2019

University of North Carolina Wilmington

Master of Science in

Computer Science and Information Systems

Proceedings

https://csbapp.uncw.edu/mscsis

AN EMPIRICAL STUDY OF FACTORS IMPACTING CYBER SECURITY ANALYST PERFORMANCE IN THE USE OF INTRUSION DETECTION SYSTEMS

William T. Roden

A Thesis Submitted to the
University of North Carolina Wilmington in Partial Fulfillment
of the Requirements for the Degree of
Master of Science in Computer Science and Information Systems

Department of Computer Science

University of North Carolina Wilmington

2019

Approved by

Advisory Committee

_____ _____

_____

Chair

Accepted By

_____

Dean, Graduate School

TABLE OF CONTENTS

ABSTRACT

Cyber security attacks are needles in a haystack. A modest computer network generates over 1,000,000 network events per day, with less than 0.1% of those events involving some sort of malicious action against the network. Human analysts cannot process the sheer volume of information travelling across a network, so organizations use Intrusion Detection Systems (IDS) to alert on abnormal or potentially malicious behavior. However, prior research has shown that it is not uncommon for 99% of IDS alerts to be false alarms. This study seeks to understand to what extent the false alarm rate of IDSes affects human analyst performance. I created Cry Wolf, a simulated IDS web application, to display and capture user responses in triaging IDS alerts. I used Cry Wolf to conduct a controlled experiment wherein 51 participants were divided into two groups, one with a 50% false alarm rate and one with a 96% false alarm rate, and asked to classify whether the alerts were benign, or malicious, in nature. I analyze participants' performance with regard to *sensitivity*, *specificity*, *precision*, and *time-on-task*.

Results indicate the group with the 50% false alarm rate had $\tilde{6}0\%$ higher *precision* and were 39% faster in *time-on-task* than the 96% false alarm rate group. The *sensitivity* of the 96% group approached 100% and the 50% group was also high, around 90%. *Specificity* appeared to be unaffected by the false alarm rate. Expertise appears to play a role in these performance measures, but more data is required to quantify the differences. These results indicate a tradeoff: IDSes that are overtuned and generate excess alarms may actually improve analyst sensitivity in identifying anomalous activity at the price of miscalssifying some false alarms as true alarms. This reflects the industry standard of placing a high priority on high *sensitivity* at the expense of low *precision* and *specificity* for intrusion detection, regardless of the circumstances for individual networks. I believe there is evidence to suggest from these results, and from personal experience, that analysts become comfortable with high false alarm rates as it reinforces what normal activity looks like and highlights abnormal activity.

ACKNOWLEDGMENTS

To my parents, Jerry and Susan: you've never been anything but supportive in all my endeavors. Thank you for all the love and encouragement throughout the years. I'm sure that sometimes it wasn't easy.

To my thesis advisor, Dr. Lucas Layman: I would like to amend a previous statement I made. "I could have done this without you, but it would not have been as good," should be struck from the record and replaced with, "This would not have been possible without you." Thank you for your guidance, encouragement, and patience.

To Dr. Jeffrey Cummings and Dr. Elham Ebrahimi: thank you, both, for always having an open door and for providing invaluable input and feedback throughout this process. You are both a pleasure to work with and your willingness to serve on my thesis committee is very much appreciated.

DEDICATION

To my wife, Jennifer: for your continuous support and encouragement throughout my studies, I cannot thank you enough. You have shouldered a hefty burden for our family with a grace and dignity that few could muster.

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

Cyber security attacks are needles in a haystack. A modest computer network generates over 1,000,000 network events per day, with less than 0.1% of those events involving some sort of malicious action against the network [1]. The sheer volume of information generated is too much for human analysts to process, thus researchers have created Intrusion Detection Systems (IDSes) to help deal with the quantity of information. Nonetheless, malicious events still wreak havoc on systems. Target and Equifax are two high profile examples of companies who have recently fallen victim to attacks that went undetected for some time [2], [3]. A 2018 study by IBM Security and the Ponemon Institute reported that the mean time to identify a data breach was 197 days (Fig 1.1) [4]. A Verizon/US Secret Service study found that 98% of data breaches occurred on network servers, and that 86% of those breaches contained evidence of the breach within the application log files [5] that went undetected despite the evidence being present.

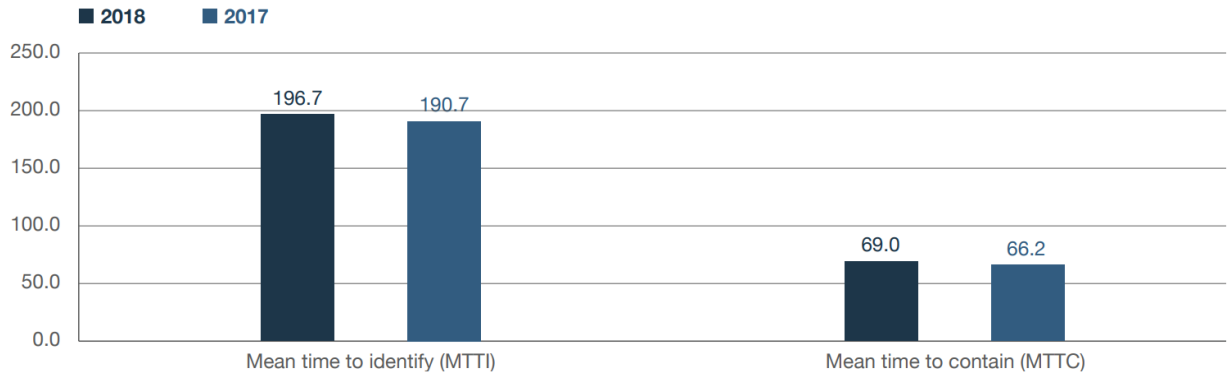


Figure 1.1: Days to identify and contain a data breach, 2017-2018 data [4]

Cyber security analysts have difficult jobs that share several parallels to Air Traffic Controllers: the job is stressful by nature; there are not enough qualified workers in the field; and no one cares about job performance until a mistake is made. These reasons, and many others, are why cyber security analyst burnout is so high [6]. Even so, cyber security analysts report to work day after day to dredge through the monotony of alerts, the vast majority of which are of benign activity, and system logs in an effort to protect the confidentiality, integrity, and availability of the systems for which they are responsible.

One challenge is the effectiveness of IDSes in supporting human analysts. IDSes examine network traffic, application logs, and computer logs against a set of pre-configured rules, then generate *alarms* when a network or host event is deemed to be suspicious [7]. However, it is not uncommon for 99% of IDS alerts to be *false alarms* [8], i.e., an alert generated for a benign network event. These statistics are accepted as the norm because in cyber security, a false positive that gets a second look by human eyes is better than a false negative that goes unnoticed. For example, UNCWs InsightIDR[1] monitoring system generates approximately 100 alarms per day but flags approximately 10,000 "notable behaviors", of which only 2-4 are actual security incidents. The accuracy of human response to alarms is influenced by the alarm accuracy of the system [9]–[11], and overly sensitive alarm systems lead to distrust in the system and psychological stress of their human operators [12]. Increasing both total alarm rate and false alarm rate decreases operator response frequency, accuracy, and speed [13]. While there has been a great deal of research to improve IDS detection capability and alarm accuracy [14]–[16], alarm rates remain unacceptably high. Human security analysts are still the last line of defense, and their needs are often forgotten in the fray.

The goal of this research is to further the scientific understanding of factors that affect cyber security analysts' performance—specifically examining the impact of false alarms generated by IDSes that are tuned to be risk averse. In this research, I will answer: *To what extent does the false alarm rate of an IDS affect an analysts' ability to detect malicious activity?* I will measure *analyst performance* in terms of specificity, sensitivity (a.k.a. recall), and precision of identifying malicious activity among the alarms generated by an IDS. I will also consider the time taken to classify alerts. I will investigate the following hypotheses:

H1. Analyst *performance* decreases as the IDS false alarm rate increases

H2. Analyst *time on task* increases as the IDS false alarm rate increases

[1]https://www.rapid7.com/products/insightidr/

The problems created by the false alarm rate of IDSes is unlikely to be solved in the near future. Much study on the issue and advances in machine learning will be required to bring this rate down to a more manageable level for analysts. This project seeks to aid in that process by better understanding the extent to which the false alarm rate of an IDS affects analysts' performance. I conduct a controlled experiment of analysts interacting with an IDS-like system, and record data to analyze how the participants' ability to detect incidents changes according to different false alarm rates. Identifying the relationship between analyst performance and IDS false alarm rate will help set benchmarks for acceptable IDS false alarm rates, and help organizations understand the trade-off between high alarm rates and the ability to detect actual attacks.

2 LITERATURE REVIEW AND ANALYSIS

The focus of my research is on how cyber security analyst performance is affected by IDS false alarm rates. In this section, I provide background on intrusion detection systems, the work processes of cyber security analysts, and the information they use to determine network attacks. I also review related findings on the *prevalence effect*, *domain knowledge*, *situated knowledge* and other possible influencing factors on analyst performance. This section ends with the motivation for my research and a formal statement of my driving research question and hypotheses.

2.1 Intrusion Detection Systems

Intrusions are an attempt to compromise the confidentiality, integrity, availability, or security measures of a system [17]. *Intrusion detection* is the process of monitoring events on a computer or network and analyzing them for signs of intrusion. Due to the overwhelming amount of data that needs to be processed for intrusion detection, automated *Intrusion Detection Systems* (IDSes) are utilized to aid cyber security analysts in their efforts to monitor for suspicious activity. IDSes are software, or hardware, systems that automate the process of monitoring the events of a network or system [17]. IDSes produce alerts for suspicious activity that require further investigation by a human. There are a variety of IDSes available, both commercially and open-sourced, for example, Snort (Fig 2.1), Rapid7 (Fig 2.2), Suricata, etc. The two most common types of IDSes are network and host-based IDSes. Network IDSes monitor the entire network for suspicious traffic by analyzing protocols. Host-based IDSes are installed on and monitor a single machine, i.e., the host.

Early concepts for IDSes can be traced back to 1980 when the National Security Agency developed tools to aid analysts to sift through audit trails created by access logs, file logs, system event logs, etc. [18]. In 1987, Denning published an IDS model that many IDSes follow today [7]. Her model used *anomaly detection* and resulted in an early IDS called the Intrusion Detection Expert System, which also incorporated *rule-based detection*. *Anomaly*

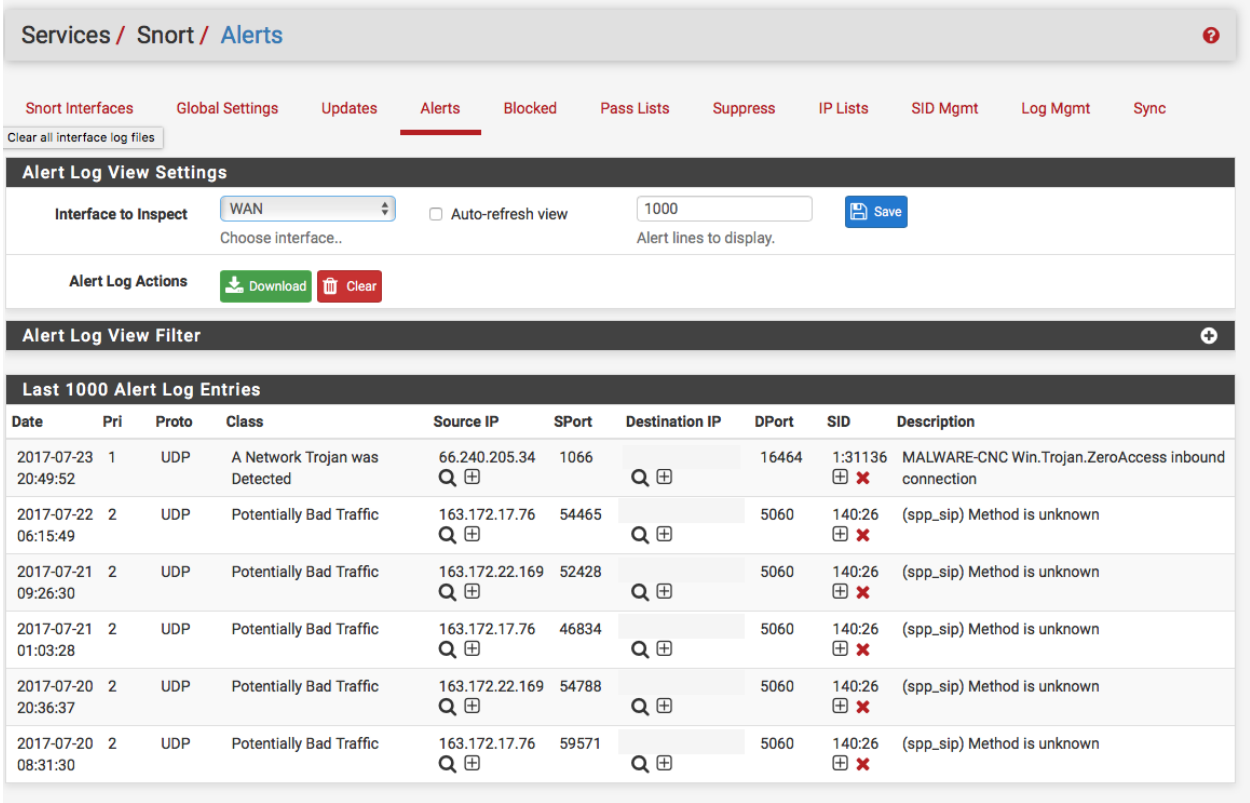


Figure 2.1: Dashboard of the Snort IDS system showing several alert entries.

detection is analysis that looks for abnormal traffic in the network utilizing machine learning [8]. *Rule-based detection*, sometimes known as misuse detection, targets traffic that is known to be bad because it violates an established rule [8]. These two methods of detection are still the basis for modern IDSes, but have troubling issues—most notably the enormous false alarm rates that plague even the best IDSes. Individual IDS false alarm rates will vary between organizations and are dependent on how the IDS is configured (i.e. which rules are set to trigger an alarm), but typical false alarm rates are above 90%, with many as high as 99% [8]. These statistics are accepted as the norm because in cyber security, a false positive that gets a second look by human eyes is better than a false negative that goes unnoticed.

Axelsson [1] discusses the base-rate fallacy of *intrusion detection*. He argues that the ratio of actual attacks to benign traffic is low, citing one study where the proportion of log entries showing an attack is approximately 1 in 100,000. Thus, IDSes must be extraordinarily accurate to have acceptable detection performance, but this requirement is compromised by

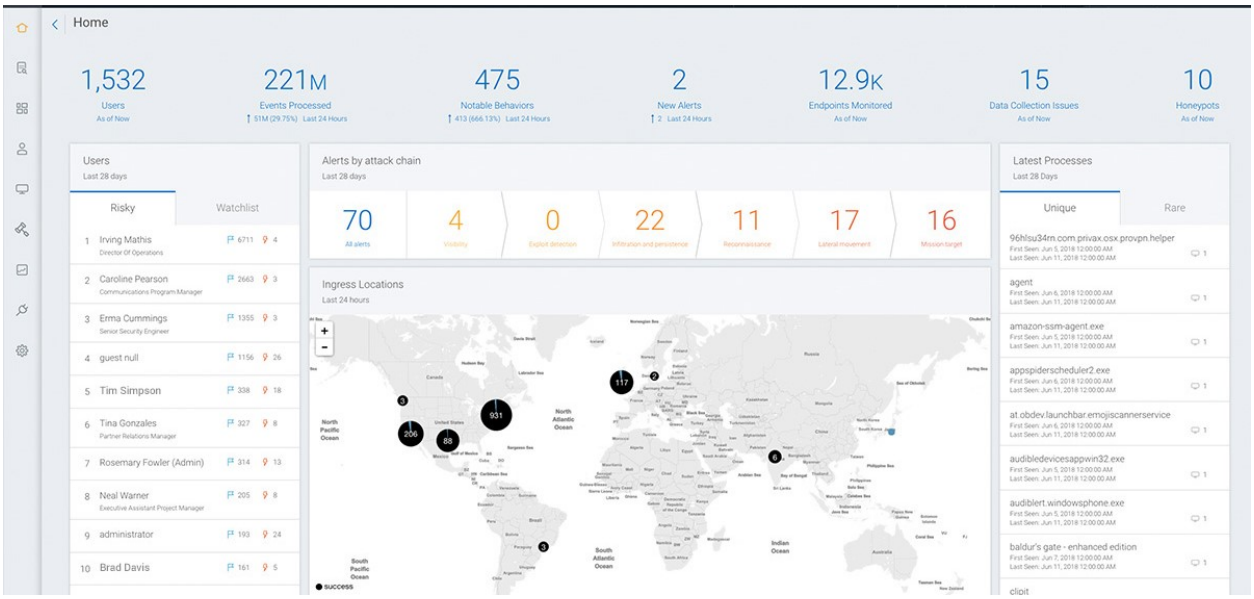


Figure 2.2: The Rapid7 dashboard providing alert summaries from a variety of network and host-based security monitors. Rapid7 is used by UNCW.

a high rate of false alarms. He states that *"a crucial question... is the capacity of the human operator to correctly respond to the output of the system, especially his/her capacity to tolerate false alarms"* [1]. Other studies of human/machine pairings indicate that the threshold for human tolerance of false alarms is approximately 50% [19]–[21]. This sets an extreme challenge for cyber security analysts—having to sift through hundreds, if not thousands, of false positives daily in search of a few true incidents. Lippmann et al. [22] suggest that new techniques must be developed that favor *anomaly detection* over *rule-based detection*. McHugh [23], while not necessarily disagreeing with Lippmann on his assertions, has strong criticism of his experimental test data and data validity. McHugh opines that plotting true positives against false positives is a poor manner in which to characterize IDS performance as it provides no insight into the reasons for that performance. McHugh states that *"there is a need for calibrated and validated artificial test data sets or test data generators. As long as the false alarm rate is used as a measure of system effectiveness, it must be possible to make sure that the false alarm rate for synthetic data has a well-understood relationship to the false alarm rate of 'natural' data for systems under test"* [23].

2.2 Work Processes of Cyber Security Analysts

IDSes play an integral part in network defense. The quantity of activity and information, even on small networks, is too great for humans to process in a tractable manner. While IDSes have improved with the addition of machine learning, there is much improvement that needs to be made before humans can be removed from network monitoring. Until that day, network defense will be the work of human/machine pairing. With this reality in mind, efforts should be made not only to understand and improve the machine aspect of network security, but also the human element.

D'Amico et al. [24] performed a Cognitive Task Analysis (CTA) [25] that investigated workflows, decision processes, and cognitive demands of cyber security analysts. CTA attempts to capture what people think about, are paying attention to, the strategies they use to make decisions or detect problems, what they are trying to accomplish, and what they know about a process [26]. Forty-one cyber security analysts from the U.S. Department of Defense and industry were interviewed; participants' expertise varied from novice to expert and covered a variety of analyst roles. The CTA examined the different roles of cyber security analysts, all whom have duties that the Department of Defense terms *computer network defense* (CND). Participants were posed with a hypothetical scenario exercise to avoid issues of confidentiality or classification, thus enabling the participants to share the kind of knowledge, techniques, and abstract connections they would make as a part of their jobs. Job titles varied considerably among participants, yet had strong similarities in analysis tasks and duties. Due to a lack of common functional job descriptions, D'Amico et al. categorized the actual tasks into these six broad roles that encapsulate the cognitive work of an analyst:

- *Triage* - this is the initial look at the data. In this step an analyst discards the false alarms in IDSes and other network monitoring system, and escalates suspicious activity for further analysis.

- *Escalation Analysis* - a closer look at the events that were not immediately removed

from initial triage.

- *Correlation Analysis* - searching for patterns or trends in current and historical data.
- *Threat Analysis* - the gathering and analysis of intelligence to support CND. This might involve attacker profiling and an attempt to identify the attacker's identity and motive.
- *Incident Response Analysis* - recommending and/or implementing a course of action in response to a confirmed incident.
- *Forensic Analysis* - collecting and preserving evidence to support a law enforcement investigation.

D'Amico et al. also observed participants progressing through three distinct stages as they fused data together (Fig 2.3). Those stages were:

- *Detection* - the first stage of analysis. Analysts are primarily concerned with inspection and associating data with normal or suspicious activity.
- *Situation Assessment* - the next step where events are escalated. Analysts are incorporating data from other sources (e.g. logs, web searches, other reports) in order to fuse and process information as the assessment gets refined.
- *Threat Assessment* - the last step where an incident is confirmed. Analysts are assessing damage, attempting to identify attacker identities and motives

The CTA also identified included goals, decisions, knowledge and barriers to success in the analysis process. Cyber security analysts deal with many cognitive challenges as a result of the inherent difficulty of the analysis domain and the shortcomings of current tools available. D'Amico et al. list four challenges of note: massive data, fusion of complex data, building site-specific knowledge, and maintaining multiple mental models.

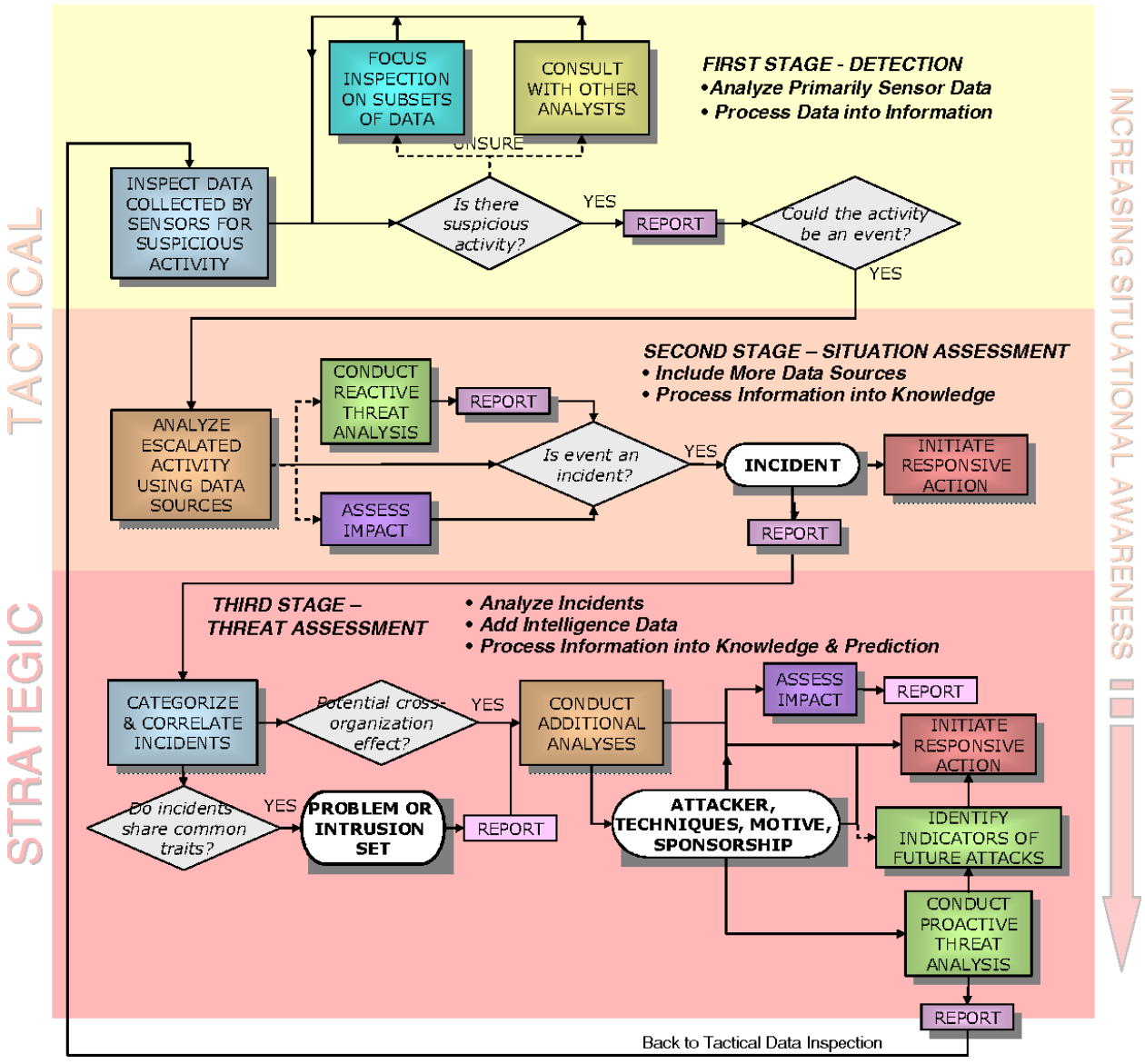


Figure 2.3: Stages of CND situation awareness and cognitive data fusion [24]

D'Amico and Whitley [27] extend their previous work by investigating daily interactions and analyses of cyber security analysts. For this study they focused on understanding how analysts *examine data* and build conclusions as events go through varying stages of triage. This was done in an effort to help tool developers who might otherwise not have easy access to cyber security analysts design effective visualizations. D'Amico and Whitley state that *"as analysis proceeds, data are transformed as an analyst's confidence in an analytic conclusion increases. The cognitive transformation can be seen as a hierarchy of data filters*

with the levels reflecting the increasing certainty that a reportable security violation has been identified."[27]. The data hierarchy (Fig!2.4) involves includes the following:

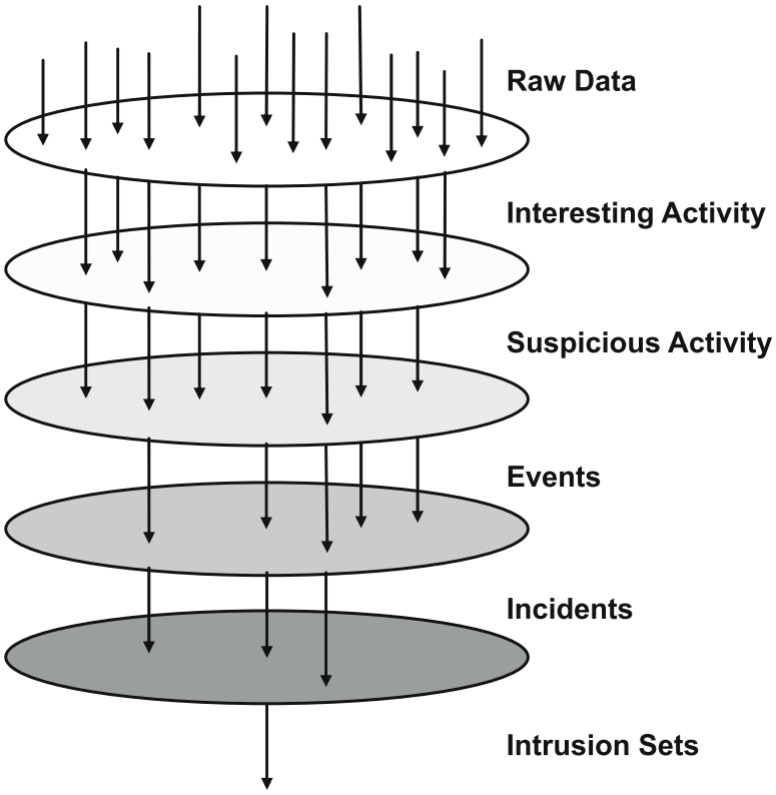


Figure 2.4: Data hierarchy as data are transformed into security situation awareness [27]. Note that data volume decreases as levels progress.

- *Raw data* is the most basic data in the hierarchy. This can be network packet traffic, netflow data, or log data. Due to the sheer volume of raw data, analysts typically do not look at it. Instead it passes through and is filtered by some automated system (e.g. IDS).

- *Interesting activity* is the data that the automated system has flagged for inspection. This is where an analyst begins to *triage* the data. The data may contain false alarms at this point.

- *Suspicious activity* is what remains after the *triage*. Data here is anomalous to normal activity, though it does not necessarily constitute an attack.

- *Events* are suspicious activities that an analyst should report. At this level data has been significantly reduced and analysts begin to group and categorize data based on common characteristics (e.g. source/destination IP addresses, time, attacker behavior, etc).

- *Incidents* are the points at which an analyst can confirm that an attack has happened based on one or more events.

- *Intrusion sets* are "sets of related incidents"[27] that have been collated over time, e.g., a series of incidents from a single actor or targeting a particular resource.

Both [24] and [27] give an overview of typical processes one might expect to see an analyst perform and clearly define several terms that will be used throughout this paper. Their work also highlights the requirement for cyber security analysts to rely upon automated systems to process raw data, but also to address possible false alarms produced by those systems.

Zhong et al. [28] addressed the need to capture the fine-grained cognitive processes in order to deepen our understanding of analysts' reasoning. They performed a CTA on 30 industry professional and doctoral students who specialize in cyber security. Zhong et al. found that analysts conduct *traces* of network activities as part of the *triage* process, and that these tracing operations could be formally defined (Fig 2.5). These detailed tracing steps formalize several of the processes identified in [27]. To address the difficulty of performing CTAs for cyber security, they propose an integrated computer-aided collection method that relies on automated capture and situated self-reports. Zhong et al. [29] also make the case for senior analysts as front line defenders in data triage because they have the expertise to distinguish false alarms from real incidents. Since the industry typically has data triage performed by novice analysts, they propose a data triage support system that provides novice analysts with suggestions based on the historical analytic processes of senior analysts. They used this proposal to develop a tool to that provides on-the-job suggestions for data triage

that were developed by senior analysts. Their results showed that their retrieval system could effectively identify relevant data based on the analyst's current analytic process.

Operation	Description
BROWSE	BROWSE(D_i), $D_i \subseteq \bigcup DS_i$*: Browse the data sources.
FILTER	FILTER($DS_i, Cond.$): Filter the source data DS_i based on condition $Cond.$
SEARCH	SEARCH(D_i, K), $D_i \subseteq \bigcup DS_i$: Search K in data D_i.
INQUIRE	INQUIRE(T_m): Inquire about a term T_m
SELECT	SELECT(D_i), $D_i \subseteq \bigcup DS_i$:Select the data of interest in D_i .
SELECTED *	*(Come in pairs with SELECT) SELECTED(D_i), $D_i \subseteq \bigcup DS_i$: The selected data of interest
LINK	LINK(D_i, L), $D_i \subseteq \bigcup DS_i$: 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.

Figure 2.5: Eleven types of operations conducted by analysts [28]

.

Goodall et al. [30] sought to understand the socio-technial aspect of humans pairing with IDSes. They conducted a field study to explore the the task of network intrusion detection. They found that while a high degree of domain *expertise* is needed, cyber security analysts rely heavily on *situated knowledge*. Expertise is an accumulation of knowledge and problem-solving strategies related to a particular task [31]. Situated knowledge is not defined by Goodall, but it likely related to the concept of *situated action* [32], [33] and is the idea that knowledge is formed as part of performing a physical task [34]. Similarly, as it relates to cyber security, is knowledge specific to a particular network, or system. Such knowledge is dependent on the organization, implicit, and difficult to articulate [35]. Experts acquire it through on-going interactions within a specific network environment. It is often learned through continually tuning and adjusting the IDS to meet an organizations network

security demands without interfering with legitimate network users [36]. This means that the participants in this study will need proper training on the tool presented to them, but also a clear understanding of the scenario.

2.3 Information Used to Determine An Attack

Table 2.1 succinctly summarizes several studies on what information analysts need to see to make informed decisions. I believe these data fields to be the most relevant, and crucial, to an analyst for decision making. I will use this information, and dashboard information from real-world IDSes, in the development of the mock IDS for my experiment.

data name	**short description**	**Layman14**	**Mahoney10**	**Gutzwiller16**	**Erbacher10**	**Snort IDS**
network topology	map of the network				x	
network traffic	where from to	x	x	x	x	x
port info	what port was used	x			x	x
server log	who touched what	x		x	x	

Table 2.1: Data sources used in CND activities

Mahoney et al. [37] conducted a CTA on an industry professional in an effort to develop a cyber situational awareness tool for users at varying levels of responsibility and expertise of an organization, not just for network administrators. Their CTA was driven by a series of analytical knowledge acquisition sessions to understand the situational awareness required for such a tool. Layman et al. [38] studied log fields in an effort to understand which fields held the most value for attack detection. They experimented with 65 IT professionals and novices. The data they gathered was used to analyze attack detection accuracy, identify the most valuable fields for attack detection, and to understand how participants processed information and the techniques they used. Erbacher et al. [39] re-emphasized the importance of *situational awareness* and sought to create a visualization tool to help analysts make better, faster decisions by understanding the current state of the environment they work in. They interviewed analysts to determine what information they needed to see to make effective decisions. This lead to the development of new task-flows representing the activities of analysts. Gutzwiller et al. [40] performed a CTA to determine the abstract factors at

play that make up Cyber Cognitive Situation Awareness for cyber security analysts.

Ultimately, I focused on one type of IDS alarm, *impossible travel*, that I experienced frequently in my positions as a UNCW cyber security analyst. The IDS alarms I generate in my study include network traffic information. However, I decided to simplify the types of alerts presented by my simulated IDS given the time constraints and training required to perform the analysis task. These issues are described in Section 3.

2.4 Factors Impacting Cyber Security Analysts Performance

Sawyer and Hancock [41] examined the *prevalence effect* of cyber attacks on humans. The *prevalence effect* states that, as signals become less frequent, they become harder to detect regardless of how important those signals are [42]. They tested the prevalence effect in the context of email cyber attacks, specifically phishing attempts and emails with malicious executable attachments. For their experiment, Sawyer and Hancock placed thirty participants into three groups that received email cyber attacks at different signal probabilities - 1%, 5%, and 20%. The goal of this experiment was to observe how accuracy, in terms of attack detection, and response time changed with varying signal probabilities (Fig 2.6).

In one session, each participant examined 300 emails, at his or her own pace, to determine if each email constituted an attack. Prior to the experiment, participants were presented with 20 training emails, 10 of which were attacks, and were required to correctly identify at least 80% of the attacks in order to continue. Benign and attack emails were delivered at random, though each participant was presented attacks at a rate corresponding to his or her assigned signal probability. Once participants had completed their session they were asked to complete demographic information and were then debriefed by the research associate. The average time for participants to complete their session was approximately one hour.

The results of their experiment suggest that as the ratio of malicious emails to benign emails becomes lower, our ability to detect malicious emails decays at a logarithmic rate

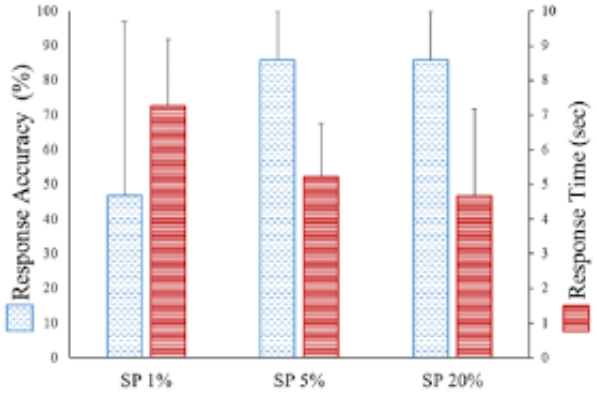


Figure 2.6: When the prevalence of attacks was raised, response time (striped bars) decreased while response accuracy (dotted bars) increased. Results additionally show that the lowest signal probability led to the highest levels of response time and lowest levels of accuracy. [41]

(Fig 2.7), confirming findings from other domains [43]. Sawyer and Hancock state, *"While there is presently no work that explicitly explores this link among prevalence effects, teaming with automation, and trust, there is evidence that such a relationship does exist"*[41]. This statement goes to the heart of this study.

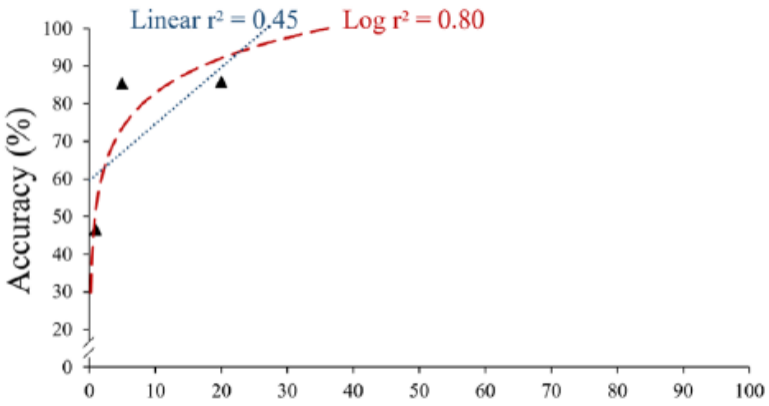


Figure 2.7: Analyst accuracy in detecting malicious emails vs. signal probability suggesting a logarithmic fit [41]

Ben-Asher and Gonzalez [36] studied how the roles domain knowledge and network operations play in intrusion detection. They developed a simple IDS and had participants of varying expertise (i.e. novice - expert) decide if a sequence of events constituted a cyber attack. Prior to their experiment, they had participants answer a questionnaire to establish their expertise level. The questionnaire assessed participants' level of domain knowledge

through questions regarding networks and information security. Questions were also designed to assess participants' theoretical and practical knowledge on those subjects, and to gauge the participants' experience with IDSes.

For their experiment, Ben-Asher and Gonzalez developed five scenarios, each representing a different type of attack: deface website, sniffer detected, denial-of-service, stealing confidential data, and a no attack scenario. Each scenario was comprised of a sequence of 20 network events and were presented utilizing the IDS with a new event appearing every 10 seconds (Fig 2.8). As these events appeared, participants decided whether they believed the scenario represented an attack. Average time to completion about 60 minutes for novices and 25 minutes for experts. Experts were also more likely to be able to describe the nature of the attacks they encountered.

Is threat	ID	Alert	Description
☐	1		The web server is running ftpd and httpd services. The traffic is 3.3 Mbps between internet and web server, 3.3 Mbps between web server and file server, and 3.3 Mbps between web server and workstation.
☐	2	ftpd has started running on web server	The web server is running ftpd and httpd services. The traffic is 3.3 Mbps between internet and web server, 3.3 Mbps between web server and file server, and 3.3 Mbps between web server and workstation. An ftpd operation has been executed.
☐	3		The workstation is running a user process. The traffic is 3.3 Mbps between file server and workstation, and 3.3 Mbps between workstation and web server.

Figure 2.8: Network events with description and alerts from the IDS [36]

Their results showed that while general expertise is helpful in attack detection, situational knowledge of the specific network is especially helpful to accurately make decisions regarding detection. This reinforces the finding of Goodall et al. [30]

Sabottke et al. [44] compared the strategies and features of clustering systems and humans in detecting attacks. Their study also reminds us that attacks do not happen in a vacuum. At some point, there is a human on the other end of an attack, and they have goals and strategies, as well. They will leverage their own resources to exploit any vulnerability, both human and machine, in order to achieve those goals.

3 METHODOLOGY

In the section, I discuss the experiment I designed to test my hypotheses. The goals of this experiment, which I nicknamed "Cry Wolf", were to provide a meaningful answer to my research question, to be as realistic as possible, and to last no more than an hour so as not to tire the participants. I provide a detailed explanation of the process of my experiment and who participated. This experiment was designed with factors in mind that guarded the integrity of the experiment's results.

3.1 Research Question and Hypotheses

The problem presented by the false alarm rate of IDSes is unlikely to be solved in the near future. Much study on the issue and advances in machine learning will be required to bring this rate down to a more manageable level for analysts. This project seeks to aid in that process by better understanding the extent to which the false alarm rate of an IDS affects analysts' performance—specifically observing how analysts' ability to detect incidents and their response times change with an increasing false alarm rate. Identifying the relationship between analyst performance and IDS false alarm rate can help set benchmarks for IDS performance, and help organizations understand the trade-off between high alarm rates and the ability to detect actual attacks.

Research question: How does the false alarm rate of an IDS affect the performance of a cyber security analyst?

Hypotheses:

H1. Analyst *performance* decreases as the IDS false alarm rate increases

H2. Analyst *time on task* increases as the IDS false alarm rate increases

The central hypothesis, **H1**, is derived from the base-rate fallacy described by Axelsson [1] and the prevalence effect identified by Sawyer [41]. Analyst performance is measured in terms of *sensitivity, specificity, and precision* as described below. The second hypothesis, **H2**, is chosen because response time was also measured in Sawyer's study on the prevalence

effect [41]. Furthermore, I hypothesize that analysts will become distrustful when a system that generates a high number of false alarms, and will have to spend more time confirming the decision of the IDS rather than accepting its conclusion. This follows observations by Bliss [9] that speed in responding to and acting upon alarms is influenced by the false alarm rate.

3.2 Analysis Variables

A *true alarm* is an alert from the IDS that represents a malicious event (i.e. an "Escalate" event), whereas a *false alarm* is an alert from the IDS that represents a benign event (i.e. "Don't escalate"). The main independent variable in my analysis is the $false\ alarm\ rate = \frac{\#\ false\ alarms}{\#\ of\ alarms}$. I examined this with respect to the the human analyst's performance in correctly classifying alarms from an IDS. The study participants were asked to classify all alarms as "Escalate" or "Don't escalate". Binary classification yields four measures that are used in performance calculations:

- True Positive (TP) - the analyst escalates a true alarm;
- False Positive (FP) - the analysts escalates a false alarm;
- True Negative (TN) - the analyst does not escalate a false alarm;
- False Negative (FN) - the analyst does not escalate a true alarm.

I computed and examined the following dependent variables derived from these measures:

- $sensitivity = \frac{TP}{TP+FN}$ - a measure capturing whether an analyst recognizes true alarms among all alerts.
- $specificity = \frac{TN}{TN+FP}$: a measure of how accurate the analyst is in identifying false alarms

- $precision = \frac{TP}{TP+FP}$: a measure capturing misidentification of false alarms as malicious events. Also known as *positive predictive value*.

- $time\ on\ task = time\ of\ finish - time\ of\ begin$. The time it takes the subject to complete the experiment, measured in *minutes*.

3.3 Subject Selection

Subject selection for this experiment was challenging, especially finding cyber security experts. The majority of participants came from two courses taught in the Fall of 2019, MIS 310–Web Page Development Languages and MIS 322–Information Assurance and Security. The remainder of participants came from email and in-person recruitment to CSC and MIS students, my colleagues in the MSCSIS program, friends and family. Participation was optional for students in the MIS courses. For compensation for their participation, participants from the MIS courses had the option to choose extra course credit or be entered into a raffle for one of five $30 Amazon gift cards; the other participants were entered into the raffle. A total of 52 participants were recruited for the experiment, 21 from MIS 310, 20 from MIS 322, nine from a recruitment email to the CSC/IT undergraduates and my MSCSIS colleagues, and two were family members.

All participants filled out a questionnaire prior to the experiment; answers were used to categorize the expertise of each subject, e.g., expert vs. novice. Particular areas of interest for the questionnaire are:

- Domain knowledge about network behaviors and concepts

- Familiarity with IDSes

- Self-report of network experience (in years)

Institutional Review Board (IRB) approval[1] was granted for our study procedures and artifacts prior to conducting any human subjects research. Analysis of the participants is

[1] https://uncw.edu/sparc/integrity/irb.html

provided in Chapter 4.

3.4 Test Scenario and Test Instrument

I wanted to create an experimental scenario that was meaningful, intuitive, and quasi-realistic in order to test my hypotheses. Participants were presented with the following scenario: "You are a junior cyber security analyst at Company XYZ. Your job is to perform initial *triage* on a list of network events that your system's IDS has deemed an alert. You must determine if the alert could be an attack against the network, requiring further investigation by a senior analyst, or is normal network activity that can be dismissed." I chose this scenario to inject realism into my experiment based on its relevance to this study, and because it is a scenario I am familiar with as a cyber security analyst for UNCW.

I decided to focus on the *impossible travel* type of alerts that IDSes often generate. This is where a user authenticates from two geographic locations in an amount of time that would be considered impossible for them to do so. For an example, a participant might see an event whose authentications came from Seattle, WA and New York City, NY with a time between authentications of 30 minutes. This is obviously an impossible feat, unless there are other means that explain the authentications, such as utilizing a VPN. An example of an *impossible travel* event is shown in Fig 3.1. Some guidance for evaluating this event, and the decision-making form, are shown in Fig 3.2. This example event was given as part of the training to subjects along with four other training events. The training events and reference materials provided to participants are discussed further in Section 3.5.

The difficult part of event synthesis was deciding on what data fields to present participant. Data fields included:

- **City of Authorization** — There were two of these fields, one for each geographic location from which the IDS detected an authorization.
- **Number of Successful Logins** — The number of successful logins from each location

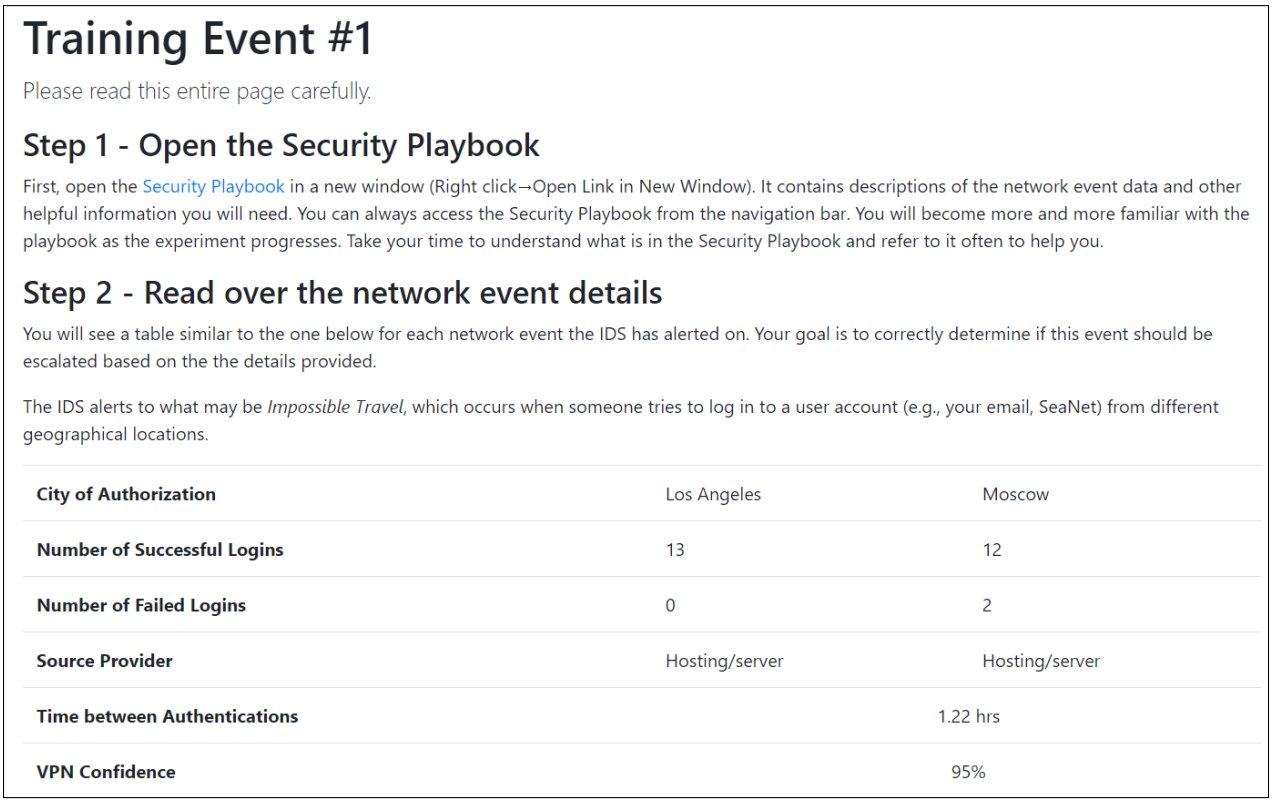


Figure 3.1: Training Event 1. Step 1 invites the participant to use the Security Playbook to aid in decision making. Step 2 presents the data that each event will consist of.

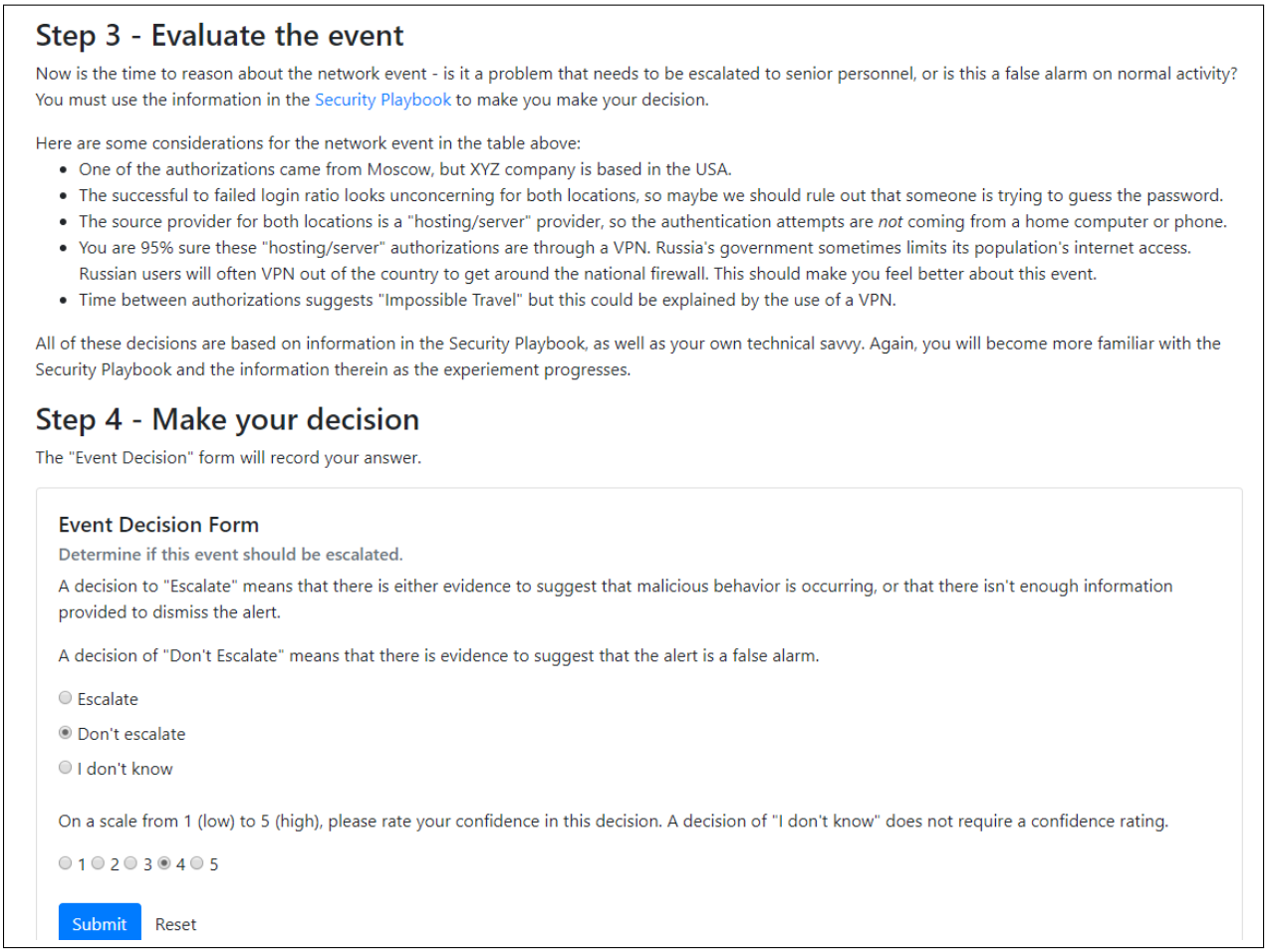


Figure 3.2: Training Event 1. Step 3 walks the participant through my thought process on how to evaluate this event. Step 4 presents the decision form the participants will utilize for the experiment.

in the past 24 hours. A successful login is one where a correct username+password combination was entered.

- **Number of Failed Logins** — The number of failed login from each location in the past 24 hours. A failed login is when an incorrect password was used for an account.
- **Source Provider** — The type of Internet Provider the authorizations came from at each location. Possible values were "Telecom", "Cellular/Mobile", or "Hosting/server".
- **Time Between Authorizations** — This is the field that typically triggers the alarm. The IDS correlates an authorization from each location and determines that they happened too quickly.
- **VPN Confidence** — A percent likelihood, based on evidence not given, that the user utilized a Virtual Private Network (VPN).

I picked these data fields from my experience as a security operations analyst at UNCW. These were primary data points that I would consider in which I could typically arrive at a decision as to whether activity was normal. I picked designed several variations of the *impossible travel* scenario that I experienced that were both concerning events needing escalation or remediation (i.e., true alarms) and non-issues (i.e., false alarms). Data for events was generated in Excel.

I designed and deployed a web application (app) as the vehicle in which to deliver the experiment. I used a web app to simplify the administration and records keeping of the experiment. The web app, also called Cry Wolf, administered the entirety of the experiment with the exception of collecting informed consent. Cry Wolf was developed with Flask[2], a micro-framework for Python web development. Cry Wolf was deployed on a Heroku web app server[3] and utilized a Postgres database for data persistence. The project is comprised of over 1500 lines of HTML and 1000 lines of Python (manually- and automatically-generated

[2]https://palletsprojects.com/p/flask/
[3]https://cry-wolf.herokuapp.com/

code). The steps of the experiment, and how Cry Wolf presented them, are described in the next section.

3.5 Experiment

The overall workflow for the experiment is shown in Fig 3.3. The entirety of this workflow was implemented in the Cry Wolf web app, allowing participants to complete it with no intervention. The experiment was conducted on four separate occasions with different participant groups: one MIS 320 class, one MIS 322 class, and two open sessions for individuals recruited via email and in-person. The experiment was proctored by me, and participants were allowed to ask clarifying questions, but were not allowed to collaborate or use Internet sources to assist in completing the experiment.

3.5.1 Introduction and Pre-test Questionnaire

After welcoming participants I introduced myself and explained that I was conducting a study on human/machine pairings as it relates to cyber security. I briefly described the general flow of events and that I would be guiding them through the training portion of the experiment. I welcomed all questions until training was completed and informed the participants that I would be unable to assist them in their determinations during the experiment. The experiments were conducted in computer labs at UNCW.

Participants were required to sign consent forms prior to experimentation. Once a signed and dated consent form was returned to me, I provided them with a slip of paper with their unique login username and directed them to the web app's URL. The home page of the web application presented a short overview of the expected work and time required as well as the required IRB disclaimers. When the participants logged in, they were redirected to a landing page which primarily served to present the scenario to participants. In the event of any disruption to the experiment, this page would also provide links that directed the participants back to where they were to continue experimentation.

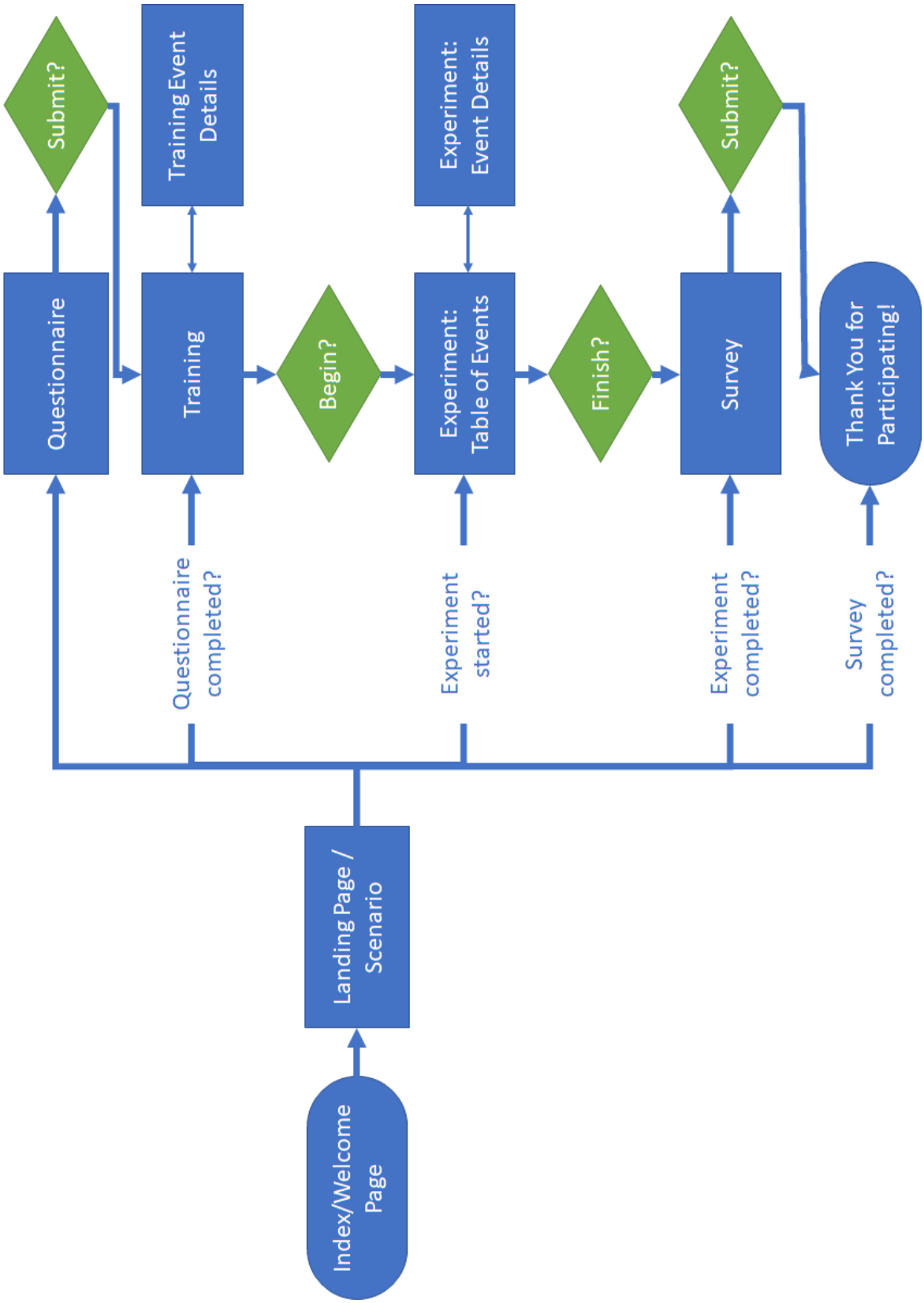


Figure 3.3: Flowchart of the experiment

From the landing page a participant would move on to the questionnaire about the participants' background and experience, and also some basic networking questions. The full text of the questionnaire is provided in Appendix A. Participants were required to complete all questions in the pre-questionnaire, and their responses to the questions were not scored for correctness and did not impact their ability to continue with the experiment. Participants could not change their answers after they submitted a completed questionnaire.

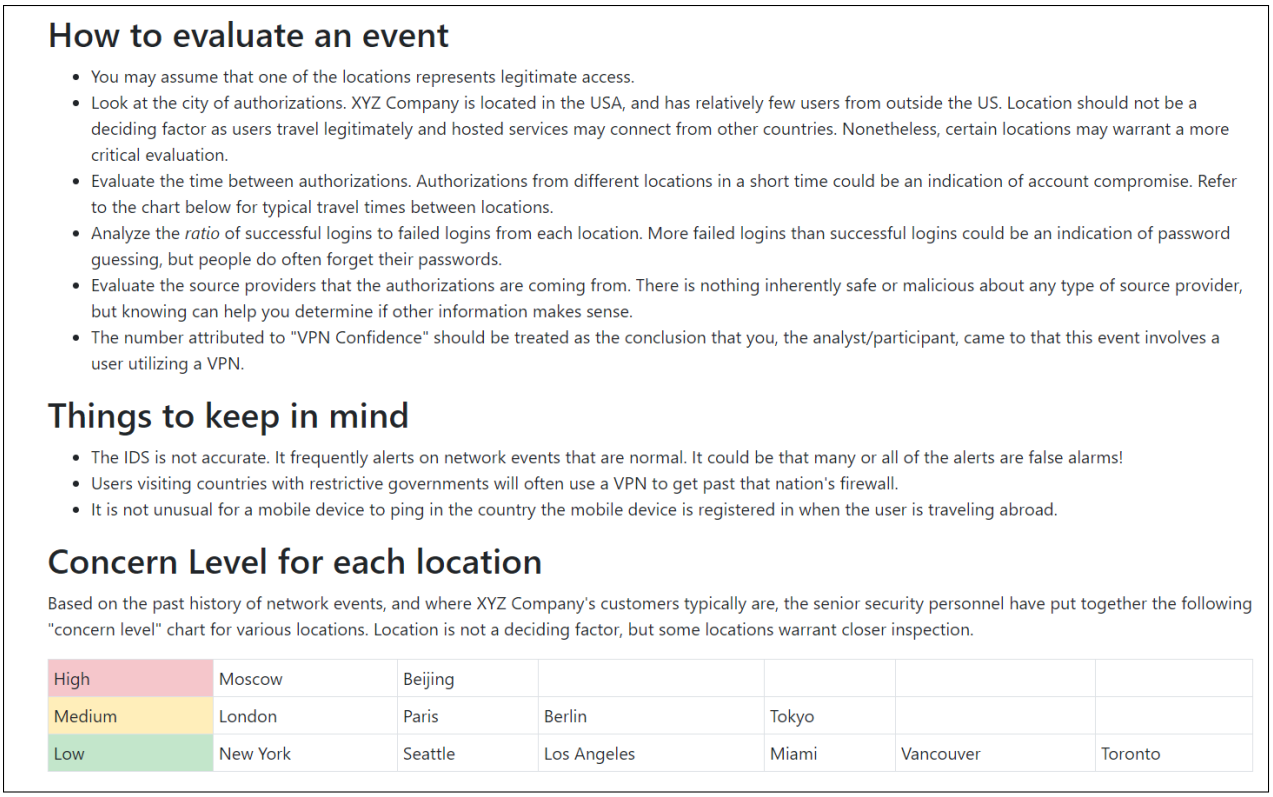
3.5.2 Training

Once the participants completed the questionnaire they began a short training exercise. Five events were provided for training. The first two were thorough in explanation on how to evaluate events while the remainder were presented as they would appear on the experiment. The first training event is shown in Figs 3.1-3.2. All training event pages included a button to reveal the correct answer and the rationale for that answer.

For each training event (and for events in the main task) participants were presented data pertinent to the individual event and were then required to determine if the event should be "Escalated" to a more senior analyst for investigation–which was not part of the experiment–or, if the participant felt that the event was a false alarm on normal activity the event could be dismissed by selecting "Don't escalate". Participants were also asked, for each event, to rate their level of confidence in their decision on a five point scale. Participants could also choose "I don't know" for their decision, in which case a confidence score was not required.

To aid participants in their decision making a "Security Playbook" was offered that reminded them of the scenario, contained detailed instructions of how to evaluate an event (Fig 3.4), a section of important things to keep in mind, a concern level (low - high) for each location presented to the participant, and also a table of typical travel times between the locations (Fig 3.5).

When they decided they were ready to begin the main task, participants clicked a "Begin Experiment" button that popped up a dialog requiring second confirmation, to begin



How to evaluate an event

- You may assume that one of the locations represents legitimate access.
- Look at the city of authorizations. XYZ Company is located in the USA, and has relatively few users from outside the US. Location should not be a deciding factor as users travel legitimately and hosted services may connect from other countries. Nonetheless, certain locations may warrant a more critical evaluation.
- Evaluate the time between authorizations. Authorizations from different locations in a short time could be an indication of account compromise. Refer to the chart below for typical travel times between locations.
- Analyze the *ratio* of successful logins to failed logins from each location. More failed logins than successful logins could be an indication of password guessing, but people do often forget their passwords.
- Evaluate the source providers that the authorizations are coming from. There is nothing inherently safe or malicious about any type of source provider, but knowing can help you determine if other information makes sense.
- The number attributed to "VPN Confidence" should be treated as the conclusion that you, the analyst/participant, came to that this event involves a user utilizing a VPN.

Things to keep in mind

- The IDS is not accurate. It frequently alerts on network events that are normal. It could be that many or all of the alerts are false alarms!
- Users visiting countries with restrictive governments will often use a VPN to get past that nation's firewall.
- It is not unusual for a mobile device to ping in the country the mobile device is registered in when the user is traveling abroad.

Concern Level for each location

Based on the past history of network events, and where XYZ Company's customers typically are, the senior security personnel have put together the following "concern level" chart for various locations. Location is not a deciding factor, but some locations warrant closer inspection.

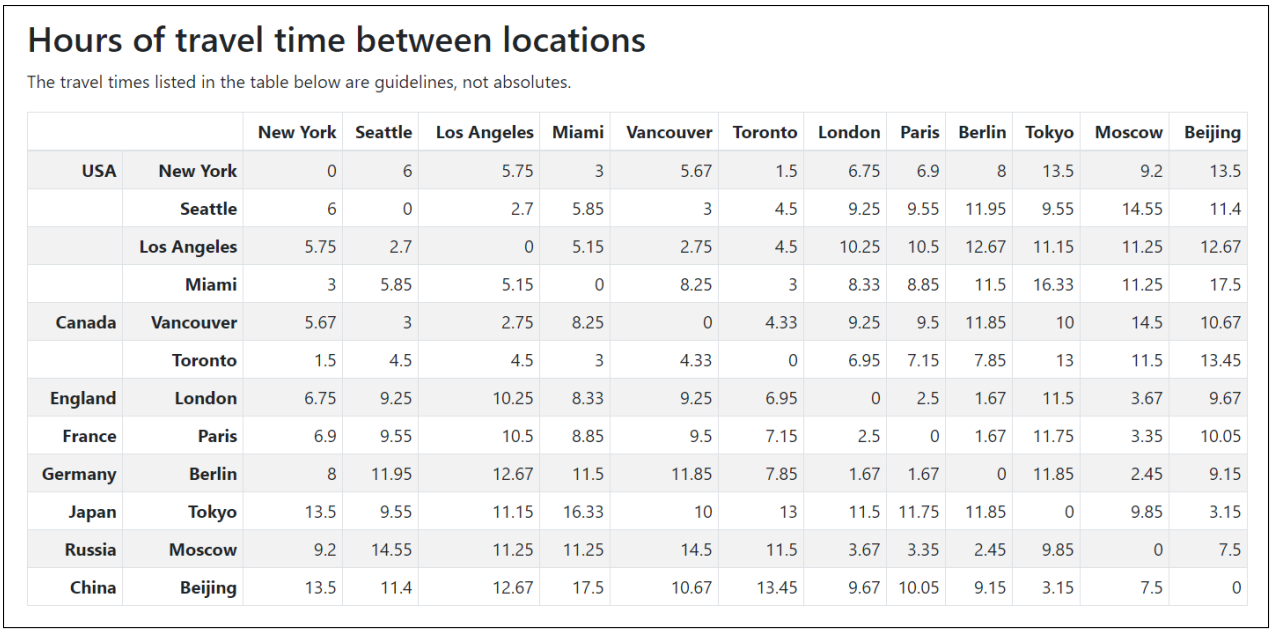
High	Moscow	Beijing				
Medium	London	Paris	Berlin	Tokyo		
Low	New York	Seattle	Los Angeles	Miami	Vancouver	Toronto

Figure 3.4: This section of the Security Playbook reminds participants how to evaluate an event, things they should consider that may not be intuitive, and the concern level for each of the geo-locations of the events

the main task. Participants were allowed to return to the training pages and the "Security Playbook" at any time while prior to completing the main task.

3.5.3 Main Task: Event Evaluation

After the participant clicked and confirmed the "Begin Experiment" button, participants were taken to a table of 52 events (Fig 3.6), and a database entry was recorded that marked the beginning of their time on the main task. Participants could open the events in any order they chose and were presented with a screen containing event information as shown in Fig 3.7. As in the training events, participants made an "Escalate/Don't Escalate/I don't know" decision and provided a confidence level. Once a decision was made, participants were automatically shown to the next event in the event table (rather than navigating back to the table). Participants could go back and change their decision on an event at any time.



Hours of travel time between locations

The travel times listed in the table below are guidelines, not absolutes.

		New York	Seattle	Los Angeles	Miami	Vancouver	Toronto	London	Paris	Berlin	Tokyo	Moscow	Beijing
USA	New York	0	6	5.75	3	5.67	1.5	6.75	6.9	8	13.5	9.2	13.5
	Seattle	6	0	2.7	5.85	3	4.5	9.25	9.55	11.95	9.55	14.55	11.4
	Los Angeles	5.75	2.7	0	5.15	2.75	4.5	10.25	10.5	12.67	11.15	11.25	12.67
	Miami	3	5.85	5.15	0	8.25	3	8.33	8.85	11.5	16.33	11.25	17.5
Canada	Vancouver	5.67	3	2.75	8.25	0	4.33	9.25	9.5	11.85	10	14.5	10.67
	Toronto	1.5	4.5	4.5	3	4.33	0	6.95	7.15	7.85	13	11.5	13.45
England	London	6.75	9.25	10.25	8.33	9.25	6.95	0	2.5	1.67	11.5	3.67	9.67
France	Paris	6.9	9.55	10.5	8.85	9.5	7.15	2.5	0	1.67	11.75	3.35	10.05
Germany	Berlin	8	11.95	12.67	11.5	11.85	7.85	1.67	1.67	0	11.85	2.45	9.15
Japan	Tokyo	13.5	9.55	11.15	16.33	10	13	11.5	11.75	11.85	0	9.85	3.15
Russia	Moscow	9.2	14.55	11.25	11.25	14.5	11.5	3.67	3.35	2.45	9.85	0	7.5
China	Beijing	13.5	11.4	12.67	17.5	10.67	13.45	9.67	10.05	9.15	3.15	7.5	0

Figure 3.5: This section of the Security Playbook provides a table of typical travel times between the locations presented to participants.

A database entry was recorded that captured each time an event was displayed, and each time a decision was made for an event. A single event could thus have multiple database entries for time displayed and decision made.

As a guard to ensure that participants were not just clicking through events, two "check" events were inserted into the table of events – one toward the beginning and one toward the end. For these events no data was presented; participants were simply asked to select a specific decision (i.e. "Escalate" or "Don't escalate") and also a specific confidence value. If the participant failed to follow the instructions on these two events I assumed that they were not paying attention to the data and therefore, their results should be excluded from analysis, unless otherwise noted in the participant's feedback specifically why they chose not to obey those instructions.

Participants were placed into one of two groups. Of the 50 actual events, Group 1 saw 25 "Escalate" events (i.e. *true alarms*) and 25 "Don't escalate" events (i.e. *false alarms*), while Group 2 saw two "Escalate" and 48 "Don't escalate" events. The false alarm rates of 50% (Group 1) and 96% (Group 2) were chosen as they reflect best-in-class and typical

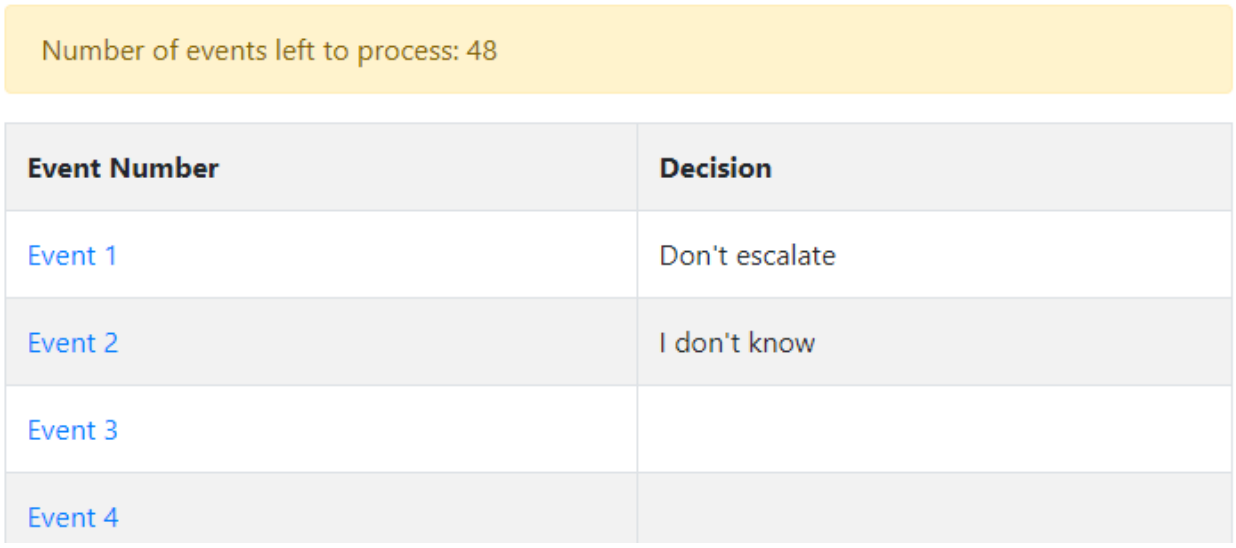


Figure 3.6: Sample screenshot of the main experiment event list.

false alarm rates respectively for IDSes according to [8]. I had intended to analyze a third group, which would evaluate have evaluated 10 "Escalate" and 40 "Don't escalate" events, but I made an error while passing out the paper slips to the MIS 322 section during the first round of experimentation and accidentally put all 20 participants in Group 1. It was decided that the best approach to remedy the mistake would be to focus on balancing the remainder of participants between Group 1 and Group 2. A participant was assigned to a group depending on which access code they were handed by me at the beginning of the experiment. The participants were not told which group they were a part of or the false alarm rate in the experiment. Groups were balanced with 25 participants assigned to Group 1 and 27 participants assigned to Group 2.

Since Group 1 needed to evaluate the most "Escalate" events, each participant was assigned every "Escalate" event I synthesized and a random sampling of 25 events from the "Don't escalate" pool. This random sample was generated once and given to every participant in Group 1. Group 2 needed to evaluate every "Don't escalate" event synthesized, and only one "Escalate" event was randomly selected from that pool. Only one was randomly selected because there was one "Escalate" event that every participant regardless of which group they were assigned to – this event is shown in Fig 3.7. This event was a "worst-case"

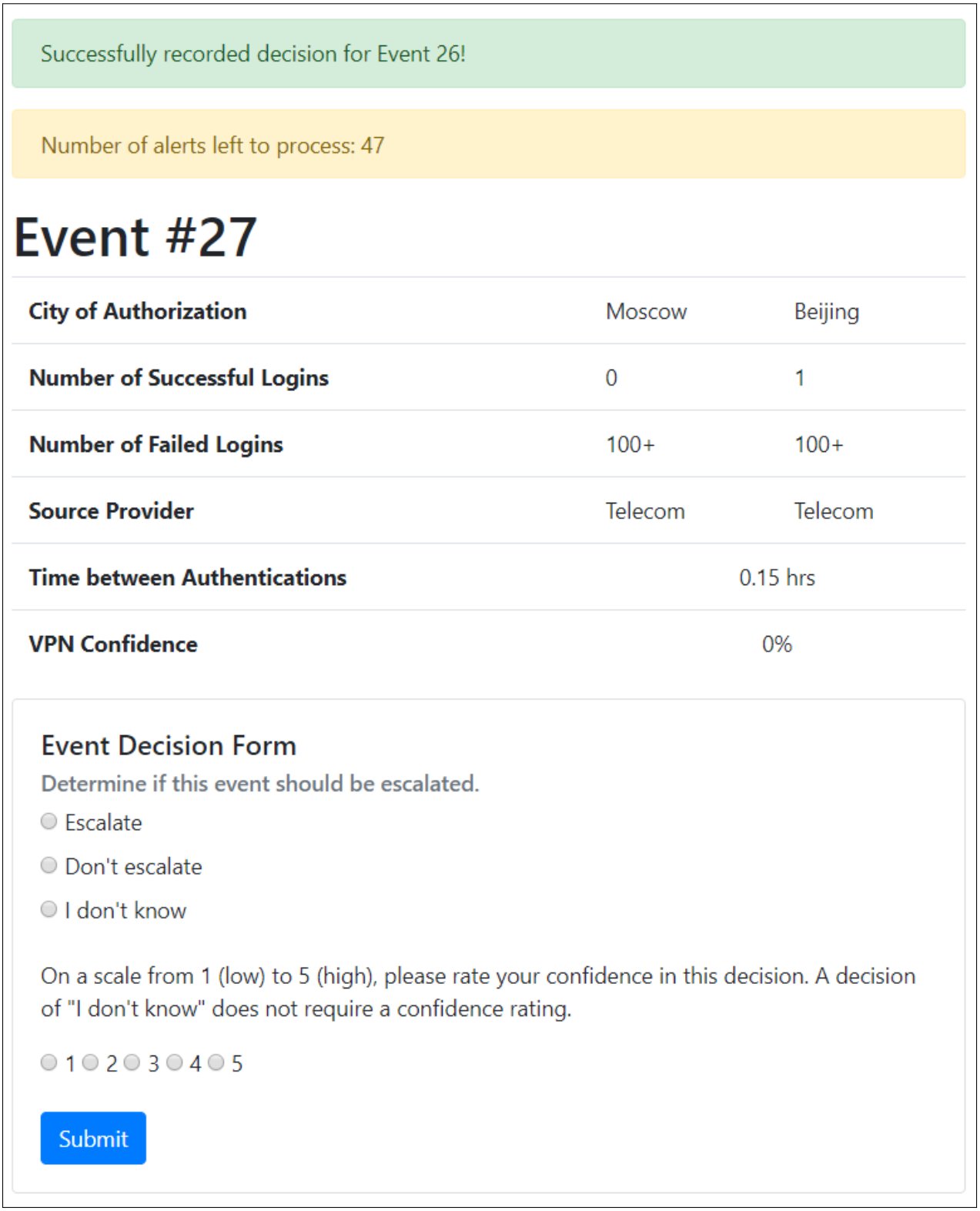


Figure 3.7: Every participant sees this event, regardless of which group they were placed in. It was designed to be an obvious "Escalate" event

type of event and intended to be an obvious decision for each participant. Once the specific events were selected for each group, those events were shuffled for each participant within groups prior to presentation in the main task table (Fig 3.6).

Participants clicked a "Complete Experiment" button once they were finished and presented with a confirmation dialog. Once confirmed, a database entry was recorded marking the end of their time on the main task, and participants were no longer able to view or edit the training or main task events.

3.5.4 Post-survey and Conclusion

After the participants completed the main task, they filled out a brief survey (see Appendix B for the complete text). This survey consisted of the NASA Task Load Index (TLX) [45] and two questions with free-form responses. The NASA TLX is a subjective, multi-dimensional assessment tool where subjects rate their perceived workload and is widely used in human factors research. In addition to being a standard assessment tool in human factors research including that of Sawyer [41], I have utilized the TLX to help ensure that my groups were balanced.

Once the survey was submitted, participants were given a completion code, which served as proof of completion, to write down on their paper slip, along with their name and email address, for entry into the raffle. After the participants turned in their paper slips the were free to leave. All of the participants completed the task in under an hour and were not pressured to hurry and finish.

Specificity, *sensitivity*, *precision* and *time on task* were calculated among all participants and within groups. I examined the relationship between these dependent variables individually within-subjects against the independent variable (false alarm rate) to determine if the false alarm rate impacts classification accuracy or time on task. I examined performance variances between categories of participants, i.e., between novices and experts as determined by the questionnaire. My analysis and discussion are supported by descriptive statistics. I also examined the NASA TLX scores with respect to participant performance to identify other possible co-factors that may account for participant performance unrelated to the false alarm rate. Finally, I examined the quantitative data through the lens of the post-study questionnaire to further help understand participants thought processes during the experiment.

4.1 Questionnaire Analysis

Prior to the experiment, all participants were required to take a questionnaire (see Appendix A for the full text of the questionnaire). The first three questions of the questionnaire consisted of self-report type questions regarding the participants' experience followed by seven multiple choice computer networking questions. The networking questions were used to evaluates the participants' experience levels. For example, I was skeptical of a participant that reported experience as a Network Administrator yet had trouble answering these basic network questions. In cases like these, those participants' were placed in the "Novice" group.

Most participants did poorly on the questionnaire. Of the 52 participants, only one answered all seven networking questions correctly. Additionally, only ten participants answered at least five questions correctly, which was the minimum to score better than 70%. Participants who answered at least five questions correctly but reported less than 1 year of experience in Network/IT Administration, Cyber Security, or Software Development were

Experience	Group 1	Group 2	Total
Cyber security	2	0	2
Network/IT	0	1	1
Software	0	4	4
Novice+	2	5	7
Novice	21	17	38
Total	25	27	52

Table 4.1: Participant experience group per treatment group

placed in a "Novice+" group. Participants who scored five or more and reported at least one year of experience in Network/IT Administration, Cyber Security, or Software Development were placed in a group labeled as such. All other participants were placed into the Novice group, even those who reported at least one year of experience in Network/IT Administration or Cyber Security but did poorly on the questionnaire. The exception to this are the participants who reported Software Development experience. I reasoned that while networking fundamentals certainly enrich a developer's understanding, it is not, strictly speaking, a requirement for the job. Each participant was placed in only one classification group, even if they reported multiple types of experiences. I placed the following priority on experiences:

1. Cyber Security
2. Network/IT Administration
3. Software Developtment
4. All other experience

Figure 4.1 lists the breakdown of how experience groups were distributed between Group 1 and Group 2. These distribution were by chance, but I feel experience among participants are fairly evenly distributed, though I would have preferred Group 1 and Group 2 to have shared the only cyber security experts.

The vast majority of participants, 41 out of 52, were sourced through MIS 310–Web Page Development Languages and MIS 322–Information Assurance and Security. The

remaining 11 participants were recruited either through an email sent to all Computer Science undergraduates or personal requests made to my technologically savvy friends and family.

4.2 Hypotheses Analysis

Data from 51 participants (n = 51) are included in this analysis. One Novice participant from Group 2 incorrectly answered one of the check events, and that individual's results are excluded from further analysis. The average *time on task* for the experiment was 16.65 minutes (median = 15.02, SD = 5.31). The average for the remainder of test variables among all participants were: *sensitivity* = 0.91 (median = 1.00, SD = 0.16), *specificity* = 0.71 (median = 0.73, SD = 0.17), *precision* = 0.45 (median = 0.33, SD = 0.31). My primary concern with these results is the average *time on task*. I had hoped participants would have spent more time considering each event during the experiment. My aim was for the average *time on task* to be approximately 30 minutes, and beta testing yielded a number much closer to that. I believe this result can be attributed to the fact that most of the participants were sourced from classrooms and that those participants were possibly uninterested in the experiment aside from the extra course credit to be awarded for their participation. In contrast, the average *time on task* for self-selected participants was 23.61 minutes (median = 21.82, SD = 4.55).

Participants were placed into one of two groups, and each group evaluated 50 events plus two "check" events. Group 1 was treated with a 50% false alarm rate where 25 events were "Don't escalate" events and 25 were "Escalate" events. Group 2 was treated with a 96% false alarm rate with 48 "Don't escalate" events and two "Escalate" events. The performance statistics per group are presented in Table 4.2; box-and-whisker charts of the data are presented in Figure 4.2. The mean *confidence* for the alarm decisions was 3.66 for Group 1 and 3.70 for Group 2, with medians both at 3.68.

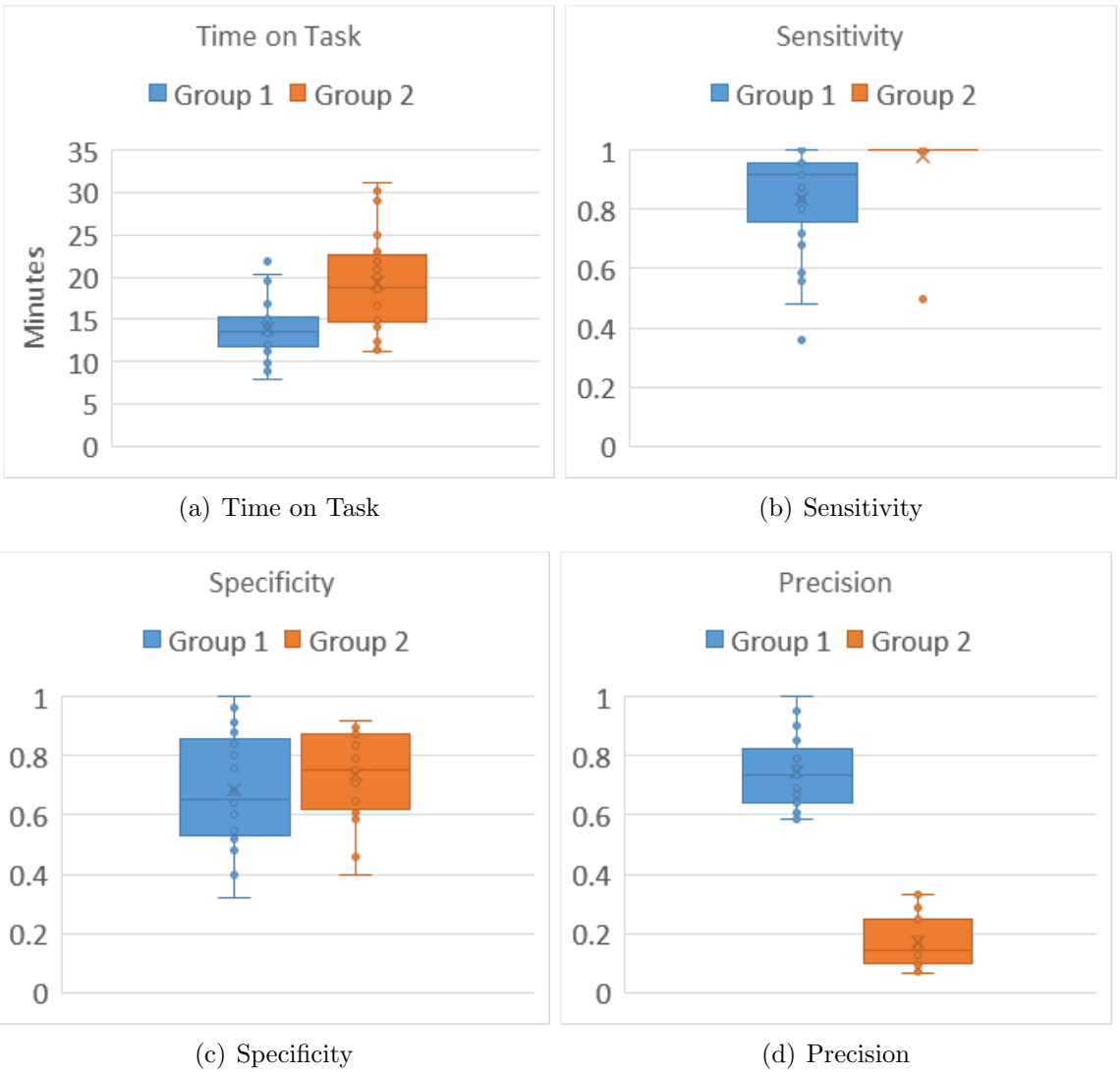
Figure 4.1(a) shows that participants who were placed in Group 2 spent, on average, approximately five and half minutes more on the experiment than participants from Group

Measure		Group 1	Group 2
count		25	26
time on task	mean	13.95	19.25
	median	13.45	18.76
	std.dev	3.43	5.50
sensitivity	mean	0.84	0.98
	median	0.92	1.00
	std.dev	0.17	0.10
specificity	mean	0.68	0.74
	median	0.65	0.75
	std.dev	0.19	0.14
precision	mean	0.75	0.17
	median	0.73	0.14
	std.dev	0.12	0.08

Table 4.2: Performance measures per treatment group

1. The reasons for this are unclear. I hypothesize this is due to the lack of true alerts. Group 2 participants were only shown two true alerts out of 50, so it is likely they spent more time debating whether an alert needed to be escalated because they did not have as good of an idea as to what a true alert looked like. Group 1, who saw an equal distribution of true to false alerts likely became more comfortable with what a true alert looked like, and therefore was able to make a determination faster. Another contributing factor to the time difference is that most of the self-selected participants, who took longer as discussed previously, were placed in Group 2.

Figure 4.1(b) shows that *sensitivity* among all participants was high, but participants from Group 2 rarely misidentified a true alert – in fact, only one true alarm from Group 2 was misclassified for all 26 participants in that group. I believe this is because they became accustomed to what benign activity looked like. Then, when a true alert presented itself, it was obvious to participants in Group 2 that those events should be escalated. Group 1, with their equal distribution of events, never achieved the same level of acclimation to either benign activity, thus leading to more frequent misidentification of true alerts.



(a) Time on Task

(b) Sensitivity

(c) Specificity

(d) Precision

Figure 4.1: Performance measures

Figure 4.1(c) shows that *specificity* between Groups 1 and 2 were similar with Group 2 having only slightly better results. Figure 4.1(d) shows a nearly 60% difference between groups in *precision*. I hypothesize that Group 1 became more accustomed to seeing events that needed to be escalated, thus they were better able to determine when an event did not need to be escalated. One participant from Group 2 stated, "I felt like there were only a few examples that seemed like they should definitely be escalated, but escalated many more than that so that a threat would not go unnoticed." This supports the hypothesis that participants from Group 2 chose "Escalate" on several events that they were unsure of

because it was the "safer" option. This mindset reflects the reality of cybersecurity practice as well: many organizations place such a priority on high sensitivity that they suffer through low specificity and precision.

4.2.1 Experience Group Performance

Table 4.3 shows the performance measures across the experience groups determined by the questionnaire. Participants with Cybersecurity or Network/IT experience tended to finish the experiment quicker than those without that experience. This could be because those participants felt more comfortable making the types of decisions that were presented during the experiment. *Sensitivity* among all experience groups were all relatively high, which indicates that most people are good at identifying true alarms; however, *precision* suffered among all participants.

Experience appears to have an impact on *precision*. The participants with cyber security experience have significantly higher *precision* than other groups. This reinforces Ben-Asher and Gonzalez's findings on expertise and performance [36]. This also supports Zhong et al.'s assertion that it should be senior analysts, not junior analysts as is typical, performing initial *triage* [29]. I believe Table 4.3 shows that the experience of participants in this study had an effect on performance. Unfortunately, the experience groups do not have enough participants for statistical analysis on this effect. In future work, I would make a concerted effort to balance the groups according to expertise.

The average confidence for participants in the Novice, Novice+, Software, and Cybersecurity groups were in the narrow range of 3.64–3.71. Only the Networking/IT group (which had only one member) exhibited a markedly different confidence: 4.67.

4.2.2 Questionable Participants

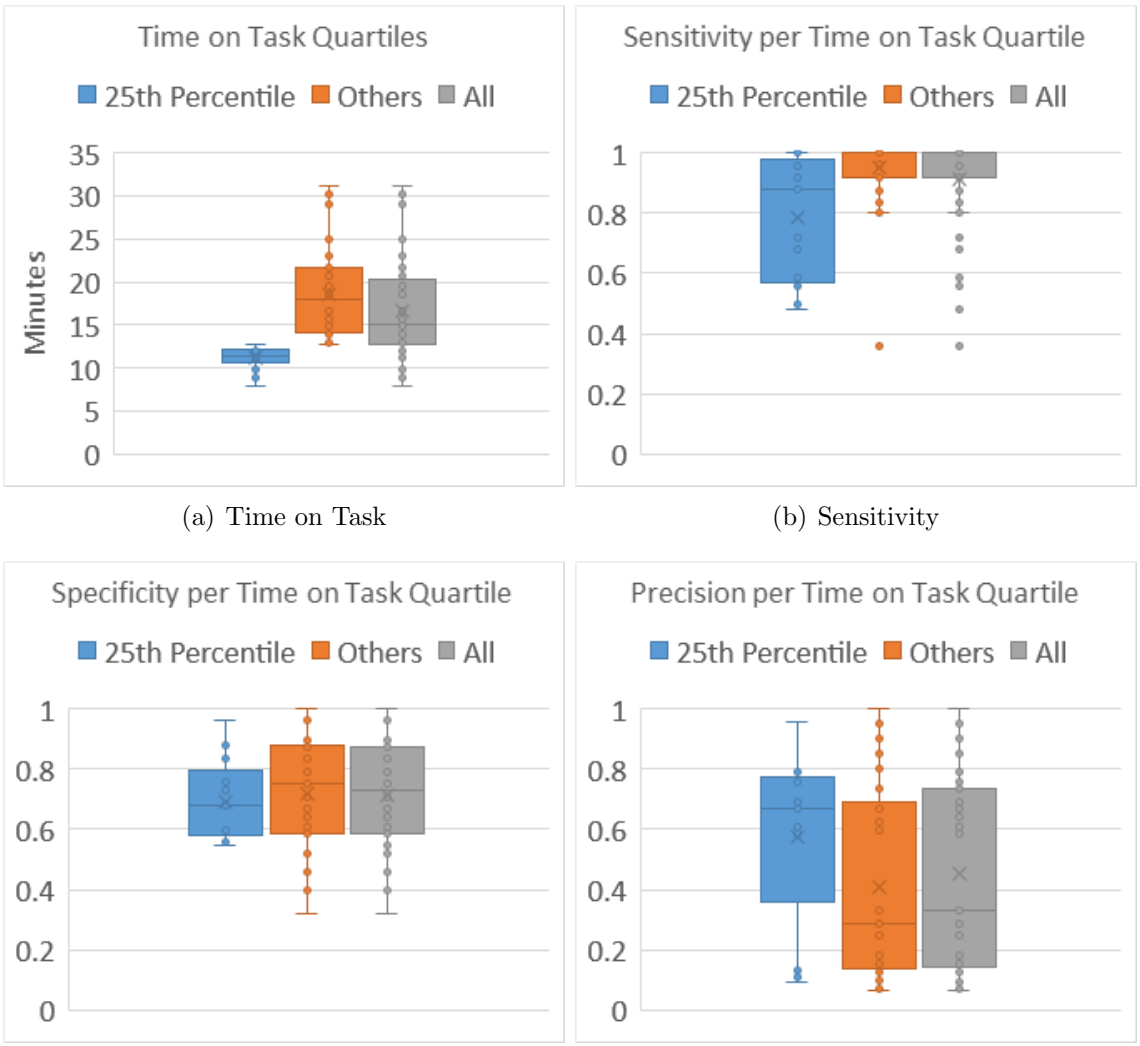
Based on classroom observation, I believe several participants sped through the experiment and question the legitimacy of their results. Had any of those participants proven to be outliers on *time-on-task*, their results would have been excluded from analysis; however,

Measure		Novice	Novice+	Software	Network/IT	Cybersecurity
count		36	7	5	1	2
time on task	mean	16.07	18.40	19.56	15.63	14.25
	median	14.59	17.20	20.23	-	-
	std.dev	4.76	6.50	6.88	-	0.77
sensitivity	mean	0.89	0.99	0.90	1.00	0.90
	median	0.98	1.00	1.00	-	-
	std.dev	0.16	0.03	0.2	-	0.1
specificity	mean	0.70	0.76	0.69	0.83	0.72
	median	0.67	0.75	0.77	-	-
	std.dev	0.17	0.16	0.19	-	0.08
precision	mean	0.49	0.37	0.23	0.2	0.77
	median	0.60	0.29	0.15	-	-
	std.dev	0.31	0.30	0.18	-	0.03

Table 4.3: Performance measures per experience group

that was not the case. With this in mind, I compared the results of the bottom *time on task* quartile (*time on task* < 12.74) against all other participants, shown in Fig C. *Specificity* between groups were similar, but *sensitivity* suffered overall for the bottom quartile. The drop in *sensitivity* for the bottom quartile is unsurprising as those participants rushed through the experiment and possibly guessing at random. The *precision* of the bottom quartile was generally higher than others; however, a closer inspection of the participants who comprised the bottom quartile shows that 75% of the those participants were placed into Group 1 (equal distribution of "Escalate" and "Don't Escalate" events). I believe this implies that this group of participants was more likely to choose "Escalate" for any given event. Being in Group 1 with the higher "Escalate" rate, compared to Group 2, may explain why the *precision* for the bottom quartile was higher compared to the rest of participants.

A reexamination of Group 1 and Group 2 results (see Fig 6.1 of Appendix C for all performance measures) with the bottom quartile for *time-on-task* removed yields notable changes in *sensitivity* (Fig 4.2.2). With the exception of one outlier, the remainder of Group 1's participants shared similar *sensitivity* levels, though the group's mean *sensitivity* changed



(a) Time on Task

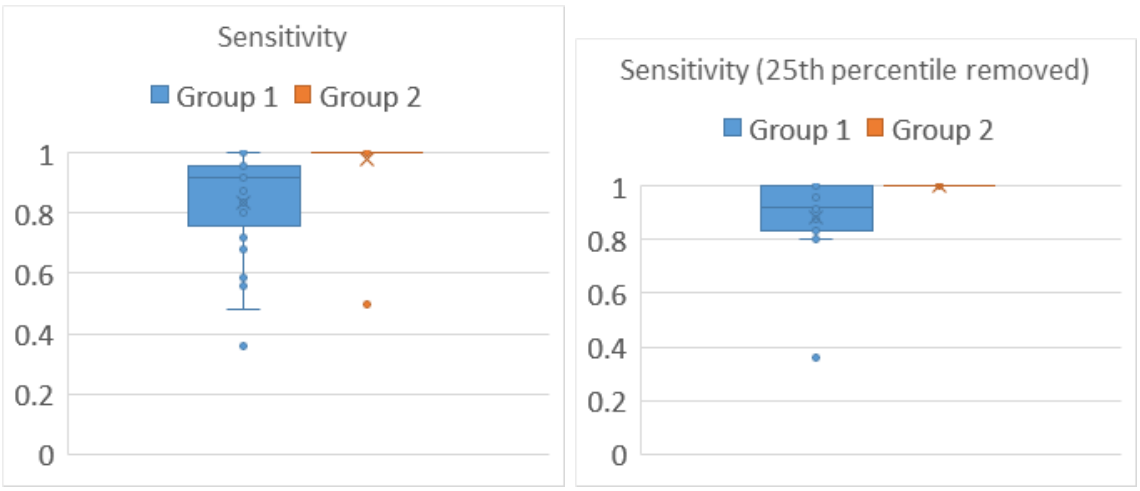
(b) Sensitivity

(c) Specificity

(d) Precision

Figure 4.2: Performance measures with 25th percentile of time on task (i.e., quickest people to complete the experiment) vs. other participants

only slightly ($\Delta = 0.04$). With the three participants in the bottom quartile from Group 2 removed, Group 2's *sensitivity* became perfect. This is further evidence that participants in the bottom quartile of *time-on-task* were more likely to select "Escalate". One interpretation of this behavior is that participants who were not invested in the experiment, or under a time constraint, were more likely to select "Escalate" because this is the "safe" option when dealing IDS alarms.



(a) Sensitivity

(b) Sensitivity with 25% removed

Figure 4.3: Sensitivity measures for entire sample and with 25th percentile of time on task removed

4.3 Survey Analysis

After experimentation, all participants completed a short survey (see Appendix B) that included the NASA Task Load Index (TLX) [45]. Results between the two groups are listed in Table 4.3; no significant difference between groups was observed. As the task asked of the participants was computer-based and lasted less than an hour, results of the TLX were unsurprising.

After participants completed the TLX, they were asked to list which information presented to them during the experiment was most helpful in making their determination on whether an event should be escalated. Table 4.5 presents a the items they listed and their frequency among participants. "Time" refers to the time between the locations presented in the events and the "Typical Travel Times" table listed in the Security Playbook. Also in the playbook was a table listing the concern level of each city presents (e.g. "Cities of concern") and a "Things to keep in mind" section. I believe "VPN" was the second most cited factor because if the analyst was confident that a VPN was being utilized it gives a legitimate explanation for the alerted impossible travel.

With regard to successful/failed login attempts, I suspect that these were not as help-

Measure		Group 1	Group 2
count		25	26
mental	mean	5.64	5.69
	median	6.00	6.00
	std.dev	1.32	1.59
physical	mean	1.76	1.23
	median	1.00	1.00
	std.dev	2.18	0.50
temporal	mean	3.52	3.12
	median	3.00	3.00
	std.dev	1.77	1.55
performance	mean	5.72	6.31
	median	6.00	6.00
	std.dev	1.73	1.23
effort	mean	5.12	4.65
	median	5.00	4.50
	std.dev	1.70	1.69
frustration	mean	3.76	3.04
	median	3.00	3.00
	std.dev	2.14	1.85

Table 4.4: Post-experiment survey TLX question data

ful because they did not convey any information related to travel times. While synthesizing events, I included those data points for each event because for a few events, in addition to the "impossible travel" nature of the alert, I wanted the failed login count to indicated that a brute force attempt was being made.

Topic	Count
Time	39
VPN	20
Location	18
# of failed login attempts	17
Source provider	16
Cities of concern	9
Successful logins	9
Login ratio	8
"Things to keep in mind"	3

Table 4.5: Most frequently appearing topics in responses to 'Which pieces of information were most useful...' survey question

5 DISCUSSION

In this study I examined the relationship of the false alarm rate of an IDS and its effect on human analysts with respect to performance (i.e. *sensitivity, specificity*, and *precision*) and *time-on-task*. I hypothesized that as IDS false alarm rate increases, human analyst performance in classifying alarms as true alarms or false alarms *decreases* and time on task *increases*.

The results indicate that analysts treated with a 96% false alarm rate had on average 39% *longer time on task* and 60% *lower precision* than participants treated with a 50% false alarm rate (Fig 4.2), i.e. analysts were more likely to misidentify a false alarm as a true alarm. Conversely, the group treated with the 96% false alarm rate had substantially *higher sensitivity* than the 50% group, though the magnitude of this difference was influenced by the speed in which some participants completed the experiment (Fig 4.2.2). The false alarm rate appears to have minimal impact on *specificity*, or the measure of accuracy in distinguishing false alarms from true alarms. In summary, I find that my hypotheses are partially supported by the data:

- *Precision* decreases as false alarm rate increases as hypothesized
- *Sensitivity* increases as false alarm rate increases, contrary to the original hypothesis
- *Specificity* does not decrease as false alarm rate increases, contrary to the original hypothesis
- *Time on task* increases as false alarm rate increases as hypothesized

The results show an increase in *sensitivity* with as false alarm rate increases, and this matches my own makes sense experience as a security operations analyst. I believe analysts who deal with an extremely high IDS false alarm rate become, in essence, an ideal anomaly detector. They become comfortable with what "normal" behavior looks like so they are keenly aware when something out of place happens. The factor at odds with this, *precision*, appears to decrease with a higher false alarm rate. Regardless of how attuned an analyst

may become with an environment, I think it is human nature "look for things that are not there." Therefore, when placed in an environment with a lower false alarm rate, the analyst is less likely to misidentify a false alarm because they become more aware of what a true alarm looks like.

The false alarm rate seems to have a clear effect on *time on task*. I believe this to be because of what is at stake if the analyst chooses incorrectly. I think analysts are more likely to give their "Don't escalate" decisions a second thought in an effort to persuade themselves that they are making the correct decision to ignore an alert. In the case of this study, a choice of "Escalate" is always the safer choice. However, we must remember that Group 2 had a higher percentage of self-selected participants, and that group was more likely to take longer.

Sawyer and Hancock found that the *prevalence effect* had a noticeable impact on his participants' ability to detect malicious emails [41]. Specifically, the rarer a malicious email, the less likely his participants were to detect it. My study found the opposite, the relatively rare false alarms in the 96% false alarm rate group were always detected, and the 50% group had high sensitivity as well. I agree with his assertions in the context of emails, but I do not think my results serve as an extension, nor a confirmation of those assertions. Our participant population was similar, comprised mostly of undergraduate students, but I think his scenario generalized better to the layman. I believe the general population is much more familiar with email and the risks associated with email than they are with cyber security, IDSes, and network attacks.

I believe the biggest factor to consider in interpreting these results, as found by Ben-Asher et al. [36], is experience. It is a lot to ask of a participant with little-to-no experience in security operations to accurately make judgements in an unfamiliar environment with only 15 minutes of training. Unfortunately, cyber security professionals are difficult to come by and their time is usually not cheap.

With all factors considered, I agree with the industry's current standard of placing

a high priority on high *sensitivity* at the expense of low *precision* and *specificity* for intrusion detection, regardless of the circumstances for individual networks. I believe there is evidence to suggest from these results, and from personal experience, that analysts become comfortable with high false alarm rates as it reinforces what normal activity looks like and highlights abnormal activity.

5.1 Threats to Validity

In this section I discuss the most relevant threats to validity in this study and how I addressed them according to the categories described by Wohlin et al.[46].

Conclusion validity concerns itself "...with issues that affect the ability to draw the correct conclusion about relations between the treatment and the outcome of an experiment. [46]" These are the threats to conclusion validity that I believe are relevant to my experiment and how I have addressed these concerns:

- Reliability of the treatment implementation — ensuring that implementation is similar over different participants and occasions. All participants were given similar instructions and directed to the same URL for the experiment which had a clear, linear flow from start to finish.

- Random irrelevancies in experimental setting — things outside the experimentation setting that may have disturbed the results. All participants were provided a quiet room for a testing environment. I was present for the duration of all experimentation and attest that there were no significant distractions or interruptions.

- Random heterogeneity of subjects — the level of heterogeneity, or its inverse, homogeneity, affects our ability to draw conclusions on a larger group based on the results of an erratic smaller group. The participants in my study are largely homogeneous in that most of them are undergraduate students in a CS/MIS/IT major. This is why I analyzed results with respect to experience levels — to extract some level of hetero-

geneity out of this group. Even so, most participants were classified as Novices. Thus, I have confidence that the results obtained were not unduly influenced by heterogeneity of the subjects.

Internal validity threats are "influences that can affect the independent variable(s) with respect to causality, without the researcher's knowledge."[46] Below are what believe to be the relevant internal validity threats:

- Maturation — this is the effect that participants react differently as time passes. Fig 5.1 shows a clear learning curve across all participants with respect to the average time to make a decision for each event. Participants becomes more comfortable with the task as time went on, though the curve shows signs of flattening out. I had originally planned to apply both treatments (50% false alarm rate and 96% false alarm rate) to each subject, but it is clear that the maturation effect would have skewed such results.

- Instrumentation — this essentially states that a poorly designed or constructed test will likely yield bad results. The design of this study was review and approved by my thesis committee. Beta testers were utilized after construction of the experiment and participants were trained on how to utilize the testing tool.

- Selection — this is the effect of natural variation in human performance. This was one of the more challenging threats to consider for this experiment. Ideally, I would have had an abundance of participants who were cyber security experts. In reality, it was very difficult to find experts in this field who were willing to participate. Even an invitation to all CSC/IT/MIS undergraduate students and CSIS graduate students to attend one of two open experimentation sessions only yielded nine participates. Thankfully, Dr. Cummings offered his MIS-310 and MIS-322 sections to me for experimentation, but this was a double-edged sword. While it greatly boosted my participant count, these participants were not self-selected and were more likely to not give a good-faith effort during testing. Every participant from the bottom quartile in *time-on-task* was

sourced from one of these two sections. By eliminating their results from analysis I observed a significant improvement in *sensitivity* in Group 1.

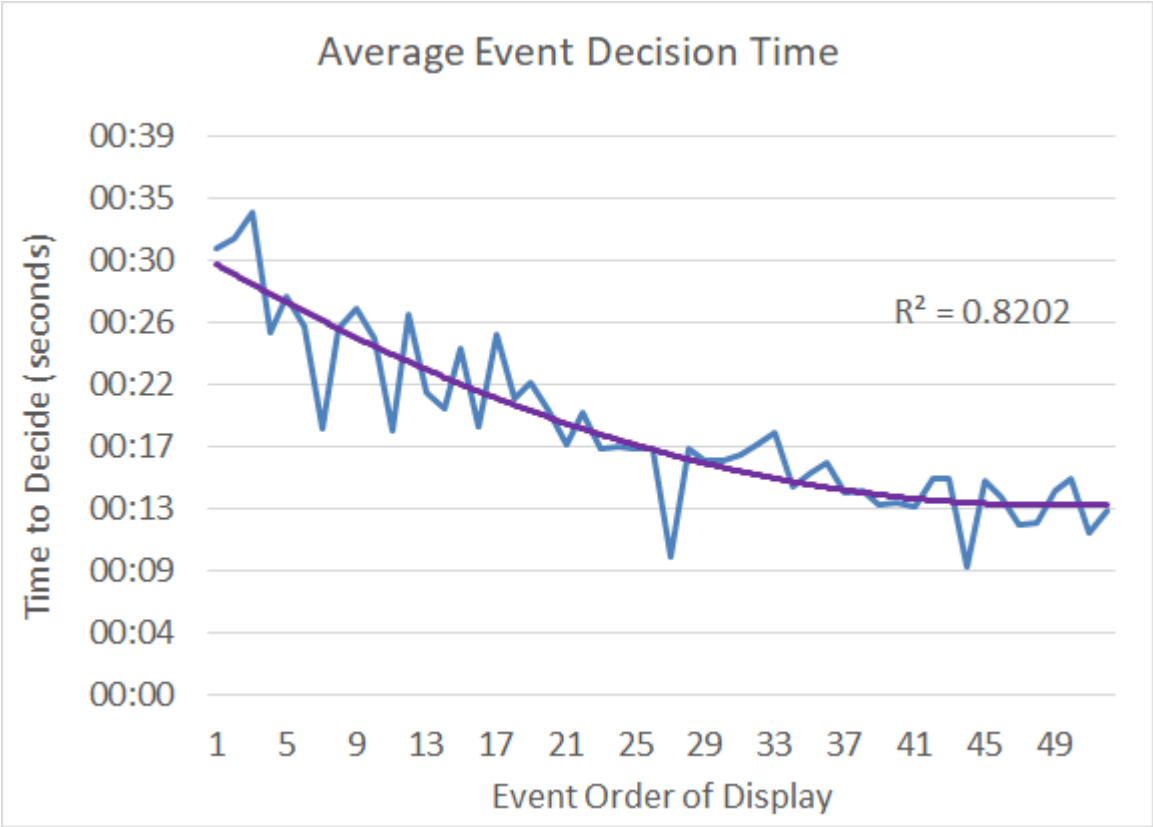


Figure 5.1: Average time to make a decision on an event across all participants.

Construct validity is the degree to which a test measures what it claims to measure. In this case, how well does this experiment test an analyst's ability accurate judge IDS alerts based on the IDS's false alarm rate. Below are the threats to construct validity I believe to be relevant to this study:

- Mono-operation bias — this is having participants perform the same task throughout the experiment. In this case, each task was to a binary decision as to whether an alert should be escalated. This limitation was accepted in order to simplify training for participants; and also, it is somewhat indicative of the monotony of investigations that cyber security analysts face each day.

- Interaction of testing and treatment — if the participants were to become sensitive to the false-positive rate they were being treated with, that could impact their performance on the test. To avoid this I explained to all participants that it is normal behavior for IDSes to generate a lot of false alarms and that the rate of false alarms generated depended greatly on how it is configured. Prior to beginning the main experimental task, participants were presented with a dialog that reminded them that a few, many, or all of the alerts there were about to see may be false alarms.

External validity concerns itself with applying generalized experimental results onto real world situations.

- Interaction of selection and treatment — this threat deals with having a participant population that does not represent the group to which we would like to generalize the study results for. Once again, ideally all participants for this study would have been cyber security experts, but that was not a realistic goal. Further, Zhong et al. [29] point out that many cyber security novices are tasked with triaging such alerts, and argue that this is not the best approach since experience is critical to performance.

- Interaction of setting and treatment — this is the effects of testing in an environment that is dissimilar to that of a realistic setting. In this case, experimentation took place in the controlled environment of a computer lab rather than a Security Operations Center (SOC). I believe in this case, the controlled environment is preferable as a testing environment as this study is narrow in scope of analyzing the affect of the false-positive rate of an IDS on an analyst. Testing in a live SOC could inject numerous uncontrolled distractions that could potentially affect performance. That makes for an interesting thought experiment, but is not ideal for this study. I believe, though, that this test experiment is remarkably similar to the day-to-day task I perform as a cyber security analyst triaging alarms at UNCW.

6 CONCLUSION

For this study I investigated to what extent the false alarm rate of an IDS affects the performance of a cyber security analyst. Prior research has shown that human analysts begin to distrust systems that consistently raise false alarms [9], [10], [12], and it is not uncommon for up to 99% of IDS alerts to be false alarms [8]. Much research has been dedicated to improving the accuracy of IDSes themselves, but it is ultimately a human analyst that must make the decision to investigate or ignore an alarm.

I developed Cry Wolf, a Flask web application that simulates a simple IDS by providing synthesized alerts to participants for evaluation. Participants were divided into groups treated with a 50% and 96% false alarm rates. I analyzed the results of 51 participants recruited primarily from the UNCW undergraduate community. Participants evaluated, based on information given, whether the IDS alerts were true alarms worth being escalated for review or false alarms that could be discarded.

Results from this experiment indicate that the false alarm rate does appear to impact analyst performance. As hypothesized, I observed a 39% increase in *time on task* and a 60% decrease in *precision* with a higher false alarm rate. Contrary to my hypotheses, *Specificity* seems largely unaffected while *sensitivity* improves with higher false alarm rate, though the latter is linked to response time.

I believe that analysts develop *situated knowledge* over time and become intimately familiar with the environment that they work in. Those who work with IDSes with extremely high false alarm rates become keenly *sensitive* to malicious behavior because of the stark contrast in how those events look when compared to benign activity. Those same analysts are plagued with a different human factor of "looking for something that is not there", so they become less *precise* in identifying malicious activity. I attribute the increase in *time on task* with regard to increased false alarm rate to analysts giving each decision to ignore an alert a second thought in order to boost confidence in their decision.

As Zhong et al. [28] and Ben-Asher and Gonazalez [36] point out, those participants with network and cyber security experience tended to perform better in this study. Contrary

to Sawyer and Hancock's results [41], the *rarity* of false-alarms seemed to improve analysts' *sensitivity* to malicious activity.

These results do give insight into the effects of the false alarm rate on analysts, but they are not without limitations. Threats to external validity include the nature of the controlled environment in which participants conducted the experiment. Threats to internal validity include the small sample size of cyber security experts.

6.1 Future Work

I think it would be a worthwhile endeavor to perform this experiment again, but with a concerted effort to find more cyber security experts. This would have been more of a priority had time limitations not been a factor. Also, to study what effect requiring participants to analyze several types of network attacks, not just limiting their analysis to one type (e.g. "Impossible travel") of alert.

It would seem that the *status quo* of cyber security analysts working in environments rife with false alarms is here for the foreseeable future. I believe the industry rightly puts a higher priority on *sensitivity* for its IDSes at the cost of *specificity* and *precision*. Until the day we are ready to relinquish more decision making control to computers, cyber security analysts will remain the last line of defense for network security. What we can do, as interest in STEM is on the rise, is to hope for an increase in the number of qualified workers in the cyber security field. I think we need to continue studying analysts in an effort to find the personal traits of a good analyst. Once we know those traits, we must work to identify those who possess them and cultivate them in the industry.

REFERENCES

[1] S. Axelsson, "The base-rate fallacy and the difficulty of intrusion detection," *ACM Transactions on Information and System Security*, vol. 3, no. 3, pp. 186–205, Aug. 2000.

[2] N. Manworren, J. Letwat, and O. Daily, "Why you should care about the Target data breach," *Business Horizons*, vol. 59, no. 3, pp. 257–266, 2016.

[3] H. Berghel, "Equifax and the Latest Round of Identity Theft Roulette," *Computer*, vol. 50, no. 12, pp. 72–76, Dec. 2017.

[4] IBM Security and the Ponemon Institute, "2018 Cost of Data Breach Study: Global Overview," Tech. Rep., 2018.

[5] Verizon, "2010 Data Breach Investigations Report, http://goo.gl/28pPGM," Verizon, Tech. Rep., 2010.

[6] S. Chandran Sundaramurthy, A. G. Bardas, J. Case, X. Ou, M. Wesch, J. McHugh, and S. Raj Rajagopalan, "A Human Capital Model for Mitigating Security Analyst Burnout," in *Eleventh Symposium On Usable Privacy and Security (SOUPS 2015)*, 2015, pp. 347–359.

[7] D. E. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering*, vol. 13, no. 2, pp. 222–232, 1987.

[8] K. Julisch, "Clustering intrusion detection alarms to support root cause analysis," *ACM Transactions on Information and System Security*, vol. 6, no. 4, pp. 443–471, 2003.

[9] J. P. Bliss, R. D. Gilson, and J. E. Deaton, "Human probability matching behaviour in response to alarms of varying reliability," *Ergonomics*, vol. 38, no. 11, pp. 2300–2312, Nov. 1995.

[10] J. Meyer, "Effects of Warning Validity and Proximity on Responses to Warnings," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 43, no. 4, pp. 563–572, Dec. 2001.

[11] M. R. Endsley, *Designing for Situation Awareness*. CRC Press, Apr. 2016.

[12] S. Breznitz, *Cry wolf: the psychology of false alarms*. Hillsdale, NJ: Lawrence Erlbaum Associates, 1984.

[13] J. P. Bliss and M. C. Dunn, "Behavioural implications of alarm mistrust as a function of task workload," *Ergonomics*, vol. 43, no. 9, pp. 1283–1300, Sep. 2000.

[14] M. Sazzadul Hoque, "An Implementation of Intrusion Detection System Using Genetic Algorithm," *International Journal of Network Security & Its Applications*, vol. 4, no. 2, pp. 109–120, 2012.

[15] J. Ryan, M.-J. Lin, and R. Miikkulainen, "Intrusion Detection with Neural Networks," in *AI Approaches to Fraud Detection and Risk Management: Papers from the 1997 AAAI Workshop*, Providence, RI, 1997, pp. 72–79.

[16] H. Om and A. Kundu, "A hybrid system for reducing the false alarm rate of anomaly intrusion detection system," in *2012 1st International Conference on Recent Advances in Information Technology, RAIT-2012*, Dhanbad, India, Mar. 2012, pp. 131–136.

[17] R. Bace and P. Mell, "NIST Special Publication on Intrusion Detection Systems," National Institute of Standards and Technology, Tech. Rep., 2001.

[18] James P. Anderson Co., "Computer Security Threat Monitoring and Surveillance," Tech. Rep., 1980.

[19] J. Rasmussen, *Information processing and human-machine interaction : an approach to cognitive engineering*. New York: Elsevier Science Ltd, 1986, p. 215.

[20] C. D. Wickens and J. G. Hollands, *Engineering Psychology and Human Performance*, Second. New York: Harper Collins, 1992.

[21] B. H. Deatherage, "Auditory and other sensory forms of information presentation," in *Human Engineering Guide to Equipment Design*, Washington, DC, 1972.

[22] R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, and M. Zissman, "Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation," in *Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00*, Hilton Head, SC, 2000, pp. 12–26.

[23] J. McHugh, "Testing Intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by Lincoln Laboratory," *ACM Transactions on Information and System Security*, vol. 3, no. 4, pp. 262–294, Nov. 2000.

[24] 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," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 49, no. 3, pp. 229–233, Sep. 2005.

[25] J. M. Schraagen, S. F. Chipman, and V. L. Shalin, *Cognitive task analysis*. L. Erlbaum Associates, 2000.

[26] B. Crandall, G. A. Klein, and R. R. Hoffman, *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*. MIT Press, 2006.

[27] A. D'Amico and K. Whitley, "The real work of computer network defense analysts: The analysis roles and processes that transform network data into security situation awareness," in *Proceedings of the Workshop on Visualization for Computer Security*, Berlin, Heidelberg, 2008, pp. 19–37.

[28] C. Zhong, J. Yen, P. Liu, R. Erbacher, R. Etoty, and C. Garneau, "An integrated computer-aided cognitive task analysis method for tracing cyber-attack analysis processes," in *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security - HotSoS '15*, New York, New York, USA, 2015, pp. 1–11.

[29] C. Zhong, T. Lin, P. Liu, J. Yen, and K. Chen, "A cyber security data triage operation retrieval system," *Computers & Security*, vol. 76, pp. 12–31, Jul. 2018.

[30] J. R. Goodall, W. G. Lutters, and A. Komlodi, "Developing expertise for network intrusion detection," *Information Technology & People*, vol. 22, no. 2, pp. 92–108, Jun. 2009.

[31] J. Bédard and M. T. Chi, "Expertise," *Current Directions in Psychological Science*, vol. 1, no. 4, pp. 135–139, Aug. 1992.

[32] C. W. Mills, "Situated Actions and Vocabularies of Motive," *American Sociological Review1*, vol. 5, no. 6, pp. 904–913, 1940.

[33] L. Suchman, *Plans and situated actions: The problem of human-machine communication*. Cambridge, MA: Cambridge University Press, 1987.

[34] W. J. Clancey, "The Conceptual Nature of Knowledge, Situations, and Activity," in *Human and Machine: Expertise in Context*, Menlo Park, 1997, pp. 247–291.

[35] F. L. Schmidt and J. E. Hunter, "Tacit Knowledge, Practical Intelligence, General Mental Ability, and Job Knowledge," *Current Directions in Psychological Science*, vol. 2, no. 1, pp. 8–9, Feb. 1993.

[36] N. Ben-Asher and C. Gonzalez, "Effects of cyber security knowledge on attack detection," *Computers in Human Behavior*, vol. 48, pp. 51–61, Jul. 2015.

[37] S. Mahoney, E. Roth, K. Steinke, J. Pfautz, C. Wu, and M. Farry, "A Cognitive Task Analysis for Cyber Situational Awareness," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 54, Sep. 2010, pp. 279–283.

[38] L. Layman, S. D. Diffo, and N. Zazworka, "Human factors in webserver log file analysis," in *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security - HotSoS '14*, New York, New York, USA, 2014, Article No. 9.

[39] 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, Sep. 2010.

[40] R. S. Gutzwiller, S. M. Hunt, and D. S. Lange, "A task analysis toward characterizing cyber-cognitive situation awareness (CCSA) in cyber defense analysts," in *2016 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, San Diego, CA, Mar. 2016, pp. 14–20.

[41] B. D. Sawyer and P. A. Hancock, "Hacking the Human: The Prevalence Paradox in Cybersecurity," *Human Factors*, vol. 60, no. 5, pp. 597–609, Aug. 2018.

[42] P. A. Hancock and J. S. Warm, "A dynamic model of stress and sustained attention," *Human Factors*, vol. 31, no. 5, pp. 519–537, 1989.

[43] S. R. Mitroff and A. T. Biggs, "The Ultra-Rare-Item Effect," *Psychological Science*, vol. 25, no. 1, pp. 284–289, Jan. 2014.

[44] C. Sabottke, D. Chen, L. Layman, and T. Dumitraş, "How to trick the Borg: threat models against manual and automated techniques for detecting network attacks," *Computers & Security*, vol. 81, pp. 25–40, Mar. 2019.

[45] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," *Advances in Psychology*, vol. 52, no. C, pp. 139–183, Jan. 1988.

[46] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Berlin Heidelberg: Springer, 2012.

APPENDICES

A Prequestionnaire

The following questionnaire was administered to all participants via the Cry Wolf platform prior to engaging in the training task:

Your background
Answer these questions honestly. Your answers have no impact on the rest of the experiment or your eligibility to receive compensation.

1. Which role best describes your current experience?
    - Student
    - Researcher
    - IT/Network Administrator
    - Software Engineering (developer, tester, project management, etc.)
    - Cyber Security Specialist
2. How many years of professional experience do you have in the following (select all that apply):
    - Researcher (No Experience, <1, 1-5, 5-10, 10+)
    - IT/Network Administrator (No Experience, <1, 1-5, 5-10, 10+)
    - Software Engineering (No Experience, <1, 1-5, 5-10, 10+)
    - Cyber Security Specialist (No Experience, <1, 1-5, 5-10, 10+)
3. How familiar are you with Internet cyber security attacks? (select all that apply)
    - None/very little
    - I have read about how attacks work
    - I have attacked or defended against an attack in a controlled setting
    - I have defended or investigated attacks on a public network
    - I have engineered software that explicitly involves attacks or defense of network cyber security attacks

Networking Questions
Answer these questions to the best of your ability. Do not use any outside resources, including the Internet or another student. Your answers have no impact on the rest of the experiment or your eligibility to receive compensation.

5. What is the subnet mask for 173.67.14.127 / 24?
    a) 173.67.14.127

b) 173.67.14.0
c) 255.255.255.0
d) 255.255.255.24
e) I don't know

6. What is the network address for 173.67.14.127 / 24?

a) 173.67.14.127
b) 173.67.14.0
c) 255.255.255.0
d) 255.255.255.24
e) I don't know

7. True or False: TCP is a faster, more efficient transfer protocol compared to UDP?

a) True
b) False
c) I don't know

8. Which port does HTTP use by default?

a) 80
b) 443
c) 587
d) 5000
e) I don't know

9. A network security device that monitors incoming and outgoing network traffic and decides whether to allow or block specific traffic based on a defined set of security rules is the definition of a(n):

a) Honeypot
b) Firewall
c) Botnet
d) Intrusion Detection System
e) I don't know

10. The combination of an IP address and the port used is called a:

a) Socket
b) MAC Address
c) Protocol

d) Ping

e) I don't know

11. Network, Internet, Transport, Application are the layers of which networking model?

a) OSI

b) TCP/IP

c) UML

d) HTTPS

e) I don't know

B Post-task survey

The following questions were administered via the Cry Wolf platform after participants completed the main task:

NASA TLX Questions [45]

1. *Mental Demand*: How much mental and perceptual activity was required (e.g. thinking, deciding, calculating, remembering, looking, searching, etc.)? Was the task easy or demanding, simple or complex? (Very Low 0–10 Very High)

2. *Physical Demand*: How much physical activity was required (e.g. pushing, pulling, turning, controlling, activating, etc.)? Was the task easy or demanding, slack or strenuous? (Very Low 0–10 Very High)

3. *Temporal Demand*:How much time pressure did you feel due to the rate or pace at which the tasks occurred? Was the pace slow and leisurely or rapid and frantic? (Very Low 0–10 Very High)

4. *Performance*: How successful do you think you were in accomplishing the goals of the task set by the experimenter (or yourself)? How satisfied were you with your performance in accomplishing these goals? (Failure 0–10 Perfect)

5. *Effort*: How hard did you have to work (mentally and physically) to accomplish your level of performance? (Very Low 0–10 Very High)

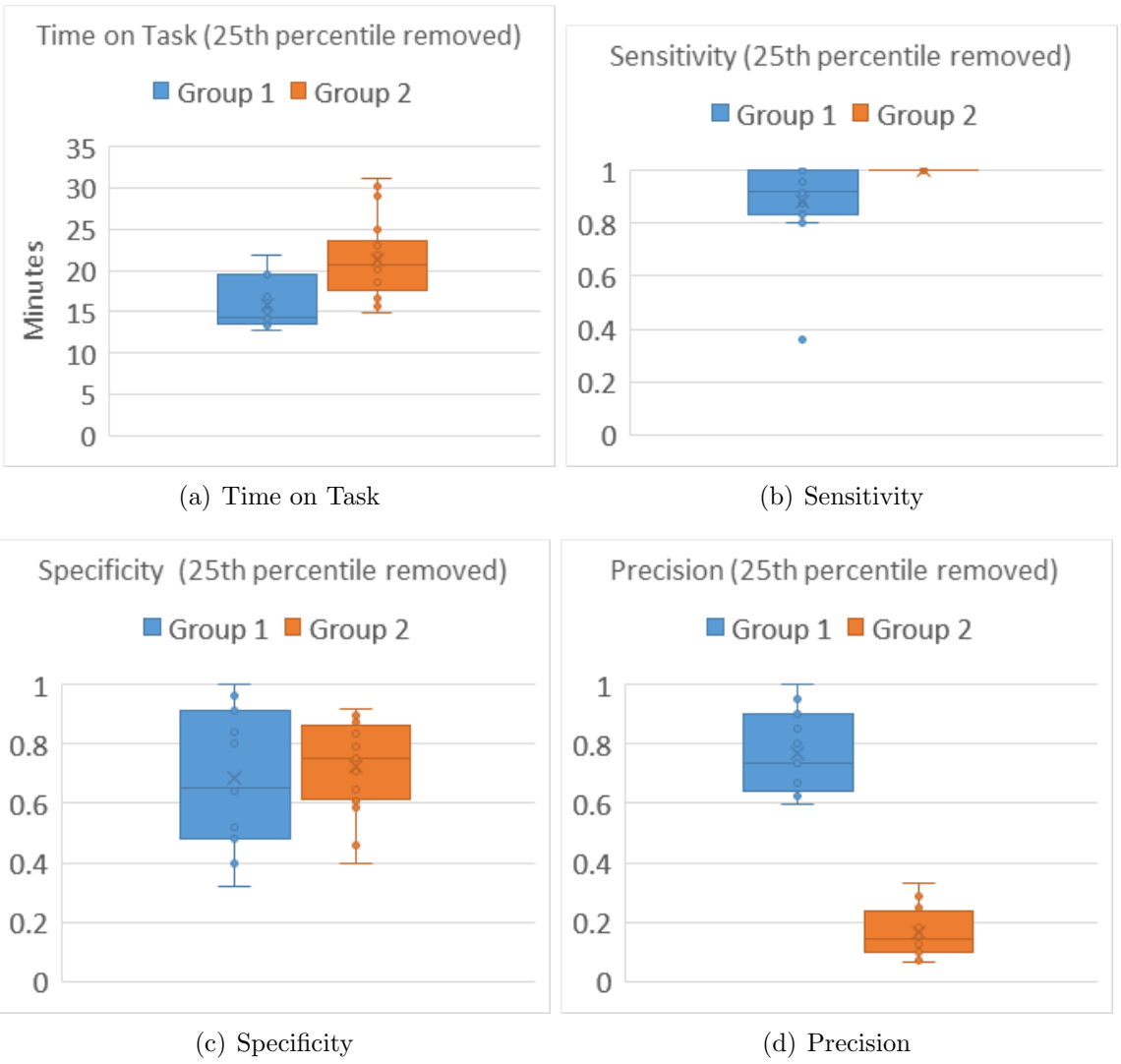
6. *Frustration*: How insecure, discouraged, irritated, stressed, and annoyed versus secure, gratified, content, relaxed, and complacent did you feel during the task? (Very Low 0–10 Very High)

Other feedback

8. Which pieces of information were most useful in determining whether or not an alert should be escalated? What did you focus on the most? You can refer back to the Security Playbook to remind yourself of the alert details. (Free text)

9. Is there any other feedback you would like to provide? (Free text)

C Supplemental Charts and Tables



(a) Time on Task

(b) Sensitivity

(c) Specificity

(d) Precision

Figure 6.1: Performance measures with 25th percentile of time on task removed

BIOGRAPHICAL SKETCH

William Troy Roden is originally from Albertville, Alabama, but now calls Wilmington, NC home. He earned his Bachelor of Science in Commerce and Business Administration from the University of Alabama with a major in Small Business Management/Entrepreneurship in May 2011. After college he commissioned as an officer in the United States Marine Corps and served as a Logistics Officer until November 2014. In 2017 he returned to school to pursue a Master of Science in Computer Science and Information Systems at the University of North Carolina – Wilmington.