An Integrated Computer-Aided Cognitive Task Analysis Method for Tracing Cyber-Attack Analysis Processes

Chen Zhong, John Yen, Peng Liu
College of Information Sciences and Technology
Pennsylvania State University
{czz111, dok5181, jyen, pliu}@ist.psu.edu

Rob Erbacher, Renee Etoty, Christopher Garneau
Army Research Lab
{robert.f.erbacher.civ,renee.e.etoty.civ,christopher.j.garneau.civ}@mail.mil

ABSTRACT

As cyber-attacks become more sophisticated, cyber-attack analysts are required to process large amounts of network data and to reason under uncertainty with the aim of detecting cyber-attacks. Capturing and studying the fine-grained analysts' cognitive processes helps researchers gain deep understanding of how they conduct analytical reasoning and elicit their procedure knowledge and experience to further improve their performance. However, it's very challenging to conduct cognitive task analysis studies in cyber-attack analysis. To address the problem, we propose an integrated computer-aided data collection method for cognitive task analysis (CTA) which has three building blocks: a trace representation of the fine-grained cyber-attack analysis process, a computer tool supporting process tracing and a laboratory experiment for collecting traces of analysts' cognitive processes in conducting a cyber-attack analysis task. This CTA method integrates automatic capture and situated self-reports in a novel way to avoiding distracting analysts from their work and adding much extra work load. With IRB approval, we recruited thirteen full-time professional analysts and seventeen doctoral students specialized in cyber security in our experiment. We mainly employ the qualitative data analysis method to analyze the collected traces and analysts' comments. The results of the preliminary trace analysis turn out highly promising.

INTRODUCTION

According to a recent Kaspersky report, Brazil faced 87,776 cyber-attack attempts during the period of World Cup 2014 [1]. Cyber-attack analysis, as a newly emerged complex cognitive task, has been increasingly vital for organizations to tackle various cyber threats. Due to the lack of uniformed terminology, there are many other terms used such as network security analysis or intrusion detection analysis.

As cyber-attacks become more sophisticated, many security technologies have been developed to help organizations enhance cyber situational awareness. Intrusion detection and prevention systems (IDS/IPS) are widely deployed to monitor the organizations' network traffic and to be alert to possible intrusion attempts, policy violation, or malicious activities. Firewalls are also widespread network security systems that inspect the incoming and outgoing network connections and allow or deny them using one or more sets of rules. With the disparate technologies applied in the network, cyber-attack analysis is a human-in-the-loop process, in which cyber analysts (shortened as "analysts" in the remaining part) identify malicious network activities by analyzing the data sources (e.g. IDS alerts and firewall logs). For example, analyzing firewall logs and IDS alerts in a correlated way can help the analysts identify false alarms.

Cyber-attack analysis places high cognitive load on individual analysts for information foraging and analytical reasoning [2]. The amount of data generated by the network security technologies is enormous and among which are abundant in false alerts [3]. 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 identify the "true signals" and to correlate the network activities in depth to gain insights into cyber threats [2]. Furthermore, nowadays more attackers would conduct coordinated activities to achieve their malicious goals at different points of time and/or different segments of a network. Detecting such attacks requires analysts' ability to reason under uncertainty and connect their insights generated in the analysis process ("connect the dots").

Potential Benefits of Studying the Analysis Processes

As cyber-attack analysis is a very complex cognitive task, it requires analysis to draw on their knowledge and experiences, which involved various cognitive activities in its accomplishment. There are several important benefits for capturing and analyzing the cyber-attack analysis processes of analysts and these benefits can be best illustrated in the following example.

A Motivating Example. Figure 1 shows an organization's network and the data sources with more than 200 IDS

alerts, and 100,000 Firewall logs generated in 10 minutes. To detect the attack events, Alice, an analyst, started with browsing the IDS alerts. She first noticed some alerts about SMB (Sever Message Block) buffer overflow attempts on the DNS server inside the network, which made her generate a hypothesis about DNS attack. Meanwhile, she thought another possibility could be that the alerts were caused by normal DNS updates. To investigate the possible DNS attack, she further turned to the firewall logs to check whether there was any suspicious network connection to the DNS server. Instead of finding any suspicious DNS connection, she noticed some connections were built between the inner-network IP addresses and outside IP addresses using the port "6667". According to her domain knowledge, port "6667" is usually used for IRC (Internet Relay Chat) service and could be exploited by a botnet. To verify it, she went to the IDS alerts and found the alerts about the IRC connections between the same set of IP addresses. This observation strengthened her hypothesis that the IRC connections indicate the botnet traffic. After that, she easily linked her previous hypothesis about the DNS attack to this one, believing the DNS attack as a pre-step of the botnet traffic. Finally, she wrote an incident report about the attack.

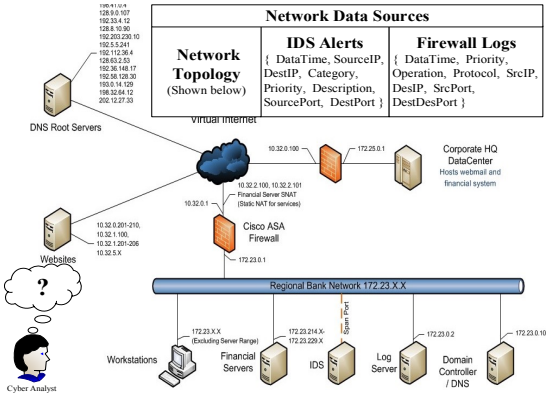

Figure 1. An analyst is performing a cyber-attack analysis task. The data is from VAST Challenge 2012 [4].

Analyzing Alice's analysis process, we can know the steps Alice took in exploring the data sources, generating and verifying hypotheses, and "connecting the dots". Several benefits for understanding an analyst's analysis process can be illustrated by this Example. Firstly, knowing how Alice generates a hypothesis of an attack event is helpful for understanding her conclusion about the event in her report and for assessing how valid and reliable the conclusion is, which enhances the **accountability** of decision making. Secondly, through reflecting on the reasoning process, Alice may realize there is no sufficient evidence showing that the DNS server attack is a pre-step of the botnet traffic, which means the her original hypothesis about the relationship between the DNS attack with botnet traffic may be invalid. She relied too heavily on her first finding (i.e. the suspicious connections to the DNS server), which is referred to as anchoring bias in cognitive science [5]. Therefore, lessons can be learned through **reflecting** on the analysis process and thus improving Alice's performance over time. Thirdly, if Alice was working together with other analysts, knowing Alice's analysis process could help others better comprehend her observations and hypotheses of attack events so that they can better **collaborate** to counter cyber-attacks, especially the multistep cyber-attacks that may last a long period of time on a large number of machines. Fourthly, we can extract some **practical knowledge and experience** from Alice's analysis process. For example, the suspicious IRC connections were found based on the port of interest (i.e. port "6667"); and Alice strengthened her hypothesis of malicious IRC connections by seeking for consistent evidence in different data sources. The knowledge and experience can be transferred into **cognitive aids and tools** for improving the agility of cyber defense. Besides, understanding the analysis process of the analysts made it possible for the comparison between high-performance analysts and less experienced ones so that their differences can be recognized and better **training** tools can be developed to address these differences.

Need for Capturing Fine-Grained Cognitive Processes

Given the important potential benefits, it's highly desirable to capture and analyze the analysts' fine-grained cognitive processes of cyber-attack analysis. The term "fine-grained cognitive process" refers to a cyber-attack analysis process involving the following detailed activities: (1) ***actions*** performed by the analyst for filtering the data sources and for searching for evidence of attack events; (2) the analysts' ***observations*** of suspicious network traffic and events; (3) the analysts' ***hypotheses*** about suspected the attack events that are generated based on the existing observations.

The reason why we need to focus on the fine-grained cognitive process comes from the characteristics of cyber-attack analysis tasks. First of all, cyber-attack analysis tasks are by nature data-driven and labor-intensive [6], considering that most network technologies are prone to data flooding [3]. Handling a large amount of data sources to identify the attack events plays a dominate role in analysts' analysis process [2, 7]. It is necessary to capture the ***actions*** that an analyst explores the data sources, including searching, filtering and aggregation [8]. Meanwhile, it is also important to know what ***observations*** the analyst obtain by data exploration, what he/she knows from the observation, and how he/she evaluates the cyber situation (i.e. the analyst's ***hypothesis*** of the attack events) [9]. Secondly, 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 [6]. How analysts think and choose actions mainly depends on the current analysis process (i.e. the observations they have gained after conducting the previous actions). Specialized knowledge and experience is needed to generate a reasonable hypothesis or to choose an appropriate action under a certain context. Therefore, it is necessary to highlight the connections between an analyst's

hypotheses about attack events and his/her data exploration actions and observations, and it again requires fined-grained capturing of cognitive activities.

Related Work

Cognitive Task Analysis (CTA) is a family of methods that "*help researchers understand how cognition makes it possible for human to get things done and then turning that understanding into aids—low or high tech—for helping people get things done better*" [10]. There are multiple techniques used by CTAs, including interviews (e.g. unstructured/semi-structured/structured interviews [11]), observations (e.g. shoulder surfing), self-reports (e.g. questionnaires, surveys and diaries), process tracing (e.g. eye-tracking, think-aloud protocols [12, 13]). Each technique has its own advantages and disadvantages [10, 14]. Wei and Salvendy carefully compared these advantages and disadvantages and provided 11 useful guidelines in selecting CTA techniques according to a particular task [15].

Several CTAs have been conducted for cyber-attack analysis, which provide a basis for this research. Priolli and Card proposed a notional model of sensemaking process that involves leverage points dealing with data overload and attention management, based on interviews and think-aloud protocols with intelligence analysts [9]. D'Amico and Whitley described how the data sources are transferred into situation awareness in the cognitive process of computer network defense (CND) analysis and identified three stages of CND analysis in a generalized CND workflow, using interviews, observations and self-reports [16]. 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 [7].

Most CTA studies give description of the macro-level cyber-attack analysis processes (e.g. classifying analysts' activities and identifying them in the task workflow), which provides valuable insights into the analysts' cognitive activities in these processes. However, few of them capture the fine-grained cognitive activities and the contextual links among them due to several unique challenges of conducting CTA studies of cyber-attack analysis.

First of all, analysts are fully concentrated on accomplishing their cyber-attack analysis tasks and are sensitive to interruptions [7]. Using common think-aloud protocols during the task could be an unacceptable 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 [10]. However, analysts may be unable to give a complete and accurate report of their own cognitive processes [17], especially for the experienced analysts [18]. Besides, analysts may be unwilling to self-report because they are totally engaged in their analysis tasks. The second challenge for conducting CTA of cyber-attack analysis is that the time schedule of professional analysts is pretty tight, 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 [19]. It could be difficult for them to participant in interviews. Thirdly, conducting CTAs of cyber-attack analysis processes raises several important practical issues, including the accessibility to analysts and an organization's network. Researchers outside an organization have very limited access to the real data sources and attack events as most of them are kept confidential.

Our Method: An Integrated Computer-Aided CTA

Due to the difficulties of conducting CTA studies for cyber-attack analysis, there is a gap between the need for capturing fine-grained cognitive processes of cyber-attack analysis 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 tracking the analysis process by integrating ***automated capture*** and ***situated self-reports*** in a novel way. A computer tool is developed by us to support tracking the analysis processes of analysts when they are concentrated on their cyber-attack analysis tasks. An experiment is designed to collect traces of analysts' cognitive processes in which each participant is asked to perform a simulated cyber-attack analysis task by working with the tool. This experiment setting is completely isolated from the real organizational network so that organizations' concern of confidentiality is dispelled. Meanwhile, the experiment is carefully designed to ensure that the participants' performance in our cyber-attack analysis task is close to their performance in their real-world jobs. Obtained IRB approval for conducting the proposed CTA study, we recruited thirteen professional cyber analysts and seventeen doctoral students specialized in cyber security. We collected thirty traces of their cyber-attack analysis processes and analyzed these traces using qualitative data analysis method.

The contributions of this work are two-fold. Theoretically, our study shows that fine-grained CTA can be conducted to study human's cognitive processes in an analytical reasoning task like cyber-attack analysis, and gains insights into the fine-grained cognitive processes from which human's procedure knowledge and experience can be elicited. Practically, our method may guide CTA researchers on how to effectively conduct CTA within a field where participants' time and confidentiality are set at a high value. The preliminary results provide a starting point for developing technologies to lever-age the captured processes to improve analysts' performance.

AN INTEGRATED COMPUTER-AIDED CTA DATA COLLECTION METHOD

We develop our method on the basis of three principles. Firstly, it should be able to capture the fine-grained

cognitive processes of cyber-attack analysis. Secondly, it should avoid adding too much workload and distracting their attention from the cyber-attack analysis task. Thirdly, we want to ensure the captured cognitive processes reflect the analysts' performance in the real-world cyber-attack analysis tasks.

Figure 2 shows that our method has three building blocks: (1) a trace representation which describes the fine-grained cognitive process of cyber-attack analysis, (2) a computer tool which is developed based on the trace representation to trace analysts' cognitive processes, and (3) an experiment in which participants are asked to work with the tool to accomplish a cyber-attack analysis task. With three building blocks, this CTA method enables us collect traces of analysts' cognitive processes in performing a cyber-attack analysis tasks, without scheduling time-consuming interviews. We can analyze the collected traces after the experiment to reveal the participants' original cognitive processes in the task. Next, we explain the method in detail by describing the three building blocks.

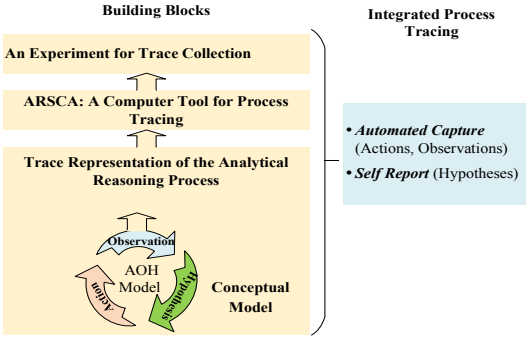

Figure 2 The framework of the CTA method.

Trace Representation

A cognitive process of cyber-attack analysis can be modeled as an analytical reasoning process, which is "*central to the analysts' task of applying human judgments to reach conclusions from a combination of evidence and assumptions.*" [20]. More specifically, given the network and the network data sources, an analyst conduct ***actions*** (e.g. searching and filtering) to explore the data; an action may result in an ***observation*** of suspicious network activities; based on the observation, the analyst may generate a ***hypothesis*** (e.g. a new insight into or an in-depth understanding of an attack event). To further investigate this hypothesis, the analyst may need to conduct a new action. Therefore, actions, observations and hypotheses form a iterative cycle [21, 22].

Based on the conceptual understanding, an analyst's analytical reasoning process is a process where actions, observations and hypotheses are created, modified and connected. These changes are caused by the operations performed by the analyst when he/she is accomplishing a task. We classify the operations into 11 categories based on the literature of existing CTA studies in cyber-attack analysis [16] and the comments from domain experts. The types of operations are shown in Table 1.

With the specified operation types, we define a trace as a sequence of operations in a time order. Each operation is performed under a certain context. Therefore, a trace can represent a complex and non-linear analytical reasoning process. Definition 1 gives the formal definition of a trace.

Table 1 11 types of operations conducted by analysts.

Operation	Description
BROWSE	BROWSE(D_i), $D_i \subseteq \cup DS_i$*: Browse the data sources.
FILTER	FILTER($DS_i, Cond.$): Filter the source data DS_i based on condition $Cond.$
SEARCH	SEARCH(D_i, K), $D_i \subseteq \cup DS_i$: Search K in data D_i.
INQUIRE	INQUIRE(T_m): Inquire about a term T_m
SELECT	SELECT(D_i), $D_i \subseteq \cup DS_i$:Select the data of interest in D_i .
SELECTED*	*(Come in pairs with SELECT) SELECTED(D_i), $D_i \subseteq \cup DS_i$: The selected data of interest
LINK	LINK(D_i, L), $D_i \subseteq \cup DS_i$: The links L among the selected data D_i (e.g. common features in D_i)
NEW_HYPO	NEW(h, O): Generate a *hypothesis* h in the context of *observation* O.
MODIFY	MODIFY(h, v_1, v_2): Modify the content of an hypothesis h from v_1 to v_2
SWITCH CONTEXT	SWITCH_CONTEXT(h_1, h_2): Change current focus of attention from *hypothesis* h_1 to *hypothesis* h_2.
CONFIRM/ DENY	CONFIRM_DENY($h_1, Y/N$): Confirm or deny an *hypothesis* h_1.

*The data set is $\mathbf{D} = \cup \boldsymbol{DS_i}, \boldsymbol{DS_i}$ is a data source.

Definition 1: A cognitive trace $\mathcal{T}\mathbf{r}$ is a sequence of items $(\mathbf{p_1}, \dots, \mathbf{p_n}), \forall \mathbf{p_i}\ (\mathbf{1} \le \boldsymbol{i} \le \boldsymbol{n})$, $\mathbf{p_i}$ is a tuple $(\boldsymbol{t_i}, \boldsymbol{op_i(I, C_i)})$, where $\mathbf{t_i}$ is the timestamp, $\boldsymbol{op_i(I, C_i)}$ is an operation on a cognitive activity $\boldsymbol{I}$ under the context $\boldsymbol{C_i.I}$ is an action, observation or hypothesis, $\boldsymbol{C_i}$ is a set of connections between $\boldsymbol{I}$ with the existing actions, observations and hypotheses.

ARSCA: The Computer Tool for CTA

Based on the trace representation, a computer tool named **ARSCA** (Analytical Reasoning Support system for Cyber Analysis) is developed by us to trace the fine-grained cognitive processes of analysts. The process tracing supported by ARSCA integrates both automated capture and situated self-reports so that they can reinforce each other, thus reducing the distraction and analysts' extra work load to analysts' analysis task [23].

Design Principles

The design principles of this tool are as follows:

- To capture analysts' fine-grained cognitive processes, the tool should be able to support analysts to perform all types of operations shown in Table 1. The reaction time of the tool should be short enough so that users can hardly notice it.
- The tool should support the integrated process tracing that minimize its distraction and analysts' work load.
- The tool should be easy for analysts to learn and use so that analysts can get familiar with it in a short training session.

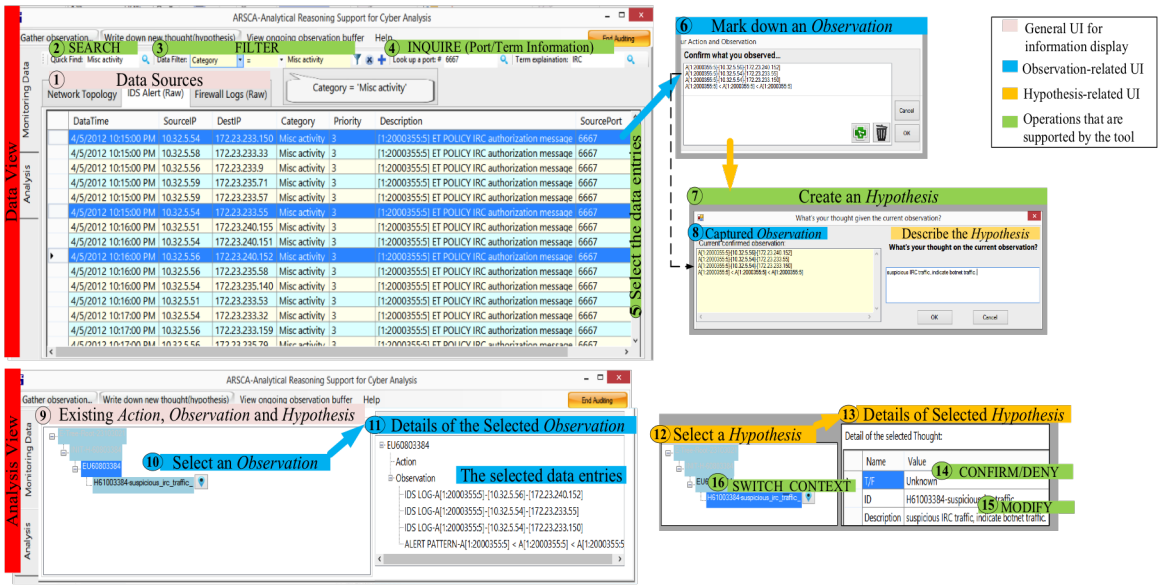

Figure 3 The ARSCA User Interface with the main components in the Data View and Analytical View

Implementation

ARSCA is developed as a Windows platform application based on the above principles. It has two views: Data View and Analysis View. Figure 3 illustrates its User Interface (UI) with the main components in these two views.

Functionalities and Features

Tracing Actions (Automated Capture). ARSCA enables analysts to manipulate the data sources (shown in the Data View) in multiple ways, thus supporting the pre-defined operations including SEARCH, FILTER, and INQUIRE (Region 2, 3 and 4). ARSCA records these operations with an analyst's inputted keywords or filtering conditions.

Tracing Observations (Automated Capture). An analyst can select the data of interest (Region 5) and the selected data are displayed in another window (Region 6) so that he/she can review and confirm them. ARSCA automatically recorded the selected data as one observation of the analyst.

Tracing Hypotheses (Situated Self-reports). Once an analyst generates a new hypothesis about an attack event based on an observation, he/she can write down it (Region 7 and 8). ARSCA records not only the hypothesis but also its connections to the relevant observations.

Process Navigation. ARSCA displays the analyst's existing actions, observations, and hypotheses in the Analysis View (Region 9). If a hypothesis is created based on an observation, ARSCA uses a nested structure to show the relationship because analysts are familiar with this display in their work places. Analysts can select each of them to see the details. Region 11 shows the details of the selected observation in Region 10, and Region 13 shows the details of a selected hypothesis in Region 12, including its truth value and description. ARSCA enables the analyst to modify the truth value and the description, thus supporting the pre-defined operation COMFIRM/ DENY and MODIFY. ARSCA also enables the analyst to change his/her current focus from one hypothesis to another by double clicking the targeted hypothesis, thus supporting the operation SWITCH _CONTEXT. After changing the focus, ARSCA will link the newly captured action and observation of the analyst to the targeted hypothesis.

Trace Output. When the analyst performs an operation, the UI events (e.g. keystroke inputs and mouse clicks) will trigger ARSCA to record the details of the operation in the background. These operations are recorded in a trace in temporal order. Once a task is finished, ARSCA outputs the captured trace in a XML file. An example of ARSCA's output is shown in Table 2.

Rationale of Integration

The process tracing supported by ARSCA integrates automated capture and self-reports, both of which have advantages and disadvantages. As for automated capture, it obtains precise quantitative data unobtrusively and the records can give useful insights into analysts' behaviors [24]. However, it can't capture analysts' comments and in-depth thoughts. Besides, the collected data don't contain any contextual information. If there are no other data provided, it will be difficult for the follow-up analysis which in turn may limit the data's utility. As for self-reports, it can gain quality data, but it relies on participates' willingness to report.

Integrated in ARSCA, the automated capture and situated self-reports can mutually strengthen. 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 contextual information for the hypothesis. Moreover, the Analysis View (i.e. Process Navigation) helps analysts reflect on their previous analysis process and organize their thoughts. Therefore, analysts are encouraged

to report their findings and thoughts emerged 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).

ARSCA Testing

Table 2 RUI and ARSCA record a same set of operations.

Operations	Filter the firewall logs based on condition "protocol='TCP'".
RUI Log	1 ▪ Elapsed Time Action X Y 2 ▪ 2014-05-24T13:24:10.726 Pressed Left 450 147 3 ▪ 2014-05-24T13:24:12.390 Pressed Left 584 74 4 ▪ 2014-05-24T13:24:13.959 Key T Shift 5 ▪ 2014-05-24T13:24:14.228 Key C Shift 6 ▪ 2014-05-24T13:24:14.325 Key P Shift 7 ▪ 2014-05-24T13:24:15.874 Pressed Left 716 79 8 ▪ …
ARSCA Trace	9 ▪ <Item Timestamp="05/24 13:24:15"> 10 ▪ **FILTER** (SELECT * FROM Task2Firewall WHERE 11 ▪ Protocol = 'TCP', Task2Firewall) 12 ▪ </Item> 13 ▪ …

ARSCA has been tested in a pilot study. We ran ARSCA and performed all the pre-defined operations. Meanwhile, we ran an existing keystroke logging tool called RUI (Record User Input) as a reference. RUI (version2.3) is a well-tested tool that can record the interface behaviors and runs in the background [25]. Running together with ARSCA, RUI recorded our low-level interactions (i.e. keystrokes and mouse clicks) with ARSCA while we were performing the operations using ARSCA. Table 2 provides a short sample of RUI logs, and one item in ARSCA trace, which is a pair of timestamp and operations. The "FILTER" operation corresponds to the RUI logs in line 2-7.

In total, we performed 126 operations, including 15 FILTER, 22 SEARCH, 15 INQUIRE, 17 SELECT, 10 LINK, 13 NEW, 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 are 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.

Experiment: Tracing Cyber-Attack Analysis Processes

Due to the concern of organization's confidentiality, we design a laboratory experiment to collect traces of the cyber-attack analysis processes. The goal of the experiment is to collect traces that represent analysts' cognitive processes in their real-world cyber-attack analysis jobs. Next, we explain how we recruit participants and design the experiment and the experimental task to achieve this goal.

Recruitment

After obtaining the IRB approvals, we recruited thirteen full-time professional cyber analysts in collaboration with Army Research Lab and seventeen doctoral students specialized in cyber security from our University. Although the doctoral students are less professional, they all have sufficient domain knowledge and experience in cyber security analysis.

Experiment Design

Pre-task Questionnaire. Before beginning a task, a participant needs to complete the pre-task questionnaire that takes about 5 minutes. The questions focus on demographic factors, domain knowledge and expertise of cyber-attack analysis, familiarity with VAST Challenge 2012, and the participant's current mental and physical status.

Tutorial Session. Considering the potential influence of the participant's proficiency with the tool on his/her performance of the task, we design a tutorial session to teach the participant about how to use ARSCA. At the end of the session, we require the participant to pass a quiz to make sure that the participant is familiar with the tool before undertaking the task.

Conducting the Task. After the tutorial session, we task to the participant to work with ARSCA to conduct a cyber-attack analysis task. Within the one-hour limit for conducting the task, participants can end the task at any time when they finished their analysis. The task will be described later.

Post-task Questionnaire. After each task, we ask the participant to complete a post-task questionnaire that takes about 15 minutes. It includes both open-ended questions and closed-ended rating questions. The open-ended questions ask the participant to report the key observations and hypotheses and explain how they gain them, which can be used to confirm the information captured in the traces. The close-ended rating questions use a five-point Likert scale and ask the participant's opinions on multiple aspects of the experiment setup and his/her performance in the task. We will discuss the questions in detail in the next section. Knowing the ground truth of the task, we can assess the participants' performance according to their answers to the post-task questions and the collected traces.

Follow-up Trace-Stimulated Recall (not required). Several participants may be invited by us to participate in a trace-simulated recall study. It is not required and depends on their performance in the task and their time schedule. The details will be discussed in the next section.

Experimental Task

The cyber-attack analysis task in our experiment should be carefully designed to ensure the quality of the traces to be captured. More specifically, the task should be of reasonable complexity for our participants, which is neither too difficult nor too easy. The data sources provided in the task should be close to the real-world data sources with

respect to the volume and complexity of noise. Meanwhile, we need to control the size of the data set to ensure the participants can finish the task within a specified time frame.

We designed our task by partially adopting the network dataset from VAST Challenge 2012 Mini Challenge 2 produced by the Visual Analytics Community [4]. The VAST dataset are of high quality [26], containing 23,711,341 firewall logs and 35,948 IDS alerts. It also provides a realistic cyber-attack scenario which is a multiple cyber-attack taking place within 40 hours on the network of an organization containing approximately 5000 hosts. Figure 1 shows the network topology and the data sources: firewall logs and IDS alerts. However, the original dataset is too huge to be analyzed by the participants in the given time. To reduce the workload, we need to select a small portion from the original dataset. Given the ground truth (i.e. the original attack events occurring during the 40 hours), we extract the data that corresponds to a critical 10-minute period with three attack events. Finally, the data sources of our task include 239 IDS alerts and 115,524 firewall logs. There are three obvious attack events underlying the data sources: (1) IRC communication between the inside workstations with a set of outside Command and Control (C&C) severs, (2) denied FTP connection attempts for data stealing, and (3) successful SSH connection for data stealing.

PRELIMINARY RESULTS OF TRACE ANALYSIS

We collected thirty traces by conducting the experiment. We analyze the traces as well as the participants' post-task questionnaires with the aim of evaluating our CTA method and gaining a basic understanding of the participants' cognitive processes of cyber-attack analysis.

Use Traces to Describe Cognitive Processes

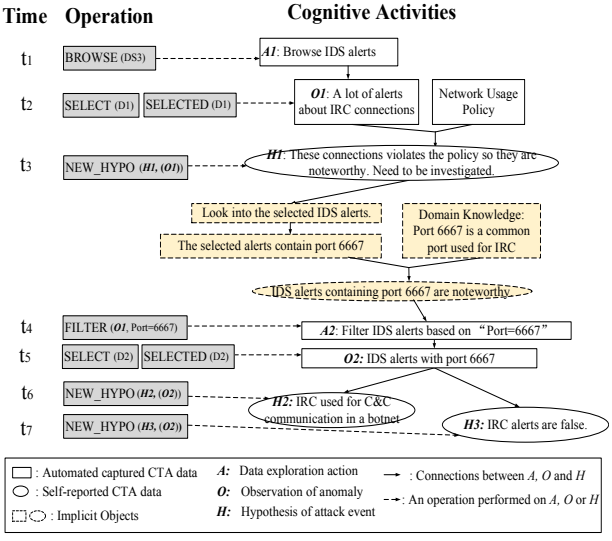

Figure 4 We use P1's trace to explain his analysis process.

We use the collected traces to explain the participants' cognitive processes to investigate what has been captured in the traces. The unit of analysis is the trace item, which is a pair of timestamp and operation. We analyze the recorded operations and explain the reason why the participant conducted one operation after another.

Our analysis of the trace starts with the activities that are explicitly recorded in the trace, such as the descriptions of *Hypotheses*. 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 has the domain knowledge that "*IRC can be used by malware*" and he/she generated this hypothesis based on this knowledge. However, the explicit cognitive activities can't explain the analytical reasoning process completely, so we need to infer the implicit activities that are not recorded in the trace. These activities can be inferred by explaining the sequence of operations.

Figure 4 shows an example of the partial trace of participant "P1". The partial trace is demonstrated as a sequence of operations in the simplified representation. It shows that P1 first browsed the IDS alerts at time t_1, and then selected a set of alerts about IRC connections and confirmed them as an observation (O1). Based on the O1, he generates a new hypothesis (H1) about policy violation. We say that network policy is an explicit observation because it is mentioned in H1. Following this operation, the trace shows that P1 filtered the IDS alerts based on "Port=6667". We notice that some connections via port "6667" have appeared in observation O1. So we can infer that P2 used this filtering condition because he thought the IDS alerts containing port "6667" are worthy of investigating. The reason why he suspected these alerts could be that he has the domain knowledge that port "6667" is a common port used by malicious C&C communication. Therefore, we created the implicit objects between the NEW_HYPO operation and the FILTER operation to explain why P1 conducted the filtering after generating the hypothesis.

After analyzing the 30 traces in the same way, we demonstrate two typical cases extracted from the traces of two participants who are professional analysts, in which experience plays an important role.

Case 1: "Following up a Clue"

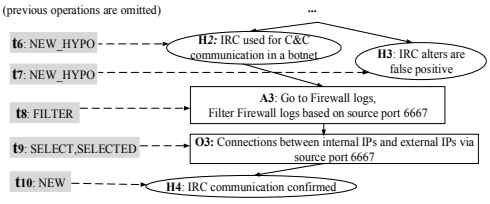

Figure 5 P1 found suspicious IRC connections in IDS alerts, and he checked for the relevant data in firewall logs.

This case is still from participant P1 and the selected part is the operations following those in Figure 4. As shown in Figure 4, P1 filtered the IDS alerts based on "Port = 6667"

and found that a lot of IDS alerts report the network connections via port 6667 (O2 in Figure 4). Based on the observation, he generated two alternative hypotheses, which are H2 and H3 (shown in both Figure 4 and Figure 5). To investigate H2, he shifted to the firewall logs for more evidence. He filtered the firewall logs based on the same condition "source port = 6667" (A3), and observed that the connections in the filtered firewall logs were also from internal IP addresses and external IP addresses (O3). So it strengthened his previous hypothesis that malicious IRC communication exists in the network. In this case, when the participant found the suspicious connections in the IDS alerts, he continued to track down such connections in firewall logs ("following up a clue"). Using this strategy, he successfully correlated the evidence in IDS alerts with those in firewall logs and confirmed his hypothesis.

Case 2: "Proceeding From One Event to its Associated Events"

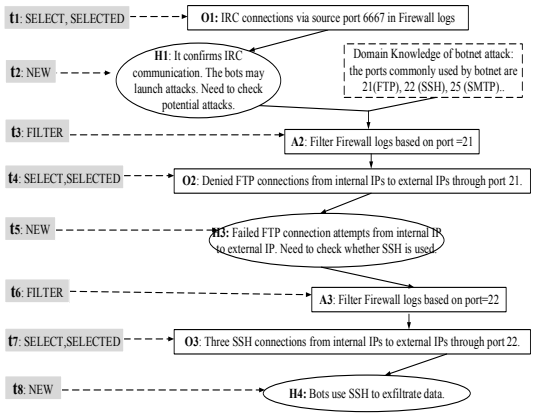

Figure 6 When P8 confirmed his hypothesis of an attack event, he speculated on the associated attack events and further investigated them.

Figure 6 demonstrates the partial trace of the participant P8. P8 first confirmed his hypothesis about IRC communications in a botnet (H1) after gaining the observation O1. According to his following operations (i.e. filtering firewall logs by port), we can infer that he knew that port "21", "22" and "25", which correspond to the service FTP, SSH and SMTP respectively, are often used by botnet based on his experience knowledge. Therefore, after confirming the existence of the botnet, he suspected that the bots conducted some malicious activities through the commonly used ports, and then investigated them one by one. He first filtered the firewall logs based on "port = 21" (A2) and found denied FTP connections from internal IP addresses to external IP addresses through port "21" (O2). Then, he knew some malicious FTP attempts indeed existed but were failed. He expected those bots would choose another way, and decided to investigate the connections using SSH (H3). He filtered the firewall logs based on port = "22" (A3), and found that three SSH connections were successfully built between internal IP addresses and external IP addresses (O3), which is very subtle. So he generated a hypothesis that the bots exfiltrate data to outside C&C servers using SSH.

In this case, after the participant confirmed his hypothesis about the IRC communication in a botnet, he further explored the follow-up attack events, and then gained the two key observations (i.e. failed FTP connections and successful data exfiltration using SSH). It largely relies on his experience which is obtained from long-term on-the-job training.

Traces Capture the Key CTA Data

We expect that the process tracing can capture the evidence and thoughts considered important by the participants in their analysis processes. In the post-task questionnaire, participants were asked to answer four open-ended questions: (1) **IMP_OBS**: *"Reflecting back, what are the 3 most important evidence that you observed in the data that contributed to your conclusion?"* (2) **FD_OBS**: *"Please explain how you find the above evidence."* (3) **IMP_HPY**: *"Reflecting back, what are the 3 most important thoughts in your mind that contributed to your conclusion?"* (4) **EVTS**: *"Based on your analysis, please create one or more narratives that describe the events on the network (i.e. tell the storyline of the potential events)".*

Given participants' answers to these questions, we checked whether the information in the answers had been captured in the traces. We need to analyze the content 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 generates a preliminary coding. After that, the evaluator proofreads the coding and revises it if 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 7 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. Different themes, which corresponding to different attack events, are marked in different colors. 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" are captured in traces. As for "FD_OBS", 5 themes in its answers are not captured by the trace. The

themes are about either the domain knowledge used (e.g. "*some ports are always 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 cyber-attack analysis.

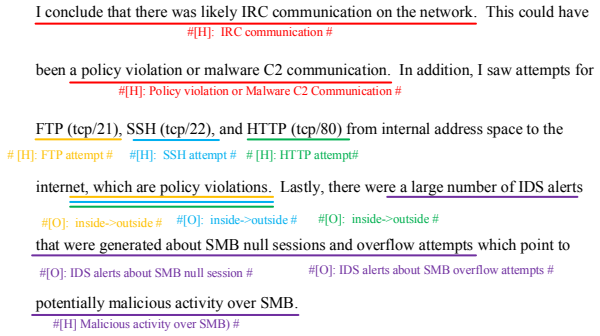

Figure 7 Extracting themes from the text of an answer to EVTS

CTA Data in Traces Complement Each Other

There are two types of CTA data in traces: the automatically captured CTA data and the self-reported CTA data. Take the trace item in Table 2 as an example. The "FILTER" operation belongs to ***the automated-captured CTA data*** because ARSCA recorded this filtering action automatically. In the 30 traces, there are balanced number of automatically captured items and self-reports: 151 items are collected by automated capture (mean=5.59, SD=3.05), and 180 are collected by self-reports (mean=6.67, SD=5.07).

In the previous two cases, the automatically captured CTA data are represented by rectangles, and the self-reported CTA data are represented by ellipses. The automatically captured and self-reported CTA data are complementary in the following ways, which enables us gain deep understanding of the participants' cognitive processes based on the traces.

Automatically Captured Action Describes Automatically Captured Observation: In Case 1, the participant first filtered the IDS alerts based on the condition "Port = 6667" at time t1, and then he selected a set of filtered alerts as his new observation at time t2. The filtering action implies that the selected alerts satisfy the filtering condition ("Port = 6667"). Therefore, the action at t1 provides us more information about the observation at t2. Without the first action, we would not be sure about the feature of the captured observation. The same cases are the action and observation at t5 and t6 in Case 1, t3 and t4, t6 and t7 in Case 2.

Automatically Captured Observation Provides the Contextual Information for Self-Reported Hypothesis: When analyzing the trace of Case 1, we learn that the participant confirmed his hypothesis about IRC network connections and reported this finding (H4) after finding the evidence in both IDS alerts (A2, O2) and Firewall logs (A3, O3). These actions and observations, which are automatically captured, provide the contextual information of the self-reported hypothesis, so that we can know under what situation the participant confirmed his hypothesis. The role of observation is also confirmed by the participant's comment in the post-task questionnaire:

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

Other actions and observations succeeded by a self-reported hypothesis in Case 1 and Case 2 also have the same role.

Self-Reported Hypothesis Explains the Motivation of the Subsequent Automatically Captured Action: In Case 1, the participant switched his focus form IDS alerts to the firewall logs and filtered the firewall logs based on the same condition as he did in IDS alerts (A3). The self-reported hypothesis (H2) between action A2 and A3 explains the motivation of A3, which is to find supporting evidence. The role of the in-between hypothesis is more significant in Case 2. The hypothesis H1 explains how the participant came up with the insight to investigate FTP network connections (A2) via port 21 when he gained the observation of IRC connections (O1). Without H1 reported by the participant, it would be hard for us to guess the reason why he performed the following action.

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 [27] by inviting participants to recall their previous cognitive processes with the aid of their traces. The procedure of trace-stimulated recall is as follows: we ask a participant to go through his/her trace by explaining each operation as what we taken before. After obtaining his/her self-explanation, we confirm our original understanding of his/her analysis process.

At this stage, only the professional participants are required to take the recall interview for the reason the their cognitive processes are more typical of the real cyber-attack analysis processes than the doctoral ones. However, it's not feasible for us to invite all the 10 professional analysts, because the recall interview requires much more time compared with the experiment we did. Therefore, we chose one representative, P1, from the professional analysts who have participated in our experiment. P1 is invited because he successfully identified all the attack events in the task and his trace is more representative than other professional analysts.

Check Trace-based Explanation with Self-Explanation

The result of the trace-stimulated recall with P1 shows that his self-explanation is consistent with our finding. The

following is an example of his comments on his decision-making in filtering IDS alerts (which is illustrated at t_4 in Figure 4):

> "*I saw strings within the IDS 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 filtering activity was based on his domain knowledge and his observation of a large amount of alerts with port 6667 (which are represented in the dotted boxes in Figure 4).

Moreover, it is founded 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 trace shows that P1 never went back to the previous hypothesis about DNS attack. In the recall study, P1 gave the reason why he didn't further explore this hypothesis: this hypothesis is not generated on his serious thinking and therefore he quickly gave up this hypothesis after a second thought but he failed to inform the tracing tool. This finding indicates that traces may not record all the thoughts emerged in the analysts' mind. Although it is impossible to capture every thought occurred in the participant's cognitive process, traces at least can provide some clues and enlightenments for interpretation which are also useful for further analysis.

DISCUSSION

Our preliminary trace analysis reveals the feasibility of understanding the analysts' cognitive process by analyzing the traces. We show that the traces not only capture the key cognitive activities but also may imply analysts' knowledge and experience. Besides, traces have the reliability in that the automated-captured and self-reports CTA data in traces can confirm and complement each other.

This work has several limitations. Firstly, some subtle thoughts occurred in a participant's mind (e.g. quickly-denied thought) might fail to be captured in the traces (implicit thoughts), which may add additional difficulty to researchers in explaining the successive operations in the traces. Regardless of it, some clues of implicit thoughts recorded in traces can enable the researchers to make an educated guess/inference on them.

Secondly, 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 schedule of cyber-attack analysts, we try to minimize analysts' work load in our CTA data collection study, however, further efforts on trace analysis are needed as the interpretation of the operations in traces is also a complex cognitive process. Fortunately, based on our preliminary trace analysis, we've observed some common patterns of sequential operations in traces. Therefore, it is highly likely to generate some guidelines or a procedure, even an automated tool for trace analysis.

Furthermore, we only confirmed our understanding of one participant's trace in the stimulated recall interview, which is not enough to conclude that the understanding gained from trace is similar to the original cognitive processes. We definitely need to conduct more recall study. However, at least we know the key cognitive activities are captured in the traces. Based on our analysis of other traces of both professional analysts and doctoral students, we did observe that there is a diversity of the strategies used by the participants for data exploration and hypothesis generation. In the future, we can identify and 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.

CONCLUSION

We proposed an integrated computer-aided CTA data collection method and conducted an experiment to capture the traces of participants' fine-grained cognitive processes in accomplishing a cyber-attack analysis task. The results of our preliminary trace analysis indicate that this method may be a feasible way to gain understanding of how analysts perform cyber-attack analysis tasks, and it may have such promising contribution to analysts' as self-reflection, elicitation of their procedure knowledge or new tool development.

ACKNOWLEDGEMENT

Supported by ARO Grant W911NF-09-1-0525 (MURI).

REFERENCES

[1] K. Lab, *Monthly Report on Online Threats in the Banking Sector*, Kaspersky Lab, 2014.

[2] A. D'Amico, K. Whitley, D. Tesone, B. O'Brien, and E. Roth, "Achieving cyber defense situational awareness: A cognitive task analysis of information assurance analysts." pp. 229-233.

[3] H. Debar, and A. Wespi, "Aggregation and correlation of intrusion-detection alerts." pp. 85-103.

[4] "VAST Challenge 2012 Mini-Challenge 2," Visual Analytics Community, 2012.

[5] J. S. Hammond, R. L. Keeney, and H. Raiffa, "The hidden traps in decision making," Harvard Business Review, vol. 76, no. 5, pp. 47-58, 1998.

[6] P. Barford, M. Dacier, T. G. Dietterich, M. Fredrikson, J. Giffin, S. Jajodia, S. Jha, J. Li, P. Liu, and P. Ning, "Cyber SA: Situational awareness for cyber defense," Cyber Situational Awareness, pp. 3-13: Springer, 2010.

[7] R. F. Erbacher, D. A. Frincke, P. C. Wong, S. Moody, and G. Fink, "A multi-phase network situational awareness cognitive

task analysis," Information Visualization, vol. 9, no. 3, pp. 204-219, 2010.
[8] J. Goldstein, and S. F. Roth, "Using aggregation and dynamic queries for exploring large data sets." pp. 23-29.
[9] P. Pirolli, and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis." pp. 2-4.
[10] B. Crandall, G. A. Klein, and R. R. Hoffman, Working minds: A practitioner's guide to cognitive task analysis: Mit Press, 2006.
[11] R. R. Hoffman, "The problem of extracting the knowledge of experts from the perspective of experimental psychology," AI magazine, vol. 8, no. 2, pp. 53, 1987.
[12] K. A. Ericsson, and H. A. Simon, "Verbal reports as data," Psychological review, vol. 87, no. 3, pp. 215, 1980.
[13] M. W. Van Someren, Y. F. Barnard, and J. A. Sandberg, The think aloud method: A practical guide to modelling cognitive processes: Academic Press London, 1994.
[14] R. E. Clark, D. Feldon, J. Van Merrienboer, K. Yates, and S. Early, "Cognitive task analysis," Handbook of research on educational communications and technology, vol. 3, pp. 577-593, 2008.
[15] J. Wei, and G. Salvendy, "The cognitive task analysis methods for job and task design: Review and reappraisal," Behaviour & Information Technology, vol. 23, no. 4, pp. 273-299, 2004.
[16] A. D'Amico, and K. Whitley, "The real work of computer network defense analysts," VizSEC 2007, pp. 19-37: Springer, 2008.
[17] T. D. Wilson, and T. D. Wilson, Strangers to ourselves: Discovering the adaptive unconscious: Harvard University Press, 2009.
[18] M. T. Chi, R. Glaser, and M. J. Farr, The nature of expertise: Psychology Press, 2014.
[19] Security Operations: Building a Successful SOC, Hewlett-Packard Development Company, hp.com/go/sioc, 2013.
[20] J. Thomas, and K. Cook, "The science of analytical reasoning," Illuminating the Path: The Research and Development Agenda for Visual Analytics, pp. 32-68, 2005.
[21] C. Zhong, D. S. Kirubakaran, J. Yen, P. Liu, S. Hutchinson, and H. Cam, "How to use experience in cyber analysis: An analytical reasoning support system." pp. 263-265.
[22] C. Zhong, D. Samuel, J. Yen, P. Liu, R. Erbacher, S. Hutchinson, R. Etoty, H. Cam, and W. Glodek, "RankAOH: Context-driven similarity-based retrieval of experiences in cyber analysis." pp. 230-236.
[23] J. Y. Chen Zhong, Peng Liu, Rob Erbacher, Renee Etoty, and Christopher Garneau, "ARSCA: A Computer Tool for Tracing the Cognitive Processes of Cyber-Attack Analysis."
[24] S. Westerman, S. Hambly, C. Alder, C. Wyatt-Millington, N. Shryane, C. Crawshaw, and G. Hockey, "Investigating the human-computer interface using the Datalogger," Behavior Research Methods, Instruments, & Computers, vol. 28, no. 4, pp. 603-606, 1996.
[25] J. H. Morgan, C.-Y. Cheng, C. Pike, and F. E. Ritter, "A design, tests and considerations for improving keystroke and mouse loggers," Interacting with Computers, vol. 25, no. 3, pp. 242-258, 2013.
[26] J. Scholtz, M. A. Whiting, C. Plaisant, and G. Grinstein, "A reflection on seven years of the VAST challenge." p. 13.
[27] J. Lyle, "Stimulated recall: A report on its use in naturalistic research," British Educational Research Journal, vol. 29, no. 6, pp. 861-878, 2003.