacks while minimizing both false positives and false negatives.

Lastly, the time required for the automated correlation was compared to data collected through interviews with cybersecurity professionals. The interview responses represent the typical correlation time in a real-world scenario, accounting for various cases and circumstances. However, it should be considered that responses may not be directly comparable to the times acquired through the controlled experiments due to the myriad of potential variables in a live environment. However, these responses still hold significant value as they offer a broader context, enabling assessment of the efficiency of the automated correlation method in a more generalized setting. An alternative approach could have been to time the manual correlation process for the same test cases used in the automated correlation experiment. However, this method raised concerns about potential threats to the validity. There is a risk of bias, as the researchers conducting the experiments may unconsciously adjust the speed or approach when manually correlating, aware of the automated process being evaluated. It was not feasible to conduct a representative number of manual correlations by impartial participants to remove all bias.

Moreover, the hypothesis is that automation will significantly improve the correlation time even in best-case scenarios, represented by interview responses regarding the minimum correlation time. Given the considerations, the most prudent approach was to compare the time taken by the automated correlation method against the average correlation times provided by the interviewed cybersecurity professionals. This approach provides a benchmark based on actual industry practice, mitigating the potential bias of the researcher while also allowing assessment of the potential time-saving benefits of the automated method.

3.4 Summary

This chapter presented a structured and comprehensive approach to address the research questions RQ3 and RQ4. A hybrid research method was employed to tackle the relative paucity of existing research data on purple teaming and the automation of correlation in purple team exercises. It was grounded in two phases: interviews and experiments.

The qualitative research design relied on data-driven content analysis, deriving patterns and themes from interviews with cybersecurity professionals. The approach utilized data from industry professionals to identify common parameters and challenges in manual correlation processes and laid the

groundwork for the development of an automated correlation method. Although the risk of bias and potential limitations exist, the approach's robustness was bolstered by its adherence to ethical principles and its focus on ensuring the trustworthiness of the collected data.

The second phase of the research was designed to employ a series of experiments to assess the effectiveness and accuracy of the correlation method. The approach has high internal validity, as it controls for potential confounding variables, uses a rigorous experimental design, and employs robust metrics for evaluation. However, the potential challenges and threats to the validity of the research were identified, including the limitations of the testing environment and the selected attack library, as well as potential bias in the manual correlation process. These considerations were taken into account in the analysis and interpretation of the experimental results.

4 Implementation

This chapter delves into the implementation of the proposed correlation method, detailing the technical decisions and components involved in demonstrating and evaluating the approach. The various aspects of the implementation, such as the testing environment, log processing pipeline, attack emulations, and automated correlation process, are discussed in detail.

4.1 Testing Environment and Log Processing Pipeline

The following Amazon Web Services (AWS) Organisation structure was set up with Terraform[i] according to the AWS' best practice recommendations:

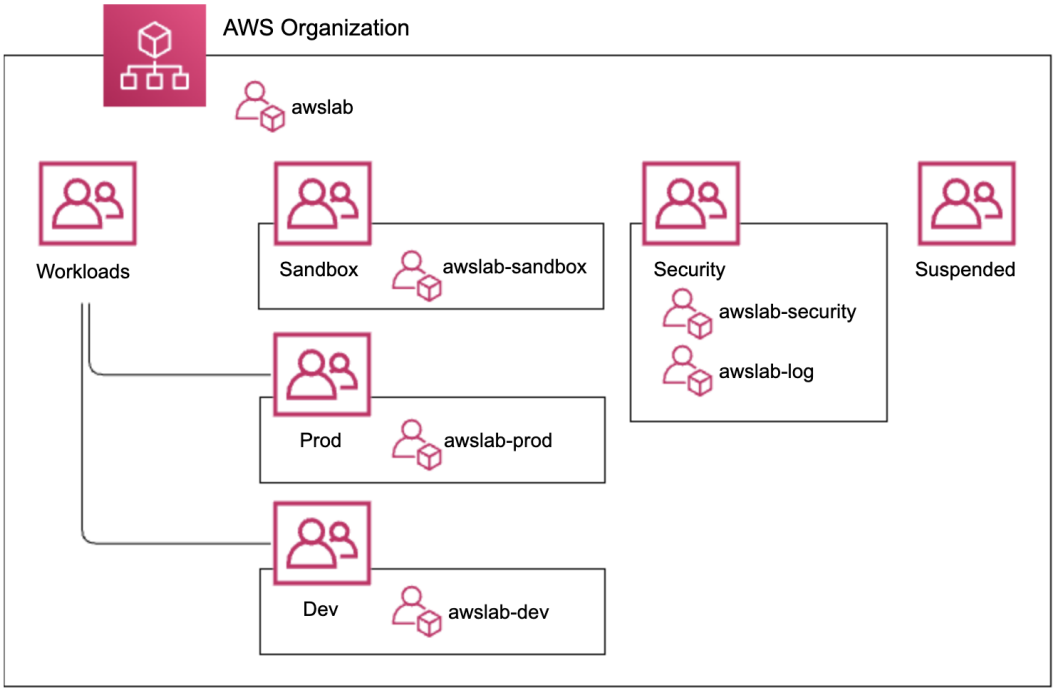


Figure 3: Lab Environment - AWS Organization Structure

Two of the main AWS logging and security event services were enabled in all accounts: CloudTrail and GuardDuty. Figure 4 illustrates the centralized logging setup for the multiple account architecture.

AWS CloudTrail and GuardDuty logs across all accounts are stored in AWS Simple Storage Service (S3) buckets in a centralized logging account. Whenever a new object is added to the bucket, S3 sends a message to the AWS Simple Notification Service (SNS) Topic, which subscribers can consume. The message contains the information necessary to retrieve the log object from the bucket. This message

[i] an open-source infrastructure-as-code software

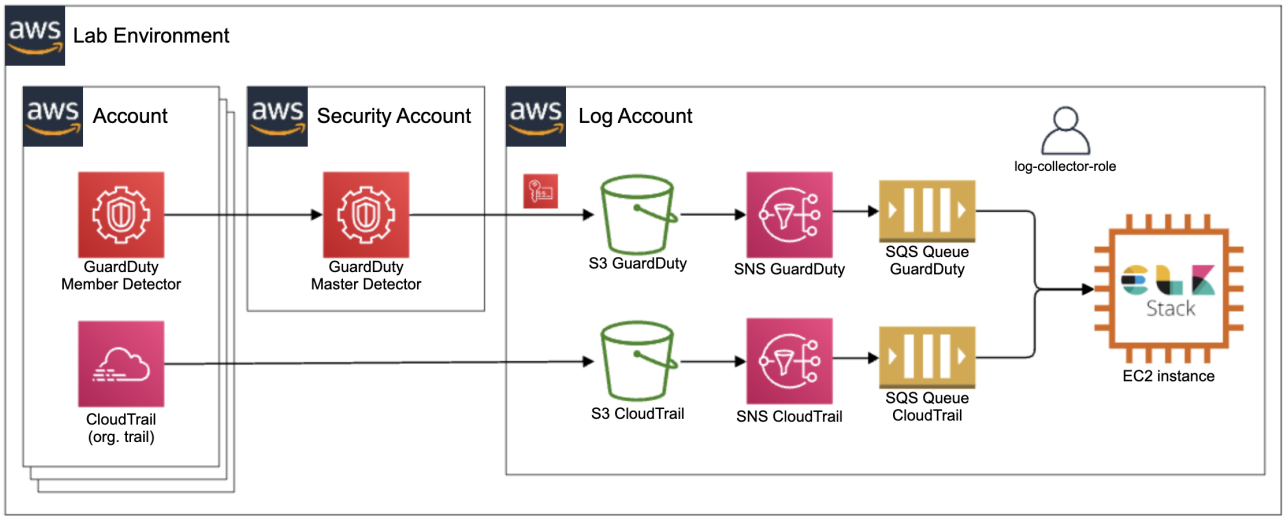


Figure 4: Log Processing Pipeline

is further sent to an Amazon Simple Queue Service (SQS) queue. Note that the SNS Topic is only necessary if there are multiple subscribers, e.g., multiple log consumers.

An ELK stack (see Section 3.3) is deployed on AWS Elastic Compute Cloud (EC2), receives SQS messages, and reads the log objects stored in the S3 buckets. Note that the ELK stack can (and should) be set up in a different AWS account or even on-premises, making it particularly useful for managing logs across multiple AWS organizations or forwarding logs to external monitoring service providers.

4.2 Attack Emulations

The Stratus Red Team test cases (detailed in Table 11, Appendix D) were incorporated into the Vectr campaign and enriched with correlation parameters as described in Section 3.3.1. As explained in the next section, these parameters were added as `blueTeamMetadata` key-value pairs and used programmatically in the automation process.

Due to internal errors within the Stratus techniques, attacks 2, 10, and 16 were omitted from the assessment. Attack 12 was excluded due to an AWS CloudTrail error, while attack 13 was precluded due to the need for a costly GPU instance, necessitating a quota increase request in the AWS account, which was beyond the researcher's available resources. The remaining 22 attacks were executed automatically within the *Sandbox* account. Each Stratus attack technique consisted of three phases:

1. **Warm-up phase**: Prerequisites for an attack technique are set up using Terraform
2. **Execution phase**: The attack technique is executed against resources created during the warm-up phase.

3. **Clean-up phase**: The resources created during the warm-up phase are terminated and destroyed.

These phases were scripted to run automatically for each test case, with a 60-second gap between the attacks to allow time for logs to be generated, transmitted, and processed and to avoid overlap. The script can be found in Appendix F. Since Vectr does not provide native support for automated attack execution, it was updated with the attack start and stop times for each test case later. Although initiating the attack script execution directly from Vectr could be automated, it was deemed irrelevant to the research questions. Consequently, the decision was made not to allocate resources for this automation but to include it as a recommendation for future work (Section 6.4).

4.3 Automated Correlation Process

Before the simulations, every Vectr test case was enriched with correlation parameters as `blueTeamMetadata` key-value pair. The objects were then programmatically extracted with GraphQL and utilized in the automation. The correlation process was designed based on the method described in 3.3.1 and tailored to the experiment variations described in Section 3.3.4.

First, the log data was filtered based on the time of the attack by extracting `StartTime` and `StopTime` of a test case from Vectr. As mentioned in Section 2.6, it is important to know timestamp semantics. While the CloudTrail logs could be filtered with the `timestamp` filed, GuadDuty logs could not. The `timestamp` field for GuardDuty logs indicates when the GuardDuty log entry was created, which is not the same as when the event actually occurred. Additionally, there was a time delay between the actual event and log entry creation. To handle this, `event.start` field was filtered on instead of `timestamp` field as the start time. CloudTrail, on the other hand, has `event.created` field, but `timestamp` field gives the correct correlation.

Next, data were filtered based on log sources. As mentioned in Section2.6, log field names may vary between log types. Correlation parameters in `blueTeamMetadata` were sorted per log type as keys to address this. Figure 5 shows an example of `blueTeamMetadata` displayed in Vectr. Values of the log type keys include identity-based parameters and/or unique correlation parameters. The identity-based parameter is the name of the IAM user created for the attack executions. To maintain consistency and mitigate bias, IOAs from Stratus detection recommendations were used as unique correlation parameters in *all* test cases. A complete overview of the correlation parameters per test case can be found in Table 11, Appendix D.

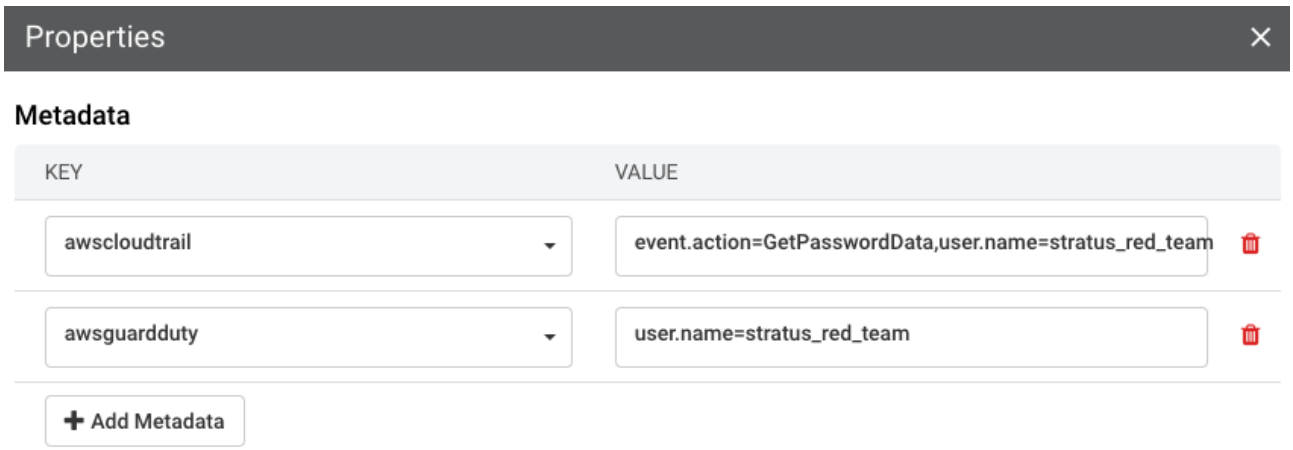


Figure 5: `blueTeamMetadata` example in Vectr

The automation script is written in Python, and the code can be found in Appendix E. It builds an Elasticsearch query based on metadata, start and stop times extracted from Vectr with GraphQL, and generates a URL pointing to the search results in the Kibana dashboard. The *Outcome Notes* of the test case in Vectr is then updated with the URL.

5 Results

Results from both the interview and experimental phases of the study are presented in this chapter. Where necessary, additional explanations are provided to aid in understanding the context or nature of the data. The chapter serves as the foundation for interpreting the finding in the subsequent chapter. The goal of the separation is to maintain the objectivity of the research process and allow for a clear distinction between the presentation of the findings and their interpretation. First, the results from the interview phase are outlined, followed by the findings from the experimental phase of the research. Each section provides a detailed account of the results collected throughout the study.

5.1 Interview Results

5.1.1 Q1 - Experience

The data collected from the interview participants regarding their experience in the field shows a wide range of experience levels, Table 2. There were 8 participants with experience levels spread across various stages of their careers, varying from 3 years to 20 years.

Table 2: Interview Question 1 Results - Years of Experience

Answer	N	Participant
3 years	1	P08
4 years 8 month	1	P01
6 years	1	P07
10 years	1	P04
14 years	1	P06
15 years	1	P02
20 years	2	P03, P05

5.1.2 Q2 - Manual Correlation Time

Figure 6 shows a representation of each participant's answers regarding manual correlation time. The data is sorted in ascending years of experience on the x-axis, and *most likely* within *minimum-maximum* time range in minutes on the y-axis. A broad gap between answers about minimum and maximum time spent resulted in a somewhat compressed representation. Therefore a section of the graph zoomed-in on the lower part of the y-axis is provided on the right. The results are also summarized with descriptive statistics in Table. 3.

Figure 6: Most Likely Time with Min-Max Range for Each Participant

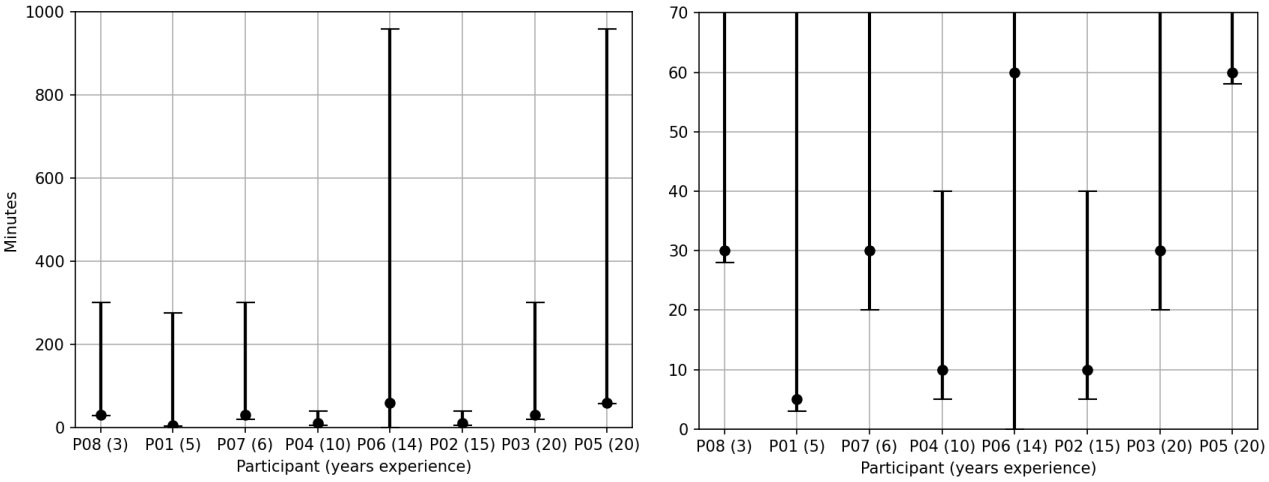


Table 3: Interview Questions 1 & 2 - Descriptive Statistics

	Experience	**Min**	**Most Likely**	**Max**
mean	11.62	12.00	29.38	367.50
std	6.65	19.68	21.45	344.95
min	3.00	2.00	5.00	30.00
max	20.00	60.00	60.00	900.00

5.1.3 Q3 - Most Common Correlation Parameters

Table 4 displays the results from interview question 3 about correlation parameters that participants said to be most common. Data analysis of the answers provided four key themes. Identity-based data was the most commonly mentioned theme, discussed by seven participants (P02, P03, P04, P05, P06, P07, and P08). Log sources and timestamps were each mentioned by five participants, with some overlap in the participant pool. Finally, IOAs and IOCs were mentioned by three participants (P01, P02, and P04).

Table 4: Themes for Interview Question 3 - Most Common Correlation Parameters

Themes	**N**	**Participant**
Identity-based data	7	P02, P03, P04, P05, P06, P07, P08
Log sources	5	P01, P02, P03, P05, P08
Timestamps	5	P02, P03, P04, P07, P08
IOAs/IOCs	3	P01, P02, P04

5.1.4 Q4 - Most Common Correlation Challenges

Table 5 presents the results from interview question 4 investigating the most common correlation challenges. Aggregated data provided nine themes. Time consumption and not knowing what to look for were the most frequently mentioned challenges, each cited by five participants. Noise and lack of experience were each identified as challenges by four participants. Log normalization was mentioned by three participants, while log availability and quality, as well as a large volume of log data, were each mentioned by two participants. Challenges related to log collection and timestamps being off were the least mentioned, with one participant citing each.

Table 5: Themes for Interview Question 4 - Most Common Correlation Challenges

Themes	**N**	**Participant**
Time consuming	5	P03, P04, P05, P06, P07
Not knowing what to look for	5	P02, P03, P04, P05, P06
Noise	4	P03, P04, P07, P08
Lack of experience	4	P02, P03, P06, P07
Log normalization	3	P02, P06, P07
Log availability and quality	2	P02, P06
Large volume of log data	2	P02, P06
Log collection	1	P06
Timestamps being off	1	P02

5.2 Experiment Results

5.2.1 Automated Correlation Accuracy

Figure 7 illustrates confusion matrix values from the results of each experiment. Table 6 lists precision, recall, and F1 score calculated for each method.

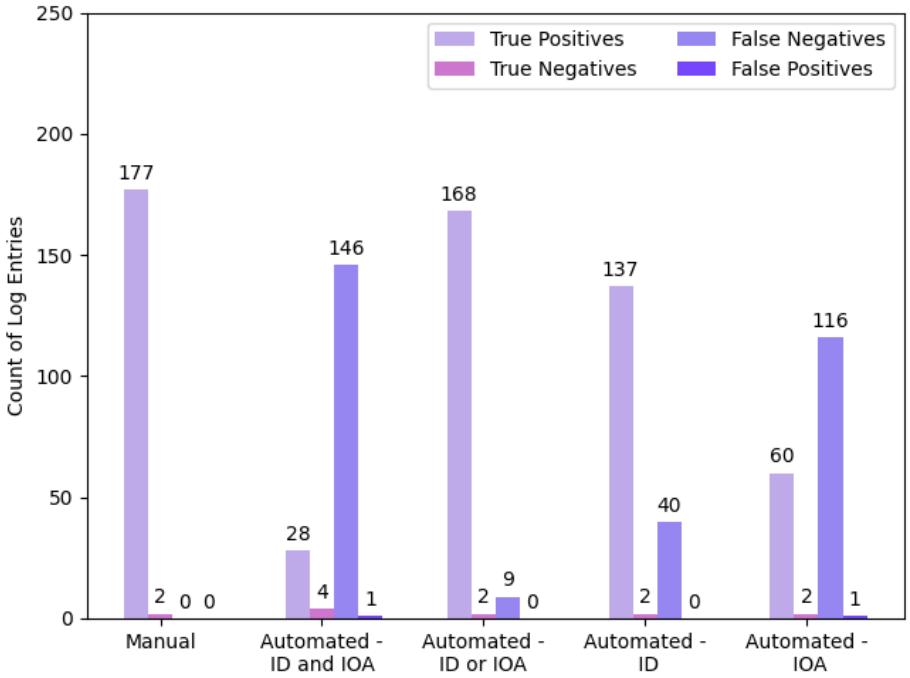


Figure 7: Confusion Matrix Results from the Experiments

Table 6: Comparison of Each Method's Accuracy

Experiment	Correlation Method	Precision	Recall	F1 Score
1	Automated (ID ∩ IOA)	0.97	0.16	0.28
2	Manual	1.00	1.00	1.00
3	Automated (ID)	1.00	0.77	0.87
4	Automated (IOA)	0.98	0.34	0.51
5	Automated (ID ∪ IOA)	1.00	0.95	0.97

Experiment 1 utilized the initially proposed method correlating log instances that meet all four correlation parameters as described in Section 3.3.1. This approach yielded a high precision of 0.97 but had a low recall of 0.16, resulting in the lowest F1 score of 0.28. Test cases 1, 9, and 20 had 0 precision and recall due to the overly specific nature of the correlation method. For example, in addition to time and log type parameters, test case 1 *"Retrieve EC2 Password Data"* utilized the following ID and IOA parameters:

- `user.name=stratus_red_team`
- `event.action=GetPasswordData`

There were, in total, 32 log entries relevant to the attack. Only 2 of those were executed by the specified user, but since none were `GetPasswordData` they were undetected. The other 30 instances were `GetPasswordData`, but since none of them were performed by `stratus_red_team` they were not detected either. This is a typical "attack via proxy" example discussed further in Section 6. Additionally, there are several cases where log entries with specified `event.action` (IOAs) executed with `stratus_red_team` user were successfully detected. However, often an attack includes several steps or actions conducted by the same user. For example, the *stratus_red_team* user could perform actions like *DescribeAttributes* or *AssumeRole*, but since they did not match specified IOA, they went undetected.

Experiment 2 employed a manual correlation and formed ground truth, serving as a reference for evaluating the accuracy of all automated correlation methods. It has perfect scores in all three metrics, with precision and recall due to the nature of the experiment. Since attacks were executed in an isolated environment, they produced feasible log volumes which could be inspected one by one, ensuring 100% accurate correlation.

Experiment 3 focused on ID-based correlation. Although it yielded a perfect precision of 1.00, the recall was 0.77. Test cases, such as the previous example of *"Retrieve EC2 Password Data"*, where `event.action` was performed by an IAM role assumed by `stratus_red_team`, resulting in false negatives. The method disregards actions performed by anything other than `stratus_red_team` used. Despite these limitations, the F1 score for this experiment was relatively high at 0.87, indicating a strong balance between precision and recall.

Experiment 4 focused on IOA-based correlation. The precision was high at 0.98, but the recall was significantly lower at 0.34. As mentioned above, an attack often consists of several actions besides the specific IOA action. Relying solely on IOA failed to capture those, leading to a lower F1 score of 0.51.

Experiment 5 (ID ∪ IOA) produced the best results across all metrics with a precision of 1.00, a recall of 0.95, and an F1 score of 0.97. The few undetected instances were due to proxy attacks where the attacker's identity was not directly observable and action did not match the IOA parameter.

In conclusion, the results demonstrate the effectiveness and limitations of different variations of the correlation method. Automated methods that use ID or a combination of ID and IOA parameters where only one of the conditions has to be met show promising results. However, further refinement is needed to improve their detection capabilities. These results and the associated challenges are further discussed in Section 6.

5.2.2 Automated Correlation Efficiency

Table 7 presents correlation time results from the experiments, comprising minimum, maximum, and mean times. For the experiments with automated correlation, the time was measured from the start of the automation script to the URL containing correlation results was posted to "*Outcome Notes*" in Vectr.

Table 7: Correlation Time Results

Experiment	**Correlation Method**	**min**	**max**	**mean**
1	Automated (ID ∩ IOA)	554 ms	813 ms	671 ms
3	Automated (ID)	564 ms	709 ms	602 ms
4	Automated (IOA)	559 ms	798 ms	688 ms
5	Automated (ID ∪ IOA)	547 ms	758 ms	655 ms

6 Discussion and Future Work

This chapter interprets the findings reported in the previous chapter. The first sections discuss findings from the interviews and experiments, respectively. Limitations, validity, and future work recommendations are considered throughout the discussions, and are consolidated into individual subsections at the end of the chapter.

6.1 Discussion of Interview Findings

6.1.1 Q1 - Experience

The data collected from the interview participants regarding their experience in the field shows a wide range of experience levels. There were 8 participants, with experience levels spread across various stages of their careers, varying from 3 years to 20 years.

The representation of experience offers a comprehensive view of the field, encompassing insights from relatively new entrants (like P08 with 3 years of experience) to seasoned professionals (such as P03 and P05 with 20 years of experience each). The diversity in years of experience allows for a more nuanced understanding of the issues being studied, as it likely brings forth varied perspectives. However, there were only 8 participants, so each experience category has only one or two participants. Thus, while the spread of years of experience is quite broad, the number of data points within each experience level is relatively low. A larger sample size within each experience category would provide a more representative set of responses.

Additionally, while the diversity in the years of experience adds depth to the analysis, it should be noted that all participants are from the same company, mnemonic AS. This could introduce a potential bias in the findings, as they might reflect company-specific practices, culture, or perspectives. However, mnemonic is one of Europe's top leading companies in the field, and their perspectives and experiences are likely to be shaped by working in a highly reputable and potentially cutting-edge environment. Their extensive years of experience in the field, combined with the company's leading position in the industry, are likely to provide insights that are not only rich but also at the forefront of the cybersecurity landscape. Therefore, while the findings might not fully represent the broader field, they can be seen as a reflection of the practices and perspectives at the industry's top tier. This brings a certain degree of weight and authority to the results.

6.1.2 Q2 - Manual Correlation Time

Interviews provided valuable insights into the manual correlation time of the cybersecurity experts. The data presented a nuanced view of how the experience impacted the time spent on manual correlation. The results suggest a general trend of increased correlation time with more experience, which was surprising. However, due to the limited sample size, it is not possible to conclusively establish this correlation. A potential explanation for this observed tendency is that more experienced analysts may engage in a deeper analysis and spend more time on the task, as they possess a broader understanding of various attack scenarios and their nuances. Nonetheless, further investigation with a larger sample size would be necessary to confirm this hypothesis and provide more definitive insights into the relationship between analyst experience and manual correlation time.

Representing the 'most likely' time within the 'minimum-maximum' time range revealed significant variance in the time taken for manual log correlation, regardless of the years of experience. This broad response gap pointed to a high degree of variability in the correlation process, hinting at its complex nature and the potential influence of other factors beyond experience alone. The mean time for minimum, most likely, and maximum time spent on correlation were 12.00 minutes, 29.38 minutes, and 367.50 minutes, respectively. The standard deviation for these times showed a high dispersion, particularly for the maximum time spent, again underlining the variability in the time taken for manual log correlation.

The variability in correlation time could be explained by a multitude of factors, including the complexity of the logs, the severity of the security incidents, and the tools and methods used for correlation. The high standard deviation values, especially for the maximum time spent on correlation (344.95 minutes), highlighted the presence of outliers, pointing towards instances where the manual correlation process was exceptionally time-consuming. These outliers could potentially represent complex cybersecurity incidents requiring significant manual effort for effective correlation.

In summary, the findings suggest that manual correlation is a time-consuming process. The time variability among answers pointed towards the complexity of the task and the potential for optimization and automation. The answers provided a reference point for evaluating the effectiveness of the automated correlation methods, which was the focus of RQ4. By establishing a baseline of manual correlation time, it became possible to make a comparison with the times achieved by automated processes.

While the data gathered provided a valuable reference point for evaluating the efficiency of automated correlation methods, a more precise comparison could have been achieved if interview question 2 had specifically requested manual correlation times for atomic attack scenarios. The automated methods in the experiments were applied to atomic test cases — single, isolated instances of attack tactics, which are less complex than multi-stage attacks. A more direct and accurate comparison between manual and automated methods could have been created by asking participants to estimate manual correlation times for similar atomic cases. Despite the potential imprecision in interview question 2, it did not prove detrimental to answering RQ4. The automated correlation methods employed in the experiments were found to outperform even the minimum manual correlation time provided by the interviewees, as discussed in Section 6.2.

6.1.3 Q3 - Most Common Correlation Parameters

The analysis of the responses to Interview Question 3, which focused on the most common correlation parameters, provides valuable insight into the practices employed by cybersecurity professionals in their day-to-day operations and contributes to the understanding of manual correlation processes. The gathered data were aggregated into four major themes: identity-based parameters, log sources, timestamps, and IOAs/IOCs.

The most frequently cited correlation was the use of identity-based data, with seven participants (P02, P03, P04, P05, P06, P07, and P08), suggesting that professionals rely heavily on information linked to specific identities, such as IP addresses, user IDs, hostnames or device identifiers, in their correlation tasks. The prominence of identity-based data as a correlation parameter can be attributed to its central role in identifying and tracing cyber threats. If a host or account is compromised, the attacker will likely use it to carry out additional malicious activities. Hence, one of the first steps would be to look for other activities originating from the same identity (e.g., hostname, IP address, user account) to track the full extent of an intrusion and understand its scope. Similarly, unusual activities from a particular user or IP address can signal a compromised account or device, making identity data vital in detecting anomalies.

As the cyber threat landscape evolves, sophisticated attackers become increasingly adept at evading identity-based detection. They employ advanced techniques such as attacks via proxies, VPNs, botnets, or even hijacking legitimate identities to mask their actions and remain undetected, making distinguishing normal operations from malicious ones based solely on identity-based parameters challenging. However, In purple teaming exercises, attacks are conducted in a controlled, visible

manner, which allows for the effective utilization of identity-based correlation. Nonetheless, it has limitations, even with the advantage of "transparency" in purple teaming exercises, as discussed in Section 6.2.

The next most commonly cited parameters were log sources and timestamps, each mentioned by five participants, albeit with some overlap in the participant pool. The sheer volume and diversity of log data generated in a typical enterprise environment can be overwhelming. By narrowing down the scope of analysis based on log source types, analysts can focus their attention on the relevant data subsets, making the correlation task significantly more manageable. Furthermore, timestamps play a crucial role in correlation by allowing the chronological sequencing of events, which is vital in piecing together the narrative of a security incident.

Finally, three participants (P01, P02, and P04) identified IOCs and IOAs as common correlation parameters. IOCs and IOAs are key components of threat intelligence. IOCs provide information about previous attacks, giving insight into the digital traces or "fingerprints" that malicious activities leave behind in a system. This information can be used to detect if a system has been compromised in a similar way. On the other hand, IOAs focus on identifying the tactics, techniques, and procedures (TTPs) of potential threats, providing early warning signs of an attack. The utility of correlating based on IOCs and IOAs is only as good as the threat intelligence available. Without up-to-date, accurate, and relevant threat intelligence, the effectiveness of correlation based on IOCs and IOAs is significantly diminished. It is also worth mentioning that while IOCs and IOAs are valuable for detecting and preventing known threats, they may be less effective against novel, previously unseen attacks.

6.1.4 Q4 - Most Common Correlation Challenges

The time-consuming nature of correlation emerged as one of the most significant challenges, as mentioned by five participants. This speaks to the complexity and depth of the task and reinforces the importance of efficient and automated correlation methods, as they could alleviate this burden and free up valuable time for other tasks. As discussed in interview question 2, manual correlation often involves a significant time commitment. The professionals must sift through voluminous log data, identify patterns, and establish connections between seemingly disparate events to uncover potential security threats. This process can be extremely time-consuming, even for experienced professionals, as they must manually analyze and correlate each data piece.

Furthermore, the time commitment is not merely a function of the volume of data but also its di-

versity. Cybersecurity professionals are dealing with a multitude of data types, formats, and sources - each requiring a unique approach for effective analysis. The responses to interview question 2 provided tangible evidence of the time-consuming nature of manual correlation. The professionals reported spending a significant range of time on manual correlation tasks, with an average "most likely" time of 29.38 minutes. These findings highlight the urgency for developing and implementing more efficient, perhaps automated, correlation methods that can help reduce the time burden and allow cybersecurity professionals to focus on other critical aspects of their roles.

Five participants noted the challenge of not knowing what to look for. This challenge could be attributed to a lack of experience identified by four participants (P02, P03, P06, P07) and inadequate threat intelligence. Lack of experience can impact an analyst's ability to identify key patterns and correlations. Experienced analysts often develop an intuitive understanding of what represents normal behavior within a system and can more readily identify anomalies. They are also more adept at recognizing the signs of common attack vectors. There might also be a factor of familiarity with logs and correlation tools. As mentioned earlier, a lack of or adequate threat intelligence can also influence the correlation process. Without up-to-date information about the latest threats, including IOCs and IOAs, even experienced professionals may struggle to know what to look for.

Four participants identified the noise as a challenge. In today's live environments, a sheer volume of data is generated daily. When correlating on a particular parameter, there will likely be many instances of the same type of traffic or activity that are benign. This 'noise' can obscure real threats and lead to false positives, where normal activities are mistakenly flagged as malicious, and false negatives, where actual threats go unnoticed because they're lost in the sea of benign data.

Log-related challenges were most frequently mentioned, including log collection, normalization, availability and quality, and timestamps being off. This highlights the complexity inherent in log analysis and threat correlation tasks, which we discussed in Section 2.6. Collecting logs from diverse sources across an enterprise network requires efficient coordination and integration of various systems and protocols, each with unique configurations and requirements. Normalizing logs of various formats into a consistent, standardized form for analysis adds another layer of complexity. Ensuring continuous access to relevant, accurate, and complete logs requires sophisticated log management systems and processes. Time zone differences, system clock discrepancies, inaccuracies, and semantics introduce an additional level of intricacy. These complexities emphasize the need for advanced skills, knowledge, threat detection, and log analysis tools.

6.2 Discussion of Experiment Findings

To answer RQ3: the automation of attack correlation in purple team exercises is indeed possible, as evidenced by the results of the experiments, although some methods yield more effective results than others. The low recall in Experiment 1 (0.16) reflects the key challenge with over-specific parameters, missing malicious activities that do not align neatly with predefined correlation parameters. Experiments 3 and 4 highlight the limitations of correlating based on a single parameter, be it ID or IOA.

The IOA-based correlation method yields unsatisfactory results (experiment 4) due to its design, which focuses on detecting only the most critical instances to minimize noise within threat detection systems. As discussed in Section2.6, the ideal correlation outcome in purple team exercises involves correlating *all* traces of an attack. In the context of purple teaming, even commonplace activities such as `DescribeInstanceAttribute`, which occurs in almost every test case, can provide invaluable insights into the nuances of daily operations and help avoid pitfalls in detection rule engineering. For instance, when crafting a detection rule for reconnaissance attack tactics, incorporating the `DescribeInstanceAttribute` could lead to an abundance of false positives and unnecessary noise. However, by correlating *all* instances of an attack in purple teaming exercises, a more comprehensive understanding of the cyber threat landscape is gained, providing the knowledge to prevent such inaccuracies.

Moreover, the effectiveness of IOA-based correlation is significantly dependent on the quality and extent of threat intelligence regarding the attack. It is important to note that the experiments in this study were based on threat intelligence provided by the Stratus Red Team. As deliberated in Section 3.3.3, this could introduce a degree of bias. Furthermore, threat intelligence often tends to focus more on detailing the execution of an attack rather than its detection, a common scenario in Red Team exercises. In such cases, it becomes particularly critical for the Blue Team to have a complete picture of the attack instances and context to develop the most effective detection strategies.

Identity-based correlation (experiment 3) provides better results than IOA-based correlation in the purple team exercise as it captures all instances of an attack except for the attacks via proxy, where another identity executes activity than the one the correlation method relays on, as evident in the example of test case 1 "*Retrieve EC2 Password Data*". In such cases, the IOA-based detection yields better results, as it captures the most critical instances of an attack and does not rely on identity.

Experiment 5 demonstrated the most promising approach to automated correlation. It achieved near-perfect scores across precision, recall, and F1 scores by correlating based on either ID or IOA, allowing for a broader yet accurate capture of attack instances. Nonetheless, this method faced the same propagating challenge as detecting proxy attacks, which are complex and often employed in real-world threats. Additionally, it is essential to note that the method has yet to be tested in a live and noisy environment. A correlation method that works perfectly in a lab environment may not function as expected in a production environment with much noise and the potential for false positives. While ID-based correlation is likely to have the same results, as long as no other activity is conducted with the same user, IOA-based correlation, on the other hand, is prone to generating false positives in a live environment. For example, test case 11 "*Download EC2 Instance User Data*" had an IOA correlation parameter `event.action=DescribeInstanceAttribute`. In the isolated lab environment, where there was only one instance correlated by this parameter. `DescribeInstanceAttribute` is a prevalent event action often executed as a part of normal operations in an AWS environment, so in a live environment, it will most likely result in more false positives than in an isolated environment. It is therefore recommended as the next step to test the two experiments that yielded the best results (3 and 5) in a live environment.

Automated correlation across various methods demonstrated exceptional time efficiency, with relatively minor variations. The high efficiency level can be attributed to the correlation not requiring fetching actual log data but instead generating a Kibana URL linked to the results. As a result, the time spent on correlation is largely contingent upon the Vectr API's response time and the execution time of the script rather than any specific correlation parameters. The time taken for automated correlation was consistently below one second during the experiments. This rapid performance far exceeds the mean and minimum manual correlation times of 29.38 and 12.00 minutes documented during interviews, accentuating the robust speed advantages of automated methods.

It is important to highlight, however, that although a Kibana URL link is promptly generated and posted to Vectr, the actual availability of the logs within the ELK stack may experience delays. These delays are induced by factors external to the purview of this study, such as network latency, bandwidth constraints, system configurations, and processing power. These external factors and potential delays should be considerations when implementing such methods in a real-world context and could be investigated further in the future. Nonetheless, the experiment results indicate that automated correlation methods provide a significantly more efficient alternative to manual approaches.

Automated correlation's effectiveness hinges on the uniformity of correlation parameters in Vectr, which should match the format found in the logs. This challenge is not exclusive to purple teaming correlation but extends to detection and log systems at large (as discussed in Section 2.6). The diversity of correlation parameters across different systems, applications, and devices adds a layer of intricacy that could be explored in the future. The study conforms to the formatting of AWS logs and the Elastic Common Schema. To extend the correlation approach to systems other than the ELK stack, it is advisable to explore strategies for parameter mapping and normalization that integrate with a range of log systems. However, this study does not delve into this issue due to its broad application in log systems.

6.3 Limitations of the Study

The identified potential constraints and limitations in the study that could possibly influence the outcomes or deductions are as follows:

- **Novelty of the Purple Teaming Concept and Limited Preceding Research:** Purple Teaming has yet to be extensively explored as a nascent concept in cybersecurity, resulting in a limited body of research available that could serve as a robust foundation for this study. The need for more comprehensive scholarly work on the topic might challenge the broadness of the results, particularly concerning RQ1 and RQ2, and some findings may require validation as the field matures. Nonetheless, this emphasizes the need for further research of its underlying principles, methodologies, and tooling.
- **Interview Sample Size and Profile:** This study builds upon insights gathered from a limited set of interviews with cybersecurity professionals from the same company, as detailed in Section 6.1.1. Nevertheless, the diversity in terms of years of experience and the interviewees being from the industry's top tier adds depth to the findings.
- **Third-Party Tooling:** This study utilized the existing purple teaming platform, Vectr. Consequently, any limitations, bugs, or constraints inherent in this platform could potentially exert influence on the implementation of the correlation method. Alterations or updates to the Vectr platform could also impose implications on the reproducibility of the study. Certain shortcomings in Vectr were brought to light during the experiments, such as an incomplete API, limited mapping to the MITRE ATT&CK framework, limited automation capabilities, and a constraint to atomic test cases, thereby stressing the need for further research of purple teaming tooling.

Furthermore, the Stratus Red Team self-contained binary was utilized in the adversary simulations, a factor which, as discussed in Section 3.3.3, could also influence the experimental results.

- **Atomic Test Cases:** The experimental design utilized atomic test cases, inherently restricting the simulation of consecutive events typically observed during a cyber attack. This constraint emanates from the atomic structure of the Vectr test cases (discussed in Section 2.5) and the Stratus Red Team library (discussed in Section 3.3.3) and thus should be taken into consideration when evaluating the scope and applicability of the experimental results.
- **Testing Environment:** The research experiments were executed within a managed, isolated laboratory setting. While providing substantial control over experimental variables, this methodological approach does not fully replicate the intricate and dynamic circumstances inherent to real-world production environments, a constraint discussed in Section 3.3.2. Furthermore, the experiments were conducted within a cloud-based environment, potentially limiting the applicability of findings to non-cloud attack vectors or solutions that accommodate local hosting.
- **Format of Correlation Parameters:** Automated correlation methods tested in this study are contingent upon the consistency of correlation parameters entered in Vectr, as discussed in Section 6.2. Given that this is a general challenge extending to detection and log systems at large (as discussed in Section 2.6), this study does not delve into this particular issue.

6.4 Recommendations and Future Work

In light of the findings, several recommendations and potential avenues for future research are proposed:

- **Testing in Different Environments:** Future research could aim to validate the correlation method in a live production environment. This could provide a more realistic assessment of the method's performance and help to identify potential real-world issues not present in the controlled, isolated laboratory environment used in this study. Additional testing could be performed in different types of environments, such as local servers or hybrid cloud environments, to assess the robustness and generalizability of the correlation method.
- **Utilizing Correlated Data for Attack Pattern Recognition:** The correlated data produced by the correlation method could be used to train machine learning models for attack pattern recognition, contributing to more effective detection engineering.

- **Development of Purple Teaming Platform and Tooling:** There is a clear need for more comprehensive, seamless, and automated tooling to support purple teaming exercises. Future work could aim to develop improvements to existing tools such as Vectr or explore alternative solutions that better meet the needs of purple teams.
- **Correlation Parameter Mapping:** Future research could aim to develop strategies for parameter mapping and normalization to overcome the challenge of diverse correlation parameters across different systems. This could facilitate the integration of the correlation method with a more comprehensive range of log systems and contribute to more effective and efficient correlation.
- **Support of Advanced (Non-Atomic) Attacks:** As cyber threats continue to evolve in complexity, there is a need for methods that can effectively correlate more advanced multi-stage attacks. Future work could aim to develop such methods or explore a balanced approach that effectively integrates automation with human expertise in purple teaming exercises.

7 Conclusion

Purple Teaming represents an evolutionary step in cybersecurity, incorporating Red and Blue Teaming elements to create a more synergistic approach to identifying and mitigating security threats. The key innovation in Purple Teaming is the focus on collaboration and communication. Red and Blue teams no longer work in isolation but instead, share information and collaborate closely throughout the exercise, leading to more efficient and effective detection and remediation of vulnerabilities. The concept has become quite popular in the governmental and private sectors. However, it is yet to be extensively explored, resulting in a limited body of research available and a limited number of open-source tools to support the collaborative activities of purple teaming. The scarcity of literature and tooling emphasizes the need for further research on purple teaming and its underlying principles, methodologies, and tooling.

Through interviews with cybersecurity professionals, this study highlighted the most common challenges of manual correlation of events, emphasizing the need for further exploration and innovative tooling in purple teaming and threat detection in general. Interview findings revealed how time-consuming and tedious the manual correlation of events is. Through experiments, it was demonstrated that this task can successfully be automated. Several correlation methods were proposed based on the most common correlation parameters provided by the interviewees, and a proof-of-concept tooling was developed to validate them. Methods were rigorously tested and compared within a controlled cloud environment. Notably, one particular approach stood out with exceptional results in accuracy and efficiency. This promising outcome highlights the potential for improving the manual correlation process through automation and offers a clear path for further research and development.

References

Ajmal, A. B., Shah, M. A., Maple, C., Asghar, M. N., & Islam, S. U. (2021). Offensive security: Towards proactive threat hunting via adversary emulation. *IEEE Access*, *9*, 126023–126033.

Atomic Red Team. (n.d.). GitHub. Retrieved 2023-05-02, from `https://github.com/redcanaryco/atomic-red-team`

Babayeva, G., Maennel, K., & Maennel, O. M. (2022). Building an ontology for cyber defence exercises. In *2022 ieee european symposium on security and privacy workshops (euros&pw)* (pp. 423–432).

Blue team definition. (n.d.). In *Computer Security Resource Center*. National Institute of Standards and Technology. Retrieved 2022-11-22, from `https://csrc.nist.gov/glossary/term/blue_team`

Booz, A. H. (2020). The future is purple: Creating a proactive cyber threat program.. Retrieved 2022-09-01, from `https://www.sans.org/white-papers/39190/`

Brilingaitė, A., Bukauskas, L., Juozapavičius, A., & Kutka, E. (2020). Information sharing in cyber defence exercises. In *European conference on cyber warfare and security* (pp. 42–49).

Brilingaitė, A., Bukauskas, L., Juozapavičius, A., & Kutka, E. (2022). Overcoming information-sharing challenges in cyber defence exercises. *Journal of cybersecurity*, *8*(1), tyac001.

Carey, M. J., & Jin, J. (2020). *Tribe of hackers blue team: Tribal knowledge from the best in defensive cybersecurity*. John Wiley & Sons.

Chaplinska, S., et al. (2022). A purple team approach to attack automation in the cloud native environment. Aalto University.

Chhajed, S. (2015). *Learning elk stack*. Packt Publishing Ltd.

Chowdhury, S. (2019). *Perceptions of purple teams among cybersecurity professionals* . Master thesis, Purdue University Graduate School.

Dale, C. (2019). Red, blue and purple teams: Combining your security capabilities for the best outcome. In *SANS White Papers*. SANS Institure. Retrieved 2022-11-22, from `https://www.sans.org/white-papers/39190/`

Definition of purple. (2022). In *Oxford Advanced American Dictionary*. Oxford University Press. Retrieved 2022-11-21, from `https://www.oxfordlearnersdictionaries.com/definition/american_english/purple`

Diogenes, Y., & Ozkaya, E. (2022). *Cybersecurity - attack and defense strategies: Improve your security posture to mitigate risks and prevent attackers from infiltrating your system*. Packt Publishing

Ltd.

Ickler, K., & Orchilles, J. (2020). Atomic purple team framework and lifecycle. GitHub. Retrieved 2022-09-29, from https://github.com/DefensiveOrigins/AtomicPurpleTeam

Ilca, L. F., & Balan, T. (2021). Purple team security assessment of firmware vulnerabilities. In *International incollection on remote engineering and virtual instrumentation* (pp. 370–379).

Lee, R. M. (2020). SANS Cyber Threat Intelligence (CTI) Survey 2020. In *SANS White Papers*. SANS Institure. Retrieved 2022-11-23, from https://www.sans.org/white-papers/39395/

Mansfield-Devine, S. (2018). The best form of defence–the benefits of red teaming. *Computer Fraud & Security*, *2018*(10), 8–12.

MITRE ATT&CK. (2015-2022). In *MITRE ATT&CK online knowledge base*. The MITRE Corporation. Retrieved 2022-11-21, from https://attack.mitre.org/

Oakley, J. G. (2018). *Towards improved offensive security assessment using counter apt red teams*. Towson University.

Oakley, J. G. (2019). *Professional red teaming: Conducting successful cybersecurity engagements*. Apress.

Olsen, X. (2022). *Enterprise purple teaming: An exploratory qualitative study* . PhD dissertation, Marymount University.

Orchilles, J. (n.d.).

In *Jorge Orchilles' presentation slides*. SlideShare from Scribd. Retrieved 2022-11-21, from https://www.slideshare.net/jorgeorchilles/presentations

Orchilles, J. (2021). Purple Team Exercise Framework 2.0 (PTEF). In *SCYTHE online download library*. SCYTHE Inc. Retrieved 2022-11-24, from https://www.scythe.io/ptef

Paper, P. W. (2021). The power of purple teaming. CyberRisk Alliance. Retrieved 2022-11-21, from https://plextrac.com/whitepaper/the-power-of-purple-teaming/

Plextrac - purple teaming tool. (n.d.). Retrieved 2022-11-22, from https://plextrac.com/

Red team definition. (n.d.). In *Computer Security Resource Center*. National Institute of Standards and Technology. Retrieved 2022-11-22, from https://csrc.nist.gov/glossary/term/red_team

Rehberger, J. (2020). *Cybersecurity attacks–red team strategies: A practical guide to building a penetration testing program having homefield advantage*. Packt Publishing Ltd.

Reiber, J., Opel, B., & Wright, C. (2021). *Purple teaming for dummies, attackiq special edition*. John Wiley & Sons.

Routin, D., Thoores, S., & Rossier, S. (2022). *Purple team strategies - enhancing global security posture*

through uniting red and blue teams with adversary emulation. Packt Publishing Ltd.

Saarainen, V. (2021). *Red teaming : Regulatory and non-regulatory frameworks used in adversarial simulations* . Bachelor's thesis, South-Eastern Finland University of Applied Sciences.

Scythe - purple teaming tool. (n.d.). Retrieved 2022-11-22, from http://scythe.io/

Seker, E., & Ozbenli, H. H. (2018). The concept of cyber defence exercises (cdx): Planning, execution, evaluation. In *2018 international conference on cyber security and protection of digital services (cyber security)* (pp. 1–9).

Stratus Red Team. (n.d.). Retrieved 2023-05-02, from https://stratus-red-team.cloud/

Svacina, J., Raffety, J., Woodahl, C., Stone, B., Cerny, T., Bures, M., … Tisnovsky, P. (2020). On vulnerability and security log analysis: A systematic literature review on recent trends. In *Proceedings of the international conference on research in adaptive and convergent systems* (pp. 175–180).

Threat intelligence definition. (n.d.). In *Computer Security Resource Center*. National Institute of Standards and Technology. Retrieved 2022-11-23, from https://csrc.nist.gov/glossary/term/threat_intelligence

Towards a definition of red teaming. (2019). In *The Center for Advanced Red Teaming*. University at Albany, State University of New York. Retrieved 2022-11-22, from https://www.albany.edu/sites/default/files/2019-11/CART20Definition.pdf

TTP Definition. (n.d.). In *Computer Security Resource Center*. National Institute of Standards and Technology. Retrieved 2022-11-23, from https://csrc.nist.gov/glossary/term/tactics_techniques_and_procedures

VECTR - purple teaming tool. (n.d.). Retrieved 022-11-22, from https://vectr.io/

Vykopal, J., Vizváry, M., Oslejsek, R., Celeda, P., & Tovarnak, D. (2017). Lessons learned from complex hands-on defence exercises in a cyber range. In *2017 ieee frontiers in education conference (fie)* (pp. 1–8).

What is ELK Stack? (n.d.). Elasticsearch B.V. Retrieved 2023-03-02, from https://www.elastic.co/what-is/elk-stack

Appendices

Appendix A Interview Script

Intro

This is the interview with Participant number XX. It is XX AM/PM, weekday XX, the XX of March 2023. The participant has been informed about the study's purpose, the interview procedure, confidentiality, and anonymity of the interview. They have also been informed that participation is voluntary and that they can withdraw at any time. For the record, do you consent?

Interview Questions

- (Q0) How many years of experience do you have in cybersecurity?
- (Q1) In your experience, how long does it take to correlate detection data to an attack technique manually?
- (Q2) In your experience, what are the three most common parameters used to correlate detection data to an attack technique?
- (Q3) In your experience, what are the most common challenges with correlating detection data to an attack technique?

Thank you for your participation.

Appendix B Interview Transcripts

B.1 Interview with P01

The Researcher (R): "This is the interview with Participant number one. It is twelve PM, Monday, the 13th of March 2023. The participant has been informed about the study's purpose, the interview procedure, confidentiality, and anonymity of the interview. They have also been informed that participation is voluntary and that they can withdraw at any time. For the record, do you consent?"

The Participant 01 (P01): "Yes."

R: "Thank you. How many years of experience do you have in cybersecurity?"

P01: "Four years and eight months."

R: "Thank you. In your experience, how long does it take to correlate detection data to an attack technique manually? If possible, please answer with numbers of minimum, maximum, and most likely."

P01: "Everything from a couple of minutes to several hours, depending on the type of attack. It is hard to give a specific number. Maybe 5 minutes. So 2 min, 5 min, and say 5 hours."

R: "Thank you. In your experience, what are the three most common parameters used to correlate detection data to an attack technique?"

P01: "Vendor type, log type, and different data based on the indications of compromise for that type of the attack."

R: "Thanks. In your experience, what are the most common challenges with correlating detection data to an attack technique?"

P01: "Knowing all the techniques and deciding on which one fits the best. Or having a complete picture of all the MITRE techniques and then how to map those to the attack. Because what I see is that we, for example, put all into bucket techniques because they are easy to map to that. For example *Valid Accounts*, we see it a lot of because it is easy to map to, but more techniques are not as easily mapped."

R: "Let me see if I understand you correctly. Are you saying it is hard to know what MITRE technique detection data maps to?"

P01: "Yes."

R: "I see. Thank you. I do not have any further questions. Thank you for your participation."

P01: "No problem!"

B.2 Interview with P02

R: " This is an interview with Participant number two. It is Tuesday, the 14th of March, 2023. The participant has been informed about the study purpose, the interview procedure, confidentiality, and anonymity of the interview. They have also been informed that participation is voluntary and that they can withdraw at any time. For the record, do you consent?"

The Participant 02 (P02): "I consent."

R: "Thank you. Could you state how many years of experience you have in cybersecurity?"

P02: "Almost 15 years. I have mainly worked with analysis of log data and network events, also development of detection capabilities and normalization and management of log data."

R: "Thank you. In your experience, how long does it take to correlate detection data to an attack technique manually? If possible, please answer with numbers of minimum, maximum, and most likely."

P02: "That is actually a really difficult question as it will depend on how good your log data is and what kind of attack it is. But normally, it takes about 5min if you are lucky, but it can take longer if you have to search through all the logs. So if you don't have any good indicators, you have to go through all of the data, it takes time. So I would say the maximum time will be about 30 minutes."

R: "And most likely?"

P02: "Most likely 10 minutes.

R: "Thank you. The next question is: in your experience, what are the three most common parameters used to correlate detection data to an attack technique?"

P02: "Timestamp, I would say, is number one. Then identificators such as the source IP or source user. So anything that is identity-based data. And then finally, type of log depending on the type of attack, and like IoCs."

R: "Thanks. And the last question: in your experience, what are the most common challenges with

correlating detection data to an attack technique?"

P02: "One thing we are struggling with often is that we don't get the logs, but if you have the logs, they can be of poor quality. For instance, you are missing some fields to correlate events. If logs are not properly normalized, you must dig through raw logs. Another challenge can be timestamped being off. For example, if you have different time zones or misconfigurations, it could be challenging. And, of course, if you do not have enough knowledge to know what logs or fields to look for, it is a challenge. So lack of experience."

R: "Thank you. I do not have any further questions. Thank you for your participation."

B.3 Interview with P03

R: " This is an interview with Participant number three. It is ten AM, Tuesday, the 14th of March 2023. The participant has been informed about the study purpose, the interview procedure, confidentiality, and anonymity of the interview. They have also been informed that participation is voluntary and that they can withdraw at any time. For the record, do you consent?"

The Participant 03 (P03): "Yes."

R: "Thank you. Could you state how many years of experience you have in cybersecurity?"

P03: "I have 20 years of experience in cybersecurity. Manly working with detection engineering."

R: "Thank you. In your experience, how long does it take to correlate detection data to an attack technique manually? If possible, please answer with numbers of minimum, maximum, and most likely."

P03: "It is highly dependent on the type of the attack and complexity, but it can take anywhere from 10 minutes to several hours, say 4 hours. Most likely about 30 minutes. That is just from the top of my head."

R: "Thank you. The next question is: in your experience, what are the three most common parameters used to correlate detection data to an attack technique?"

P03: "The most important one is always timestamps. Then IP addresses, user name, and hostname. Those are generic properties. Then you have quite often some criteria per log type. For instance, for process logs, you have the command line, but that would not be relevant for other types of attacks. So I would say log source."

R: "Thanks. The next question is: in your experience, what are the most common challenges with correlating detection data to an attack technique?"

P03: "One quite common challenge is that it might seem like you are able to correlate something in a clean demo environment, if like developed a new detection, but once you try to correlate it in the same way in a production environment with normal traffic, you see that the hypothesis of the correlation method might turn out coincidental. Another challenge is the experiences with how logs look like, what fields are of interest, what is normal, and what is not. And, of course, the challenge of it being time-consuming and that it is a lot of manual effort. We don't have the tools to present some suggestions of what could be relevant. That is a challenge."

R: "Thank you. I do not have any further questions. Thank you for your participation."

B.4 Interview with P04

R: "This is an interview with Participant number four. It is twelve PM, Tuesday, the 14th of March 2023. The participant has been informed about the study purpose, the interview procedure, confidentiality, and anonymity of the interview. They have also been informed that participation is voluntary and that they can withdraw at any time. For the record, do you consent?"

The Participant 04 (P04): "Yes."

R: "Thank you. Could you state how many years of experience you have in cybersecurity?"

P04: "10 years."

R: "Thank you. In your experience, how long does it take to correlate detection data to an attack technique manually? If possible, please answer with numbers of minimum, maximum, and most likely."

P04: "5 minutes. Maximum 30min. 10-15min."

R: "Thank you. The next question is: in your experience, what are the three most common parameters used to correlate detection data to an attack technique?"

P04: "Unique identifiers, metadata based on the type of the attack technique, and of course time-based correlation."

R: "Thanks. The last question is: in your experience, what are the most common challenges with correlating detection data to an attack technique?"

P04: "Correct data collection is the biggest challenge. Being able to identify what log sources you need to be able to sport the aspects of an attack. That is very hard. There are always new zero days, and you must look into different types of logs. So the collection of correct data and correlation across different log sources. And also, if you are not testing in a controlled environment, you will have a lot of struggles with a lot of noise. Also, the time spent executing and verifying manually is often very long, so it is time-consuming."

R: "Thank you. I do not have any further questions. Thank you for your participation."

B.5 Interview with P05

R: "This is an interview with Participant number five. It is one PM, Tuesday, the 14th of March 2023. The participant has been informed about the study purpose, the interview procedure, confidentiality, and anonymity of the interview. They have also been informed that participation is voluntary and that they can withdraw at any time. For the record, do you consent?"

The Participant 05 (P05): "Yes."

R: "Thank you. Before we begin, could you state how many years of experience you have in cybersecurity?"

P05: "20 years."

R: "Thank you. The first question is: in your experience, how long does it take to correlate detection data to an attack technique manually? If possible, please answer with numbers of minimum, maximum, and most likely."

P05: "It depends a lot on what kind of detection data you are looking at. So it would vary a lot. So maybe from 2 minutes to hours or even days. But for the kind of middle ground, I would say an hour. It would also depend on tooling etc."

R: "Thank you. The next question is: in your experience, what are the three most common parameters used to correlate detection data to an attack technique?"

P05: "Type of log sources based on the type of technique, identification metadata such as host or user, and timestamp."

R: "Thanks. The next question is: in your experience, what are the most common challenges with correlating detection data to an attack technique?"

P05: "It is time-consuming, the ability to detect or trace the activity, and having to keep up with the threat actors to have the correct insight."

R: "Thank you. I do not have any further questions. Thank you for your participation."

B.6 Interview with P06

R: "This is an interview with Participant number six. It is twelve PM, Tuesday, the 15th of March 2023. The participant has been informed about the study purpose, the interview procedure, confidentiality, and anonymity of the interview. They have also been informed that participation is voluntary and that they can withdraw at any time. For the record, do you consent?"

The Participant 06 (P06): "Yes."

R: "Thank you. Before we begin, could you state how many years of experience you have in cybersecurity?"

P06: "14 years."

R: "Thank you. The first question is: in your experience, how long does it take to correlate detection data to an attack technique manually? If possible, please answer with numbers of minimum, maximum, and most likely."

P05: "Depends on the data but from hours to days. So minimum an hour and maximum about 2 days. And most likely 1 hour."

R: "Thank you. The next question is: in your experience, what are the three most common parameters used to correlate detection data to an attack technique?"

P06: "IP, user and host."

R: "Thanks. The next question is: in your experience, what are the most common challenges with correlating detection data to an attack technique?"

P06: "Normalizing data. The worst part is that it is time-consuming. Takes forever. Then the volumes of logs. And of course, gathering the data, knowing where to look, where the logs are etc."

R: "Thank you. I do not have any further questions. Thank you for your participation."

B.7 Interview with P07

R: "This is an interview with Participant number seven. It is twelve PM, Tuesday, the 15th of March 2023. The participant has been informed about the study purpose, the interview procedure, confidentiality, and anonymity of the interview. They have also been informed that participation is voluntary and that they can withdraw at any time. For the record, do you consent?"

The Participant 07 (P07): "Yes."

R: "Thank you. Before we begin, could you state how many years of experience you have in cybersecurity?"

P07: "6 years."

R: "Thank you. The first question is: in your experience, how long does it take to correlate detection data to an attack technique manually? If possible, please answer with numbers of minimum, maximum, and most likely."

P07: "Depends on the technique. But I would say a minimum of 10 minutes and up to hours, say 4 or 5. And most likely, I would say, 30 minutes."

R: "Thank you. The next question is: in your experience, what are the three most common parameters used to correlate detection data to an attack technique?"

P07: "Account data and asset identifications such as user and host. And time-correlation is also important, so like timestamps."

R: "Thanks. The next question is: in your experience, what are the most common challenges with correlating detection data to an attack technique?"

P07: "Noise, normalization of log data, lack of log experience, and that it is time-consuming."

R: "Thank you. I do not have any further questions. Thank you for your participation."

B.8 Interview with P08

R: "This is an interview with Participant number eight. It is ten AM, Friday, the 17th of March 2023. The participant has been informed about the study purpose, the interview procedure, confidentiality, and anonymity of the interview. They have also been informed that participation is voluntary and that they can withdraw at any time. For the record, do you consent?"

The Participant 08 (P08): "Yes."

R: "Thank you. Could you state how many years of experience you have in cybersecurity?"

P08: "3 years."

R: "Thank you. In your experience, how long does it take to correlate detection data to an attack technique manually? If possible, please answer with numbers of minimum, maximum, and most likely."

P08: "2min. 4-5 hours. 30mins."

R: "Thank you. The next question is: in your experience, what are the three most common parameters used to correlate detection data to an attack technique?"

P08: "Timestamps, ID-based data such as username, event id, etc., and then log source type and device product relevant for the type of the attack."

R: "Thanks. The next question is: in your experience, what are the most common challenges with correlating detection data to an attack technique?"

P08: "The hardest thing is finding relevant data and distinguishing normal traffic or behavior from malicious activity."

R: "Thank you. I do not have any further questions. Thank you for your participation."

Appendix C Interview Results - Manual Correlation Time

Table 8: Interview Question 2 Results - Minimum Manual Correlation Time

Answer	N	Participant
2 minutes	3	P01, P05, P08
5 minutes	2	P02, P04
10 minutes	2	P03, P07
1 hour	1	P06

Table 9: Interview Question 2 Results - Maximum Manual Correlation Time

Answer	N	Participant
30 minutes	2	P02, P04
4-5 hours	4	P03, P07, P08, P01
2 days	2	P05, P06

Table 10: Interview Question 2 Results - Most Likely Manual Correlation Time

Answer	N	Participant
5 minutes	1	P01
10 minutes	2	P02, P04

Table 10: Interview Question 2 Results - Most Likely Manual Correlation Time

Answer	**N**	**Participant**
30 minutes	3	P03, P07, P08
1 hour	2	P05, P06

Appendix D Experiment Correlation Parameters per Test Case

Table 11: Correlation Parameters per Test Case

ID	Test Case	Correlation Parameters
1	Retrieve EC2 Password Data	`event.action=GetPasswordData` `user.name=stratus_red_team`
2	Steal EC2 Instance Credentials	`user.name=stratus_red_team`
3	Retrieve a High Number of Secrets Manager secrets	`event.action=GetSecretValue` `user.name=stratus_red_team`

4	Retrieve And Decrypt SSM Parameters	`event.action=GetParameter`
		`event.action=GetParameters`
		`user.name=stratus_red_team`

5	Delete CloudTrail Trail	`event.action=DeleteTrail` `user.name=stratus_red_team`
6	Disable CloudTrail Logging Through Event Selectors	`event.action=PutEventSelectors` `user.name=stratus_red_team`
7	CloudTrail Logs Impairment Through S3 Lifecycle Rule	`event.action=PutBucketLifecycle` `user.name=stratus_red_team`
8	Stop CloudTrail Trail	`event.action=StopLogging` `user.name=stratus_red_team`
9	Attempt to Leave the AWS Organization	`event.action=LeaveOrganization` `user.name=stratus_red_team`

Table 11: Correlation Parameters per Test Case

ID	Test Case	Correlation Parameters
10	Remove VPC Flow Logs	`event.action=DeleteFlowLogs` `user.name=stratus_red_team`
11	Download EC2 Instance User Data	`event.action=DescribeInstanceAttribute` `user.name=stratus_red_team`
12	Execute Discovery Commands on an EC2 Instance	`user.name=stratus_red_team`
13	Launch Unusual EC2 instances	`event.action=RunInstances` `user.name=stratus_red_team`

14	Execute Commands on EC2 Instance via User Data	`event.action=StopInstances`
		`event.action=ModifyInstanceAttribute`
		`user.name=stratus_red_team`

15	Open Ingress Port 22 on a Security Group	`event.action=AuthorizeSecurityGroupIngress` `user.name=stratus_red_team`
16	Exfiltrate an AMI by Sharing It	`event.action=ModifyImageAttribute` `user.name=stratus_red_team`

17	Exfiltrate EBS Snapshot by Sharing It	`event.action=ModifySnapshotAttribute`
		`event.action=SharedSnapshotCopyInitiated`
		`event.action=SharedSnapshotVolumeCreated`
		`user.name=stratus_red_team`

18	Exfiltrate RDS Snapshot by Sharing	`event.action=ModifyDBSnapshotAttribute` `user.name=stratus_red_team`
19	Backdoor an S3 Bucket via its Bucket Policy	`event.action=PutBucketPolicy` `user.name=stratus_red_team`

Table 11: Correlation Parameters per Test Case

ID	Test Case	Correlation Parameters
20	Console Login without MFA	`event.action=ConsoleLogin` `user.name=stratus_red_team`
21	Backdoor an IAM Role	`event.action=UpdateAssumeRolePolicy` `user.name=stratus_red_team`
22	Create an Access Key on an IAM User	`event.action=CreateAccessKey` `user.name=stratus_red_team`

23	Create an administrative
	IAM User
		`event.action=CreateUser`
		`event.action=AttachUserPolicy`
		`event.action=CreateAccessKey`
		`user.name=stratus_red_team`

24	Create a Login Profile
	on an IAM User
		`event.action=CreateLoginProfile`
		`event.action=UpdateLoginProfile`
		`user.name=stratus_red_team`

25	Backdoor Lambda Function
	Through Resource-Based Policy
		`event.action=AddPermission20150331`
		`event.action=AddPermission20150331v2`
		`user.name=stratus_red_team`

26	Overwrite Lambda Function Code	`event.action=UpdateFunctionCode*` `user.name=stratus_red_team`
27	Create an IAM Roles Anywhere trust anchor	`event.action = CreateTrustAnchor` `user.name=stratus_red_team`

Appendix E Automated Correlation Script

```
import os
from _datetime import datetime


import requests


# Config
api_key = os.environ["API_KEY"]
vectr_gql_url = os.environ["VECTR_GQL_URL"]
target_db = os.environ["target_db"]
elk_host = "http://localhost:5602"
vectr_test_case_ids = [
    "74ca6375-a213-40e0-bc3b-da94fd08c7c5",
    "6ffd1e0c-6068-4db1-906d-f11321c5382b",
    "3ad07124-6b20-470e-b14f-c5e48ac53cfb",
    "2ebbb469-6aa6-4aa9-8fa7-9d835a17b9dd",
    "fa67554e-dcf1-4642-a859-b425ecb25594",
    "d94fb180-f41a-43ca-93f0-34d4407f9206",
    "faeedc95-2773-487d-96f9-82bc81c1d804",
    "7669e2eb-93f4-4943-bb16-e6f8d6e5ff00",
    "2a2be3a1-7223-46eb-ad82-415570d2f984",
    "d55b223e-5282-4627-bb89-8219a35d5a0c",
    "d5b4868b-8529-48e3-994a-39816152eb13",
    "71147858-d06f-4f15-bcea-547783943b8a",
    "650a4116-ba25-4b92-aa40-9b4a8efe5919",
    "eec812d1-1150-4a04-9374-9c28ffaf77cd",
    "5838a886-c29c-4786-b376-14a0c9e9e456",
    "9628c4c6-4762-4ef5-aa07-6d503cea2249",
    "4f89afae-229e-48ba-9f42-d74ca1a4c6a5",
    "bd24b9c2-f94d-4908-bb14-526a709c3527",
    "39e9e799-9712-4ed2-b888-b68eadf34c66",
```

```
    "e0ff158d-007c-48ec-90f1-55ca42c8edd4",
    "9a6119de-19ed-4a7a-9ed9-34378091db86",
    "cebd729d-dbbf-4174-bf08-cd1596785f3f",
    "f91f5199-35fd-48a0-aa44-4e8c079390db",
    "417363c4-e0f2-41b9-b34c-407967e7f027",
    "ceac5b62-7bfa-489b-9ab4-7cbf52e31b5d",
    "77c04fea-c9de-4057-95e7-57fafedcd6dd",
    "e5fc836c-32f0-4502-b486-e58b2b2598a1"
]


def get_test_case_from_vectr(test_case_id: str):
    response = requests.post(vectr_gql_url + '/sra-purpletools-rest/graphql',
                             verify=True,
                             headers={"Authorization": api_key},
                             json={
                                 "query": "query($testCaseId: String!, $db:
                                  ↪  String!) {\n  testcase(id:$testCaseId,
                                  ↪  db:$db) {\n    id, name,
                                  ↪  attackStart{\n\t\t\tcreateTime\n\t\t},
                                  ↪  \n\t\tattackStop{\n\t\t\tcreateTime\n\t\t},
                                  ↪  \n\t\tblueTeamMetadata{\n\t\t\tkey,
                                  ↪  value\n\t\t}\n  }\n}",
                                 "variables": {"testCaseId": test_case_id,
                                  ↪  "db": target_db}})

    # Extract relevant data
    attack_start =
     ↪  response.json()["data"]["testcase"]["attackStart"]["createTime"]
    attack_stop =
     ↪  response.json()["data"]["testcase"]["attackStop"]["createTime"]
    if attack_stop is None or attack_start is None:
        raise ValueError("Start and stop time must be set.")
```

```
    start_time = datetime.fromtimestamp(attack_start / 1000).isoformat()
    stop_time = datetime.fromtimestamp(attack_stop / 1000).isoformat()
    metadata = response.json()["data"]["testcase"]["blueTeamMetadata"]

    return start_time, stop_time, metadata


def add_elk_url_to_vectr(test_case_id: str, url: str):
    # Create note
    outcome_notes = "See ELK at \n" + url
    requests.post(vectr_gql_url + '/sra-purpletools-rest/graphql',
                  verify=True,
                  headers={"Authorization": api_key},
                  json={
                      "query": "mutation ($input: UpdateTestCaseInput!)
                       ↪  {\n\ttestCase {\n\t\tupdate(input: $input)
                       ↪  {\n\t\t\ttestCases
                       ↪  {\n\t\t\t\tid\n\t\t\t\}\n\t\t\}\n\t\}\n\}\n",
                      "variables": {"input": {"db": target_db,
                       ↪  "testCaseUpdates": [
                          {"testCaseId": test_case_id, "outcomeNotes":
                           ↪  outcome_notes}]}}})


def build_query(start_time, stop_time, metadata):
    # Data set queries
    queries = []

    # Build query
    for dic in metadata:
        key = dic["key"]
        value = dic["value"]
        query = "("
```

```
        if key == "awscloudtrail":
            query += "event.dataset%20:%20aws.cloudtrail%20and%20" +
             ↪  "%20and%20".join(
                ["@timestamp%20%3E%3D%22" + start_time + "%22",
                 ↪  "@timestamp%20%3C%3D%20%22" + stop_time + "%22"])
        elif key == "awsguardduty":
            query += "event.dataset%20:%20aws.guardduty%20and%20" +
             ↪  "%20and%20".join(
                ["event.start%20%3E%3D%22" + start_time + "%22",
                 ↪  "event.start%20%3C%3D%20%22" + stop_time + "%22"])
        else:
            continue
        filter_params = {}

        # Group similar keys
        for query_filter in value.split(","):
            key = query_filter.split("=")[0]
            if key not in filter_params:
                filter_params[key] = []
            filter_params[key].append(query_filter.replace("=", "%20:%20"))
        for filters in filter_params.values():
            query += "%20and%20" + "(" + "%20or%20".join(filters) + ")"

        query += ")"
        queries.append(query)

    return "%20or%20".join(queries)


def create_elk_url(start_time, query):
```

```
    return elk_host +
     ↪ "/app/discover#/?_g=(filters:!(),refreshInterval:(pause:!t,value:0),
     ↪ time:(from:'" + start_time +
     ↪ "',to:now))&_a=(columns:!(),filters:!(),index:'logs-*',
     ↪ interval:auto,query:(language:kuery,query:'" + query +
     ↪ "'),sort:!(!('@timestamp',desc)))"


def correlate_test_case(test_case_id: str):
    now = datetime.now()
    print(f"Running for test_case_id={test_case_id} - started at
     ↪ {now.isoformat()}")
    start_time, stop_time, metadata = get_test_case_from_vectr(test_case_id)
    query = build_query(start_time, stop_time, metadata)
    url = create_elk_url(start_time, query)

    # Post to Vectr
    add_elk_url_to_vectr(test_case_id, url)
    print(f"Finished for test_case_id={test_case_id} - took {(datetime.now()
     ↪ - now).microseconds / 1000}ms")


def main():
    # Correlate all test cases
    for test_case_id in vectr_test_case_ids:
        correlate_test_case(test_case_id)


# Run correlation script
main()
```

Appendix F Automated Attack Simulations Script

```
import os
import subprocess
import time
from datetime import datetime

aws_profile = "stratus_red_team"
aws_region = "eu-west-1"
target_file = f'dist/attack_{datetime.now().strftime("%Y%m%d%H%M%S")}.csv'
silent = False
delay = 60
attacks = [
    "aws.credential-access.ec2-get-password-data",
    "aws.credential-access.ec2-steal-instance-credentials",
    "aws.credential-access.secretsmanager-retrieve-secrets",
    "aws.credential-access.ssm-retrieve-securestring-parameters",
    "aws.defense-evasion.cloudtrail-delete",
    "aws.defense-evasion.cloudtrail-event-selectors",
    "aws.defense-evasion.cloudtrail-lifecycle-rule",
    "aws.defense-evasion.cloudtrail-stop",
    "aws.defense-evasion.organizations-leave",
    "aws.defense-evasion.vpc-remove-flow-logs",
    "aws.discovery.exc2-enumerate-from-instance",
    "aws.discovery.ec2-download-user-data",
    "aws.execution.ec2-launch-unusual-instances",
    "aws.execution.ec2-user-data",
    "aws.exfiltration.ec2-security-group-open-port-22-ingress",
    "aws.exfiltration.ec2-share-ami",
    "aws.exfiltration.ec2-share-ebs-snapshot",
    "aws.exfiltration.rds-share-snapshot",
```

```
    "aws.exfiltration.s3-backdoor-bucket-policy",
    "aws.initial-access.console-login-without-mfa",
    "aws.persistence.iam-backdoor-role",
    "aws.persistence.iam-backdoor-user",
    "aws.persistence.iam-create-admin-user",
    "aws.persistence.iam-create-user-login-profile",
    "aws.persistence.lambda-backdoor-function",
    "aws.persistence.lambda-overwrite-code",
    "aws.persistence.rolesanywhere-create-trust-anchor"
]

# Setup environment for attacks
env = {
    'AWS_PROFILE': aws_profile,
    'AWS_DEFAULT_REGION': aws_region
}
env.update(os.environ)

# for attack in attacks:
#     result = subprocess.run(f"/opt/homebrew/bin/stratus cleanup
 ↪  {attack}",
#                             shell=True,
#                             env=env)
# exit(0)

# Create output dir if not exists
if not os.path.exists("dist"):
    os.makedirs("dist")
log_file = open(target_file, "w+")
log_file.write("attack;start;stop;duration;status\n")

# Loop over attack commands
for attack in attacks:
```

```
stdout = subprocess.DEVNULL if silent else None
error = False
warmup_cmd = f"stratus warmup {attack}"
detonate_cmd = f"stratus detonate {attack}"
cleanup_cmd = f"stratus cleanup {attack}"

print(f'')
print(f'Running attack: {attack}')

#################
# Warmup
#################
print(f'    ..Warming up..')
result = subprocess.run(warmup_cmd,
                        shell=True,
                        stdout=stdout,
                        env=env)

if result.returncode != 0:
    error = True
    print(f'    ....FAILED')
    print(f'    ....ROLLING BACK')
    subprocess.run(cleanup_cmd,
                   shell=True,
                   stdout=stdout,
                   env=env)
    log_file.write(f"{attack};NULL;NULL;NULL;ERROR\n")
    continue
else:
    print(f'    ....SUCCESS')
print(f'    ....waiting {delay} seconds')
time.sleep(delay)
```

```
################
# Detonation
################
print(f'    ..Detonating')

start = datetime.utcnow()
result = subprocess.run(detonate_cmd,
                        shell=True,
                        stdout=stdout,
                        env=env)
stop = datetime.utcnow()

if result.returncode != 0:
    error = True
    print(f'    ....FAILED')
else:
    print(f'    ....SUCCESS')
print(f'    ....waiting {delay} seconds')
time.sleep(delay)

################
# Cleanup
################
print(f'    ..Cleaning up')

result = subprocess.run(cleanup_cmd,
                        shell=True,
                        stdout=stdout,
                        env=env)

if result.returncode != 0:
    error = True
    print(f'    ....FAILED')
```

```
        log_file.write(f'{attack};{start.isoformat()}Z;
        ↪ {stop.isoformat()}Z;{(stop - start).total_seconds()};ERROR\n')
    else:
        print(f'    ....SUCCESS')
        log_file.write(
            f'{attack};{start.isoformat()}Z;{stop.isoformat()}Z;{(stop -
            ↪ start).total_seconds()};{"SUCCESS" if error is False else
            ↪ "ERROR"}\n')

    print(f'')
    print(
        f'    Duration: {stop - start} - start: {start.isoformat()}Z - stop:
        ↪ {stop.isoformat()}Z - {"SUCCESS" if error is False else
        ↪ "FAILED"}')

log_file.close()
print('')
print(f'Log file created {target_file}')
```