A COGNITIVE PROCESS TRACING APPROACH TO
CYBERSECURITY DATA TRIAGE OPERATIONS AUTOMATION

A Dissertation in
Information Sciences and Technology
by
Chen Zhong

© 2016 Chen Zhong

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

August 2016
The dissertation of Chen Zhong was reviewed and approved* by the following:

Peng Liu  
Professor of Information Sciences and Technology  
Dissertation Co-Adviser, Chair of Committee

John Yen  
Professor of Information Sciences and Technology  
Dissertation Co-Adviser

John M. Carroll  
Distinguished Professor of Information Sciences and Technology

David Hunter  
Professor of Statistics

Robert F. Erbacher  
Research Scientist at Army Research Laboratory  
Special Member

Andrea Tapia  
Associate Professor of Information Sciences and Technology  
Director of Graduate Programs, College of Information Sciences and Technology

*Signatures are on file in the Graduate School.
Abstract

Security Operations Centers (SOCs) not only employ various cyber defense technologies to continually monitor and control network traffic, but also rely heavily on cybersecurity analysts to make sense of the resultant network monitoring data for attack detection and incident response. As the network monitoring data are usually generated at a rapid speed and contain a lot of noises, analysts are so far bounded by tedious and repetitive data triage tasks that they can hardly concentrate on in-depth analysis to generate timely and quality incident reports. These difficulties result in a great disparity in force between overwhelmed cybersecurity analysts and aggressive attackers. Therefore, there is an urgent need to liberate cybersecurity analysts from the tedious data analytics to focus on the higher-level cyber situational awareness.

Our work is aimed to reduce analysts’ workloads by leveraging analysts’ previous cognitive processes of data triage. With this goal in mind, a fundamental step is to trace analysts’ cognitive processes at the fine-grained level while they are performing data triage tasks. We defined data triage as a dynamic cyber-human system and proposed a trace representation of a fine-grained data triage cognitive process. Based on the trace representation, we developed a tracing method which integrates automated capture and self-reports to capture an analyst’s cognitive process with the minimum interference. An interactive toolkit, named ARSCA, was developed as a specific realization of the tracing method. In collaboration with Army Research Lab, an experiment on professional analysts was conducted in which participants’ cognitive processes were traced by ARSCA while they were performing a simulated cybersecurity analytics task. The results of the experiment show that the proposed tracing method is a feasible way to conduct on-the-job cognitive task analysis studies with less influence on the human performance. Besides, the preliminary trace analysis indicates that the collected traces are abundant in
information and shed some insights into the cognitive process of how the participants perform data triage task.

To the best utilization of the captured traces in the experiment, we proposed an automated trace analysis method for constructing data triage rules directly from the traces. The rules were further used to build finite state machines for automated data triage. The biggest challenge in the rule construction for the data triage was how to distinguish the effective data triage operations from the exploratory ones in traces. To solve it, a graph model was proposed to represent both the temporal and logical relationships among analysts’ data triage operations. We evaluated the automated data triage systems constructed from the traces by applying them to a large dataset and comparing the data triage results with the ground truth. The results of the study illustrate the rules mined from the traces are useful to conduct effective data triage, which further validates the practical value of the proposed tracing method. Besides, the results also show that the automated system built on the traces from the analysts with better task performance have a better data triage performance, which implies that a better data triage performance can be achieved by tracing and utilizing experts’ operations.

In conclusion, an initial step had been taken towards leveraging human analysts’ previous cognitive processes to facilitate data triage. Its contribution lies in three aspects. Firstly, the proposed tracing method realizes the possibility of tracing human analysts’ cognitive processes in a less intrusive manner while analysts are performing cybersecurity analytics tasks. Secondly, the proposed trace analysis method was shown to be effective and useful in constructing useful data triage rules from the analysts’ operation traces in a largely automated way. Thirdly, the constructed data triage rules can be used to construct data triage automation for reducing analysts’ workloads.
Table of Contents

List of Figures x
List of Tables xii
Acknowledgments xiii

Chapter 1
Introduction 1
1.1 Background and Motivation ........................... 1
  1.1.1 A Critical Role of Cybersecurity Analysts .......... 1
  1.1.2 Challenges Faced by Cybersecurity Analysts ........ 2
1.2 My Approach ........................................... 4
  1.2.1 Method for Tracing Analysts’ Cognitive Processes ... 4
  1.2.2 Empirical Study of Fine-Grained Analysts’ Cognitive Process of Data Triage ............................ 6
  1.2.3 Automated System for Generating Data Triage Rules from Cognitive Traces .......................... 8
1.3 Outline .................................................. 9

Chapter 2
Literature Review 11
2.1 Multi-Level Cybersecurity Analytics .................... 11
  2.1.1 Network Event Level ............................... 13
  2.1.2 Incident Level ..................................... 14
2.2 Related Theories of Cybersecurity Analytics ........... 15
  2.2.1 Situational Awareness ............................. 16
  2.2.2 Sensemaking ....................................... 17
Chapter 6
Automated Cyber SA Data Triage System by Leveraging Analysts’ Cognitive Traces

6.1 Introduction ......................................................... 92
6.2 The System Model .................................................. 96
   6.2.1 Characteristic Constraint of Data Triage Operation ........ 96
   6.2.2 Relationship between Characteristic Constraints ........... 98
      6.2.2.1 Characteristic Constraint Graph ....................... 99
6.3 The Automated Data Triage Approach ............................ 99
   6.3.1 Step 1: Identifying Data Triage Operations ................. 100
   6.3.2 Step 2: Constructing Characteristic Constraint Graph ...... 102
   6.3.3 Step 3: Mining the Characteristic Constraint Graphs ........ 105
      6.3.3.1 Extracting the Critical Endpoints from “isSub” Subgraphs 106
      6.3.3.2 Mutually Exclusive Characteristic Constraints ......... 107
      6.3.3.3 Grouping the “Candidates” based on Heuristics .......... 108
   6.3.4 Step 4: State Machine for Real-Time Pattern Matching ... 108
      6.3.4.1 Attack Path Pattern ................................ 108
      6.3.4.2 State Machine .................................. 109
6.4 Evaluation ......................................................... 110
   6.4.1 Experiment Dataset ........................................... 111
      6.4.1.1 ARSCA Traces Collected from an Experiment ........... 111
      6.4.1.2 Task Data Sources .................................. 112
   6.4.2 DT-SM Construction ......................................... 113
      6.4.2.1 Data Triage Operation Identification .................. 113
      6.4.2.2 Characteristic Constraint Graph Analysis .............. 113
      6.4.2.3 Attack Path Pattern Construction ..................... 116
   6.4.3 Performance of DT-SM ...................................... 118
      6.4.3.1 Data Source and Attack Scenario for Testing .......... 118
      6.4.3.2 Performance of DT-SM ................................ 118
      6.4.4 Effect of Analysts’ Task Performance on the DT-SM’s Performance 120
6.5 Related Work .................................................... 122
6.6 Limitation and Evasion .......................................... 123
6.7 Conclusion ....................................................... 124
## List of Figures

2.1 Levels of cyber situational awareness ............................................. 12

3.1 A series of data triage operations conducted to detect evidence of the malicious activities in attack chains. Each data triage operation filters or correlates network events based on a characteristics constraint defined by the analyst. .............................................. 26

3.2 The framework of the interaction between human analysts and the Cyber SA data. .......................................................... 29

3.3 An example of analytical reasoning process represented by the AOH Objects and their relationships. ................................................. 34

3.4 The H-Trees corresponds to the AOH-Trees in Figure 3.3. .................. 34

4.1 The architecture of the method for tracing an analyst’s cognitive process of data triage .......................................................... 47

4.2 Main components of the ARSCA user interface. It includes (a) a Data View, which presents the network monitoring data and supports data triage operations; and (b) a Analysis View, which displays the existing AOH-Trees and H-Trees and supports the operations related to H-Trees. ................................................. 49

4.3 Extracting themes from the text of an answer to EVTS ....................... 62

5.1 An empirical study of the analysts’ cognitive process of Cyber SA data triage. We collected traces of analysts’ cognitive processes in an experiment, using the computer-aided tracing method described in Chapter 4, and conducted case studies of how analysts’ perform data triage. .......................................................... 70

5.2 An analyst’s operation sequence is interpreted by the underlying AOH objects and their logic relationships ................................. 79

5.3 The case of identifying an observation .............................................. 81

5.4 The case of creating a hypothesis .................................................. 82

5.5 The case of hypothesis investigation .............................................. 84
5.6 A case of a participant detecting suspicious network events by “gradually narrowing down” .......................... 85
5.7 A case of a participant locating suspicious network events by “following a cue” ........................................ 86
5.8 A case of a participant “proceeding from one event to its related events” ...................................................... 87

6.1 The overview of the automated data triage approach. .... 100
6.2 ARSCA functions that support data triage operations .... 101
6.3 A partial characteristic constraint graph constructed from an analyst’s trace. The nodes are the data triage operations, with the index of their temporal order. A squared node refers to a data triage operation based on which the analyst generated a following hypothesis about the possible attack path. However, circle nodes are the data triage operations that did not result in any hypothesis. 106
6.4 The characteristic constraint graphs of the professional analysts’ traces. ......................................................... 114
6.5 Motif profiles of “isSub” and “isCom” subgraph .......... 116
6.6 Identifying suspicious network event sequences based on multiple DT-SMs .................................................. 118
6.7 The performance of DT-SMs with different thresholds of least support. 119
## List of Tables

2.1 Discriminability and response bias are two factors in decision making. 19

4.1 An example of a sequence data triage operations recorded by ARSCA 54
4.2 RUI log of a same set of operations recorded in Table 4.1. 55
4.3 A partial sequence of data triage operations from a professional analyst’s trace 58
4.4 The four open-ended questions in the post-task questionnaire. 61
4.5 The procedure of trace-stimulated recall 63
4.6 The results of the trace-stimulated recall of four participants 64

5.1 Task data sources tailored from VAST Challenge 2012 data 76
5.2 The four rating questions about experiment setting and task performance. 77
5.3 Participants’ responses to the questions in Table 5.2 78

6.1 The data triage operations recorded by ARSCA 103
6.2 Identifying the data triage operations from ARSCA trace 104
6.3 Attack ground truth of 3 selected time windows 111
6.4 Summary statistics for the Characteristic Constraint Graphs constructed from the professional analysts’ traces 113
6.5 Top 10 characteristic constraints in the attack path pattern by analyzing the 29 analysts’ data triage operations 117
6.6 Performance of the DT-SMs built on the traces from analysts with different task performance. The threshold of least support is 2 120
Acknowledgments

I would like to express my special appreciation and thanks to my advisers Dr. Peng Liu and Dr. John Yen. You have been tremendous mentors for me. I would like to thank you for your advice on both research as well as on my career. I can still clearly remember my first conversation with Dr. Liu over the phone five years ago, from which my journey of PhD started. During the past five years, I have been continuously encouraged and inspired by your research enthusiasm. I feel so lucky to have you as a role model as I’m growing as a researcher.

I would also like to thank my committee members, Dr. John M. Carroll and Dr. David Hunter and Dr. Robert F. Erbacher for serving as my committee members even at hardship. I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

I would like to give my special thanks to Dr. Robert F. Erbacher, who is my most important collaborator. His mentorship deepened my understanding in cognitive task analysis and cyber operations. The inspiring discussions with him have improved my research a lot.

I would like to thank every one of my collaborators: Deepak S. Kirubakaran, Renee Etoty, Christopher Garneau, Gaoyao Xiao, Steve Hutchinson, Hasan Cam, Sam Gur, Xiaoxiao Bai, William Glodek and Mingyi Zhao. Without any of them, I cannot make my way through this dissertation.

I would like to thank Dr. Nicklaus Giacobe for his endless advice on teaching when I was his teaching assistant.

I would like to thank all my labmates, colleagues and friends at Penn State. Jun Wang, Fang Dong, Bin Zhao, Tao Zhang, Jun Dai, Xiaoyan Sun, Mingyi Zhao and Gaoyao Xiao all offered me a great help, especially in my first year at Happy Valley. Thanks Kai Chen, Jun Dai for their advise on both life and career. Xin Peng has celebrated five new years together with me when we were away from home. Best wishes to Chi Ping for her new life and career.

Finally, I give my deepest thanks to my mother Dr. Chunxia Xu and my father
Dr. Ming Zhong for their love, support and encouragement. Words cannot express how grateful I am. Thank you for everything and wish you the best.
This dissertation is dedicated to my mother Dr. Chunxia Xu and my father Dr. Ming Zhong for their endless love, which gives me the ability to love others.
Chapter 1

Introduction

1.1 Background and Motivation

1.1.1 A Critical Role of Cybersecurity Analysts

The big breaches of Target, JPMorgan Chase, Sony Pictures, and Primera Blue Cross constantly remind us that the risk of cyber attacks can hardly be underestimated. Many prominent companies, government organizations and military departments have invested a lot of money to construct Security Operations Centers (SOCs) against the increasingly sophisticated cyber attacks. Typically, SOCs are cyber defense systems for 24*7 monitoring, intrusion detection, and diagnosis (on what is actually happening) [78]. In a military setting, CNDSP (Computer Network Defense Service Provider) centers have already been established and operating for quite a few years. SOCs usually employ multiple automated security measures, such as traffic monitors, firewalls, vulnerability scanners, and Intrusion Detection/Prevention System (IDS/IPS), all of which continually generate network monitoring data.

Nowadays SOCs rely heavily on human analysts (i.e., cybersecurity analysts) to make sense of these data to achieve cyber situational awareness (Cyber SA).
More specifically, the following questions need to be answered through analysts’
cybersecurity analytics: Whether a network is under attack? How does attacks
happen? What will attackers do next?
Analysts are playing a critical and indispensable role because the automated
measures are in many cases unable to “comprehend” sophisticated cyber attack
strategies even with advanced correlated diagnosis. Specifically, analysts need to
conduct a series of analyses, including data triage, escalation analysis, correlation
analysis, threat analysis, incident response and forensic analysis [17]. Each stage
of analysis involves very complicated analytical reasoning performed by analysts
with their knowledge and experience gained through years of on-the-job training.
For example, data triage, as the fundamental step, encompasses examining the
details of a variety of data sources (e.g., IDS alerts, firewall logs, OS audit trails,
vulnerability reports, and packet dumps), weeding out the false positives, and
grouping the related indicators so that different attack campaigns (i.e., attack
plots) can be separated from each other. Data triage provides a basis for closer
inspection in the subsequent analyses to finally generate confidence-bounded attack
incident reports. These incident reports will serve as the primary basis for further
decision-making regarding how to change current security configuration and how
to act against the attacks.

1.1.2 Challenges Faced by Cybersecurity Analysts
Cybersecurity analytics presents the analysts with multiple challenges in today’s
SOCs. First of all, the network monitoring data, which are being continuously gen-
erated, are too overwhelming for analysts to process. Compared with a computer,
human brains have smaller orders of magnitudes of data processing throughput,
and human beings have such disadvantaged weaknesses as fatigue, anxiety and de-
pression. However, neither the network nor the attack campaign is waiting for the
human brains. In addition, the data coming from various sources contain many
false alerts so that analysts need to apply their domain knowledge and experience to make high quality decisions regarding which parts of the data are worth further investigation and what are suspected malicious events to report as an incident. According to a research by the Ponemon Institute in January 2015 [71], the average number of malware alerts an organization receives in a week is 17,000, and only 19% or less of them are reliable. About two-thirds of the time of cybersecurity analysts are wasted for investigating the false alerts. FireEye reported that the average annual operational spending of an organization due to false positives is $17.816 million given the default resource capacity informed through common deployments [28].

Secondly, analysts have to hasten their data triage under the pressure of limited time. Given the rapid influx of network monitoring data, analysts usually have to make quick decisions because earlier detection of cyber attacks is the premise of timely incident response. Besides, analysts have to rotate on shifts on a 24/7 schedule (24 hours a day, 7 days a week) in order to ensure uninterrupted monitoring. Especially for data triage, analysts need to make decisions within a very short period for filtering the incoming data to identify indicators for suspicious events, weeding out false alerts, generating hypotheses regarding malicious events, and investigating data from different sensors concerning the suspicious events.

Considering those challenges, several major companies and government organizations have adopted advanced cybersecurity analytics systems, such as Security Information and Event Management (SIEM) products, to support integrated cross-source data analysis and incident management. Although the SIEM systems take a big leap forward in cybersecurity analytics, these SIEM systems are extremely expensive not only for the high cost of its license and deployment but also for the large amount of time and expertise required in regular system management and customization. Due to SIEM’s high cost, most organizations can’t afford good protection of their networks, despite the potential perils and loss of the cyber attacks
to their networks.

1.2 My Approach

There is a big gap between the demands for enhancing the cybersecurity analytics capability of analysts and the limited resources of expert analysts and the extremely high cost of advanced cyber analytics systems. My research is motivated to bridge that gap. My dissertation work takes the first step to facilitate cybersecurity analytics by understanding and leveraging analysts’ analytical reasoning processes. To hone in on a reasonable research scope, I mainly focus on data triage which is a fundamental but the most tedious and time-consuming stage in cybersecurity analytics.

My approach is built on three important insights. Firstly, it is possible to trace human analysts’ cognitive processes in a less intrusive manner while they are performing analytics tasks. Secondly, useful information can be mined from the captured cognitive processes in a largely automated way. Thirdly, data triage automatons can be constructed to reduce analysts’ workloads by utilizing the mining results.

1.2.1 Method for Tracing Analysts’ Cognitive Processes

The first objective of my research is to gain understanding of human analysts’ cognitive processes of data triage at a fine-grained level. A critical factor influencing the success of a SOC in tackling the cybersecurity analytics challenges is the effectiveness of the cybersecurity analysts’ cognitive processes in data analysis tasks. However, the detailed cognitive process of data triage analysts is rather complicated and far from well-understood.

I choose to focus on capturing the fine-grained cognitive processes because it can better explain how the analysts deal with the challenges in data triage. One
main challenge for analysts exists in detecting the attack evidence among a large volume of massive data sources. It is likely to deduce the way how an analyst manages to make it on the ground of examining the data triage operations and the related hypotheses that triggered these operations.

Cognitive Task Analysis (CTA) is a traditional method for studying human working processes. Some researchers with the access to the cybersecurity analysts have conducted several CTA studies using various techniques, such as observation and interviews. Most of these studies focused on macro-level descriptions of cybersecurity analytics (e.g., the stages of data filtering and the roles of analysts), had gained some valuable insights into the analysts’ cognitive processes. However, few have attended on the fine-grained cognitive activities due to several real-world difficulties in conducting CTA studies in cybersecurity. For instance, CTA studies can be too time-consuming as analysts, who have to rotate through day shift and night shift on a 24/7 schedule (24 hours a day, 7 days a week) may have little time to participate in interviews. Besides, data triage tasks are memory-intensive and require vigorous concentration so that it is hard for analysts to give complete and accurate reports on their cognitive processes with a commonly-used think-aloud protocol.

To overcome the above disadvantages, I propose an integrated process tracing method for capturing analysts’ fine-grained cognitive processes. First of all, a trace representation is proposed to specify a cognitive process at the fine-grained level. Drawn on the sensemaking theory in cognitive science, the trace representation identifies the key elements in an analyst’s cognitive process of data triage, including actions of data triage, observations of suspicious network data, and hypotheses about the possible cyber attack activities. To capture the analysts’ cognitive processes specified in the trace representation, this method integrates automatic capture with situated self-reports. On one hand, it automatically capture an analyst’s data filtering and correlating actions and the resultant observations of suspicious
network events. On the other hand, the analyst spontaneously reports his/her hypotheses about attacks based on the current observations. Each hypothesis is linked automatically to the corresponding observation once generated. The automatically captured and self-reported information is mutually complementary.

As a specific measure of the tracing method, an interactive toolkit has been designed and developed to support the proposed tracing method. The biggest problem in design lies in that the toolkit should not detract the analyst from performing his/her task as is usual in think-aloud protocols. Therefore, I adopted a user-centered approach (i.e., scenario-based design) at the design stage and then to test it in the following empirical studies.

### 1.2.2 Empirical Study of Fine-Grained Analysts’ Cognitive Process of Data Triage

A sophisticated understanding of the fine-grained cognitive process of an analyst can provide several critical benefits for enhancing the effectiveness of SOC. The fine-grained cognitive process of an expert also offers the opportunity to allow other junior analysts to leverage it to improve their own analysis efficiency. Moreover, such understanding can enhance the accountability of decision making, improving the effectiveness in analysts training, developing better cognitive aids and collaboration supports to address the three challenges described above. Furthermore, understanding the fine-grained cognitive process of an analyst is the basis for developing automation toolkits to facilitate data triage.

Several CTA studies have been conducted which provide valuable insights into the high-level processes of analysts such as their roles and the work-flows [18, 17], their cognitive demands [24], and their performance in Cyber SA data analysis [34]. However, it still remains unclear as for analysts’ fine-grained cognitive activities in data triage [96]. With the proposed tracing method, it is possible to trace analysts’ fine-grained cognitive processes of data triage. Therefore, an empirical
study was conducted to capture analysts’ traces to gain deeper understanding of their cognitive processes. An experiment had been carefully designed to collect the traces of analysts’ cognitive processes while they performed a data triage task. The duration of each experiment lasts about 100 minutes including four sessions: a pre-task questionnaire, a tutorial session (with a quiz), a data triage task and a post-task questionnaire. During the task session, each participant was asked to work with the toolkit to accomplish the assigned task. With the close collaboration with Army Research Lab (ARL), we were able to recruit thirty professional cybersecurity analysts and to collect the traces of the analysts’ cognitive processes in the experiment within about seven months. The experiment results verify the feasibility of the proposed method for capturing analysts’ cognitive process. Moreover, several important assumptions have emerged as a result of this work:

- The captured traces contain the details of the critical data triage operations and hypotheses of analysts while they were performing the data triage task [100].
- Several common behavior patterns emerge in these traces [101] and various analytics strategies (e.g., data triage and correlation strategies) are implied by the traces [100].
- Traces help analysts with self-reflection so that they can learn from their previous experience [101], which is quite promising in training by utilizing traces.

The results of the experiment in return shows that the proposed method can be an effective way to conduct on-the-job CTA studies with less intrusion on the human performance being studied. Besides, the captured traces are a good source for further in-depth analysis to gain better understanding of how the subjects perform the assigned task.
1.2.3 Automated System for Generating Data Triage Rules from Cognitive Traces

Given the traces collected from the experiment, I proposed a trace analysis method for constructing useful data triage rules from traces which can help analysts identify and correlate the key evidence of malicious events from raw network monitoring data. Extensive research has been conducted on alert correlation, which is a milestone in automated data triage. Motivated by the benefits of cross-data-source analysis, SIEM systems have been focusing on security event management and correlation across multiple data sources. The complicated SIEM rules can be decomposed into a set of basic rules (for data filtering or correlation) and logic connectors (“AND” or “OR”). Rule generation requires large amount of time and expertise, which makes SIEM systems extremely expensive. Therefore, it is desirable to automate the rule generation process.

One important research question is how to distinguish the critical data triage operations recorded in the traces from the exploratory ones. We tackled the problem with a graph-based analysis method. A graph model was developed which represents the temporal and logical relationships among the data triage operations. The proposed approach has been evaluated through a human-in-the-loop study. The result of the study illustrates the rules mined from the traces are helpful for analysts to conduct data triage. The result validates the practical value of the captured traces. Although this method still can’t generate the rules as complicated as those generated by experts in SIEM systems, the analysts can embark on with the from the automatically generated rules to construct more complicated rules instead of from scratch.

To the best of our knowledge, this is the first method that directly leverages traces of human analysts’ cognitive processes to develop data triage automation. Relating this method to the rule-based SIEM systems, it can greatly reduce analysts’ workloads in generating SIEM rules and thus render the cost of data triage
automaton generation orders of magnitudes smaller.

1.3 Outline

The remaining of this dissertation is organized as follows. Chapter 2 is a literature review of the studies of multilevel Cyber SA and related theories of cybersecurity analytics. Chapter 3 provides a definition of Cyber SA data triage and presents a trace representation of the fine-grained cognitive process of data triage. In Chapter 4, we propose a computer-aided method for tracing fine-grained analysts’ cognitive process of data triage. Chapter 5 presents an empirical study in which we capture analysts’ cognitive processes in an experiment and explore the captured traces using a qualitative analysis method. In Chapter 6, we present an automated trace analysis method for constructing automated data triage system. Finally, Chapter 6 concludes this dissertation.

Chapter 2 is organized as follows. Section 2.1 reviews the existing studies of cybersecurity analytics at network event level and incident level. Section 2.2 investigates the related theories of cybersecurity analytics, including situation awareness, sense making, information foraging, data fusion and decision under uncertainty.

Chapter 3 is organized as follows. Section 3.1 presents the main characteristics of Cyber SA data triage. Section 3.2 defines data triage as a dynamic cyber-human system and describes the human-data interaction within the system. Section 3.3 presents the trace representation of the human analysts’ fine-grained cognitive processes of data triage.

Chapter 4 is organized as follows. Section 4.1 proposes the problem of tracing analysts’ cognitive process. Section 4.2 provides a review of the related work. Section 4.3 describes the main components of the tracing method and the computer toolkit design. Section 4.4 presents the implementation and testing result of the toolkit. In Section 4.5, we evaluate the tracing method based on analysts’ feedback.
and a preliminary trace analysis. Section 4.6 discusses the pros and cons of the tracing method and compared it with reactive and non-reactive tracing methods.

Chapter 5 is organized as follows. Section 5.1 gives an introduction of the empirical study. Section 5.2 describes the experiment designed for capturing analysts’ cognitive processes using the tracing method proposed in Chapter 4. Several critical design issues are discussed. In Section 5.3, we conduct a case study of the traces collected in the experiment and gain a deep understanding of the participants’ operation patterns and data triage strategies. In Section 5.4, we discuss about what in the participants’ cognitive processes can or cannot be recovered from the captured traces.

Chapter 6 is organized as follows. Section 6.1 motives the work of building automated data triage system directly form the capture traces. Section 6.2 describes our automated trace analysis method for constructing data triage “rules” from the traces. We also present the state machine built on these “rules” for automated data triage. We evaluate the performance of the state machines developed from analysts’ traces. Section 6.4 reviews the related work. In Section 6.5, we discuss about the limitation and evasion of the automated data triage system. Section 6.6 concludes this work.
Chapter 2

Literature Review

2.1 Multi-Level Cybersecurity Analytics

The jobs of cybersecurity analysts are conceptualized differently across organizations, which leads to diversity in expectations, tasks and responsibilities assigned to the analysts at different levels [17, 59]. Nevertheless, they are all driven by the same goal of gaining Cyber Situational Awareness (Cyber SA) [17, 4, 21].

However, cybersecurity analytics is a complicated process involving multiple levels of Cyber SA. Previous Cognitive Task Analysis (CTA) studies on cyber defense have provided valuable insights into the high-level workflow and data reduction process of cybersecurity analytics.

D’Amico et al. identified six types of cybersecurity analytics within the U.S. Department of Defense: triage analysis, escalation analysis, correlation analysis, threat analysis, incident response, and forensic analysis [18]. The process of transforming raw data into Cyber SA is further investigated and a generalized workflow of analysis process is developed which contains both tactical and strategic goals[17]. A cyber command and control task flow diagram is developed to highlight the primary tasks based on interviews of individuals with different levels of decision-making responsibility within cyber operations at Pacific Northwest Na-
Figure 2.1. Levels of cyber situational awareness

tional Laboratory, which includes organization/mission impact assessment, data cleanup and detailed analysis, external and internal response, audit, and big picture of cyber attacks [25]. Based on the results of these CTA studies, Figure 2.1 synthesizes and summarizes the high-level analysis process at different level of Cyber SA. It shows that a cybersecurity analytics process transforms a huge amount of network monitoring data (e.g., IDS alerts, firewall logs, and packet dumps) into understandings of incidents at the network event level, which further lead to in-
incident correlation and response and predictions of further attacks at the incident level.

2.1.1 Network Event Level

Existing automated cybersecurity measures (e.g., Intrusion Detection/Prevention Systems (IDS/IPS), firewall, traffic monitors, and vulnerability scanners) collect network traffics, host-based or network-based log data. The raw data can be represented by network events. (Please refer to Chapter 3 for details). As the first step, suspicious event detection identifies suspicious events by weeding out false alerts or irrelevant logs. Most automated suspicious event detection systems match network traffic to signatures of known malicious activity patterns [51, 89], or use statistical methods for anomaly detection [43, 63, 73]. Despite the help of all kinds of automated system, human analysts are still playing a significant role in inspecting the network events and making fast decisions on which events are worth of further investigation. D’Amico and Whitley categorized this role of analysis as triage analysis [17].

The identified suspicious events feed to event escalation to further investigate, interpret, and assemble data from multiple sources. Event correlation discovers connections in current or history network events and searches for patterns in the current or history data. Many automated methods have also been developed to facilitate event escalation and correlation by conducting real-time data aggregation and correlation. Alert correlation is an important milestone in automated event correlation, which aggregate alerts by their common characteristics (e.g., source address, destination address, server name or port number)[15, 20] or use heuristics rules to correlate the alerts that indicate a same malicious activity in a multi-step attack [46, 83]. However, alert correlation has its limitations in that it can only analyze one data source (i.e., IDS alerts) while cross-data-source analysis is necessary in most cases considering that nowadays more attackers conduct coordinated
activities at different time points or different segments of a network to achieve an ultimate malicious goal. Logical attack graphs are proposed to help analysts understand the logical dependencies among attack steps, as well as the relationships between network configuration and a possible attack chain [66]. Sometimes the Security Information and Event Management (SIEM) products (e.g., ArcSight) may be used to conduct real-time initial data aggregation and correlation. Even with the aid of SIEM, analysts still need to conduct data filtering and triage to correlate the network events in depth to gain insights into incidents. Escalation and correlation rely heavily on the prior knowledge of analysts to make sense of the detected events.

These three types of analyses at the level of network event are described as an interactive process by D’Amico and Whiteley in a workflow of computer network defense analysts[17]. The goal of the analysis at the network event level is to describe incidents based on the collected network monitoring data. Incidents refer to the network-based attack that need to be responded, which are implied by a sequence of correlated and malicious network events.

2.1.2 Incident Level

Compared with the analysis at the network event level, the goal of the analysis at the incident level is to achieve a higher-level perception of cyber attacks and a broader understanding of the implications of related incidents by assessing the incidents in a community and adding intelligence data sources [17]. Provided with the confirmed incidents, analysts perform incident correlation and threat analysis by correlating the incidents (which may be detected in different regions or far in time), which are parts of a larger attack scheme, with additional intelligence information to understand attackers’ true identify and intent. With the improved comprehension, analysts take responsive actions to the incidents and gather evidence of the confirmed attackers [47, 17]. Based on the confirmed incidents, proactive
threat analysis predicts the potential attacks in the future. The analysis result at the incident level is the intrusion set which represents the comprehension of the related incidents, the potential reaction to the incidents, and the prediction of future attacks [17].

2.2 Related Theories of Cybersecurity Analytics

Although the previous studies have given detailed account of different functions of analyses, the details of the cognitive processes of analysts are still not well understood. This problem is mentioned by D’Amico and Whitely regarding how analysts discover patterns during event correlation analysis:

“An analyst might not know what patterns they are looking for in advance; instead, the analyst might “know it when they see it.” When they encounter a pattern that they cannot explain, they form hypotheses about potential malicious intent, which they try to confirm or contradict via additional investigation.”[17]

The above example indicates the complexity of analysts’ cognitive processes in different types of analyses, which involve the important knowledge and experience required by these analysis roles. More specifically, an analyst’s cognitive process may reveal (1) how the analyst detected the “true signal” from the massive network monitoring data and “connected the dots” to correlate the network events, (2) what hypotheses were generated to interpret the current observation, and (3) what further actions the analyst took to verify his/her hypotheses. Therefore, it is necessary to gain a deep understanding of these cognitive processes of analysts. There are several theories relevant to the cognitive processes of cybersecurity analytics, including situational awareness, sensemaking, information foraging, data fusion, and decision making under uncertainty.
2.2.1 Situational Awareness

According to the discussion in Section 2.1, cybersecurity analytics involves Cyber SA at both the network event level and the incident level. The notion of Cyber SA is rooted in situational awareness (SA). SA is referred by the Recognition Primed Decision (RPD) model as “the expert decision makers evaluate a situation and match that situation to a situation previously encountered” [48]. Boyd’s OODA loop is a theory used in military operations that describes SA as a process of Observation, Orientation, Decision, and Action [7]. According to OODA loop model, an observation refers to the raw information which is presented before analysts’ involvement; an orientation is the process of “fusing information to build SA” [93] in which hypotheses are generated based on existing observations to interpret the current situation; a decision is the result of analyzing all the hypotheses; and action refers to the actions conducted to carry out what has been decided. Endsley proposed a well-known definition of SA as a dynamic process involving three main phases: (1) “the perception of the elements in the environment with a volume of time and space”, (2) “the comprehension of their meaning”, and (3) “the projection of their status in the near future” [22, 23]. The Cyber SA includes the awareness of the attack impact, the awareness of how an attack evolves and behaves, the awareness of why and how the attack happened, the awareness of how trustworthy of the collected information items are, and whether a decision made based on these information items is good or not, and how effective is the potential of assessment for future threats [4]. Drawing on Endsley’s theory, Cyber SA is an iterative process involving the perception of current network situation (e.g., suspicious network event detection), the comprehension of why and how an attack happened, and the projection of how the attack will evolve in the future.
2.2.2 Sensemaking

Sensemaking is a proactive aspect of Cyber SA which enables transforming network monitoring data to analysts’ awareness of cyber situation. Pirolli and Card [68] proposed a well-known sensemaking model containing a foraging loop and a sensemaking loop based on the CTA results of intelligence analysis. Both loops are iterative: the foraging loop focuses on how an analyst performs information seeking, and the sensemaking loop focuses on how the analyst’s mental model is developed in this process. This nested loop structure is “an integration of bottom-up processes and top-down processes” [68]. The bottom-up processes are processes from theory to data, such as searching and filtering, reading and extracting, schematizing, building cases and telling story. The top-down processes are processes from data to theory, in which analysts re-evaluate, search for support, search for evidence, search for relations and search for information [68].

2.2.3 Information Foraging

According to Pirolli and Card’s sensemaking model, there are three main aspects in the foraging loop: exploration (“increasing the span of new information items into the analysis process”), enrichment (“narrowing the set of items to produce higher-precision sets of data”), and exploitation (“thorough reading of documents, extraction of information, generation of inferences, noticing of patterns, etc”) [68]. Information foraging is relevant to most of the analysis at both the network event level and the incident level. Pirolli and Card point out that there exists a trade-off among exploration, enrichment and exploitation [68]. Taking the analysis at the network level as an example. On one hand, the size of network monitoring data set is continuously increasing because it will be desirable to have cyber defense technologies capture as much network events as possible (i.e., exploration). On the other hand, analysts need to narrow down their search space to locate the real evidence of incidents by weeding the false positives out of the increasing set of raw
data (i.e., enrichment). These suspicious network events further limit the set of incidents identified by the analyst because incidents consist of these suspicious network events upon which the analyst conduct event correlation (i.e., exploitation).

### 2.2.4 Data Fusion

Data fusion is also a critical aspect of Cyber SA considering that the network monitoring data are collected from multiple sensors. Joint Directors Laboratories (JDL) Data Fusion Working Group developed a general model for data fusion process across multiple fields [38]. The process model takes the sources of “information at a variety of levels, ranging from sensor data to a prior information from databases to human input”. This model breaks the data fusion process into five components: source pre-processing, object refinement, situation refinement, threat refinement, and process refinement. Most data fusion algorithms and techniques can be categorized into these JDL components. Aimed to proposed an integrated analysis framework of Cyber SA, Lan et al. proposed a data fusion method based on the Dempster-Shafer (D-S) evidence theory to gain cyber situational awareness [52]. The D-S theory is a computational model for reasoning with uncertainty which structures the uncertainties into independent items of evidences and combines beliefs of evidences [79]. Bass described data fusion by mapping the OODA decision-support processes into different level of abstractions to gain Cyber SA [5]. Data fusion is an iterative process in which the analysts perform filtering and correlating actions on the large volume of data, select data of interests as observations, describe their hypotheses about possible attacks which may further lead to additional action and observations. These activities correspond to the “observation”, “orientation”, “decision” constructs in the OODA model. Researchers have recognized the needs for a model that specifies how data fusion can be successfully applied to enhance Cyber SA [32]. The application of data fusion can contribute to intrusion detection [5], situational awareness [32], higher-level multi-step cybre
attack tracking and projection [95].

2.2.5 Decision Making under Uncertainty

The cybersecurity analytics, especially the analysis at the level of network event, requires analysts to perform effective and timely decision making in a dynamic network environment under uncertainty. Signal Detection Theory (SDT) developed in the 1950s is a widely accepted computational model in psychology applied in studies where subjects discriminate between signals and noise [36]. SDT provides a way to measure decision making under uncertainty based on statistics. Considering the task of suspicious event detection, analysts make responses (i.e., “suspicious” or “not suspicious”), given the unavoidable internal noise in their decision making process (e.g., the uncertainty of network activities and the unstable mental model of analysts). SDT assumes that the distributions of both the internal noise of response (N) and the signal plus the internal noise (SN) are normal and the two distributions generally overlap. An analysts’ responses to a set of network events (where “suspicious” present or not) are categorized into four types, as shown in Table 2.1. We not only measure the probability that the analyst responds “suspicious” when an event is suspicious (i.e., signal is present), but also care about the probability that the analyst responds “normal” when an event is not suspicious.

<table>
<thead>
<tr>
<th>Signal Present</th>
<th>Respond “Suspicious”</th>
<th>Respond “Normal”</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signal Present</td>
<td>Hit</td>
<td>Miss</td>
</tr>
<tr>
<td>Signal Absent</td>
<td>False Alarm</td>
<td>Correct Rejection</td>
</tr>
</tbody>
</table>

According to SDT, there are two factors in decision making: discriminability and response bias. Discriminability refers to the strength of the signal. D-prime


(d’) is a widely used measure of discriminability which is the distance between the
means of SN and N in standard deviation units [36]. **Response bias**, however,
has nothing to do with the measurement of the signal but with the tendency for
an analyst to response “suspicious”. Cybersecurity analysts (especially for those
taking the analysis role at the network event level) usually tend to be liberal
(i.e., respond “suspicious” more frequently) considering the severe destroy that
may caused by cyber attacks. It usually results in a high rate of false alarms.
Response bias varies from analyst to analyst according to their analysis roles.
Therefore, discriminability and response bias are independent factors of decision
making. Besides, both false alarm rate (false positive) and miss rate (false negative)
should be taken into consideration when evaluating a decision making process
under uncertainty.

### 2.3 Conclusion

Cybersecurity analytics cover different types of analyses. My dissertation mainly
focuses on the analytics at the network event level (i.e., suspicious event detec-
tion, event escalation and correlation) with the aim to gain awareness of incidents
from the network monitoring data correlated from multiple security measures. The
analysis process is a complicated sensemaking process where analysts detect and
correlate suspicious events, generate hypotheses about possible incidents, and de-
cide on further actions to investigate their hypotheses. Existing theories provide
valuable insights into the framework of the sensemaking process and the key com-
ponents of analysts’ cognitive activities, which lay the foundation for my study of
fine-grained cognitive process of analysts in Cyber SA data triage.
Chapter 3

Data Triage in Cyber Situational Awareness

3.1 Characteristics of Cyber SA Data Triage

3.1.1 Massive and Rapidly Changing Data

Aimed at gaining Cyber SA, multiple sensors are deployed in a network to monitor various network activities. Bass first pointed out that the data collected from multiple sensors, which are used as input into intrusion detection systems, are heterogeneous. The data may consist of “numerous distributed packet sniffers, system log-files, SNMP traps and queries, signature-based ID systems, user profile databases, system messages, threat databases and operator commands” [5]. Apart from the data collected by computer/network sensors, there are other important sources of data generated by human intelligence, including the data of SIEM systems (e.g., threat databases), data from external sources (e.g., external attack or threat reports) and data collected from social media (e.g., Facebook and Twitter) [54]. The heterogeneous data vary significantly in types and formats, including quantitative and qualitative data (types), structured, semi-structured, and non-
structured data (formats). In addition, the Cyber SA data change constantly over time together with the attack threats [102]. This is due to the highly dynamic nature of cybersecurity environment with the uncertain network activities and the changing of attackers’ behaviors and exploitation techniques [4]. As a result, the cyber security analysts are faced with massive and rapidly changing data.

D’Amico et al. described the workflow of how computer network defense (CND) analysts transform raw data into Cyber SA, in which raw data are gradually transformed into suspicious network events, incidents and intrusion sets [17]. D’ Amico pointed out that the raw data are filtered and gradually transformed into valuable information through different stages of analysis, including triage analysis, escalation analysis, correlation analysis, threat analysis, and incident response analysis [68]. Triage analysis is the initial stage of analysis, in which the raw data is filtered by weeding out false alerts or reports of normal network events, and provides the analysis basis of escalation and correlation. The related data are then grouped and transformed into sets of incidents. Beyond the network event level, threat and incident response analysis are targeted at a higher-level Cyber SA, in which prediction and attack forecasting are mainly relied on various types of intelligence [17].

3.1.2 Human-in-the-Loop Data Triage

It is challenging for security analysts to transform the raw data into attack incidents as the raw data are massive and rapidly changing over time with the cognitive constraints of human beings. With the need of cybersecurity analysts identified, many studies have been conducted to investigate how analysts perform cybersecurity analytics and how to improve their analysis processes. A consensus has been reached that cybersecurity analytics expert plays an important role in network intrusion detection [6]. To aid cybersecurity analysts in making sense of large amount of data, various visualization tools have been developed to visualize network moni-
toring data from different sources to facilitate analysts in undertaking the tasks of monitoring, analysis, response, pre-development and future development [75].

Human analysts play an irreplaceable role in cybersecurity analytics, despite various automated data analytics technologies. Although human analysts have much less powerful working memory and computational capability than computers, human brains are far more capable of data interpretation, context comprehension, hypothesis generation, and decision making in a more flexible manner. With the rich experience in the “on-the-job” training, senior analysts are usually capable of carrying out data triage more efficiently than novice analysts [26, 8]. Therefore, it is always beneficial to learn how analysts perform the data triage to elicit the experts’ expertise. Cognitive task analysis (CTA) is a common method for studying the cognitive processes in task performance. Related CTA studies on cybersecurity analytics provides us valuable insights about different types of analyses as mentioned in Chapter 2. The CTA study of intrusion detection experts conducted by the Air Force enabled the researchers to identify the cognitive requirements for network intrusion detection, which are essential to successful intrusion detection [6]. D’Amico et al. explored the roles of analysts and the workflow of data analysis, and identified the cognitive requirements for improving CND visualization techniques [17]. A further investigation of network analysts’ workflow was conducted through a multi-phase CTA study with a focus on the analysts’ needs for visualization, which provides detailed understanding of analysts’ tasks, concerns and goals [24].

Based on the above literature review on CTA studies, there are three types of analysis at the network event level (i.e., suspicious event detection, event escalation, and event correlation) and they sometimes overlap and interfere with each other. The suspicious event detection is the basis for event escalation and correlation, and meanwhile event escalation and correlation invoke analysts’ hypotheses on possible incidents, which in return guide the following suspicious event detection.
Therefore, the whole cycle of analysis can be regarded as an interactive process of data triage. **Data triage process** is a fundamental step of cybersecuriry analytics, and most analysts in SOCs are playing relevant roles.

### 3.1.3 Reporting Incidents for Incident Response

An important form of output of data triage is incident reports, which feeds the following analysis at the incident level (e.g., threat analysis, forensic analysis, etc.) [17]. An incident is defined as “a violation or imminent threat of violation of computer security policies, acceptable use policies, or standard security practices” [35]. An incident report usually contains the following information [42, 30, 72]:

- Status (whether the report is complete or incomplete)
- Reporter
- Incident type (e.g., Account compromise, Denial-of-Service, Malicious code, Misuse of systems, Reconnaissance, spam, phishing, scams, 0-day attack, unauthorized access, etc.)
- Source IP (where the attack packets came from)
- Incident scope (e.g., which machines have been affected)
- Incident time line
- Incident description (e.g., explain how and why this attack incident happened)
- Evidence data (i.e., which data sources, alert, flow, connection, or payload provide evidence for the incident)
- Remediation actions (e.g., recommendations)
- Correlated incidents
In the process of data analysis, analysts need to quickly assess the network events reported in the raw data. Incident reports are generated to report the possible incidents, which consist of related suspicious network events. These incident reports serve as a data source for further analysis at the incident level to detect intrusion sets.

3.2 Definition of Data Triage in Cyber SA

3.2.1 Data Triage: A Dynamic Cyber-Human System

In order to investigate the data triage process, a definition of data triage in Cyber SA is proposed to present the unique characteristics of this process. A typical data triage process of an analyst is illustrated in Figure 3.1. The data sources collected over time are presented to an analyst in sequence. Each item in the data sources reports either a malicious or a normal network event. The malicious events may belong to different attack chains. On the basis of the data sources, an analyst performs a sequence of data triage operations (which will be described in details in this section) and narrow down the searching scope to a smaller subset of interest. These data triage operations are carried out based on their existing observation of the network events and their domain knowledge and experience. As a result, the analyst may report his/her hypotheses about the possible attack chains in incident reports or revises the existing incident reports accordingly.

Given a network, an analyst’s data triage process is a dynamic Cyber-Human System (CHS) evolving over time. This dynamic CHS composes of (1) the attack activities happening in the network, and (2) the massive and rapidly changing monitoring data collected from multiple sources, and (3) a set of reported incidents and the inferred temporal and casual relationships into attack kill chains, and (4) a set of “world knowledge” (e.g., the intelligence about the attacks and the mission to defend the network), and (5) the mental model of the analyst through
Figure 3.1. A series of data triage operations conducted to detect evidence of the malicious activities in attack chains. Each data triage operation filters or correlates network events based on a characteristics constraint defined by the analyst.
which the hypotheses about the possible incidents are generated, and (6) the data triage operations performed by the analysts that gradually filter the data indicating suspicious network events.

Data triage is defined by each “state of the CHS”. Assume an analyst is performing data triage of a network. At a certain point of analysis time $t$, a state of data analysis process can be defined by a tuple,

$$S(t) = (t', D(t')_t, A(t')_t, J_t, K_t, H_t, O_t)$$

, where

- $t$ is the current time of analysis system.
- $t'$ is the time when the network events occurred.
- $D(t')_t$ is the data sources to be analyzed at time $t$, which can be generalized to a sequence of network events occurred at time $t'$.
- $A(t')_t$ is the attack chains to be discovered at time $t$, which may leave evidence in $D(t')_t$.
- $J_t = \{e(t')_t\}, R_{e(t')_t}$ is a set of incidents detected at time $t$. $\{e(t')_t\}$ is a set of suspicious network events occurred at time $t'$ which is analyzed at time $t$, and $R_{e(t')_t}$ consists of the temporal and causal relationships among the events in $\{e(t')_t\}$.
- $K_t$ is the analyst’s domain knowledge about the network and attacks at time $t$, as well as the experience knowledge of data analysis.
- $H_t = \{h_t\}, R_{h_t}$ is the analyst’s mental model at time $t$. $\{h_t\}$ is a set of the analyst’s hypotheses about possible attacks. $R_{h_t}$ contains the relationships between the hypotheses in $\{h_t\}$ which are determined by the analyst at time $t$. 
• $\mathcal{O}_{t}$ is a set of data triage operations conducted by the time $t$, which are explained in Section 3.3.

This definition indicates that the data triage, as a CHS, changes from one state to another over time. In this CHS, the data sources ($\mathcal{D}(t')_t$) collected by the sensors report the events in the network, which are determined by the network activities and the attackers’ behaviors ($\mathcal{A}(t')_t$) in the physical world. The time $t'$ refers to the occurring time of network events in the data sources $\mathcal{D}(t')_t$, which is different from the CHS time $t$. The analyst interacts with the collected data ($\mathcal{D}(t')_t$) with the aim of detecting the evidence of $\mathcal{A}(t')_t$. By performing the data triage operations ($\mathcal{O}_{t}$), the analyst gradually filters the suspicious network events and generates hypotheses about the possible attack chains based on his/her observations, and in return updates his/her mental model ($\mathcal{H}_{t}$). Based on the updated mental model, the analyst will be able to report the identified attack incidents ($\mathcal{I}_{t}$). Meanwhile, the detected incidents may further deepen the analyst’s domain knowledge and experience knowledge ($\mathcal{K}_{t}$).

3.2.2 Data Triage Input: Massive and Rapidly Changing Data Sources

Figure 3.2 illustrates the massive and rapidly changing data sources in Cyber SA. The data are categorized into six different dimensions (which can be further extended) and they are as follows:

• **Sensor.** The network monitoring data can be categorized based on the sensors from which they are collected. The common data sources include alerts of intrusion detection systems (IDS) alerts, firewall logs, traffic packages, vulnerability reports, network configurations, server logs, system security reports and anti-virus reports.
• **Data Format.** The format of data can be categorized into structured, semi-structured, and non-structured data.

• **Level of Monitoring Scale.** The data can be divided into the activities of network, host, database, application, and directory.

• **Accessibility.** There are two types of data in this dimension: internal and external data. Internal data refer to the data which can be directly accessed by the analysts within a SOC, while external data refer to the data outside the SOC which will be available by request only.

• **General Type.** This type of data include both qualitative and quantitative data.

• **Timing.** According to whether the data are time-sensitive or not (i.e., timing), the data can be divided into stable and streaming data. Stable data are relatively fixed and not necessarily changing over time, e.g., network configurations and vulnerability reports. Streaming data refer to the data sources
which are continuously collected throughout the run of the network, such as 
IDS alerts and firewall logs. The large volume and time sensitive features 
of the streaming data bring significant challenges for data triage. We thus 
focus on streaming data.

Most of the streaming data are well-formed but there may be different formats 
across various sources. The common ground of streaming data sources is that they 
can be viewed as a sequence of data entries in temporal order considering that 
they are collected over time. A data entry can be an alert, a report or a log item.

The Cyber SA raw data report the network events perceived by the monitoring 
sensors (including human intelligence). From the view point of Cyber SA data 
analysis, we define a network event as a unit of analysis. A network event can be 
identified as one or more data entries from different data sources. For example, an 
IDS alert and an entry in firewall log correspond to a same network event. The 
following elaboration illustrates the way how a network event is defined.

**Definition 3.2.1.** Given a network, a **network event** \( e \) is a multi-tuple that 
specifies the characteristics of a connection activity happened in the network,

\[
e = \langle \text{occurTime}, \text{detectTime}, \text{eventType}, \text{attackType}_{\text{prior}}, \text{srcIP}, \text{srcPort}, \text{dstIP}, \text{dstPort}, \text{protocol}, \text{sensor}, \text{severity}, \text{confidence}, \text{msg} \rangle
\]

Where \( \text{occurTime} \) is the occurring time of the event; \( \text{detectTime} \) is the earliest 
timestamp of the event being detected; \( \text{eventType} \) is the type of network connection 
(e.g., built, teardown and deny); \( \text{attackType}_{\text{prior}} \) is the attack type of an event 
detected and specified by the sensor/agent with the help of prior knowledge; by 
default, \( \text{attackType}_{\text{prior}} \) is null; \( \text{srcIP} \) and \( \text{srcPort} \) are the IP address and port 
of the source of the network connection, respectively; \( \text{dstIP} \) and \( \text{dstPort} \) are the 
IP address and port of the target, respectively; \( \text{protocol} \) is the network protocol; 
**sensor** refers to the sensors who detected this event; \( \text{severity} \) and \( \text{confidence} \)
specify the level of severity and confidence of the event, respectively; \( msg \) specifies other important characteristics of the event, determined by the sensor.

### 3.2.3 Data Triage Output: Incidents in Attack Chains

Data triage, as the initial stage of Cyber SA data analysis, undertakes a number of functions: (1) coordinate diverse data sources and identify the noteworthy network events, and (2) connect these events from the perspective of multi-step attack to support intelligent response. Attack chain is modeled as a sequence changes of network state upon exploitation over exposure [19, 44]. In the context of Cyber SA data analysis, an attack chain can be presented by a sequence of network events reported by multiple (heterogeneous) sensors.

As data from different sources may reflect different network activities, human analysts need to detect the true “signals” from them and “connect the dots” to gain profound understanding of the potential network attacks. The relationships between two events are defined as follows:

**Definition 3.2.2.** Let \( e_i, e_j \) be two network events. We define the temporal and logic relationships between \( e_i \) and \( e_j \) are “happen-before” and “is-a-pre-step”.

- *happen-before*(\( e_i, e_j \))\( : e_i.\text{occurTime} < e_j.\text{occurTime} \)

- *is-a-pre-step*(\( e_i, e_j \))\( : e_i \) and \( e_j \) are in a same attack chain and \( e_i \) is a previous step of \( e_j \) in the attack chain.

The incident reports is the expected outcome of cybersecurity analysis, which specifies the suspicious network events and their relationships The instance of an attack chain is defined as a network incident and its formal definition is given below.

**Definition 3.2.3.** An attack incident is defined as a tuple, \( < \text{att}, E, R > \), where \( \text{att} \) is an attack identifier which specifies an attack chain; \( E = (e_1, \ldots, e_n) \) is a
sequence of network events occurred to carry on the attack chain \( att \). \( R = \{ \text{happen-before}(e_i, e_j), \text{is-a-pre-step}(e_i, e_j) \} \), refer to the temporal or logical relationships between two events \( e_i \) and \( e_j \) in \( E \).

### 3.2.4 Human Analysts in Data Triage: Cognitive Process

To accomplish a data triage task, an analyst have to explore the data sources by conducting a series of operations including searching, filtering and correlating and extracting, interpret the existing observations and generate hypotheses about incidents. This process involves complicated cognitive activities. We define such cognitive process as **analytical reasoning process**, which refers to “central to the analysts’ task of applying human judgments to reach conclusions from a combination of evidence and assumptions” [86]. The analytical reasoning process of an analyst is the driving force of the interaction between the analyst and the data: the analyst’s operations (e.g., filtering out the false positive alerts and identifying the data of interest) are guided by his/her hypotheses about possible attacks; meanwhile, these hypotheses are generated based on the analyst’s current observations of the suspicious events which result from the analyst’s data triage operations. Along with the process, the raw data sources are gradually transformed into the evidence of attack incidents, and meanwhile the analysts gain their Cyber SA by generating and investigating hypotheses about the attack incidents. To sum up, the analytical reasoning process produces two products: (1) the detected network events and their relationships, and (2) the analyst’s mental model which has been updated in the process of analytical reasoning.

#### 3.2.4.1 The AOH Model of Analytical Reasoning Process

The analysts’ sensemaking process (i.e., analytical reasoning process) involves information foraging and sensemaking activities [68]. Drawing on the sensemaking theory, an **AOH Model** is developed to represent an analyst’s analytical reason-
ing process, which is shown on the left side of Figure 3.2. There are three key cognitive constructs in the AOH model: Action, Observation, and Hypothesis. Action refers to the analyst’s data triage operations that filter and correlate network events; Observation refers to the data identified as suspicious or noteworthy by the analyst; and Hypothesis expresses the analyst’s understanding of the current situation, such as an interpretation of the current observation, a question in mind to be further investigated, or a conclusion draw from the existing observation.

The instances of action, observation and hypothesis are called “AOH Objects”, which constitute an analyst’s analytical reasoning process[99, 100]. An analytical reasoning process is an iterative cycle: an analyst taking an action leads to a new observation; this observation may trigger the analyst to generate new hypotheses about the attack incident; investigating a new hypothesis requires the analyst to conduct further action. The relationships between the instances of action, observation and hypothesis are defined as follows.

**Definition 3.2.4.** Let $a_i$ be an instance of action, $o_j$ be an instance of observation, and $h_k$ be an instance of hypothesis, we can define three types of relationships:

- $\text{results}(a_i, o_j)$: Conducting the action $a_i$ results in the observation $o_j$.

- $\text{triggers}(o_j, h_k)$: Hypothesis $h_k$ is generated based on the observation $o_j$.

- $\text{motivates}(h_k, a_j)$: Conducting $a_i$ is motivated to further investigate hypothesis $h_k$.

The AOH objects and their relationships can be represented by tree structures, namely “AOH-Trees”. The nodes of AOH-Trees are AOH objects and edges are the relationships defined in Definition 3.2.4. The root nodes are the initial hypotheses generated by the analyst at the beginning of the data triage task based on the domain knowledge and experience. AOH-Trees may contain multiple root nodes considering that an analyst may have several alternative initial hypotheses. An
Figure 3.3. An example of analytical reasoning process represented by the AOH Objects and their relationships.

The example is shown in Figure 3.3. The analyst generates two initial hypotheses about the possible incidents at the very beginning of the data triage task based on his/her domain knowledge and experience. Elaborating on the initial hypothesis (H0.1), it triggers the analyst’s following action A1 (i.e., filtering the network connections that violate the network policy). Observation O1 is a result of action A1. Based on O1, the analyst comes up with two alternative hypotheses (i.e., H1.1 and H1.2).

Figure 3.4. The H-Trees corresponds to the AOH-Trees in Figure 3.3.
The hypotheses represent the thoughts occurred in analysts’ mind. Focusing on an analyst’s mental process, we leave out the details of the action and observation details and primarily concentrate on the hypotheses and attempt to formulate Hypothesis-Trees (H-Trees). Figure 3.4 is an example of H-Trees extracted from the AOH-Trees of Figure 3.3. The nodes of H-Trees are hypotheses and each edge represents a leads-to relationship between its nodes. The leads-to relationship is defined as follows.

**Definition 3.2.5.** Let $h_i$ and $h_j$ be two hypothesis, an edge pointing from $h_i$ to $h_j$ represents the relationship $leads-to(h_i, h_j)$. The relationship holds iff $a_p, o_q$, that motivates $(h_i, a_p)$ and results $(a_p, o_q)$ and triggers $(o_q, h_j)$

### 3.3 Trace: Cognitive Process of Data Triage

#### 3.3.1 Data Triage Operations

Data triage operations refers to the operations performed by analysts in order to achieve the following goals.

- Detect suspicious network events (i.e., weed out the false positives).
- Correlate the suspicious network events of the same attack chain.
- Generate the incident reports.

In other words, data triage operations refer to the actions taken in an analytical reasoning process, which are motivated by an analyst’s hypotheses about incidents and render the analyst new observations.

In the dynamic CHS of data triage (Section 3.2.1), analysts interact with the network monitoring data by conducting a sequence of data triage operations. A data triage operation may bring two possible aspects: (1) the data transformation
and (2) the mental model transformation, which will be elaborated in the following section.

3.3.2 Data Triage Operation: Data Transformation

As an analyst conducts data triage operations to identify a subset of suspicious events and their relationships (i.e., new observations), in the meantime the hypotheses generated based on the observations may refine the analyst’s awareness of incidents. From the perspective of data transformation, the triage operations transform the raw network data into the evidence of the attack incidents. Each data triage operation filters a subset of network events from the original set of network events specified in the provided data sources. The characteristics of the subset network events are determined by the constraints specified in the data triage operation.

Based on the understanding of data analysis at the level of network events (Section 2.1 in Chapter 2), we identify three types of data triage operations that can result in data transformation: (1) filtering the data sources based on a condition, (2) searching for data by a keyword, and (3) selecting a collection of data with common characteristics of network events. Specifically, they can be defined as follows.

- \textit{FILTER}(D_{input}, D_{output}, C):\ Filtering a set of network events \(D_{input}\) based on a condition \(C\) and resulting in a subset \(D_{output}\).

- \textit{SEARCH}(D_{input}, D_{output}, C):\ Searching a keyword \(C\) in a set of network events \(D_{input}\) and resulting in a subset \(D_{output}\).

- \textit{SELECT}(D_{input}, D_{output}, C):\ Select a subset \(D_{output}\) from a set of network events \(D_{input}\). The network events of subset \(D_{output}\) have a common characteristic \(C\).
3.3.3 Data Triage Operation: Refining Analysts’ Mental Model

A data triage operation performed by an analyst results in a new observation, which may (1) trigger the analyst’s new hypotheses or (2) confirm/invalidate his/her previous hypotheses. In either case, the corresponding H-Trees (representing the analyst’s mental model) is refined by gaining better Cyber SA. We define the following H-Trees operations to represent the operations that modifies an analyst’s mental model.

- **NEW_HYPO**($h, O$): Generate a new hypothesis $h$ in the context of current observation $O$.
- **MODIFY**($h, v_1, v_2$): Modify the content of an existing hypothesis $h$ from $v_1$ to $v_2$.
- **CONFIRM/DENY**($h, T/F$): Confirm or deny an existing hypothesis $h$.

3.3.4 Data Triage Trace Representation

We represent an analyst’s analytical reasoning process of Cyber SA data triage as a trace, including the AOH-Trees, H-Trees and a sequence of data triage operations in temporal order. The definition of trace is shown below.

**Definition 3.3.1.**

$$\mathcal{T} = (\mathcal{G}_{AOH}, \mathcal{G}_H, \mathcal{S}_{op})$$

where

- $\mathcal{G}_{AOH}$ is the AOH-Trees, which is a heterogeneous tree structure involving an analyst’s actions (i.e., data triage operations), observations of suspicious network events, and hypotheses about possible incidents. The edges are the relationships between the AOH objects (Definition 3.2.4).
• $S_H$ is the H-Trees which consists the hypotheses in AOH-Trees. H-Trees represent the analyst’s mental model of Cyber SA.

• $S_{op}$ is a sequence of data triage operations $(p_1, p_n)$ in temporal order. $\forall p_i (1 \leq i \leq n)$, $p_i$ is a tuple $(t_i, (op)_i(I, C_i))$, where $t_i$ is the timestamp of $p_i$, and “$(op)_i(I, C_i)$ is an operation on an AOH object (i.e., data triage operations and H-Trees operations) $I$ under context $C_i$. $C_i$ consists of a set of network events that have been investigated by the analyst [100].

4.1 Introduction

As data triage is a very complicated cognitive task, which involves various cognitive activities, it requires analysts to draw on their knowledge and experiences. There are several important benefits for capturing and analyzing an analyst’s cognitive process (i.e., analytical reasoning process) of data triage, which can be best illustrated in the following example.

A Motivating Example. An analyst Alice was conducting data triage given two data sources (which are IDS alerts and firewall logs). To detect malicious network events, Alice started with browsing the IDS alerts. She first noticed some network events about SMB (Sever Message Block) buffer overflow attempts on the internal DNS server, which made her generate a hypothesis about DNS exploit. Meanwhile, she thought another possibility could be that normal DNS updates triggered the alerts. To verify the first hypothesis about DNS exploit, she further
turned to the firewall logs to investigate the network connections to the DNS server. Instead of finding any suspicious DNS connection, she found out some connections were built between the internal IP addresses and external IP addresses through the port “6667”. According to her domain knowledge, port “6667” is usually used for IRC (Internet Relay Chat) service which may serve as a communication channel in a botnet. Based on this observation, she generated a new hypothesis about malicious IRC communication. To verify this hypothesis, she filtered the IDS alerts based on the port “6667” and found some network events about the IRC connections between the same set of IP addresses. This observation strengthened her previous hypothesis about the IRC communication in a botnet. She also linked this new finding to her hypothesis about the DNS exploit, believing that the DNS exploit was an antecedent cause of the malicious IRC communication. Finally, she described this attack chain in her incident report.

Based on the analysis of her cognitive process, we can identify the steps involved in Alice’s exploring the data sources: generating and verifying hypotheses, and “connecting the dots”. There are several benefits for understanding an analyst’s cognitive process. Firstly, knowing how Alice generates a hypothesis of an attack event is helpful for understanding her conclusion about the incident and for assessing how valid and reliable the conclusion is, which can enhance the accountability of decision making. Secondly, through reflecting on the cognitive process, Alice may realize there is no sufficient evidence showing that the DNS server attack is an antecedent cause of the malicious IRC communication, which means that her reasoning about the causal relationship between these two hypotheses may be invalid. She relied too heavily on her first finding (i.e., the suspicious events related to the DNS server), which is called anchoring bias in cognitive science [39]. Therefore, lessons can be learned through her reflection and thus Alice’s performance can be accordingly improved over time. Thirdly, Alice’s domain knowledge and experience play a critical role in her analytical reasoning process. For example,
the suspicious IRC communication events were detected based on a port of interest (i.e., port “6667”); and Alice strengthened her hypothesis by seeking for consistent evidences in data sources. Such domain knowledge and data triage strategies can be transferred into cognitive aids and tools for bolstering the agility of cyber defense. Besides, a comparison between the analytical reasoning processes of experienced analysts and less-experienced analysts help researchers recognize the differences between experts and novices, and more effective programs for training can be developed. Fourthly, if Alice was working together or work on shifts with other analysts, the presentation of Alice’s analytical reasoning process can help others better comprehend her observations and hypotheses so that they can collaborate better in data triage.

4.1.1 Need for Studying Fine-Grained Cognitive Processes

To achieve these benefits, it is highly desirable to capture and study the analysts’ fine-grained cognitive processes of data triage. The term “fine-grained cognitive process” refers to an analytical reasoning process of data triage involving the following detailed activities: (1) actions performed to detect and correlate suspicious network events (i.e., data triage operations), (2) observations of suspicious network events; (3) hypotheses about suspicious and related network events that are generated/updated under a current context.

The reason why we are interested in the fine-grained cognitive process comes from the consideration of characteristics of data triage. First of all, data triage are by nature data-driven and labor-intensive [4]. The network monitoring data are overwhelming because most cyber defense technologies (e.g., IDS, firewall, and network traffic monitor) are prone to data flooding [20]. Therefore, it places high demands on analysts’ expertise to process the large amount of data effectively to detect the evidence of incidents. To capture an analyst’s expertise, we should audit the data triage actions performed by an analyst for detecting and correlating
suspicious network events. Meanwhile, it is also important to know what the resultant observations are, how the analyst interpret these observations, and what hypotheses about the potential incidents are generated by the analyst based on the existing observations. Secondly, the relationships between an analyst’s hypotheses should also be taken into account in the investigation of the analyst’s cognitive process. The dynamic evolution of network and threat behaviors requires analysts to connect their findings to maintain a holistic and history-aware understanding of the network situation [4]. An analyst mainly depends on his/her domain knowledge or experience to decide what hypothesis need to be further investigated and how the new observations (i.e., findings) may influence the previous hypotheses. Capturing such relationships also requires fine-grained capturing of analysts’ cognitive processes.

4.1.2 Difficulties with Data Triage Cognitive Task Analysis

Cognitive Task Analysis (CTA) is a commonly used method that “help researchers understand how cognition makes it possible for human to get things done and then turning that understanding into aids or high tech for helping people get things done better” [13]. Many studies have used various CTA methods to identify cognitive activities [68], to elicit mental models [13], or to study expert/novice difference in performing various complex cognitive tasks [41]. Our work is motivated by the difficulties in conducting CTA studies on Cyber SA data triage. First of all, analysts are fully concentrated on accomplishing their data triage tasks and are sensitive to interruptions. Using typical think-aloud protocols during data triage tasks could be an undesirable distraction for analysts because it is likely to influence their performance. Besides, self-report techniques and think-aloud rely on analysts’ motivation and “self-CTA” capability to report their cognitive activities [13]. However, analysts may be unable to give a complete and accurate report of their own cognitive processes [92], especially for the experienced analysts [10]. Be-
sides, analysts may be unwilling to self-report because they are totally engaged in their analysis tasks. The second challenge for conducting CTA on Cyber SA data triage is the tight schedule of professional analysts, considering that professional analysts conduct 24/7 (24 hours a day, 7 days a week) security operations and rotate through day shift and night shift [78]. It renders some difficulty for them to participate in interviews. Thirdly, conducting CTAs on Cyber SA raises several important practical issues, including the accessibility to analysts and an organization’s network. Researchers outside an organization may have limited access to the real data sources and attack events as most of them are kept confidential.

4.1.3 Integrated Method for Tracing Analyst’s Cognitive Process

Due to the difficulties in conducting CTA studies on data triage, there is a gap between the need for capturing fine-grained cognitive processes of data triage and the limited capability of existing CTA methods. To fill the gap, we propose an integrated computer-aided CTA data collection method to trace the analysts’ fine-grained cognitive processes. This method mainly relies on tracing the analytical reasoning process by integrating automatic capture and situated self-reports in a novel way. A computer toolkit is developed by us to support tracing the analytical reasoning processes of analysts while they are concentrated on their data triage tasks. An experiment has been conducted to evaluate the feasibility of the method, in which participants were asked to perform a simulated data triage task by working with the tool. The experiment is carefully designed to ensure the participants’ task performance to accord with their performance in the real-world jobs.

Our study shows that the proposed tracing method enables running fine-grained CTA study of analysts’ cognitive processes of data triage at a lower cost. The preliminary results not only provide valuable insights into the analysts’ analytical reasoning processes but also is a basis for more in-depth trace analysis to develop
new techniques to facilitate Cyber SA data triage.

4.2 Related Work

There are multiple techniques used by CTAs, including interviews (e.g., unstructured, semi-structured, structured interviews [40]), observations (e.g., shoulder surfing), self-reports (e.g., questionnaires, surveys and diaries), process tracing (e.g., eye-tracking, think-aloud protocols [27, 82]). Each technique has its own advantages and disadvantages [13, 11]. Wei and Salvendy carefully compared these advantages and disadvantages and provided 11 useful guidelines in selecting CTA techniques according to a particular task [90].

Existing CTAs focused on analytics tasks involve multiple CTA techniques. Priolli and Card conducted interviews and think-aloud with intelligence analysts and proposed a notional model of sensemaking process which involves leverage points dealing with data overload and attention management [68]. D’Amico and Whitley leveraged interviews, observations and self-reports to investigate how the data sources are transferred into SA through the cognitive process of computer network defense (CND) analysis and identified three stages of CND analysis in a generalized CND workflow [17, 18]. To identify the needs of analysts for visualizations in their tasks, another study followed a multi-phase CTA using interviews and self-reports to analyze analysts’ cognitive activities and proposed a task-flow diagram [24].

To improve the effectiveness of CTA studies, some automated tracing techniques have been developed to capture human behaviors automatically, such as eye movement tracking [13] and mouse-click keystroke logging [49]. Although such techniques obtain quantitative data unobtrusively and collect complementary data that may provide useful insights into the cognitive process of interest, these behavior tracking techniques fail to capture the information about subjects’ rationale for
choosing a course of action, which is critical for the subjects’ cognitive processes. Without such information, the data could be incomplete and have limited utility.

Process tracing is an effective approach for capturing participants’ cognitive processes by asking them to go through a task [13, 27]. However, we have observed that the existing techniques supporting process tracing (e.g., think-aloud protocols and the above behavior recording) suffer from various disadvantages in the CTAs of data triage. Therefore, there is a need for more efficient and less interruptive method for tracing the cognitive processes and capturing the main elements in analysts’ cognitive processes.

4.3 Computer-Aided Cognitive Process Tracing

Method

4.3.1 Guidelines

To achieve the benefits of gaining better understanding of analysts’ cognitive process, we developed a computer-aided method for tracing analysts’ fine-grained cognitive processes on account of the difficulties with CTA studies on Cyber SA data triage. To achieve our ultimate goal of gaining deep understanding of analysts’ cognitive processes, three important guidelines are identified in the method design.

- (G1) In order to capture the fine-grained cognitive activities of data triage, the method should capture the details of an analyst’s data triage operations, observations of suspicious events, and hypotheses about possible incidents. Besides, it should record the temporal order of the analyst’s data triage operations and the relationships among the analyst’s actions, observations, and hypothesis (Definition 3.2.4).
• (G2) The method should be minimum reactive. Reactivity refers to the influence of the process of observing analysts’ analysis on their behaviors being observed [70]. Cybersecurity analysts are working under extremely high time pressure regarding the rapidly changing cyber environment, and maintaining a working memory is critical to identify the relationships between recognized network events. Any distraction caused by the capture method could affect their performance on the cyber analysis task. Therefore, the tracing method should avoid unnecessary interference in analysts’ job/task performance.

• (G3) It should be feasible to launch the tracing method in the real-world setting considering the concern from organizations about the schedule of analysts and confidential information.

### 4.3.2 The Architecture of the Tracing Method

Figure 4.1 shows the architecture of the computer-aided method for tracing analysts’ fine-grained cognitive processes of data triage, which integrates both automatic capture and situated self-reports. The trace representation defined in Section 3.3.4 specify what fine-grained cognitive activities should be captured, considering the first guideline G1. Following the second guideline G2, a computer toolkit has been developed to capture analysts’ cognitive processes in a passive manner in order to reduce unnecessary interference caused by CTA data collection in analysts’ task performance. The functions provided by the toolkit will be described in details in the following sections. In terms of the third guideline G3, the tracing method lessens organizations’ concerns in the following two aspects: (1) the analysts (i.e., subjects) can work merely with the toolkit while they are accomplishing a data triage task, without interacting with the researchers outside their organizations; (2) the captured cognitive processes of analysts are retained in formats that are easy for organizations to review for security clearance.

The toolkit, named **ARSCA** (Analytical Reasoning Support for Cybersecurity
Figure 4.1. The architecture of the method for tracing an analyst’s cognitive process of data triage

Analytics), is a specific measure of the tracing method. It not only automatically captures analyst’s cognitive process of data triage but also manages analysts’ existing cognitive processes (i.e., actions, observations and hypotheses, and their relationships). It consists of three main modules: (1) a cognitive element management module, which manages the analyst’s existing actions, observations and hypotheses and their relationships; (2) a user interface module, which supports various human-data interactions and provides a Data View of the network data and a Analysis View of an analyst’s existing cognitive process; and (3) a CTA data collection module, which integrates both automatic capture and self-report techniques to capture an analyst’s cognitive process based on the trace representation. The modules are described as follows.
4.3.3 Cognitive Element Management

To accomplish a data triage task, an analyst usually explores the data sources to detect suspicious network events and to gain insights of the possible incidents, which is modeled as an analytical reasoning process. We mainly focus on the three main elements in an analyst’s cognitive process: action, observation and hypothesis. Described in Section 3.2.4.1 in Chapter 3, actions refer to an analyst’s data triage operations; observations are the suspicious clues or evidence detected by the analyst from the data sources; and the hypotheses refer to the analyst’s thoughts generated during the task, such as an interpretation of the existing observation, a plan for further investigation and an insight/conclusion of the incident.

The cognitive element management module maintains the analyst’s actions, observations and hypotheses generated along with his/her analytical process. Besides, this module also connects the elements based on their relationships (Definition 3.2.4): an observation is connected to an action for being resulted from the action; a hypothesis is connected to an observation for being generated to interpret the observation; and an action is connected to a hypothesis for being conducted to further investigate the hypothesis. The elements and their relationships will then be presented in the Analysis View in the user interface module, as discussed below.

4.3.4 User Interface

ARSCA provides a Data View and an Analysis View. Figure 4.2 shows the main interface of ARSCA. The Data View (Figure 4.2 (a)) presents the sources of network events and support analysts’ data triage operations. The Analysis View (Figure 4.2 (b)) provides a overview of the analyst’s existing cognitive processes by visualizing the AOH-Trees (consists of the analyst’s actions, observations and hypotheses) and H-Trees (consists of merely hypotheses), which enables the analyst to review and modify his/her previous hypothesis about the potential incidents.

More specifically, ARSCA provides three types of functions: (1) supporting the
Figure 4.2. Main components of the ARSCA user interface. It includes (a) a Data View, which presents the network monitoring data and supports data triage operations; and (b) a Analysis View, which displays the existing AOH-Trees and H-Trees and supports the operations related to H-Trees.

analyst’s data triage operations, (2) capturing the analyst’s cognitive process, and (3) managing the AOH-Trees and H-Trees created by the analyst during his/her data triage process.

- **Supporting Data Triage Operations.** ARSCA supports the data triage...
operations defined in Section 3.3.1, including \textit{SEARCH}, \textit{FILTER}, and \textit{SELECT}. Besides, ARSCA also provides several functions for facilitating data analysis, including \textit{INQUIRE} and \textit{LINK}. In Figure 4.2, Region 2 enables an analyst to search for a keyword (i.e., \textit{SEARCH} operation), and Region 3 lets an analyst to specify a condition to filter the current data set (i.e., \textit{FILTER} operations). Region 4 is a helper for analysts to inquire about a certain port or a specific term appearing in the data sources (i.e., \textit{INQUIRE} operation). Region 5 shows that an analyst can select the suspicious network events from the data sources. Once selected, the selected data entries will be displayed in another window (Region 6) for the analyst to review and confirm them (i.e., \textit{SELECT} operation). ARSCA also enables analysts to identify the common characteristics of the selected data among the selected data in Region 6 (i.e., \textit{LINK} operations). With the help of this toolkit, the analyst can write down his/her hypotheses based on the current observation. This function is shown in Region 7 and 8, which enables the \textit{NEW HYPO} operation. ARSCA visualizes the analyst’s existing hypotheses and enables the analyst to modify them. Region 13 presents the details of the analyst’s previous hypotheses and allows the analyst to modify the description of or to confirm/deny a selected hypothesis (i.e., \textit{MODIFY} and \textit{CONFIRM/DENY} operation).

- **Capturing Cognitive Process of Data Triage.** ARSCA integrates both automatic capture and self-report to capture an analyst’s cognitive process based on the trace representation, which will be detailed in Section 4.3.5.

- **Managing AOH-Trees and H-Trees.** ARSCA captures an analyst’s actions, observations and hypotheses, and its cognitive element management model connects them according to their mutual relationships, which forms the AOH-Trees and H-Trees (defined in Section 3.2.4.1 in Chapter 3. Given
the AOH-Trees and H-Trees, ARSCA presents them to the analyst as an overview of the existing cognitive process of the analyst. ARSCA displays AOH-Trees and H-Trees using a nested structure because analysts are familiar with this display in their work settings. Region 9 shows AOH-Trees, where an action appears together (as a node) with an observation indicates the action resulted in this observation; a hypothesis nested under an observation means the analyst created the hypothesis based on the observation; an action nested under an hypothesis indicates the hypothesis motivated the analyst to perform this action for further investigation. Region 12 shows H-Trees, which is visualized in the same nested tree structure. To view the details of an existing action, observation or hypothesis, an analyst can select the node (Region 10 or Region 12) and view the details in Region 11 and Region 13. Besides, ARSCA enables the analyst to change his/her current focus of attention from one hypothesis to another by double clicking the targeted hypothesis. After changing the focus, ARSCA will nest the following action and observation under the analyst to the targeted hypothesis.

4.3.5 Integrated CTA Data Collection

The CTA data collection module integrates leveraging automatic capture and self-report in the following way.

- **Tracing Actions (Automatic Capture).** An analyst’s data triage operations are captured automatically: ARSCA records the condition specified in `FILTER` operation, the keyword used in `SEARCH` operation, or the data entries selected in `SELECT` operations, once a data triage operation is performed.

- **Tracing Observations (Automatic Capture).** Once an analyst recognizes something interesting from the result of a data triage operation, he/she
can select the suspicious subset as an observation on ARSCA, and meanwhile ARSCA will record all the selected data entries behind the scenes.

- **Tracing Hypotheses (Situated Self-reports).** Once an analyst generates a new hypothesis about an incident based on an observation, he/she can type it down in ARSCA so that ARSCA can link the newly-generated hypothesis to the current observation. At the back end, ARSCA records not only the details of this hypothesis but also its relationship with the observation.

### 4.3.6 Rationale of Integration

The process tracing supported by ARSCA integrates automatic capture and self-reports, both of which have their advantages and disadvantages. As for automatic capture, it obtains precise quantitative data unobtrusively and the records can give useful insights into analysts’ behaviors [91]. However, it can’t capture analysts’ comments and in-depth thoughts. Besides, the collected data don’t contain context information. If there are no other data provided, the follow-up analysis can be difficult which may limit the utility of the data. As for self-reports, it may gain quality data, but it depends heavily on participates’ willingness to report. ARSCA, combined the automatic capture with situated self-reports, can overcome these shortcomings. After an action or observation is automatically captured, the following hypothesis can provide the analyst’s comments on the observation. Meanwhile, the automatically captured data provide the context information for the hypothesis. Moreover, the Analysis View (i.e., Process Navigation) helps analysts reflect on their previous analytical reasoning process and organize their thoughts. Therefore, it encourages analysts to report their findings and thoughts in the tasks. (However, the Analysis View is not indispensable. Analysts can also accomplish their tasks without using the Analysis View if they don’t like it).
4.4 ARSCA Implementation and Testing

4.4.1 ARSCA Implementation and Output

ARSCA is implemented as an interactive Windows platform application which runs on .Net Framework 4.0 or above versions 1. The data sources contain more than 100,000 entries which can be loaded in a tabular view within one second. The cognitive element management module and the CTA data collection module work saliently in the background.

ARSCA’s output (i.e., a trace) includes the AOH-Trees, H-Trees, and a sequence of data triage operations in temporal order, which are written in XML format. A example of a sequence of data triage operations is shown in Table 4.1. The first operation is a FILTER operation which was conducted to filter the firewall log based on the condition “protocol = TCP”. The second one is a SELECT operation for selecting two entries in firewall log as an observation of suspicious network events. The last one is an operation that the analyst wrote down a thought (i.e., “a thought”). All these operations are recorded with timestamps in a trace.

4.4.2 Comparison of ARSCA with Automated Logging Tool

We make a further comparison of ARSCA traces (i.e., the sequence of data triage operations) with the log of an automated keystroke logging tool called RUI (Record User Input). RUI (version2.3) is a well-tested tool that can record the interface behaviors (i.e., keystrokes and mouse clicks) and runs in the background [61]. In the test, we ran both ARSCA and RUI to record a same set of operations. Table 4.2 shows the RUI log, each line of which records the timestamp, type of action, and the coordinates of a mouse click action. Mapping the RUI records in Table 4.2 to the ARSCA trace in Table 4.1, the RUI records of line 2-7 are for the FILTER operation; line 8-10 correspond to the SELECT operation; and line 11-20 capture the NEW_HYPO operation.
Table 4.1. An example of a sequence data triage operations recorded by ARSCA

<table>
<thead>
<tr>
<th>#</th>
<th>Data Triage Operations in a Trace</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><code>&lt;Trace ID=&quot;5333322&quot;&gt;</code>&lt;Item Timestamp= &quot;05/24/2014 13:24:15&quot;&gt;FILTER(SELECT * FROM Firewall Log WHERE Protocol = TCP)`</td>
<td>Filter firewall log based on Protocol = TCP</td>
</tr>
<tr>
<td>2</td>
<td><code>&lt;Item Timestamp=&quot;05/24/2014 13:25:29&quot;&gt;SELECT(FIREWALL-[4/5/2012 10:15:00 PM]-[Built]-[TCP] (172.23.240.254, 10.32.5.59), FIREWALL-[4/5/2012 10:15:00 PM]-[Built]-[TCP] (172.23.30.220, 10.32.0.100),)</code></td>
<td>Select a subset of network events (i.e., the underlined firewall log entries)</td>
</tr>
<tr>
<td>3</td>
<td><code>&lt;Item Timestamp=&quot;05/24/2014 13:34:41&quot;&gt;NEW_HYPO(H_(3524121), H_(44524411) &quot;a thought&quot;)&lt;/Item&gt;</code></td>
<td>Generate a new hypothesis “a thought”</td>
</tr>
</tbody>
</table>

In total, we performed 126 operations, including 15 *FILTER*, 22 *SEARCH*, 15 *INQUIRE*, 17 *SELECT*, 10 *LINK*, 13 *NEW_HYPO*, 10 *MODIFY*, 14 *SWITCH_CONTEXT*, 10 *CONFIRM*-*DENY* operations. We first went through each item in the ARSCA traces to investigate whether the operations were correctly captured. We found that the descriptions of the operations recorded in the traces are correct. We further checked whether the timestamps in ARSCA traces were accurate by matching them to the timestamps in RUI logs. According to a Wilcoxon signed-rank test, the median absolute time difference is significantly less than 1000 (Wilcoxon Statistic = 535.0, P-value = 0.000) in the unit of millisecond. It indicates no difference because timestamp in RUI log is accurate to .001 second while the trace timestamp are accurate to 1 second.
Table 4.2. RUI log of a same set of operations recorded in Table 4.1.

<table>
<thead>
<tr>
<th>No.</th>
<th>Elapsed Time</th>
<th>Action</th>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>2014-05-24T13:34:37.780</td>
<td>Key a</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>2014-05-24T13:34:38.198</td>
<td>Key t</td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>2014-05-24T13:34:38.301</td>
<td>Key h</td>
<td></td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td>2014-05-24T13:34:38.526</td>
<td>Key u</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18.</td>
<td>2014-05-24T13:34:38.608</td>
<td>Key g</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19.</td>
<td>2014-05-24T13:34:38.701</td>
<td>Key t</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.5 Evaluation

We evaluate the proposed tracing method by considering four main questions: (1) whether the tracing method is feasible for tracing analysts’ cognitive processes of Cyber SA data triage? (2) whether the tracing method influences participants’ performance of data triage? (3) to what extent the traces collected by ARSCA capture the participants’ cognitive processes in the task?

4.5.1 Feasibility of Tracing Cognitive Processes of Data Triage

To evaluate the feasibility of the tracing method, an experiment was designed and conducted to trace cybersecurity analysts’ cognitive process when they were performing a Cyber SA data triage task. The Cyber SA data triage task was carried
out in a simulated task with each participant given two data sources (i.e., IDS alerts and firewall logs) and the analysts were asked to detect and correlate suspicious network events. The task should to be of reasonable complexity that is close to the real-world one and also be workable for participants to complete within a limited experiment time. This is a main challenge we have to take into consideration in designing a data triage task, which will be detailed in the following Chapter 5. We designed our task by partially adopting the network dataset from VAST Challenge 2012 Mini Challenge 2 produced by the Visual Analytics Community [94]. The VAST dataset are of high quality [77], containing 23,711,341 firewall logs and 35,948 IDS alerts. It also implies a realistic attack scenario which is a multi-step attack taking place within 40 hours on the network of an organization containing approximately 5000 hosts. However, the original dataset was too huge to be analyzed by the participants in the given time. Given the ground truth (i.e., the original attack events occurring during the 40 hours), we extracted a small portion from the original data set that corresponded to a critical 10-minute period with four types malicious network events, including 239 IDS alerts and 115,524 firewall logs. The malicious correlated network events include: (1) the IRC communication between several inside workstations with a set of outside Command and Control (C&C) severs, (2) denied FTP connection attempts for data exfiltration, and (3) successful data exfiltration using SSH.

We were able to get access to the professional analysts with the collaboration of Army Research Lab. After obtaining the IRB approvals, we recruited thirteen full-time professional cybersecurity analysts from Army Research Lab and seventeen doctoral students specialized in cyber security from our University. All participants in our experiment were screened to ensure their possession of sufficient domain knowledge and expertise in Cyber SA data triage.

Considering the potential influence of the participant’s proficiency with the ARSCA on his/her task performance, a tutorial session was provided before the
task to train the participant of using ARSCA. At the end of the session, we required each participant to pass a quiz to make sure that the participant is familiar with the ARSCA before undertaking the task. After the tutorial session, we asked each participant to accomplish the task by working together with ARSCA.

At the end of the experiment, we asked each participant to complete a 15-minute post-task questionnaire. The post-task questionnaire included several rating questions using a five-point Likert scale for their opinions on their performance by working with ARSCA. Moreover, there were several open-ended questions for a participant to report his/her key findings and conclusion about the incidents and to explain his/her analytical reasoning process. The participant’s answers to these questions were evaluated as an important component of his/her task performance. Besides, the answers enabled us to evaluate the trace captured by ARSCA.

In this experiment, we successfully captured the traces of the data triage cognitive processes of the thirty participants. The realization of the experiment lied in the greatly reduced cost of running CTA studies and its reviewing convenience of the captured data for the organizations out of confidential concern with our tracing method.

4.5.2 Analysts’ Operation Patterns Revealed by Traces

Given the traces collected in the experiment, we observed several common patterns of the participants’ operations. It indicates that the proposed tracing method captures several typical cognitive activities, including identifying observation, creating hypotheses and hypothesis investigation, which are consistent with the results of the existing CTA studies on Cybersecurity analytics [18, 17]. We used a qualitative analysis method to explain the cases of the common patterns. The unit of analysis is operation. We analyzed a sequence of operations in a trace from two aspects: (1) whether this operation causes any changes to the participant’s actions, observations or hypotheses, and (2) why the participant conduct this operation in that
4.5.3 Complementing of Automatic Capture and Self-Report

Table 4.3. A partial sequence of data triage operations from a professional analyst’s trace

<table>
<thead>
<tr>
<th>#</th>
<th>Data Triage Operations in a Trace</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SELECT * FROM Firewall Log WHERE DstPort = 6667</td>
</tr>
<tr>
<td>2</td>
<td>FIREWALL-[4/5/2012 10:15:00 PM]-[Built]-[TCP] (172.23.233.57, 10.32.5.58),</td>
</tr>
<tr>
<td></td>
<td>FIREWALL-[4/5/2012 10:15:00 PM]-[Teardown]-[TCP] (172.23.233.52, 10.32.5.59),</td>
</tr>
<tr>
<td></td>
<td>FIREWALL-[4/5/2012 10:15:00 PM]-[Teardown]-[TCP] (172.23.233.58, 10.32.5.51),</td>
</tr>
</tbody>
</table>
| 3  | NEW_HYP0(44524411) "It could be an indicator of compromise because malware can leverage IRC for Command to Control (C2) communication."
| 4  | FILTER(SELECT * FROM IDS Alert WHERE DstPort = 6667)                                             |

specific context. The case studies are presented in Chapter 5.
The integrated tracing method collects two types of CTA data in traces: the automatically captured CTA data and the self-reported CTA data. Take the partial trace in Table 4.1 as an example. The \textit{FILTER} and \textit{SELECT} operation belongs to the \textbf{automatically captured CTA data} because it was recorded automatically, but the \textit{NEW\_HYPO} operation belongs to the \textbf{self-reported CTA data} because it relied on the analyst to type down his/her hypothesis. Among the 331 operations captured in the 30 traces, there are balanced number of automatically captured items and self-reports: 151 items were automatically captured (mean=5.59, SD=3.05), and 180 were collected by self-reports (mean=6.67, SD=5.07). We found that the automatically captured and self-reported CTA data complemented each other while analyzing the traces. Next, we will discuss how these two types of data complement each other based on a partial trace of an participant’s data triage cognitive process as an example (shown in Table 4.3). In the partial trace, a \textit{FILTER} operation was first conducted to filter the firewall log based on the condition “DstPort = 6667”, which was followed by a \textit{SELECT} operation that selected a subset from the firewall log as an observation. Based on the observation, a hypothesis about malicious IRC communication was generated by performing a \textit{NEW\_HYPO} operation, which triggered the following \textit{FILTER} operation for further investigation.

**Automatically Captured Action Describing Automatically Captured Observation.** In this case, the participant first filtered the firewall log, and then selected a subset from the filtered data set as his new observation. The \textit{FILTER} operation (#1) implies that the data selected in the \textit{SELECT} operation satisfy the condition specified in the \textit{FILTER} operation (i.e., “DstPort = 6667”). It means \textit{FILTER} operation provides us more information about the following \textit{SELECT} operation: we would not be clear about what common features are shared by the data recorded \textit{SELECT} operation.

**Automatically Captured Observation Providing the Context Infor-
mation for Self-Reported Hypothesis. According to the $NEW\_HYPO$ operation in Table 4.3, we learn that the participant generated his hypothesis (i.e., $NEW\_HYPO$) right after the $SELECT$ operation. The $SELECT$ operation records the details of selected suspicious network events, which provide the context information (i.e., evidence) of the self-reported hypothesis. In this way, we can easily learn under what situation the participant generated this hypothesis while analyzing the trace. The role of observation was also confirmed by a participant’s comment in the post-task questionnaire:

“Sometimes when exploring multiple thought processes I would add to a small bit of context information to confirm a hypothesis (find initial IRC traffic, then later expand on that).”

Self-Reported Hypothesis Explaining the Motivation of the Automatically Captured Action. The hypothesis generated by $NEW\_HYPO$ can help us explain why the participant performed the following $FILTER$ operation #4: the participant filtered the IDS alerts using the same condition (i.e., DstPort = 6667) which was used in the $FILTER$ operation #1 because he may want to search for more evidence in the IDS alerts that support this hypothesis generated based on the observation in the firewall log. Therefore, the hypothesis specified in $NEW\_HYPO$ operation explains the motivation of the participant performing the following data triage operation.

4.5.4 Traces Capturing the Key CTA Data

We expect that the process tracing can capture the evidences and thoughts that are considered important by the participants in their cognitive processes of data triage. In the post-task questionnaire, participants were asked to answer four open-ended questions, which are shown in Table 4.4.
Table 4.4. The four open-ended questions in the post-task questionnaire.

<table>
<thead>
<tr>
<th>Code</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMP_OBS</td>
<td>“Reflecting back, what are the three most important evidences that you observed in the data that contributed to your conclusion?”</td>
</tr>
<tr>
<td>FD_OBS</td>
<td>“Please explain how you find the above evidences.”</td>
</tr>
<tr>
<td>IMP_HPY</td>
<td>“Reflecting back, what are the three most important thoughts in your mind that contributed to your conclusion?”</td>
</tr>
<tr>
<td>EVTS</td>
<td>“Based on your analysis, please create one or more narratives that describe your conclusion of the incident (i.e., tell the storyline of the possible incidents)”</td>
</tr>
</tbody>
</table>

To check whether the participants’ answers to the questions in Table 4.4 are captured in the traces, we performed a thematic analysis of the free-text answers to extract the themes in each of them and then match them to the traces. The procedure of our analysis is as follows. We first weeded out the undesirable answers, including irrelevant answers (e.g., “the tool is helpful”) and vacuous answers (e.g., “I don’t have any comment”). Next, we analyzed the content of each answer and extracted themes from it (i.e., coding the answer). The unit of analysis is sentence. We have one main coder and one evaluator. Given a sentence in an answer, the main coder and the evaluator first read through it respectively, and then the main coder extracts the themes and generate a preliminary coding. After that, the evaluator proofread the coding and revised it when necessary. At last, we matched the codes to relevant items in the traces.

We have 30 answers to “IMP_OBS”, 27 answers to “FD_OBS”, 29 answers to “IMP_HPY”, and 29 answers to “EVTS” after data cleaning. Figure 4A–5 presents an example of a coded answer. Themes are marked by “#”, and “[H]” indicates a theme about hypothesis/thought and “[O]” indicates a theme about observation/evidence. We use different colors to illustrate the themes that cor-
I conclude that there was likely IRC communication on the network. This could have been a policy violation or malware C2 communication. In addition, I saw attempts for FTP (tcp/21), SSH (tcp/22), and HTTP (tcp/80) from internal address space to the internet, which are policy violations. Lastly, there were a large number of IDS alerts that were generated about SMB null sessions and overflow attempts which point to potentially malicious activity over SMB.

Figure 4.3. Extracting themes from the text of an answer to EVTS

respond to different attack events. In total, we generated 318 themes in these answers. Fortunately, we found that all the themes in the answers to “IMP_OBS”, “IMP_HPY” and “EVTS” were captured in traces. As for “FD_OBS”, 5 themes in its answers were not captured by the trace. The themes were about either the domain knowledge used (e.g., “some ports can be used by attacker”) or implicit assumption made by the participants (e.g., “outbound connections are not allowed by the network policy”), which are the implicit parts and can be inferred by analyzing the trace. In conclusion, the results of the content analysis show that traces can capture the key evidence and thoughts in the cognitive processes of data triage.

4.5.5 Retrospection on Traces

Comparing the traces with the participants’ retrospection is another way to evaluate to which extent the tracing method captures the participants’ data triage cognitive processes. It is not hard to see that most participants failed to remember the details of their cognitive processes in retrospect. Moreover, we did not
Table 4.5. The procedure of trace-stimulated recall

<table>
<thead>
<tr>
<th>Step 1: Recall without any aid.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q1: What can you recall in your previous analytical process on mm/dd/yyyy?</td>
</tr>
<tr>
<td>(hint: think about the evidence you found in the data, how you explore them, thoughts or conclusions)</td>
</tr>
<tr>
<td>Q2: Can you remember what strategies you used in your analysis?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 2: Recall with the aid of trace.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Go through the trace and answer Q1 and Q2 with the aid of the trace.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Step 3: Evaluate the effect of trace on recall.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q3: What do you think improve your recall?</td>
</tr>
</tbody>
</table>

A five-point Likert Scale:
- Strongly Disagree (1), Disagree (2), Neutral (3), Agree (4), Strongly Agree (5)

| Q4: I think the notes I typed down in that task to describe my hypotheses are helpful for my recall. |
| Q5: The actions and observations captured in the trace (e.g., searching by a keyword and a set of selected data entries) are helpful for my recall. |
| Q6: The ARSCA captures the key evidence and insights generated in my analytical process. |

interview the professional analysts for the retrospection right after the experiment due to our limited access to them (which is a common issue for outside researchers). Given this concern, we conducted a trace-stimulated recall after the experiment with the focus on two questions: (1) whether the traces can help the participants recall the details of their original cognitive processes in the task? and (2) how each type of the CTA data in traces (i.e., automatically captured or self-report data) help participant recall? The procedure of the trace-stimulated recall interview is shown in Table 5.3.

We selected four doctoral students in this study with subject ID “G1”, “G2”, “G3”, and “G4” respectively. These four participants were selected for three reasons: (1) they all successfully identified the major attack events in the task; (2) none of them had access to any information of their traces after the experiment; and (3) their cognitive processes have similar scale (measured by the task duration and
Table 4.6. The results of the trace-stimulated recall of four participants

<table>
<thead>
<tr>
<th>Time distance since the last experiment (month)</th>
<th>G1</th>
<th>G2</th>
<th>G3</th>
<th>G4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scale of cognitive process</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task Duration (min)</td>
<td>56</td>
<td>1</td>
<td>55</td>
<td>36</td>
</tr>
<tr>
<td># of operations</td>
<td>54</td>
<td>38</td>
<td>47</td>
<td>49</td>
</tr>
<tr>
<td># of words in hypothesis descriptions</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean</td>
<td>10</td>
<td>29</td>
<td>31</td>
<td>10</td>
</tr>
<tr>
<td>SD</td>
<td>5</td>
<td>11</td>
<td>12</td>
<td>3</td>
</tr>
<tr>
<td>% of operations</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fully-explained (%)</td>
<td>96.3</td>
<td>92.9</td>
<td>91.9</td>
<td>91.8</td>
</tr>
<tr>
<td>Partially-explained (%)</td>
<td>3.7</td>
<td>7.1</td>
<td>8.1</td>
<td>8.2</td>
</tr>
<tr>
<td>Feedbacks using a 5-point Likert Scale</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-reported hypotheses help recall (Q4)</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Automatically captured actions and observations help recall (Q5)</td>
<td>5</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>ARSCA captures the key evidence and insights (Q6)</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

number of operations), which is considered because the scale of cognitive processes may have an effect on the recall performance. However, the spans between the time of recall study and their last experiment are different so that we need to estimate the effect of time span on recall.

Regarding Step 1 in Table 5.3, the answers of the four participants to Q1 and Q2 fall into four categories: (1) the key attack events (G1 recalled 4 events, G2 and G3 recalled 2 events, G4 recalled 1 event); (2) the strategies used to explore
the data sources (only G1 could recall the general strategy used by him: “I mainly investigated the ports of interest”).); and (3) the basic background information of the task scenario (recalled by G1). The time distance since the last experiment of G1 was the shortest among them and G1 performed the best on the recall. However, all of them failed to recall most of the observations and hypotheses they generated and the specific strategies used for data exploration or hypothesis investigation during the task.

In Step 2, we went through the trace with each participant operation by operation (i.e., the unit of analysis is operation). We examined the meaning of each operation to the participant to make sure he/she can understand it, however we did not present any explanation or our understanding to avoid the effect on the recall performance of the participant. Then, we asked the participant to analyze each operation by elaborating the purpose and the corresponding actions, observations or hypotheses, as we did in the previous trace analysis. The operations can be classified into three categories according to the degree of confidence of the participants in their elaboration of the operations: (1) the ones that can be fully explained; (2) the ones that can be partially explained; and (3) the ones that cannot be explained. The percentage of operations in each category of the participants is summarized in the fifth row of Table 4.5.5. It shows that most operations in their traces can be fully explained, some can be partially explained, and none can’t be explained. The “fully-explained” operations are usually those on the observations or hypotheses which are viewed important by the participants. Regarding the “partially-explained” operations, all the four participants mentioned that they were not sure but could infer the purpose of conducting those operations. G2 said “It feels like I’m doing a second-round analysis. I’m inferring my intention based on the recorded behaviors in the trace this time.” G4 said “I could infer multiple motives behind this operation.” For example, G3 was not sure about why he did not immediately investigate a hypothesis after creating it, he explained “There could
be an intermediate thinking. I might want to reflect on my previous hypotheses first to think carefully about what to do next, so here I stopped.” Their feedback gives us some insights into what information in the cognitive processes can/cannot be captured by ARSCA, and therefore helps us refine ARSCA and our trace analysis method.

In Step 3, we evaluated how the traces help recall. Q4 and Q5 are interested in how the self-reported and the automatically captured cognitive data in the traces help them on recall. Q6 asks whether they think ARSCA captures the key evidence and insights in their analysis process. The four participants’ feedbacks are shown in Table 4.5.5, using a 5-point Likert Scale. Although the samples are quite limited, the positive feedback shows that it is very promising to use the traces captured by ARSCA for self-reflection.

4.6 Discussion

Our tracing method focuses on the intermediate level of the unit of analysis, compared with the macro-level CTA studies and the micro-level behavioral tracking or neural activity study. The macro-level CTA usually focuses on the human task performance, cognitive load (e.g., short-term memory), thoughts, and cognitive bias, e.g., the study of the situational awareness analysts’ jobs and cognitive needs [18, 17, 24], and the study of the Cyber SA measurements [56, 31]. These studies usually involve interviews, observations, verbal protocols and questionnaires, in which the cognitive assessment procedure has an impact on the analysts’ behaviors being assessed as “reactive recording” methods in [70]. Reactive recording usually involve higher time investment and may bring about interference to participants’ task performance. Besides, verbalizing the key aspects of the thought process is also difficult for analysts because some thoughts are “so automated as a result of their prior knowledge and/or training” [87]). On the contrary, the
micro-level analysis usually collect data by nonreactive recording, which may involve automatic keystroke logging [53, 50], eye tracking [53, 2] or even brain-level EEG/fMRI recording [74]. One major limitation of the micro-level study lies in that many important behaviors may be failed to be captured or recovered in the subsequent reflective analysis due to the lack of the analysts’ situated interpretation [70].

On account of the real-world challenges of conducting CTAs on Cyber SA, the tracing method we proposed focuses on the intermediate level of analysis (i.e., analysts’ operations) and integrates both reactive and nonreactive recording techniques. The advantages of the integration lies in that it enables us to capture the key actions, observations, and hypotheses in analysts’ cognitive processes at low time cost: ARSCA supports real-time CTA data collection while analysts are conducting data triage tasks, but it does not require analysts to explain their operations. Besides, the tracing method capture analysts’ hypotheses about incidents together with the context information, which enhances the audit trail for response agents in a SOC to make decisions based on the analysts’ conclusions.

However, there is always a trade-off between CTA data collection (i.e., capturing traces) and CTA data analysis (i.e., analyzing traces). Considering the tight time schedule of cybersecurity analysts, it is desirable to minimize analysts’ workloads in our CTA data collection. However, it adds the workload of trace analysis for tracking back the analysts’ cognitive processes. Fortunately, based on our preliminary trace analysis, we have identified some common patterns of sequential operations in traces. Therefore, it is highly likely to generate some guidelines or a procedure, even with an automated tool for trace analysis.

Furthermore, we only confirmed our understanding of four participants’ traces in the stimulated recall interview, which is not adequate enough to draw the absolute conclusion that the understanding gained from traces is identical to the original cognitive processes. We definitely need to conduct more retrospective study.
However, some tentative conclusion can be drawn with key findings and thoughts captured in the traces of the participants that some domain knowledge and data triage strategies can be learned by tracking back the cognitive processes from the traces. In the future, we can identify and further categorize these strategies and compare them across the participants. It will not only contribute to elicitation of procedure knowledge but also help us understand the difference between experts and novices and develop tools to assist or train analysts.

4.7 Conclusion

We proposed a minimum-reactive method for tracing cybersecurity analysts’ fine-grained cognitive processes of data triage. It is aided by a computer toolkit which integrates two main CTA data collection techniques (i.e., automatic capture and self-report) for collecting complementary CTA data about analysts’ cognitive processes. We have shown that the this is a feasible method for gaining deep understanding of how analysts perform a data triage task, and the tracing method may have such promising contribution to analysts’ as self-reflection, elicitation of their procedure knowledge or new tool development.
Empirical Study of Fine-Grained Cognitive Traces of Cyber SA Data Triage

5.1 Introduction

A fine-grained understanding of analysts’ cognitive processes of data triage helps researchers learn analysts’ domain knowledge and data triage strategies and develop technologies to facilitate Cyber SA data triage. Therefore, it is highly desirable to capture and analyze the analysts’ fine-grained cognitive processes of data triage. Cognitive Task Analysis (CTA) is commonly used method for investigating human cognitive process in a task. However, CTA studies of Cyber SA analytics can be limited due to their high time cost or the organizations’ concern about the disclosure of their confidential information. Although several existing CTA studies of Cyber SA focus on the macro-level cybersecurity analytics, the fine-grained analyst’s cognitive process of data triage is not well-studied.

To study the analysts’ fine-grained cognitive processes, we have proposed a trace representation of the fine-grained cognitive processes of data triage. The
term “**fine-grained cognitive process**” refers to an analytical reasoning process involving (1) data triage actions performed by the analyst; (2) the analysts’ observations of suspicious network traffics and events; (3) the analysts’ hypotheses about suspected the attack events that are generated based on the existing observations (described in Section 3.2.4.1 in Chapter 3). We further developed a computer-aided tracing method which provides a feasible way to conduct fine-grained CTA study on data triage.

Based on the trace representation and the data triage tracing method, we are able to conduct an empirical study of analysts’ cognitive processes of data triage. There are several real-world challenges to be considered when designing the empirical study. First of all, the cybersecurity analysts can hardly find time to participate in interviews considering that they have to work in a 24/7 shift with

---

**Figure 5.1.** An empirical study of the analysts’ cognitive process of Cyber SA data triage. We collected traces of analysts’ cognitive processes in an experiment, using the computer-aided tracing method described in Chapter 4, and conducted case studies of how analysts’ perform data triage.
a very tight time schedule. Secondly, analysts’ cognitive processes of data triage are bounded together with the network monitoring data being processed and the network being monitored, which are confidential information of an organization. Therefore, organizations may limit outside researchers’ access to such information.

To tackle these concerns, we design a laboratory experiment with a simulated Cyber SA data analysis task. Neither the task data sources nor the network topology reveal any information of the real-world setting of an organization’s network, thus eliminating the organizations’ confidentiality concern. In addition, this experiment is designed to be time-efficient for analysts by leveraging ARSCA in data collection. It means analysts do not need to make extra efforts on reporting or summarizing their analysis behaviors (e.g., interviews, verbal protocols), except for writing down some thoughts for the benefit of their own analysis process in the task (which is not mandatory in the experiment).

Analysts’ cognitive processes were captured by traces in the experiment. A case study is conducted to analyze the collected traces to gain a better understanding of the analysts’ cognitive processes. The case study focuses on interpreting the data triage operations in the context of analysts’ analytical reasoning processes and recognizing some common patterns in the operation sequences. Figure 5.1 shows that our empirical study includes (1) an experiment conducted for collecting traces of analysts’ cognitive process and (2) a case study focused on the qualitative analysis of the traces collected from the experiment.
5.2 Experiment: Tracing Analysts’ Cognitive Process of Cyber SA Data Triage

5.2.1 Experiment Design

An experiment is designed to trace analysts’ cognitive processes while they are performing a data triage task, using the tracing method described in Chapter 4. Considering the real-world challenges of conducting CTA studies on Cyber SA, we list several guidelines for the experiment design.

- It is necessary to learn participants’ background, expertise, working experience, physical and mental status at the time of the experiment. These are factors that can influence an analyst’s task performance.

- The analysts should be trained to get familiar with the experiment environment before performing the task so that we can assume their task performance is close to their performance in the real-world job setting.

- It is possible that the experiment conductor can’t get the permission to interview the participants in a face-to-face manner if the analyst identifies are confidential in an organization.

- The cybersecurity analysts (event experts) may not be good at or can not afford enough time expressing the critical thoughts in their minds.

- The average time needed for accomplishing the task should be limited in a time span which is acceptable for cybersecurity analysts.

- Participants’ feedback is important after the task. We should ask for their conclusions about the possible incidents after they finish the task so that we can evaluate their task performance afterwords. Besides, the participants’ comments about their cognitive processes also provide a valuable reference for the following trace analysis.
• The experiment data need to be stored in an explicit format so that it is easy for organizations to review them before passing them to the researchers.

Given these guidelines, the experiment is designed with four main stages: (1) pre-task questionnaire (5 minutes), which gathers the information of participants’ domain knowledge, expertise and their physical and mental status, (2) tutorial session (20 minutes), in which participants are trained to get familiar with the experimental environment to perform data triage, (3) a data triage task (at most 60 minutes), in which the analyst works on a Cyber SA data triage task together with ARSCA, and (4) a post-task questionnaire (15 minutes), which contains both open-ended and close-ended questions asking about the findings and conclusion about the possible attack chains.

5.2.1.1 Pre-task Questionnaire

A pre-task questionnaire taking 5 minutes is the first stage of the experiment. The first part of the pre-task questionnaire includes the demographic questions about age, gender, ethnicity, and native language. The second part of questions asks about the domain knowledge and expertise, including the work title, working years, five rating questions using five-point Likert scale for assessing domain knowledge in cybersecurity, two questions asking about familiarity with security techniques and security certificates, a question about the familiarity with VAST challenges 2012 data (which is used in our simulated task), and two more rating questions assessing the analyst’s current mental and physical status.

5.2.1.2 Tutorial Session

A tutorial session is added in the experiment procedure to avoid the participants’ task performance being influenced by their familiarity with the experiment environment. Participants are shown five short videos (the average length is 5 minutes) which introduce the details of the experiment environment and task description.
Introducing the ARSCA toolkit is the most important part in training sessions. Participants are asked to interact with ARSCA to complete a mock-up task after watching the videos. At last, participants have to pass a quiz which includes a list of practical questions for checking whether the participants are familiar with the experiment environment.

5.2.1.3 Data Triage Task

Before starting the task, we need to make sure that they understand the task correctly. Therefore, at the beginning of the task, we explain the network configuration and the responsibility of the participant in the task. To avoid influencing participants’ performance, participants are instructed that only data triage is their responsibility in the task but not interacting with ARSCA (e.g., managing hypotheses in the Analysis View) unless it is out of their own will. In addition, we explain the meaning of every attributes of the data sources used in the task. Once a task is started, each participant interacts merely with the ARSCA toolkit without being disturbed by others.

5.2.1.4 Post-task Questionnaire

After finishing the task, each participant needs to take a post-task questionnaire which takes about 15 minutes. It includes both open-ended questions and closed-ended rating questions. The open-ended questions ask a participant to explain the key observations and hypotheses of his/her cognitive processes and to report his/her conclusion of the incidents. These questions are “IMP_OBS”, “FD_OBS”, “IMP_HPY”, and “EVTS”, which have been described in Table 4.4 in Section 4.5.4. Regarding the close-ended rating questions, some ask for a participant’s opinions on the experiment setup, and some focus on the participants’ performance in the task. We discuss the questions in details in Section 5.2.4.
5.2.2 Recruitment

With this experimental design, we successfully obtained the IRB approval and recruited 30 professional analysts from Army Research Lab and collected the traces of their data triage process. These analysts have various domain expertise at different levels. In addition to the professional analysts, we also recruited 20 doctoral students specialized in cybersecurity. Although they were not professionals, they were invited because they have sufficient domain knowledge and experience in cybersecurity analysis.

5.2.3 Experiment Task Design

The data triage task in our experiment should be carefully designed so that the participants can perform as well as they do in their daily jobs. More specifically, the task should be of reasonable complexity (i.e., neither too difficult nor too easy for the participants). The data sources used in the task should match the scalability of the real-world data sources with respect to the volume and complexity of noise. On the other hand, we have to control the size of the data set to ensure the participants can finish the task within a specified experiment time.

We designed our task by partially adopting the dataset of VAST Challenge 2012 Mini Challenge 2 [94], which is of high quality and contains 23,711,341 firewall logs and 35,948 IDS alerts in total. The task scenario underlying the dataset is provided as a ground truth, which is a multi-step attack taking place within 40 hours on the network of an organization containing approximately 5000 hosts.

However, the original dataset was too huge to be processed by the participants given the limited experiment time. Therefore, we tailored the original dataset by selecting some small portions out of it. Two subset of data were extracted with the reference of the ground truth so that each subset implied critical incidents. Table 5.1 shows that the two subsets correspond to two sequences of network events happening in two 10-minute time windows (i.e., Time Window I and Time Window
Table 5.1. Task data sources tailored from VAST Challenge 2012 data

<table>
<thead>
<tr>
<th>Time Window</th>
<th>Attack Scenario</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scenario from the beginning (started from 4/5/2012 20:25):</td>
<td>An internal workstation got infected with a botnet due to a USB insertion.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>The botnet spread fast.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(2) The botnet located sensitive data.</td>
<td># Firewall Log: 115,524</td>
</tr>
<tr>
<td></td>
<td>(3) Failed attempted to exfiltrate data using FTP.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(4) Successful attempted to exfiltrate data using SSH.</td>
<td></td>
</tr>
<tr>
<td>II: 4/6/2012 18:05-18:15 (10 min)</td>
<td>(1) The botnet spread (inquiring the network hardware)</td>
<td># IDS Alerts: 228</td>
</tr>
<tr>
<td></td>
<td>(2) New IRC communications.</td>
<td># Firewall Log: 48,012</td>
</tr>
<tr>
<td></td>
<td>(3) New attempt to exfiltrate data using FTP.</td>
<td></td>
</tr>
</tbody>
</table>

II) in the 40-hour attack scenario of the VAST challenge. We use either one of these datasets in our data triage task. No matter which dataset used in the task, it would not influence the task’s complexity because the datasets contain similar number of network events (reported in IDS alerts and firewall log), and the attack scenario corresponding to these datasets were at the same level of complexity.

5.2.4 Participants’ Feedbacks on the Experiment

Table 5.2 lists four rating questions (using a five-point likert scale) designed in the post-task questionnaire. “TASK_CMP” asks whether the task is of reasonable complexity. “SET_CFT” asks whether the participant felt comfortable with the
Table 5.2. The four rating questions about experiment setting and task performance.

<table>
<thead>
<tr>
<th>Code</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>TASK_CMP</td>
<td>“The task is of reasonable complexity regarding the analysis activities it involves (e.g., data exploration, thinking reasoning, making decisions).”</td>
</tr>
<tr>
<td>SET_CFT</td>
<td>“I felt comfortable with the setup of the experiment (e.g., the provided software tool and physical environment), and my performance is not hindered by the experiment setup.”</td>
</tr>
<tr>
<td>EXP_RFL</td>
<td>“My expertise of cyber analysis is fully leveraged and is reflected in accomplishing the task.”</td>
</tr>
<tr>
<td>CONC</td>
<td>“I’m fully concentrated on accomplishing the task.”</td>
</tr>
</tbody>
</table>

experiment setup. “EXP_RFL” asks how much expertise is leveraged in the task. “CONC” asks how a participant was concentrated in accomplishing the task.

Table 5.2 summarizes the percentage of participants’ responses to each level of these rating questions, with the four questions as columns and the five levels of Likert Scale as rows. Considering that we have 30 professional analysts and 20 students, we calculate the percentage of the participants among each group who responded with a particular rating level for each question. The numbers in brackets refer to the response of professional analysts and doctoral students respectively. We find that most participants in both groups rated “Agree (4)” to all the questions. No other participant rates “Strongly Disagree” except for one participant rating “Strongly Disagree” for question “SET_CFT”. We find the participant encountered a computer crash during the task. Comparing the responses of professional analysts and doctoral students, we find there are more diversity in the professional analysts. Generally, professional responses responded less “Agree” than the
Table 5.3. Participants’ responses to the questions in Table 5.2

<table>
<thead>
<tr>
<th>Likert Response</th>
<th>Likert Scale</th>
<th>Post-task Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TASK_CMP</td>
<td>SET_CFT</td>
</tr>
<tr>
<td>Disagree</td>
<td>1</td>
<td>(0%, 0%)*</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>(10%, 0%)</td>
</tr>
<tr>
<td>Neutral</td>
<td>3</td>
<td>(23.3%, 25%)</td>
</tr>
<tr>
<td>Agree</td>
<td>4</td>
<td>(46.7%, 35%)</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>(20%, 40%)</td>
</tr>
</tbody>
</table>

*Numbers in brackets correspond to professional analysts and students respectively.

doctoral students did, regarding the question “TASK_CMP” and “SET_CFT”. A reasonable explanation is that the professional analysts may be accustomed to the facilities (e.g., command line) in their workplaces so that it is harder for them to switch to the experiment environment. Besides, the professional analysts may have higher criteria for expertise utilization and concentration compared with doctoral students. We need to continue the experiment to enlarge the sample size to draw more solid conclusions.

5.3 Case Study

We collected thirty professional analysts’ traces and twenty doctoral students’ traces of data triage cognitive processes in the experiment. One trace of a professional analyst was excluded because the analyst did not finish the task due to a technical issue. A case study was conducted to analyze the collected traces of analysts’ data triage operations to recover the participants’ cognitive processes
captured in the traces. In this case study, we mainly paid special attention to the traces of professional analysts in order to learn the domain knowledge and data triage strategies of experts.

### 5.3.1 Qualitative Trace Analysis Method

![Diagram]

**Figure 5.2.** An analyst’s operation sequence is interpreted by the underlying AOH objects and their logic relationships

Given an analyst’s trace, our trace analysis starts with the sequence of operations (i.e., actions, observations, and hypotheses) that are explicitly recorded in a trace. These operations may imply the domain knowledge of an analyst. For
example, a participant wrote in a hypothesis: “Websites are communicating with financial servers. They are communicating over what appears to be IRC which is commonly used by malware.” We can know that the participant generated the hypothesis because of the domain knowledge that “IRC can be used by malware”. However, the explicit AOH objects or operations in the trace can hardly explain the analyst’s analytical reasoning process completely. Figure 5.2 shows an example of the partial trace of participant “P1”. The trace includes the sequence of P1’s operations and AOH objects. It shows that P1 first browsed the IDS alerts at time $t_1$, and then selected a set of network events of IRC connections as a new observation (O1). Based on O1, he generated a new hypothesis (H1) about policy violation. It is reasonable to infer that network policy is one basis for P1 to generate the hypothesis H1. After that, P1 filtered the network events in IDS alerts based on the condition “Port = 6667”. We notice that O1 has included some connection using port “6667”. To explain why P1 conducted a further FILTER operation at $t_4$, we infer that P1 has the domain knowledge that port “6667” can be used for malicious purpose (i.e., IRC communication in a botnet) so that P1 has an implicit hypothesis that the network events using port “6667” are worthy of further investigation. The above example shows that we can recover an analyst’s cognitive process from a trace based on an operation-by-operation manner of analysis. Given an operation, we analyze it from two aspects: (1) the actions, observations or hypothesis based on which the operation is performed, and (2) the reason why the operation is conducted after the previous operation. A directed graph can be constructed, named Trace Analysis Graph, by analyzing each operation in a trace from the above two aspects. The nodes are actions, observations, or hypotheses in an analyst’s analytical reasoning process. A directed edge between two nodes indicates that the source node helps explain the target node. Figure 5.2 is the trace analysis graph for explaining P1’s partial operation sequence. It explains how P1’s operations are related to P1’s AOH objects and under what context each
operation was carried out.

5.3.2 Patterns in Cognitive Traces

We analyzed the traces collected by ARSCA with the help of trace analysis graphs. Several common patterns are observed in the traces and correspond to identical sub graphs in the trace analysis graphs. These patterns indicate several typical cognitive activities, including identifying observation, creating hypotheses and hypothesis investigation. These activities have been identified in the existing study of sensemaking process and the CTA studies on cybersecurity analytics workflows. We will discuss these patterns with real cases to evaluate how the collected traces capture these cognitive activities in a fine-grained way. Besides, identifying these patterns indicates that it is possible to analyze the traces in a more automatic way.

5.3.2.1 Identifying Observation

\[
\begin{align*}
\text{t1. FILTER} & \quad \text{(Firewall logs, Port=21)} \\
& & \text{A1: Filter the firewall logs based on Port=21} \\
\text{t2. SELECT} & \quad \text{(Firewall logs via port=21)} \\
& & \text{O1: Connections via Port 21} \\
\text{t1. FILTER} & \quad \text{(Firewall logs, Port!=6667)} \\
& & \text{A1: Filter the firewall logs based on Port != 6667} \\
\text{t2. SELECT} & \quad \text{(Firewall logs via port=21)} \\
& & \text{O1: Connections via Port 21}
\end{align*}
\]

Figure 5.3. The case of identifying an observation

This pattern, FILTER/SEARCH \( \prec \) SELECT, is a SELECT operation following a FILTER or SEARCH operation, indicating the activity of identifying observation. Figure 5.3 shows two cases of this pattern. In Figure 5.3 (a), the participant first located the evidence (i.e., the denied FTP connection attempts) by conducting FILTER based on “port = 21”, and then directly selected the filtered data as a noteworthy observation (SELECT operation at t2). The other case is
shown in Figure 5.3 (b). The participant first weeded out the network events using port 6667 from the firewall logs (FILTER at t1). It resulted a subset of network events which include some FTP connections via port 21, so the participant further selected these network events as an observation (SELECT at t2). Unlike the first case, the participant selected a subset of the filtered data. Both cases demonstrate that an observation was identified after locating some suspicious network events. It was the most frequent pattern of analysts’ operations of data triage. Given the sequences of the operations, we can know how a participant filter the data sources to locate the suspicious network event.

5.3.2.2 Creating Hypotheses

This pattern, SELECT ≺ (LINK) ≺ NEW_HYPO, describes an analyst creates a new hypothesis about an incident after gaining an observation. The analyst may also identify the common features among the network events selected in the observation (LINK). Figure 5.4 shows a case of this pattern. The participant first identified a new observation in firewall logs (i.e., the FTP connections using port 21), and he claimed that the selected network events are all from inside IP addresses to outside IP addresses (LINK). Based on the observation, the participant created a new hypothesis about suspicious FTP connections that may be used for data exfiltration. Similar to the above example, we select the following hypotheses from the traces which were typed down by the participants.
for observation interpretation.

“Workstations participating in IRC chat port 6667. This type of activity is not authorized on the network. The 10.32.5.* IP range is an external server used for IRC chat.”

“User at 172.23.1.110 is violating policy by using IRC chat and connecting to external FTP server ...”

“IDS SMB Null session events are false positives: with this rule enabled on an IDS it normally requires quite a bit of tuning prior to getting useful results because it is prone to false positives.”

“Via Firewall logs we can see that there was an FTP connection attempt from an internal workstation IP to the external Internet. This type of communication is a policy violation.”

“In this situation this could indicate that this could also be an indicator of compromise because malware can leverage IRC for Command to Control (C2) communication.”

This case demonstrates that a previous observation can provide the context information for the creation of a hypothesis. Existing CTA studies have shown that analysts usually integrate their observations, experience, and practical knowledge to assess and interpret the observations and to generate hypotheses about potential incidents. Therefore, linking hypothesis to its relevant observations helps us understand why an analyst generates a hypothesis given an existing observation, and learn the domain knowledge underlying the hypothesis.

5.3.2.3 Hypothesis Investigation

This pattern, NEW_HYPO \prec FILTER/SEARCH/SELECT, consists of a NEW_HYPO operation and an following data triage operation (i.e., FILTER,
The participant first generated a hypothesis, which is the same one as in Figure 5.4 (H1). To further investigate the hypothesis H1, the participant filtered the firewall logs to check whether there were FTP connections not denied by firewall (FILTER at t2).

This pattern illustrates that analysts may search for support after generating a hypothesis. In this way, the analyst can strengthen or weaken this hypothesis in the course of following actions. Besides, if the current hypothesis is about one step in an attack chain, an analyst may start from this hypothesis to further investigate other possible steps (either “previous” or “next” steps) in this hypothesized chain. Some participants mentioned their next-step plans when creating hypotheses during the task. Several selected hypotheses are listed as follows.

“Based on this entry in the Firewall logs it seems an SSH connection attempt was taken down. This communication is from an Internal IP to an external IP over TCP/22 the default port for SSH. This type of communication is worthy of further investigation because it is a policy violation.”

“Financial Servers are prime targets for cyber crime (so I’d better filter based on the financial servers)”

“The IDS generated a lot of events related to SMB NTML SSP unicode asn1 overflow attempts. This could be an attempt at some form of mali-
cious code or attacker to propagate through a network over SMB. Most notably I would start to suspect a worm that propagates over SMB.”

5.3.3 Cases of Data Triage Strategy

5.3.4 Case 1: “Gradually Narrowing Down”

![Diagram showing the cognitive processes of data triage strategy]

Figure 5.6. A case of a participant detecting suspicious network events by “gradually narrowing down”

Figure 5.6 shows the cognitive processes recovered from a participant’s trace. The participant first performed a FILTER operation on the network events in IDS alerts at $t_1$ and observed a set of network connections from external IPs to internal IPs (O1). He thought this set of network events were highly suspicious (H1) so that he performed another FILTER operation at $t_4$ to further narrow down his search space by adding another condition “DstPort = 6667” to the filtering
condition used in \textit{FILTER} at $t_1$. In this way, the participant detected a set of malicious network events that indicates malicious IRC communication from several Command & Control (C&C) servers to a group of internal workstations. This case indicates the participant took the strategy of gradually filtering narrowing down the search space to locate key evidence, which is the most common strategies used by the professional analysts in the experiment.

\subsection{5.3.5 Case 2: “Following a Cue”}

Figure 5.7. A case of a participant locating suspicious network events by “following a cue”

Figure 5.7 is a follow-up of P1’s cognitive process shown in Figure 5.2, which indicates a “following a cue” data triage strategy. According to our discussion of the case in Figure 5.2, he generated the hypothesis (H2) about malicious IRC communication in a botnet. Starting from H2, Figure 5.7 shows that P1 filtered the network events recorded in firewall logs based on a same condition “SrcPort = 6667” to search for more details that support H2. It resulted in a set of network connections between a same set of internal IPs using source port 6667, and therefore strengthened P1’s hypothesis about malicious IRC communication. In this case, “following up a cue” refers to the strategy used by analysts in data triage to search
for network events in different data sources that indicate a same step in an attack chain based on a clue (e.g., same set of IP addresses involved in the network events).

5.3.6 Case 3: “Proceeding From One Event to its Related Events”

![Diagram of Case 3]

**Figure 5.8.** A case of a participant “proceeding from one event to its related events”

Figure 5.8 demonstrates a participant’s cognitive process of obtaining three key observations based on the knowledge of the events’ relationships in attack chains. At the beginning of the partial operation sequence, the participant first confirmed his hypothesis about IRC communications in a botnet (H1) after gaining the observation O1. After confirming the hypothesis about malicious IRC communication, he thought creatively about what could be the next step of the attacker. One common following step was exfiltrating data from internal hosts, and FTP is a common service used for file transfer (using port 20 or 21). Therefore, he continued data triage by filtering the network events using FTP (A2), which resulted
in the observation O2. O2 let him know that there were indeed malicious FTP attempts but were failed. He expected those bots would choose another way, and decided to search for the connections using SSH (using port 22). He filtered the firewall logs based on “port = 22” (FILTER at $t_6$), and found that three SSH connections were successfully built between internal IP addresses and external IP addresses. So he generated a hypothesis that the bots exfiltrate data to outside C&C servers using SSH.

According to our understanding of the participant’s cognitive process, we can infer that he is familiar with the common attack chains and has the domain knowledge of the services used for Command & Control communication and data exfiltration. Thinking about the related events in an attack chain enabled him performed data triage in a very efficient way, which reflects his expertise gained from long-term on-the-job training.

### 5.3.7 Trace-stimulated Recall

To check whether our understanding of the participants’ cognitive processes based on their traces is close to their original ones, we conducted a follow-up trace-based stimulated recall interview by inviting participants to retrospect on their previous cognitive processes with the aid of their traces. The procedure of trace-stimulated recall was as follows: we asked a participant to go through his/her trace by explaining each operation as what we did in Section 4.5.5 in Chapter 4. After obtaining his/her self-explanation, we confirmed our original understanding of his/her cognitive process of data triage.

Our trace analysis focuses on professional analysts because their cognitive processes in the experiment are closer to the real-world processes compared with the doctoral students. However, the trace-stimulated recall requires in-depth interview with our participants, which is difficult for professional analysts. Therefore, we chose one representative from the professional analysts, coded as “P1”. P1 was
invited because he successfully identified the incidents in the task and his cognitive process was representative and of high quality.

5.3.7.1 Trace-based Explanation and Self-Explanation

The result of the trace-stimulated recall with P1 shows that his self-explanation is consistent with our explanation. The following is an example of his comments on his cognitive process of filtering IDS alerts (which is illustrated at $t_4$ in Figure 5.2):

“I saw strings within the alerts that meant IRC communication traffic that was based on my prior experience this is a default port for IRC. I usually start with large chunk of port communications and it happened to be port 6667. I also wanted to see if there was one or more hosts communicating.”

It indicates that P1’s FILTER operation was based on his domain knowledge and his observation of a large amount of network events using port 6667.

Moreover, it is found that the cognitive process of the participants can be inferred from the successive operations in the trace. For instance, one part of P1’s trace indicates that he generated a new hypothesis about DNS attack by self-reporting: “Some malware attacks on DNS from inner network, need further exploration”. Right after making the hypothesis, he switched his focus of attention to another hypothesis, which was generated previously with the note: “Suspicious IRC connections from outside”. The following part of the operation sequences showed that P1 never went back to the previous hypothesis about DNS attack. In the recall study, P1 gave his explanation about why he did not further investigate this hypothesis: this hypothesis was not generated on his serious thinking and therefore he quickly gave up this hypothesis after a second thought but did not inform the tracing toolkit (ARSCA). This finding indicates that traces may not record all the thoughts emerged in the analysts’ mind. Although it is impossible
to capture every hypothesis occurred in the participant’s cognitive process, traces at least capture some clues and enlightenment for interpretation which are also useful for further analysis.

5.4 Discussion

Our preliminary trace analysis shows that it is feasible to capture analysts’ data triage cognitive processes at the real-time and to infer them based on captured traces offline. It is not surprising that the captured traces is vulnerable to capture the subtle thoughts in analysts’ minds (e.g., quickly-denied thoughts or implicit hypotheses). However, we can identify useful clues in traces and infer about the implicit hypothesis or domain knowledge through explaining the analyst’s sequence of operations in traces.

We found that the participants’ data triage cognitive processes recovered from the captured traces may imply the data triage strategies used for efficiently detecting suspicious network events, as well as the participants’ domain knowledge. It indicates traces not only can help analysts with self-reflection to learn from their experience, but also enable researchers to elicit analysts’ domain knowledge and data triage strategies to develop better cybersecurity analytics techniques and training programs.

The current work still has several limitations. Given the current experiment results, we have observed a lot of differences among individual participants, even they belong to the same group (i.e., professional analysts or doctoral students). Therefore, it is hard for us to compare the differences of data triage cognitive processes between professional analysts (senior analysts) and students (junior analysts) without enlarging the size of the trace collection. In addition, manual efforts are still required in this work on analyzing the semi-formal traces captured by ARSCA to understand analysts’ cognitive processes. Considering that common
patterns of operations have been observed in these traces, it is possible to develop an automated method for trace analysis.

5.5 Conclusion

In conclusion, an empirical study was conducted to capture and study analysts’ cognitive process of Cyber SA data triage. An experiment was designed to capture analysts’ data triage cognitive processes, in which we asked our participants (including 30 professional analysts and 20 doctoral students) to accomplish a simulated data triage task and used the tracing method described in Chapter 4 to capture their cognitive processes. Given the traces collected from the experiment, we reconstructed the participants’ cognitive processes by analyzing the traces in an operation-by-operation manner, which helped us gain a deep understanding of the participants’ fine-grained analytical reasoning processes. Some common patterns in the participants’ operation sequences have been observed in our trace analysis. However, we also noticed that a great individual difference lied in the participant’s data triage strategies and hypotheses about potential incidents. In general, the collected traces contain rich information of the participants’ domain knowledge and data triage strategies, which can contribute to analysts’ self-reflection, knowledge elicitation and engineering and new cybersecurity analytics tool development.
Chapter 6

Automated Cyber SA Data Triage System by Leveraging Analysts’ Cognitive Traces

6.1 Introduction

The data, coming from a variety of data sources, are being continuously generated and the data volume is overwhelming. Therefore, data triage is labor-intensive and mostly manually operated by analysts [17]. Compared to a computer, human brains have orders of magnitudes smaller data processing throughput. In addition, human beings face unique challenges such as fatigue, anxiety and depression, which nothing of that sort a computer would face. However, neither the network nor the attack campaign is waiting for the human brains. Besides, the alert portion of the data contains a large number of false positives. The analysts have to leverage their domain expertise and experience to make fast judgments on which parts of the data sources are worthy of further investigation. Under the great time pressure, the analysts have to resort to day-and-night rotation for 24*7 full coverage. According to a recent research report in January, 2015 [71], an organization can
receive nearly 17,000 alerts in a typical week, among which only 19% are reliable. However, only 4% of all the alerts can be investigated. The average annual cost of organizations is $1.27 million for wasting analysts’ time in investigating false alerts. FireEye reported that the average annual operational spending of an organization due to false positives is $17.816 million given the default resource capacity informed through common deployments [28].

This challenge discussed above has constituted a huge obstacle impeding timely and high-quality incident reports generated by analysts in a SOC. Analysts are bounded by the tedious data triage tasks so that they can hardly concentrate on zero-day attack or profound intrusion detection most in need. Extensive research has been done on alert correlation [76], and is an important milestone of automated data triage. Alert correlation techniques use heuristic rules (which is a simple form of automaton) to correlate alerts. Alert correlation has its limitation in that it can only analyze one type of data source (i.e., IDS alerts) while in reality security analysts have to do crossing data analysis from a variety of data sources in most cases. Alert correlation analysis has later on been integrated with other analysis [97], but the method is still along the heuristic rule angle. Motivated by the benefits of cross-data-source analysis, SIEM systems (e.g., ArcSight [3]) have been focusing on security event correlations across multiple data sources. Rules are usually employed to correlate the event log entries [64].

Although SIEM systems take a big leap forward in generating more powerful data triage automatons, SIEM systems is extremely expensive not only for its license cost but also for the large amount of time and expertise required in constantly system management and customization [60]. Analysts needs to develop and test the data triage automatons (e.g., customized filters and complicated correlation rules) that fit the organization’s setting [64, 60]. The following is an example of a complicated filtering rule used in a SIEM system [58]:

“Filters ->
EventType (In) [Failure],
AND, Context (In) [Context],
AND, Destination Port (In) [DestPort],
AND, Protocol (In) [Protocol],
AND, {Source IP (Not In) [ExcludeSrcIP1], OR, Source IP (Not In) [ExcludeSrcIP2], OR, Source IP (Not In) [ExcludeSrcIP3]},
...

This example shows that the complicated filtering rule of SIEM systems can be decomposed into a series of predicates that specify the characteristics of network events. Considering that a large number of complicated data triage automatons need to be generated, the amount of manual effort is still significant. Therefore, how to ease the analysts’ burden on data triage has been an urgent need of organizations, and it is highly desirable to generate data triage automatons so that analysts can focus on profound analysis to generate incident reports of higher quality.

This paper aims to take an initial step towards reducing cybersecurity analysts workloads of developing data triage automatons. Our previous work has proposed a feasible approach to tracing analysts’ cognitive processes of data triage in real time manner at a low cost. With their cognitive processes being traced, we propose to elicit analysts’ intelligence from their traces of performing data triage tasks and to further leverage the elicited intelligence to build automated data triage system in order to ease the burden on analysts in data triage.

This approach is enabled by there insights. Firstly, It is actually possible to do non-intrusive tracing of human analysts’ concrete data triage operations. Secondly, in spite of all these difficulties, traces of analysts’ data triage cognitive processes can be mined in a largely automated way to obtain the good “ingredients” for eliciting complicated rules. The above example of SIEM rules can be decomposed into several filtering components [58]. The “EventType (In) Failure” is a basic
component containing a data field and a constraint in its value, which is a so-called “ingredient”. The basic filtering components can be connected by logic connector “AND/OR”, and thus form a complicated filtering or correlation rule. Therefore, once the good ingredients are obtained from experts’ traces of former data triage operations, analysts can start from them to construct more complicated rules instead of from scratch.

Accordingly, our approach works as follows. First, we obtain the traces of analysts’ data triage operations collected by our existing cognitive tracing toolkit, namely ARSCA. Second, we represent analysts’ data triage operations captured in traces and their temporal and logical relationships in a Characteristic Constraint Graph (CC-Graph). Third, to mine the rule ingredients, we analyze the CC-Graphs to find the key data characteristic constraints. The key constraints are further correlated with the data sources to identify the “can-happen-before” relationships among them. The key constraints and their “can-happen-before” relationships represent various attack patterns, namely “Attack Path Pattern”. Each attack path pattern, which is formally represented, has a semantic meaning that defines a class of network events indicating multi-step attacks. Analysts can review, modify and extend them. Fourth, the formally represent attack path patterns can be directly used to initialize a finite state machine for conducting automated data triage, just as adding rules to a SIEM system. The data triage is essentially a data triage automaton.

The main contributions are as follows: (a) To the best of our knowledge, this is the first method that generates data triage automatons directly from human analysts’ operations in an automatic manner. (b) Comparing our method to the rule-based SIEM systems, it reduces analysts’ efforts on generating SIEM rules, and therefore can render the cost (of data triage automaton generation) orders of magnitudes smaller. (c) The proposed method has been designed and implemented in an automated data triage system, and this system is evaluated through a
human-in-the-loop case study. We have recruited 30 professional security analysts as human participants to work on data triage tasks designed based on the VAST challenge attack scenario[12] in our previous experiment. During the experiment, the analysts’ operation traces have been collected by a toolkit named ARSCA. These operation traces are used as inputs to construct the data triage state machine (DT-SM). We first evaluated the effectiveness of the DT-SM construction. Secondly, we fed over a much larger network dataset of the VAST network into the DT-SMs and evaluated the data triage results based on false positive and false negative rates. Furthermore, the study shows that DT-SMs built on the traces from the analysts of better task performance have better data triage results. Thirdly, we found the DT-SM built on a combination operation traces has better performance than the average performance of those built on individual traces.

6.2 The System Model

6.2.1 Characteristic Constraint of Data Triage Operation

Data triage is defined as a dynamic Cyber-Human System in Chapter 3 in which analysts’ responsibility is to make fast decisions about which incoming network events are worth further investigation. An example of data triage process is demonstrated in Figure 3.1 in Chapter 3, which describe how an analyst conducts a series of data triage operations to detect malicious network events and correlate the ones of a same attack chain. Based on his/her domain knowledge and expertise, an analyst specifies the conditions on the characteristics of network events that are viewed as suspicious or correlated in a hypothesized attack chain. It is based on these identified characteristic constraints that the analyst performs data triage operations to narrow down the search focus to detect the suspicious and related network events.

There are three types of data triage operations identified in Chapter 3 as follows.

• FILTER($D_{in}$, $D_{out}$, $C$): Filter a set of network events ($D_{in}$) based on a
condition \((C)\) on the attribute values of the network events and result in a subset \((D_{out})\).

- **SEARCH**\((D_{in}, D_{out}, C)\): Search a keyword \((C)\) in a set of network events \((D_{in})\). It results in a subset \((D_{out})\).

- **SELECT**\((D_{in}, D_{out}, C), D_j \subseteq D_i\): Select a subset of network events \((D_{out})\) in a set of network events \((D_{in})\). The subset of network events \((D_{out})\) has a characteristic \((C)\).

Each data triage operation filters out a subset of network events from an original set of network events specified in the data sources. The characteristics of the subset network events are determined by the constraint defined in the data triage operation. An atomic constraint is a predict that specifies the value range of an attribute of the network event,

\[ T_i = r_v(attr, val), \]

where \(r_v\) is the relationship between values, \(r_v = \{=, <>, >, <, <=, >=\}\). Recall the previous example of a SIEM rule in Section 6.1. The “EventType (In) Failure” is an atomic constraint represented by \(=(attr, \text{Failure})\).

A constraint on the subset characteristics can be multidimensional because a network event has multiple attributes. It is represented by a predicate in disjunctive normal form, namely “**Characteristic Constraint (CC)**”,

\[ C = \begin{cases} \bigvee(T_1 \land \ldots \land T_n), n > 1 \\ T \end{cases} \quad (6.1) \]

where \(T_i (1 \leq i \leq n)\) is an atomic predict.

Let \(C\) be a characteristic constraint specified in a data triage operation, and let \(D\) be a set of network events reported in the data sources. A data triage operation
is represented by a 2-tuple,

\[ O_1 = (t, f_C), \]

where \( t \) is the timestamp of the analyst performing this operation, \( f_C : P(D) \to P(D) \), such that \( f_C(D) = \{ e_k | e_k \in D, C \text{ holds on } e_k \} \), and \( P(D) \) is the power set of \( D \).

### 6.2.2 Relationship between Characteristic Constraints

Let \( O_1 = (t_1, f_{C_1}(D)) \) and \( O_2 = (t_2, f_{C_2}(D)) \) be two different data triage operations performed on a set of network events \( D \), we define three types of logical relationships between them based on their associated characteristic constraints: “is-equal-to”, “is-subsumed-by” and “is-complementary-with”.

\[
\text{isEql}(O_1, O_2) \iff C_1 \leftrightarrow C_2, \\
\text{isSub}(O_1, O_2) \iff C_1 \to C_2, \\
\text{isCom}(O_1, O_2) \iff C_1 \to \neg C_2 \text{ and } C_2 \to \neg C_1,
\]

where \( \to \) refers to implication, \( \text{isEql}(\cdot, \cdot) \) and \( \text{isCom}(\cdot, \cdot) \) are bidirectional relationships but \( \text{isSub}(\cdot, \cdot) \) is a unidirectional relationship. A special case of “is-complementary-with” is “is-absolute-complementary-with”: \( \text{isCom}(O_1, O_2) \) and \( O_1 \) and \( O_2 \) are absolute complement given a common set of network events, that is, \( C_1 = \neg C_2 \). For example, given a set of network events with a common characteristic “SrcPort <> 80”, the data triage operations using the characteristic constraints “SrcPort = 6667” and “SrcPort <> 80 AND SrcPort <> 6667” are absolute complement.
6.2.2.1 Characteristic Constraint Graph

Combining the logical relationships with the temporal relationships, an analyst’s triage process can be modeled as a directed graph, named “Characteristic Constraint Graph (CC-Graph),”

\[
G(\text{trace}) = \langle V, \{R_l\} \rangle,
\]

\[
V = \{O_1, \ldots, O_n\},
\]

\[
R_l \subseteq V \times V, l \in \{\text{isEq} \succ_t, \text{isSub} \succ_t, \text{isCom} \succ_t\}.
\]

The vertexes of the graph are the data triage operations, and a directed edge between two vertexes represents a conjunction of a constraint-related logical relationship and a “happen-after” relationship (i.e., \(\text{isEq} \succ_t, \text{isSub} \succ_t, \text{isCom} \succ_t\)).

The edges of CC-Graph are defined by both the temporal and logical relationships among data triage operations. The reason why the temporal relationships matter is that we are mainly interested in how a data triage operation is related to the ones that precedes it. Along with the temporal relationships, the logical relationships among data triage operations imply how an analyst switches his/her attention of focus from one subset of network events to another subset by performing these data triage operations. Therefore, the temporal and logical relationships together may imply an analyst’s data triage strategy.

6.3 The Automated Data Triage Approach

We propose an approach to building an automated data triage system based on the captured traces of analysts’ cognitive processes of data triage, including four main steps. (1) We identify the data triage operations from the collected traces along with the characteristics constrains used in these data triage operations. (2) A characteristic constraint graph is constructed to represent the logical and tempo-
6.3.1 Step 1: Identifying Data Triage Operations

ARSACA is developed to capture an analyst’s cognitive process in a less-intrusive manner when he/she is carrying out a data triage task. As it is described in Chapter 4, ARSCA provides functions that enable analysts’ data triage operations, including SEARCH, FILTER and SELECT, which are shown in Figure 6.2. Region 1 and 2 illustrate functions of filtering by condition and searching for a keyword, thus the FILTER and SEARCH. Besides, ARSCA enables an analyst to select...
Figure 6.2. ARSCA functions that support data triage operations

a subset of network events from an original set (Region 3) and to identify the common characteristics of selected network events (Region 4), thus supporting SELECT operation and LINK operations.

The data triage operations are recorded in the trace in the following structural format:

<Item Timestamp=[TIMESTAMP]>
   [ACTION_TYPE]
   ([CONTENT])
</item>

Each item contains the information of the occurring time, type and characteristic constraint of a data triage operation. “ACTION_TYPE” refers to the types of data triage operations, including FILTER, SELECT (LINK) and SEARCH. LINK is an
additional type of SEARCH, which corresponds to the LINK function provided by ARSCA which enables an analyst to specify the common characteristics of a subset of the selected network events. Several examples of the recorded data triage operations, corresponding to FILTER, SELECT (LINK), and SEARCH, are demonstrated in Table 6.1. It indicates that the “CONTENT” field of each record captures the information about the characteristic constrains specified in the data triage operation, which are the underlined parts in Table 6.1.

Focused on the data triage operations, a trace parser is developed to identify the data triage operations from the captured traces. The difficulty lies in the mapping from the ARSCA trace items to the data triage operations which is not a one-to-one matching, considering the fact that an analyst may conduct several successive actions as one data triage operation. More specifically, a FILTER or a SEARCH item could be followed by a SELECT item, which means, after narrowing down the search space by performing filtering or searching, the analyst may further select some network events in the filtered subset as suspicious events. However, sometimes analysts may directly select a subset of network events without performing any filtering or searching in advance. Both cases indicate one data triage operation. Furthermore, a LINK item may appear immediately after a SELECT item when the analyst wants to specify the common characteristics of the selected network events. Given all these situations, we mainly refer to the eight types of action sequences in Table 6.2 to parse an ARSCA trace into a set of data triage operations.

6.3.2 Step 2: Constructing Characteristic Constraint Graph

Given the data triage operations identified from the traces, we construct characteristic constraint graphs by determining the relationships among these data triage operations. The algorithm of characteristic constraint graph construction is described in Algorithm 1. An example of characteristic constraint graph is shown in
Table 6.1. The data triage operations recorded by ARSCA

<table>
<thead>
<tr>
<th>Type</th>
<th>Recorded Data Triage Operation</th>
<th>Description</th>
</tr>
</thead>
</table>
| FILTER| <Item Timestamp= 07/31/2014 13:01:41>
FILTER(
SELECT * FROM IDS Alerts WHERE SrcPort = 6667
)<Item>                                                                                                                                  | Filter the a set of network events (i.e.IDS Alerts) based on a condition (i.e. SrcPort=6667)  |
| SELECT| <Item Timestamp= 07/31/2014 13:20:29>
SELECT (FIREWALL-[4/5/2012 10:15 PM]-[Built]-[TCP](172.23.233.57/3484, 10.32.5.58/6667),
FIREWALL-[4/5/2012 10:15 PM]-[Teardown]-[TCP](172.23.233.52/5694,10.32.5.59/6667),
FIREWALL-[4/5/2012 10:15 PM]-[Built]-[TCP](172.23.233.57/3484, 10.32.5.58/6667),
FIREWALL-[4/5/2012 10:15 PM]-[Teardown]-[TCP](172.23.233.58/3231,10.32.5.51/6667),
)<Item>                                                                 | Filter the a set Select a set of network events (i.e., the underlined firewall log entries) with a common characteristics (i.e., DstPort=6667) |
| LINK  | <Item Timestamp= 07/31/2014 13:20:43>
LINK (Same DstPort)                                                                                                               | Identify the common characteristics (i.e.DstPort) of a selected set of network events               |
| SEARCH| <Item Timestamp= 08/09/2014 11:08:01>
SEARCH(Firewall Log, 172.23.233.52)                                                                                                 | Search a keyword in a set of network events specified in the Firewall log.                           |

Figure 6.3. The nodes are the data triage operations, along with the text which are indexes of them in the temporal order. The nodes in square are those triggered at least one hypotheses during an analyst’s analysis process, while circles not triggering any hypothesis. There are four different types of edges which corresponds to different logical relationships between data triage operations (i.e., “is-equal-to”, “is-subsumed-by”, “is-complementary-with” and absolute “is-complementary-with”)
Table 6.2. Identifying the data triage operations from ARSCA trace

<table>
<thead>
<tr>
<th>Item Sequence in ARSCA trace</th>
<th>Characteristic Constraint in a data triage operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILTER(cond) → OTHER(^b)</td>
<td>( C = \text{cond} )</td>
</tr>
<tr>
<td>→ SELECT({e_i})</td>
<td></td>
</tr>
<tr>
<td>→ OTHER</td>
<td></td>
</tr>
<tr>
<td>FILTER(cond)</td>
<td></td>
</tr>
<tr>
<td>→ SELECT({e_i})</td>
<td>( C = \text{cond} \cap (\text{attr} = \text{attr}_{e_i}) )</td>
</tr>
<tr>
<td>→ LINK(\text{attr})</td>
<td></td>
</tr>
<tr>
<td>SEARCH(\text{kwd}^d) → OTHER</td>
<td>( C = (\text{attr}_{\text{infer}} = \text{kwd}) )</td>
</tr>
<tr>
<td>SEARCH(\text{kwd})</td>
<td></td>
</tr>
<tr>
<td>→ SELECT({e_i})</td>
<td>( C = (\text{attr}_{e_i} = \text{kwd}) )</td>
</tr>
<tr>
<td>→ OTHER</td>
<td></td>
</tr>
<tr>
<td>SELECT({e_i}) → OTHER</td>
<td>( C = \land_i \lor_j (\text{attr}<em>{j} = \text{val}</em>{e_i,\text{attr}}) )</td>
</tr>
<tr>
<td>→ LINK(\text{attr})</td>
<td>( C = \land_i (\text{attr} = \land \text{val}_{e_i,\text{attr}}) )</td>
</tr>
</tbody>
</table>

\(^a\)Filtering condition
\(^b\)Other operation, e.g., hypothesis-related operations
\(^c\)The attribute of the events of interest
\(^d\)The keyword used for search.

combined with their temporal relationships (i.e., “happen-after”). Figure 6.3 is a partial characteristic constraint graph, with 6 nodes and their relationships. Node 8 is a data triage operation using the constraint “DstPort=6667”. It is represented in a square because it resulted in some interesting findings and triggered a hypothesis. After performing this data triage operation, the analyst screened out the network events with destination port 6667 by conducting another data triage operation (i.e., Node 9). It may indicate that the analyst switched his attention to the unexplored network events with other characteristics after investigating the network events via destination port 6667. The following data triage operations of Node 13 and Node 14 have an “is-subsumed-by” relationship with the data triage operation of Node 9. It implies the analyst gradually screened out the network events to narrow down the scope. The data triage operation of Node 14 is the end of this narrowing process and it let the analyst generate another hypothesis. Sim-
Data: $S_{\text{op}}$, a sequence of data triage operations in temporal order

Result: $\mathcal{G}$, a characteristic constraint graph

1. $V = \{ \text{op} | \text{op} \in S_{\text{op}} \}; // \text{ vertex set}$
2. $E = \emptyset; // \text{ edge set}$
3. for $i$ in 2:len($S_{\text{op}}$) do
   4. $C_i =$ characteristic constraint of $S_{\text{op}}[i]$;
   5. for $j$ in 1:$i$ do
      6. $C_j =$ characteristic constraint of $S_{\text{op}}[j]$;
      7. $rela =$ relation($C_j$, $C_i$);
      8. if label is not null then
         9. $E = E + <S_{\text{op}}[j], S_{\text{op}}[i], rela>$;
      end
   end
4. relation($C_1$, $C_2$)
   14. if $C_1 \rightarrow C_2$ and $C_2 \rightarrow C_1$ then
      15. return “isEql”;
   end
   17. else if $C_1 \rightarrow C_2$ then
      18. return “isSub”;
   end
   20. else if $C_1 \rightarrow \neg C_2$ and $C_2 \rightarrow \neg C_1$ then
      21. return “isCom”;
   end
   24. return “Other”;
end

Algorithm 1: Construct a Characteristic Constraint Graph

Similarly, the sequence of Node 9, Node 15 and Node 16 indicates the same strategy employed by the analyst.

6.3.3 Step 3: Mining the Characteristic Constraint Graphs

Several steps are taken to mine the CC-Graphs for identifying the characteristic constraints from the analysts’ traces for effective data triage. First of all, it is critical to distinguish the critical data triage operations that lead analysts to key findings from the exploratory data triage operations recorded in the traces. Considering that identified critical data triage operations could have overlap, a
Figure 6.3. A partial characteristic constraint graph constructed from an analyst’s trace. The nodes are the data triage operations, with the index of their temporal order. A squared node refers to a data triage operation based on which the analyst generated a following hypothesis about the possible attack path. However, circle nodes are the data triage operations that did not result in any hypothesis.

Following step is to adjust the identified data triage operations to make sure they contain mutually exclusive characteristic constraints. After the adjustment, data triage operations are the “candidates” for constructing state machines. To make sure each state machine is built up for detecting evidence of one attack chain, the third step is taken to group the nodes according to which attack chain they may belong to based on heuristics.

6.3.3.1 Extracting the Critical Endpoints from “isSub” Subgraphs

Given data triage operations $O_1$ and $O_2$, $isSub \succ_t (O_2, O_1)$ indicates that $O_2$ is performed after $O_1$, and $O_2$ further narrows the $O_1$’s network event set by using a more strict characteristic constraint. As the case described in Figure 6.3, analysts may conduct a series of data triage operations with the “is-subsumed-by” (isSub)
relationships to gradually narrow down the network events. The endpoints of such processes can be viewed as the critical characteristic constraints that represent a noteworthy set of network events. To investigate the endpoints, we consider the subgraph that only consists of the edges of the relationship \( isSub \succ_t \) and \( isEql \succ_t \) (isEql is a special case of isSub) and name it “isSub” subgraph.

6.3.3.2 Mutually Exclusive Characteristic Constraints

It is desirable to have the endpoints of an “isSub” subgraph be mutually exclusive. To ensure the nodes to be mutually exclusive, we examine the logic relationships between two nodes, and add additional characteristic constraints to one of them as follows. Let \( V_{ends} \) be the set of endpoints in an “isSub” subgraph, several steps are taken to make sure every two endpoints in this subgraph have an “isCom” relationship. (1) If two endpoints have an “isEql” relationship, we drop one from \( V_{ends} \). (2) Given \( V_{ends} \), we consider the subgraph induced by \( V_{ends} \) from the original CC-Graph, \( G_{induced} = < V_{ends}, \{ R_{logic} \} > \), where \( R_{logic} = \{ < v_i, v_j > | isEql(v_i, v_j) \text{ or } isSub(v_i, v_j) \text{ or } isCom(v_i, v_j) \} \). Only focusing on the edges of “isCom” relationships in \( G_{induced} \), we apply the Bron-Kerbosch algorithm to find all the maximum cliques in \( G_{induced} \). A maximum clique is a subset of the vertexes, represented by \( G_C \in V_{induced} \), such that \( \forall v_i, v_j \in G_C, isCom(v_i, v_j) \), and \( \exists v_k \in V_{induced}, v_k \notin G_C \), and \( \forall v_i \in G_C, isCom(v_k, v_i) \). The largest maximum clique is denoted as \( V_{clique} \). (3) We add each \( v \in V_{ends}, v \notin V_{clique} \) into \( V_{clique} \) by removing the overlap between \( v \) and \( v_j \in V_{clique} \). For \( v_i \notin V_{clique} \), it overlaps with a \( v_j \notin V_{clique} \), iff there exists a network event \( e \) satisfies both \( C_i \) and \( C_j \). To remove that overlap, we adjust the characteristic constraint used in \( v_i \) (say \( C_i \)) to \( C_i \land \neq C_j \). If \( C_i \land \neq C_j \) is not false, we add the adjusted \( v_i \) to \( V_{clique} \) (\( V_{clique} \in G_C \)).
6.3.3.3 Grouping the “Candidates” based on Heuristics

Given a set of adjusted data triage operations as the “candidates” for constructing state machines, we group them according to which attack chain they belong based on heuristics. One heuristic is to group the data triage operations whose characteristic constraints are set on a same IP (e.g., a suspected attack or a potential target), considering the fact that an attack chain usually has one common attacker and target. The second heuristic is that two data triage operations can be grouped if they were arranged in a same path of AOH-Trees (described in Section 3.2.4 in Chapter 3) by an analyst when he/she was performing a task. According to the design of AOH-Trees, a descendant action node is viewed as a follow-up action for investigating a hypothesis generated based on previous action. These heuristics are generated based on the domain knowledge learned from our previous experience. The characteristic constraints of these “candidates” are as readable as SIEM rules. Domain experts could also create or modify these heuristics for grouping the candidates if we apply this approach to real world.

6.3.4 Step 4: State Machine for Real-Time Pattern Matching

6.3.4.1 Attack Path Pattern

Given the critical characteristic constrains mined from the CC-Graphs (i.e., the nodes in $V_{clique}$), attack path patterns can be constructed based on the critical nodes and their temporal orders. An attack path pattern is a sequence of characteristic constraints in temporal order, which specifies a class of attack path instances. The temporal relationship is a “can-happen-before” relationship between two characteristic constraints, denoted as $\succ_{Cl} (\cdot, \cdot)$. Let $C_1$ and $C_2$ be two pre-specified characteristic constraints, $\succ_{Cl} (C_1, C_2)$ refers to the analysts’ knowledge about the attack: it happened and could happen again if a set of network events
satisfied $C_1$ occurred before the other set of events involving the same hosts that satisfies $C_2$ in an attack path. For example, one typical step in a botnet attack is the IRC communication between internal workstation with the external C&C servers. The IRC communication can be followed by a data exfiltration using FTP protocol. Therefore, this attack can be discovered by the attack pattern path: “(SrcIP = external servers AND SrcPort = 6667) OR (DstIP = external servers AND DstPort = 6667)” and “SrcIP = internal workstation AND DstPort = 21”.

The “can-happen-before” relationships of the nodes in an attack pattern is identified based on the temporal order of the corresponding network events in the task performed by the analyst. Assume that the network events come in sequence over time $E = (e_1, \ldots, e_n)$. We say this sequence of network events satisfies a sequence of characteristic constraints $(C_1, \ldots, C_m)$ defined in an attack path pattern $G_C$, iff we have, $\forall C_i(1 \leq i \leq m), \exists e_p(1 \leq p \leq n)$ satisfies $C_i \rightarrow \forall C_j(1 \leq j < i), \exists e_q(1 \leq q \leq p)$ satisfies $C_j$. Therefore, given $(C_1, \ldots, C_m)$ and $E$, the attack path instances detected based on $(C_1, \ldots, C_m)$ is a sequence of network event sets, $attack_{(C_1,\ldots,C_m)}(E) = \{(E_1, \ldots, E_m)\}$, where $E_{ik}(1 \leq i \leq m) = \{e_{ik}|e_{ik} is a network event from source_{ik1} to destination_{ik2} that satisfies C_i\}$. The algorithm for identifying the “can-happen-before” relationships is demonstrated in Algorithm 2.

6.3.4.2 State Machine

A finite state machine is constructed to automate data triage, given an attack path pattern, named “DT-SM”. A state transition is defined as $\delta : S \times D \rightarrow S$. Given the current state $S_i \in S$ and a new network event $e_i$, we have $\delta(S_i, e_i) = S(i+1)$ iff $\exists C_j$, that $\exists\hat{C}(C_0, \ldots, C_n) \in S_i$ and $C_j \in (C_0, \ldots, C_n)$, $e_i$ satisfies $C_j$ and $\exists(C_0, \ldots, C_n) \in S_i$, that $\succ_{C_t} (C_i, C_j)$. Therefore, $S_i(i + 1) = S_i - (C_0, \ldots, C_i) + (C_0, \ldots, C_j)$. Therefore, given the input of a sequence of network events in temporal order, a DT-SM takes one network event at one time and examine whether this event
Data: \( D_{task} \): a sequence of network event sets;  
\( C \): a set of characteristic constrains  
Result: \( \mathcal{G}_C = C, \{ \prec_C \} \), an attack path pattern

1 map \( \prec C \{ e \} \) // map each characteristic constraint to the  
    network events that satisfies it, the network events are in  
    temporal order

2 for \( e_i \) in \( D_{task} \) do
3    if exists \( C \) holds on \( e_i \) then
4        map.put(\( C \), \( e_i \));
5    end
6 end

7 for \( C_1, C_2 \) in \( C \) do
8    Find the first \( e_1 \) in \( \text{map}(C_1) \), and the first \( e_1 \) in \( \text{map}(C_1) \) that
9        \( e_1 \{ \text{src, dst} \} = e_2 \{ \text{src, dst} \} \);
10       if \( e_1.\text{time} < e_2.\text{time} \) then
11          add \( \prec_C (C_1, C_2) \);
12       end
13 end

Algorithm 2: Identifying “can-happen-before” relationships

triggers state transition. Once all the input network events have been processed,  
the DT-SM output the instances of the attack path pattern. Each instance is a  
sequence of network event sets that satisfy a characteristic constraint sequence in  
the attack path pattern.

6.4 Evaluation

We implemented the automated data triage system and tested it in our experiment.  
Our evaluation focuses on (1) the effectiveness of constructing the data triage  
automata (DT-SM) based on the traces, and (2) the usefulness of the data triage  
rules mined from the traces and the performance of the constructed DT-SMs.
### Table 6.3. Attack ground truth of 3 selected time windows

<table>
<thead>
<tr>
<th>Time Window</th>
<th>Attack Scenario</th>
<th>Data Sources</th>
</tr>
</thead>
</table>
| **Scenario from the beginning (started from 4/5/2012 20:25):**  
An internal workstation got infected with a botnet due to a USB insertion.  
The botnet spread fast. | (1) The botnet communicated with the external C&C servers using IRC.  
(2) The botnet located sensitive data.  
(3) Failed attempted to exfiltrate data using FTP.  
(4) Successful attempted to exfiltrate data using SSH. | # IDS Alerts: 239  
# Firewall Log: 115,524 |

| I:  
4/5/2012  
22:15-22:25  
(10 min) | | |

| **Scenario in the middle (about 20 hours, 4/5/2012 22:26-4/6/2012 17:56):**  
The IT department noticed the problem and rebooted some infected workstations several times. The data exfiltration drops, but the majority of the botnet traffic still exists. | (1) The botnet spread (inquiring the network hardware)  
(2) New IRC communications.  
(3) New attempt to exfiltrate data using FTP. | # IDS Alerts: 228  
# Firewall Log: 48,012 |

| II:  
4/6/2012  
18:05-18:15  
(10 min) | | |

| **Scenario in the middle (about 20 hours, 4/5/2012 22:26-4/6/2012 17:56):**  
The IT department noticed the problem and rebooted some infected workstations several times. The data exfiltration drops, but the majority of the botnet traffic still exists. | (1) Additional workstations were infected.  
(2) The botnet communication continued.  
(3) FTP exfiltration continued. | # IDS Alerts: 1810  
# Firewall Log: 599,489 |

| III:  
4/6/2012  
18:16-19:56  
(100 min) | | |

### 6.4.1 Experiment Dataset

#### 6.4.1.1 ARSCA Traces Collected from an Experiment

We recruited 30 full-time professional cyber analysts in a previous experiment and asked them to accomplish a data triage task. The ARSCA toolkit was used to audit the analysts’ cognitive processes while they were performing the task. We had to screen out one ARSCA trace because one of the participant failed to accomplish the task. At last, we collected 29 ARSCA traces in total with 1104 trace items. The average length of the traces is 31.17.
6.4.1.2 Task Data Sources

The cyber attack analysis task was designed based on the cyber situational awareness task in VAST Challenge 2012 with a setting of a bank’s network with approximately 5000 machines. The VAST challenge provided participants with IDS alerts and firewall log from the network. VAST Challenge 2012 required the participants to identify the noteworthy attack incidents happened in the 40 hours based on the IDS alerts and firewall logs [12]. The entire data sources were composed of 23,595,817 firewall logs and 35,709 IDS alerts. The task scenario underlying the data sources is a multistage attack path that begins with normal network events, including regional headquarters computers communicating with internal headboards financial sever and normal web browsing. The attack starts when an internal workstation was infected with a botnet due to an inserted USB. The botnet replicated itself to other hosts on the network, and meanwhile the bonnet communicated with several C&C servers. The botnet kept infecting additional computers. It attempted to exfiltrate data using FTP connections but it failed. After that, the botnet successfully exfiltrated data using SSH networks. In the following hours, the majority of the botnet communication and data exfiltration kept happening.

Task Data Sources and Scenario for ARSCA Tracing. The original data sources provided in the VAST challenge are too large for human analysts to undertake during the limited task time. Therefore, only small portions of data was selected from the original dataset and prenseted to the participants to complete the task in the experimental time (60 minutes). Two time windows, time window I and time window II shown in Table 6.3, were selected and each of them were used to make a data triage task for the participants in the experiment. In the experiment where ARSCA traces were collected, 10 analysts accomplished the task of time window I, and 19 analysts accomplished the task of time window II.
Table 6.4. Summary statistics for the Characteristic Constraint Graphs constructed from the professional analysts’ traces

<table>
<thead>
<tr>
<th></th>
<th>Nodes</th>
<th>Avg. degree</th>
<th>Max SCC size</th>
<th>Edges</th>
<th>“isEql” edges</th>
<th>“isSub” edges</th>
<th>“isCom” edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean</td>
<td>12.00</td>
<td>7.59</td>
<td>11.62%</td>
<td>38.55</td>
<td>2.069</td>
<td>10.41</td>
<td>26.07</td>
</tr>
<tr>
<td>StDev</td>
<td>6.79</td>
<td>5.32</td>
<td>6.83%</td>
<td>47.75</td>
<td>3.105</td>
<td>14.98</td>
<td>34.74</td>
</tr>
</tbody>
</table>

6.4.2 DT-SM Construction

6.4.2.1 Data Triage Operation Identification

We first evaluated the accuracy of the automated data triage operation identification by comparing with the data triage operation identified by human. We had two persons manually parse the ARSCA traces, and the identified data triage operation serves as the ground truth, and they were familiar with the task and the ARSCA toolkit. One acted as the main identifier, and the other as the evaluator: given an ARSCA trace, the main identifier first read through the trace and specified the data triage operations in the log; the evaluator then read the trace for the second time and proofreaded the data triage operations identified by the main identifier; whenever a disagreement occurred, both of them went over the ARSCA trace again together to reach an agreement by negotiation. Given the 29 ARSCA traces, 1181 log items were manually analyzed and 394 data triage operations were identified. The automatic system identified 348 data triage operations in total. Comparing them with the ground truth, there were 322 data triage operations correctly identified by the system. 62 data triage operations in the ground truth were not identified by the system. Therefore, the false positive rate is 0.075 and the false negative rate is 0.161.

6.4.2.2 Characteristic Constraint Graph Analysis

In total, 29 LINKs were constructed from the 29 traces of the professional analysts’ cognitive processes collected in our previous experiment, which are shown in
Figure 6.4. The characteristic constraint graphs of the professional analysts’ traces.
Figure 6.4. Comparing the analysts’ conclusions with the ground truth, we ranked their task performance from 1 to 5. In Figure 6.4, we categorized the CC-graphs according to the analysts’ task performance. Although there were wide individual differences among the CC-Graphs in a same category regarding the structure and complexity of these graphs, we found that all these CC-Graphs indicates that the analysts’ analytical reasoning processes of data triage are non-linear.

Table 6.4 summarizes the statistics of the CC-Graph. The large standard deviation of node number and edge number of the CC-Graphs also indicate that there exist large individual differences among analysts. The edges of “isSub” and “isCom” (i.e., corresponding to the “is-subsumed-by and happen-after” and “is-complementary-with and happen-after” respectively) account for a large portion of the CC-Graph edges. Besides, it also shows that, in average, a maximum strongly connected component (SCC) accounts for 11.62% of a CC-Graph. SCC represents a set of mutually connected data triage operations, which may imply how an analyst detected suspicious network events and update his/her mental model along with switching his/her attention from one hypothesis to another. A large SCC indicates that a large number of data triage operations are directly or indirectly connected in an analyst’s analytical reasoning process.

Considering the large individual differences among the CC-Graphs, we conduct motif analysis to investigate the small local patterns in these graphs. We mainly focus on two types of subgraphs: (1) “isSub” subgraph consisting of the nodes with merely “is-subsumed-by and happen-after” relationships, and (2) “isCom” subgraph consisting of the nodes with merely “is-complementary-with and happen-after” relationships. Figure 6.5 displays the motif profiles of the two types of subgraphs, which are developed by counting the frequency of 16 triads in each type of subgraphs. The numbers of reciprocal triads (i.e., the triads with two-way arrow) for both types of subgraphs are zero because the temporal relationships are irreversible. Besides, the number of the triad 021C in “isSub” subgraphs is also zero.
because the “isSub” edges are transitive so that the pattern always falls into the triad 030T instead. The number of triad 003 and 012 for both types of subgraphs are significant. These two triads are not viewed as interesting patterns because they contain isolated data triage operations. Therefore, we are mainly interested in the remaining triads with significant frequency in both types of subgroups, including 021D, 021U and 030T.

### 6.4.2.3 Attack Path Pattern Construction

To evaluate the effectiveness of attack path pattern construction, we constructed an attack path pattern for each characteristic constraint graph. We had to exclude 5 ARSCA traces because less than 2 data triage operations were identified in these logs. In total, we have 32 attack path patterns constructed, containing 81 characteristic constraints. The average number of the nodes in the attack path patterns is 2.781.

We evaluated these attack path patterns by checking whether its characteristic constraints are critical and mutually exclusive. To determine whether the charac-
Table 6.5. Top 10 characteristic constraints in the attack path pattern by analyzing the 29 analysts’ data triage operations

<table>
<thead>
<tr>
<th>Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 DSTPORT = 6667 AND SRCIP = 172.23.<em>.</em> AND DSTIP = 10.32.5.* AND PROTOCOL = TCP AND SERVICE = 6667_tcp</td>
</tr>
<tr>
<td>2 SRCPORT = 6667 AND SRCIP = 10.32.5.* AND DSTIP = 127.23.233.* AND PRIORITY = 3 AND DESCRIPTION = [1:2000355:5] ET POLICY IRC authorization message</td>
</tr>
<tr>
<td>3 SRCIP = 172.23.<em>.</em> AND DSTIP = 172.23.0.10 AND DSTPORT = 445 AND PRIORITY = 3</td>
</tr>
<tr>
<td>4 SRCIP = 172.23.235.* AND DSTIP = 10.32.5.* AND DSTPORT = 21 AND OPERATION = Deny AND PRIORITY = Warning AND PROTOCOL = TCP AND SERVICE = ftp</td>
</tr>
<tr>
<td>5 SRCIP = 172.23.<em>.</em> AND DSTIP = 10.32.5.* AND DSTPORT = 6667 AND OPERATION = Deny</td>
</tr>
<tr>
<td>6 SRCIP = 172.23.<em>.</em> AND DSTPORT = 53</td>
</tr>
<tr>
<td>7 SRCIP = 172.23.234.* AND DSTIP = 10.32.5.* AND DSTPORT = 22</td>
</tr>
<tr>
<td>8 SRCIP = 10.32.5.* AND DSTIP = 172.23.1.168 AND DSTPORT = 6667</td>
</tr>
<tr>
<td>9 DSTPORT = 80 AND DSTIP = 10.32.0.100 AND SRCIP &gt; 172.23.0.0</td>
</tr>
<tr>
<td>10 SRCIP = 172.23.0.108 AND DSTPORT = 6667 AND DSTIP = 10.32.5.*</td>
</tr>
</tbody>
</table>

Characteristic constraints in an attack path schema are critical, we mapped them to the analysts’ answer to the task question about the most important observations. We found 79 out of the 89 characteristic constraints in the 32 attack path patterns mentioned by the analysts that lead to important observation (88.76%). As for the 10 characteristic constraints not mentioned in analysts’ answer, we found 6 of them also lead to hypotheses in analysts’ ARSCA traces. All the characteristic constrains in the attack path pattern are mutually exclusive. Table 6.5 lists the top 10 characteristic constraints included in the attack path pattern.
6.4.3 Performance of DT-SM

6.4.3.1 Data Source and Attack Scenario for Testing

We evaluated the data triage performance of DT-SM on the data sources that were much larger than the task data sources used in the experiment. Considering the task data sources selected from a 10-minute-time window, we chose a 100-minute time window (4/6/2012 18:16-19:56) as the testing data sources from the original VAST dataset, which corresponds to the data in time window III in Table 6.3.

The second problem is to determine the ground truth before evaluating the DT-SM’s data triage result. Although the VAST Challenge provides an attack description, it is still not apparent regarding whether each alert/log entry in the data sources is related to attack or not. Therefore, we manually processed the data sources in time window III and tagged each data entry regarding to which attack step it is related.

6.4.3.2 Performance of DT-SM

A set of DT-SMs were constructed from the traces, each of them can detect a
set of event sequences. We combined the results of multiple DT-SMs by selecting the event sequences with high occurrence frequency. Figure 6.6 demonstrates an example of how the DT-SMs results are combined. A threshold can be set up to determine what is the least frequency of occurrence for the event sequences of interest (i.e., “least support”). Given the output of each DT-SM, we calculated the occurrence of frequency of each event sequence and selected the ones whose occurrence of frequency was above the threshold.

The performance of the DT-SM was measured by false positive and false negative rates. We use \( e^S \) to denote the network events in the output of the DT-SM \( S \), and \( e^T \) to denote the network events included in the ground truth \( T \). We calculated the false positive and false negative rate using the following formula:

\[
\text{false positive} = 1 - \frac{|\bigcup \{ e^T | e^T \in S \}|}{|\bigcup \{ e^S | e^S \in S \}|}
\]  
(6.2)

\[
\text{false negative} = 1 - \frac{|\bigcup \{ e^T | e^T \in S \}|}{|\bigcup \{ e^T | e^T \in T \}|}
\]  
(6.3)

Figure 6.7. The performance of DT-SMs with different thresholds of least support.
Table 6.6. Performance of the DT-SMs built on the traces from analysts with different task performance. The threshold of least support is 2.

<table>
<thead>
<tr>
<th>Group</th>
<th>Performance Score (1-5)</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>High-Performance</td>
<td>4-5</td>
<td>False Positive: 0.010</td>
</tr>
<tr>
<td></td>
<td></td>
<td>False Negative: 0.225</td>
</tr>
<tr>
<td>Low-Performance</td>
<td>2-3</td>
<td>False Positive: 0.015</td>
</tr>
<tr>
<td></td>
<td></td>
<td>False Negative: 0.350</td>
</tr>
</tbody>
</table>

We constructed a DT-SM constructed based on the data triage operations from the 29 participates and ran it on the testing data. The performance of the DT-SMs (i.e., false positive and false negative) built on their traces is shown in Figure 6.7. We set the threshold of least support as 1, 2, 5 and 10 respectively. The result shows that, in general, the false positive rates are satisfactory. However, the false negative rates are high. As the threshold increases, the false positive rate decreases while false negative rate increasing. Comparing the results at different thresholds, we found that the best result occurred when the threshold of least support was set as 2. One of the possible reasons of the high false negative rate can be that the participants failed to detect some suspicious network events in the task so that their traces don’t involve the effective data triage operations. Next, we will evaluate the effects of analysts’ task performance on the DT-SMs’ performance.

6.4.4 Effect of Analysts’ Task Performance on the DT-SM’s Performance

We know that the performance of a DT-SM is mainly determined by the quality of the attack path patterns mined from the analysts’ traces, and analysts’ traces are closely related to analysts’ task performance. Therefore, we compared the performance of the DT-SMs built on the traces from the analysts with different task performance.
In the experiment in which the traces were captured, analysts were asked about what the attack scenario underlies the provided data sources after they accomplishing the task. We evaluated the analysts’ task performance by comparing their answers with the ground truth and gave each of them a score in the range of [1, 5]. Among the 29 participants, the first 10 analysts, who participated in the first task working with the data sources of time window I, displayed a diverse range of performance, while the other 19 analysts with the data sources of time window II presented identical results with similar scores. Therefore, we only took the traces of the first 10 analysts into examination. We used each analyst’s trace to construct a DT-SM and then ran the DT-SM on the testing data sources of time window III. We divided the participants into two groups according to their task performance: Group 1 of 5 participants with high task score (i.e., either 4 or 5), and Group 2 of 5 participants with low task score (i.e., either 2 or 3). Table 6.6 shows the comparison results of the false positive and false negative rates of the DT-SMs built on the traces in these two groups. It shows that the false positive rates in both groups are very small. However, the first group of DT-SMs, which corresponds to higher task performance, have much smaller false negative. It can be explained by the fact that analysts with good performance in the task was able to detect most of the attack-related network events; their data triage operations may contain more useful characteristic constraints compared with the analysts who performed poorly in the task.

A Worst Case: We noticed a DT-SM built on the trace of an analyst (whose ID is 08239) had the highest false negative rate. We looked into the attack path patterns constructed based on his trace, and found a sequence of 3 characteristic constraints: (1) DSTPORT=6667, (2) DSTIP = 10.32.0.* AND DSTPORT = 80 AND PRIORITY = Info AND PROTOCOL = TCP, and (3) SRCIP = 10.32.5.* AND SRCPORT = 6667 AND DSTIP = 10.32.0.* AND PROTOCOL = TCP. According to the ground truth, the second characteristic constraint is not related to any attack
step of the ground truth. It suggests the characteristic constraint is misleading and results in quite a few instances with irrelevant characteristic constrains so that the false negative is much higher. This case can be avoided by combining the ARSCA traces of multiple analysts together.

6.5 Related Work

Researchers have recognized the significant role of human analysts in cyber analysis. Several exciting field studies have focused on understanding analysts’ analytical processes [17, 100, 24]. Our current work is most closely related to the ARSCA toolkit of Zhong et al. [101] because we directly borrow the operation traces collected by this toolkit (i.e., ARSCA) as the input, instead of reinventing the wheel. This work takes a big step forward by constructing semantic models based on the operation traces and eliciting useful information to automate the triage analysis.

Prior works on alert correlation and SIEM are complimentary to but distinct from our work. Regarding the works on alert correlation, a lot of methods have been proposed using a wide range of technologies. Sadoddin and Ghorbani [76] conducted an extensive survey that classifies the state-of-the-art alert correlation techniques, each of which has pros and cons. It also pointed out that knowledge acquisition is critical for most alert correlation methods. For example, the prerequisites and consequences of alerts need to be specified for rule-based correlation (e.g., [65, 15, 85]; attack scenarios models (e.g., temporal logic formalism [62]) should be defined for scenario-based correlation. The focus of our work is the automatic knowledge elicitation, with the goal of eliciting the attack path pattern from analysts’ operations in their previous analysis processes. The data triage state machine has three merits: (1) It can ease analysts’ burden by applying the data constraints they used previously to triage new data so that analysts can exert more effort on detailed investigation or on coping with new challenges, and thus in
turn improving the overall quality of incident reports in an SOC. (2) It is by nature enterprise specific and network specific. A DT-SM constructed from an analyst’s operation trace could be directly used by another analysts. (3) The DT-SM can adapt well to the highly dynamic cyber environment because it takes in the inputs of analysts’ operation traces. These traces can be collected continuously as long as the capturing toolkit is launched while analysts are working.

6.6 Limitation and Evasion

This work verifies that it is feasible to elicit attack path patterns by modeling and mining the traces of analysts’ data triage cognitive processes. In spite of that, it has some limitations. The first one lies in that the traces of the analysts’ cognitive processes of analyzing 10-minute-time-window events cannot fully embody the analysis expertise needed for analyzing 100-minute-time-window events in full play. Therefore, the DT-SMs built on the attack path pattern constructed from the traces can’t process the network events in the 100-minute time window completely. However, essentially, it is not a limitation of the automated data triage system but a problem with the limited set of the traces. In terms of the real-world practice, analysts keep on analyzing the continuous incoming data sources, meanwhile the analysts’ traces will automatically grow by ARSCA. Therefore, in this case, the input ARSCA traces will be continuously imported into our system and update the attack path pattern accordingly.

Besides, there still lies some room for improving the false negative rate in several important aspects. Firstly, it would be helpful to enlarge the set of ARSCA traces because introducing more data triage operation traces will enable the system to construct a more comprehensive attack path pattern. Meanwhile, a sufficient set of endpoints in the “isSub” subgraph will enable us to conduct frequency item mining to select the significant endpoints, which can lower the false negative rate.
In addition, we can also emphasize the non-action-related information captured in ARSCA traces, which is the analysts’ hypotheses. Working with ARSCA in a data triage task, analysts may note down their hypotheses to interpret their previous observations or to explain their action plan for further inspection. Such information ought to be useful to improve the accuracy of specifying the characteristic constraints in the identified data triage operations.

Thirdly, although the automated system is developed to reduce labor costs and to improve quality in incident reports, the automation meanwhile creates new tasks for analysts, such as working together with ARSCA for cognitive task tracing and comprehending the output of the automated data triage system. Therefore, it is necessary to conduct further evaluation of the impact of the automated data triage system on cybersecurity analysts’ work efficiency. Torkzadeha and Doll suggested a four factor instrument that measures how technology on work impact task productivity, task innovation, customer satisfaction and management control [88]. Similarly, we need to generate an instrument for measuring the impacts of the automated data triage system on the work efficiency of analysts. Based on the measurement instrument, we can further test our hypotheses about the positive influence of the automated system on reducing the analysts’ workloads and improving the quality of incident reports.

6.7 Conclusion

Based on the process tracing method for capture the traces of analysts’ data triage cognitive processes, We proposed a system that develops data triage automatons by mining the data triage operations recorded in the traces. To develop the automated data triage system, we proposed a graph-based trace mining method for discovering the critical data characteristic constrains that can be used as rules for data filtering and correlation. With these rules, finite state machines were
constructed to conduct automatic data triage. We evaluated the effectiveness of developing the state machines and their data triage performance on a much larger data set. The result shows that it is feasible to conduct automated data triage by leveraging analysts’ traces. The state machines are able to finish processing a large amount of data within a couple of minutes. Comparing the performance of automated data triage with the ground truth, we found that a satisfactory false positive rate is achieved, though the false negative rate is high and yet need further improvement. Besides, another important implication gained from the study is that selecting the traces from experts with good task performance can improve the performance of the automated data triage system.
Conclusion

The core theme of my thesis is capturing cybersecurity analysts’ data triage cognitive processes and utilizing the captured cognitive processes to develop data triage automation. It is broadly motivated by three major trends in cybersecurity analytics. Firstly, human analysts are playing a critical and indispensable role in a cyber defense system because they can perform complicated reasoning to comprehend the sophisticated cyberattack strategies. Secondly, analysts are so far bounded by the tedious data triage tasks that they can hardly concentrate on malware and intrusion detection. Thirdly, although the Security Information and Event Management (SIEMs) take a big leap forward in assisting cybersecurity analytics, they are extremely expensive due to the large amount of analysts’ time and expertise required in constantly system customization.

My work takes the first step to achieve the goal of liberating cybersecurity analysts from tedious data triage to focus on the higher-level Cyber SA. It contains four main components: (1) a data triage system model and a trace representation of analysts’ fine-grained cognitive processes, (2) a computer-aided method for tracing analysts’ cognitive processes of data triage, (3) an empirical study for gaining deep understanding of analysts’ cognitive process by capturing them in an experiment and exploring them through quantitative analysis, and (4) an automated method
for mining traces to construct data triage automaton.

As a fundamental step, a Cyber SA data triage model was proposed as a dynamic Cyber-Human System (CHS) which provides a basis for describing the interaction between human analysts and massive and rapidly-changing network data. Based on the CHS, we further developed the trace representation to define an analyst’s fine-grained cognitive process of Cyber SA data triage. The trace representation makes it clear how an analyst’s cognitive process iterates through three types of key elements (i.e., actions, observations, and hypotheses) to accomplish a data triage task.

Based on the trace representation, we developed a method for tracing analysts’ fine-grained cognitive processes. A computer toolkit was developed to aid the tracing method. Working together with an analyst, the tool automatically captured the analyst’s data triage operations and observations and recorded the hypotheses typed down by the analyst during the task. The integration of automatic capture and self-reports minimizes disturbance to analysts and lower the cost of capturing their cognitive processes.

The data triage tracing method provides a feasible way to conduct fine-grained CTA study on Cyber SA data triage. In collaboration with Army Research Lab (ARL), an experiment was designed and conducted to capture analysts’ cognitive processes of data triage in traces. We recruited thirty professional analysts from ARL and twenty doctoral students specialized in cybersecurity in the experiment and captured traces of their cognitive processes of performing a simulated data triage task. Given the traces, we reconstructed the participants’ cognitive processes by analyzing the captured traces. The analysis result shows that there exist some common patterns in analysts’ operations captured in the traces, which indicates the possibility of developing automatic trace analysis method. Besides, we gained a deep understanding of the participants’ cognitive process with the observation of different data triage strategies.
Given the traces captured in the experiment, we developed data triage automatons by leveraging these to reduce analysts’ workloads of data triage. An automated trace analysis algorithm was developed to construct data triage “rules” out of the traces, and the “rules” were used to build a finite state machine for conducting automated data triage. To the best of our knowledge, this is the first method that directly leverages traces of human analysts’ cognitive processes to develop data triage automation. Further evaluation is still needed to systematically validate the positive impact of the proposed automated data triage system on analysts’ work efficiency. However, relating this method to the rule-based SIEM system, the proposed approach can greatly reduce analysts’ workloads in generating SIEM rules and thus render the cost of data triage automation generation orders of magnitudes smaller.


Vita
Chen Zhong

Chen Zhong enrolled in the Ph.D. program in Information Sciences and Technology at Pennsylvania State University in August 2011. Prior to that, she received the B.S. degree in Computer Science from Nanjing University, China in June 2011. Her research interests include cybersecurity, artificial intelligence (knowledge representation and engineering), cognitive modeling (sensemaking and information-seeking), and human-computer interaction (interactive tool design and development, process tracing). Her work has led to seven research papers (six first-authored) published and presented at conferences including IEEE ISI, IEEE VAST, IEEE CogSIMA, ACM HotSOS, IEEE IDS, and two book chapters. She is a receiver of the (ISC)2 Graduate Scholarship and GHC Women in Computing Scholarship. She was awarded the First Place in Engineering at the Pennsylvania State University Graduate Exhibition in 2015. Besides, she was awarded two student travel grant awards from IEEE CogSIMA, and an Honorable Mention from VAST Challenge 2013. She is a member of ACM, IEEE, Science of Security VO, and IEEE Women in Engineering.