MOBILE NETWORK DEFENSE INTERFACE FOR CYBER DEFENSE AND SITUATIONAL AWARENESS

THESIS

James C. Hannan, Captain, USAF

AFIT-ENG-13-M-21

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT-ENG-13-M-21

MOBILE NETWORK DEFENSE INTERFACE FOR CYBER DEFENSE AND
SITUATIONAL AWARENESS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

James C. Hannan, B.S.E.E.

Captain, USAF

March 2013

AFIT-ENG-13-M-21

MOBILE NETWORK DEFENSE INTERFACE FOR CYBER DEFENSE AND

SITUATIONAL AWARENESS

James C. Hannan, B.S.E.E.
Captain, USAF

Approved:

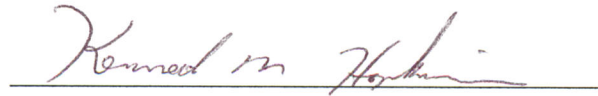_____   7 Mar 2013
Maj Kennard R. Laviers, PhD (Chairman)   Date

_____   7 MAR 2013
Lt Col Brett J. Borghetti, PhD (Member)   Date

_____   7 Mar 2013
Kenneth M. Hopkinson, PhD (Member)   Date

AFIT-ENG-13-M-21

Abstract

Today's computer networks are under constant attack. In order to deal with this constant threat, network administrators rely on intrusion detection and prevention services (IDS) (IPS). Most IDS and IPS implement static rule sets to automatically alert administrators and resolve intrusions. Network administrators face a difficult challenge, identifying attacks against a vast number of benign network transactions. Also after a threat is identified making even the smallest policy change to the security software potentially has far-reaching and unanticipated consequences. Finally, because the administrator is primarily responding to alerts they may lose situational awareness of the network.

During this research a Mobile Network Defense Interface (MNDI) was created that visualized a live network under cyber attack. MNDI allowed test subjects to take actions and make configuration changes in real time on the network. The interface was designed to take advantage of state-of-the-art touch technology engaging the network administrator in the defense of the network. MNDI increased administrator's ability to make time-sensitive decision quickly and accurately on their network.

MNDI was tested against a set of open source network administration tool implemented on a desktop system. Both systems used an automated system that polled an expert system (ES) to resolve zero to 75% of the alerts. The amount of alerts resolved is referred to as level of automation (LOA). During the experiment MNDI outperformed the desktop configuration at all LOAs. The test results showed a statistical difference between the percentage of alerts correctly resolved and the time between actions on MNDI versus desktop test subjects.

Table of Contents

List of Figures

Figure   Page

List of Tables

List of Acronyms

Acronym	Definition
AP	access point
BASE	Basic Analysis and Security Engine
CPU	centrel processing unit
ES	expert system
FTP	file transfer protocol
GB	gigabytes
GHz	gigahertz
IDS	intrusion detection systems
IID	interactive incident diagram
iOS	iPhone Operating System
IP	Internet protocol
LOA	level of automation
MNDI	Mobile Network Defense Interface
Nmap	network mapper
NIMBLE	Network Intrusion Management Benefiting from Learned Expertise
OS	operating system
RAM	random access memory
SUT	system under test
XML	extensible markup language

MOBILE NETWORK DEFENSE INTERFACE FOR CYBER DEFENSE AND
SITUATIONAL AWARENESS

I.   Introduction

1.1   Anatomy of an Attack

Defining and understanding the network computer security posture during a
common attack methodology is vital to successfully implementing an attack pattern
visualization interface. Most cyber attacks follow a common methodology. Cyber
attacks typically occur in five phases: reconnaissance, scanning, gaining access at
the operating system and application level, gaining access at the network level, and
covering tracks [32]. This thesis refers to phases three through five as the attack
phase.

Reconnaissance is the process of gathering information about the target [28].
During this phase attackers gain an appreciation of their target's organizational
structure and business practices. Attackers attempt to learn everything from who
is responsible for maintaining the network to what network security policies are in
place to protect network services. Attackers accomplished this through a variety of
means from sophisticated social engineering scams to researching publicly available
websites. They also resort to less sophisticated means such as searching through a
company's trash. The end product for the attacker is a list of potential targets such
as telephone numbers, domain names, Internet protocol (IP) addresses, usernames
and passwords.

Once the attacker has gathered information they are ready to begin phase two,
scanning. During this phase the attacker is testing the defenses of the target.

1

Scanning phase favors the attacker because they only have to be right about a single vulnerability, conversely security professionals must address every vulnerability [9]. While companies are busy with daily business and securing their infrastructure attackers have the luxury of time on their side to perform methodical scans over numerous days, week or even months [9]. Common tactics in this phase include war driving and network mapping. War driving is a tactic to identify vulnerable wireless access point (AP) by taking advantage of the fact that most AP's broadcast their existence for ease of accessibility. Network mapping involves using publicly available tools, such as network mapper (Nmap), to ping the network to identifying live host and open ports. Identifying a live host requires a more sophisticated tool, such as Nessus. Nessus is also able to identify misconfigured or unpatched host providing a list of exploitable vulnerabilities.

The final phase is the attack phase where all the vulnerabilities are tested in an attempt to gain unauthorized access to the network [15]. The Metasploit framework is a publicly available software suite that provides attacks with a remarkable amount of preconfigured exploits and allows attackers to create their own exploits.

The task of securing the network is daunting, especially given the availability of numerous, free and sophisticated tools. From an attacker's perspective, performing an attack is relatively low risk and cost, the cost to the victim could be crippling. Fortunately for network administrators an attack rarely goes unseen. All three phases of the attack methodology leave footprints in event logs up until a root kit compromises the host. Timely, actionable information is critical to an administrator's ability to thwart the attacker's attempts. Timely, actionable information is critical to a security professional's ability to thwart the attackers attempts.

1.2 Network Security

Many businesses and government agencies depend on networked computer systems to perform their daily tasks. Compromised networks severely degrades the ability of the organization to achieve their objectives, and in the case of the military, may endanger lives [3, 22]. Today's networks are under constant attack. The CSI Computer Crime and Security Survey is a polling of the 522 computer security practitioners in corporations, government, financial institutions, medical institutions and universities in the United States. According to the 2008 survey, 46% of the 522 entities identified security events within their network, costing an average of $500,000 to remedy [26]. Countless resources are poured into network security configurations, monitoring of network assets, identifying damage done by malicious activity and restoring the network after malicious events or negligent behavior occurs. However, very few, if any attacks, go unnoticed by the multitude of network monitoring systems that are implemented on any given network [17]. The root problem lies in sifting through the benign traffic to quickly and accurately identify the malicious or negligent behavior that may compromise the network [14].

From an administrator's perspective, network security is comprised of the policies and configuration in place to control the flow of data in and out of the network [30]. Network security is comprised of three primary components: requirements, policy and mechanisms [4]. Defining security requirements involves identify what the organization's security goals are. For example, consider the contrasting goals of a university versus an organization that focuses on cryptology [4]. A university's primary goal is the free flow of information with some protected information, such as student's grades or social security numbers. An organization that focus on cryptology, in contrast, is more interested in securing information versus open sharing [4]. The contrasting requirements drive the policies and configuration that are put into place

regarding how users may access the network. The policies and configurations become both the mechanism for enforcement, as well as the vulnerabilities the administrator must address to keep the network operational.

Once the requirements and policies are established the mechanisms to enforce the policy are put into place. Common mechanisms include intrusion detection systems (IDS), firewalls and user authentication methods. The aforementioned systems and mechanisms, as well as other entities create event logs that trace the entities behavior on the network. The event logs also document all the user's actions on the network. Alarms are configured to alert administrators of user's attempts to circumvent the network's policies. However, for the administrators to configure alarms and account for every vulnerability or user action poses a significant challenge. Diligent administrators attempt to identify malicious behavior amongst all recorded data and also address all vulnerabilities in the policy and configurations that allow unwanted behavior in order to protect the network. The mobile network interface developed for this research seeks to minimize the time required to gain situational awareness of the network. Seeking to increase administrator's effectiveness in alert resolution and making any policy or configuration changes to better protect the network.

1.3 Network Security Visualization

Network security visualization seeks to address this issue by presenting the networks state visually augmenting traditional review of textual log files. Evaluations by Goodall and Rasmussen concluded that humans with some networking experience are able to quickly grasp well designed visualization tools and more importantly, identify malicious behavior on the network quicker than those without visualization tools [9, 24]. While visualization tools have shown that they can be a vital asset on the network, they have two major pitfalls: they often do not provide the administrators

4

with the ability to take action on the network and, like the textual logs before them, they often create information overload for the administrators. Network administrators need a simplified, interactive interface that provides actionable data in a timely manner that leads to accurate decision making on network security events.

Due to the 24-hour nature of cyber warfare, network professionals can benefit from mobile, actionable access to their networks. Advances in mobile technology have created an environment well suited to address this shortfall. Faced with limited processing power and less visual real estate, mobile software developers have become adept at focusing on aspects of the applications that are most important to the user. This developmental mindset helps ensure successful products focus on what the user needs to accomplish their task and filters extraneous information. An actionable, mobile visualization of network status that takes advantage of intuitive touch-based visualization, could aid network administrators in making prompt, accurate decision regarding network security incidents. The hypothesized gain would be expected to reduce the damage done and recovery time needed to return the network to normal operations.

1.4 System Overview

To address the gaps in current cyber defense methodology, a novel system was conceived to increase the effectiveness of administrators. The system consists of a network interface, network modeler, action/plan generation and an automation setting, Figure 1.1. The network interface polls the network model for the status of the network and visualizes it. When an alert occurs on the network, the alert impact level is compared to the automation setting and a plan is constructed. If the impact level is below the threshold of the current automation setting the user is presented with a plan to resolve the alert. The plan contains actions that would reasonably resolve the alert. If the impact exceeds the threshold of the current automation

setting an action is automatically generated by an expert system (ES) to resolve the alert. This research focuses on the network interface, planning system and the user's interaction with the network.
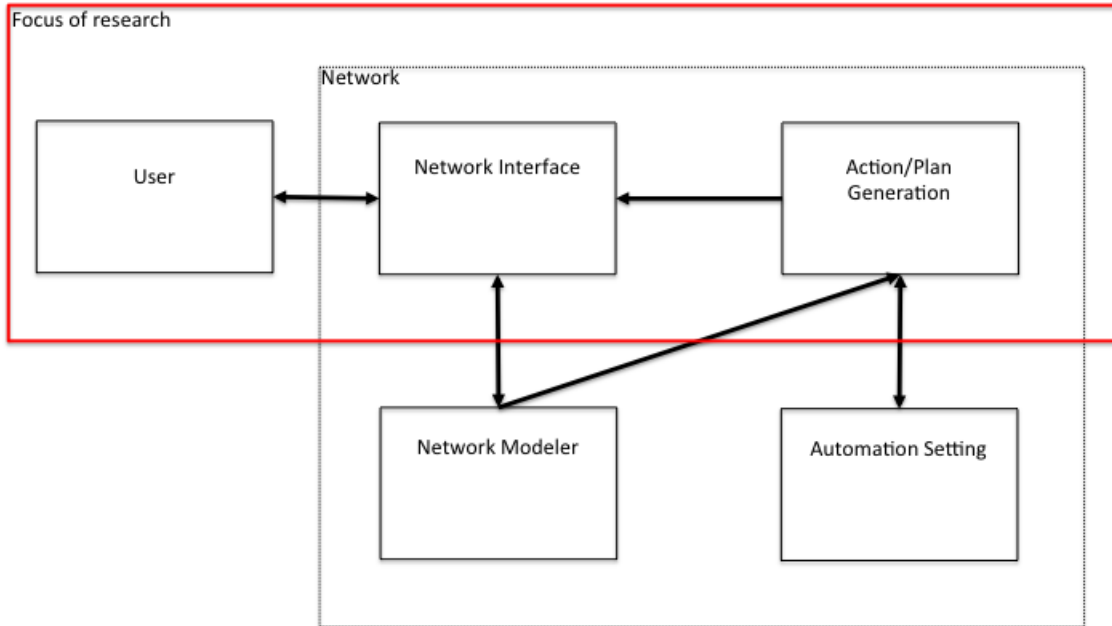
Figure 1.1: System Overview

1.4.1   Mobile Network Defense Interface (MNDI).

Keeping pace with the dynamic and fast paced environment of network administration requires integration of mobile device within network management. Advances in mobile technology from both platform and security aspects make this a viable option in today's environment [10]. However, simply maintaining situational awareness of the network is not enough. Network managers must be able to interact with the network to make time sensitive decisions.

MNDI melds visualization techniques with an implementation that is able to interact with the network. MNDI is an iPhone Operating System (iOS) application

that leverages a touch-based graphical interface for increased user interaction and visually appealing graphics, which enhances understandability. MNDI provides many benefits to administrators, such as the capability to take actions on the network or alter configurations, even if they are away from their desk. Administrators are often required to sift through benign traffic and false alarms to identify which alerts can actually harm the network. MNDI mitigates this task by pulling in data from a network modeler that classifies the impact level against the current network configuration [25] This ensures the administrator is aware of the potential damage the alert could cause. Finally, if MNDI receives an alert with a known IDS signature it presents the administrator with a list of actions that reasonably resolve the alert.

1.5 Overview

Chapter 2 is the literature review of the components that are incorporated into the MNDI. This chapter includes a review of network visualization, mobile interface design and automation techniques. Chapter 3 discusses MNDI interface design and the experiment that was developed to test it against a more traditional desktop interface. Chapter 4 contains the results of the experiment. Analysis is conducted on test subject's abilities to resolve alerts correctly and quickly at varying levels of automation. The data set is further broken down into the skill level of the individuals and analyzed. Chapter 5 summarizes the results and provides an evaluation of the quality of the research's MNDI interface versus the desktop interface. It is the conclusion of this research that MNDI's capabilities significantly improve administrator abilities to defend a network during a cyber attack.

II.   Literature Review

Network administrators face a difficult challenge, identifying malicious traffic amongst all the benign network transactions. Traditionally, administrators relied upon event logs and automated systems to maintain the network [18]. Three significant challenges exist in the current environment:

1. Numerous alerts make text-based manual analysis tiresome and error prone

2. Threshold adaptation is difficult because small changes have unpredictable effects

3. Missing contextual information makes interpreting the alerts difficult leading to misidentification of alerts [18]

Sorting through intrusion detection systems (IDS) logs is a cumbersome task and even the best network administrators struggle to identify malicious activity buried amongst everyday traffic. The problem compounds with the sheer number of alerts generated by the network monitoring systems [18]. Currently, the monitoring systems are not sophisticated enough to filter out benign activity and only present malicious activities for resolution. The combination of the above capability gaps leaves the administrators either missing key data or wasting valuable time researching false positives.

Maintaining situational awareness in the cyberspace domain is vital to the administrator's ability to effectively manage the network. Establishing cyber situational awareness involves answering the following questions [27]:

1. Am I under attack; what is the nature and origin?

2. What are the attackers doing; what might they do next?

3. How does it affect my mission?

4. What defenses do I have that will be effective against this attack?

5. What can I do about it? What are my options?

6. How do I choose the best option?

7. How do I prevent such attacks in the future?

Visualizing events logs is a promising technique for increasing network administrator's situational awareness [19]. Visually analyzing data helps network administrators perceive patterns, trends, structures, and exceptions in complex data sources [29].

Correlating information quickly is vital, each analyst may be monitoring multiple systems with only a few minutes to devote to each alarm [24]. Attackers are now using automated systems to make their attacks more sophisticated and widespread. Botnets create a large portion of unwanted, potentially malicious network traffic, most of which is malicious [7]. It is estimated that 27% of unwanted network traffic is generated by botnets [7].

Visually displaying log files allows network administrators to visually recognize abnormal behavior, reducing the processing time of manually reviewing multiple text log files. However, visualizations are not immune to the shortcomings of other network management software: information overload, counter intuitive interfaces and lack of actionable information.

2.1 Visualization

Visualizing network logs provides network administrators with a more sophisticated method to monitor and react to attacks on their networks. Administrators who have well defined tasks perform more accurately using visualization interfaces versus

textual interfaces [9]. Using the visualization interface users were more apt to draw correlations between events and the data being visualized, than when simply parsing through a textual interface [9].

When developing a strategy for cyber attack resolution, network administrators must address the following six items [8]:

1. Identify the incident

2. Evaluate the incident

3. Determine how prevalent the problem is and what else is being affected

4. Drill down data to identify patterns and test hypotheses

5. Mark and report results to communicate information to others

6. Direct a response

Modern human factor theory suggests that the effectiveness of a visualization interface is measured by its ability to present information that is consistent with the user's perceptual, cognitive and response-based mental representations [8]. When the information presented is consistent with cognitive representation, performance is more rapid, accurate and consistent [8]. For visualization interfaces to be effective they must resemble the cognitive model the network administrator has in their mind. Failing to address the administrator's perception will lead to erroneous information analysis and detract from the current process.

2.1.1 Visualization Interfaces.

Shiravi's [29] survey of security visualization created a taxonomy based upon the functionality incorporated in the interfaces. The use-case classification process is a departure from the traditional data driven approach. For example rather than

seeking to visualize an IDS log, a system designer should seek to visualize a denial of service attack.

Focusing on a specific task gives the visualization interface the freedom to incorporate multiple data sets to maximize effectiveness [29]. The survey used the following criteria to select visualization interfaces: relevance to network security, contribution of the system, visual techniques and satisfactoriness of evaluation [29]. The visualization systems were broken into five use-case categories: host/server monitoring, internal/external monitoring, port activity, attack pattern and routing behavior [29]. The Mobile Network Defense Interface (MNDI) developed for this research seeks to visualize attack patterns, therefore only the analysis of attack pattern visualization interfaces is relevant to this research. Common data sources for attack pattern visualization interfaces are packet traces, intrusion alerts, Domain Name Server traces and bandwidth monitoring.

Three attack pattern visualization IDS Rainstorm, VizSec and NIMBLE form the framework around which the MNDI was built. The following visualization techniques provide the lesson's learned upon which the basic implementation of the this research's MNDI is based. The three visualization are described in the following sections.

2.1.2   IDS Rainstorm.

IDS Rainstorm was developed at Georgia Tech, from school system administrator requirements and tested on the campus network alert logs. The system administrator's requirements were: identifying high-priority alarms, locations, and appropriately allocating human resources to resolve it [1]. The administrator's method for determining the significance of an alarm was based on the alarm count, severity and time of day of the alert [1]. Historically, the network administrators had chosen to rely on the parsing of textual alert logs to monitor network status [1].

IDS Rainstorm visualizes alarm data from an IDS and presents it in an overview screen allowing administrators to easily detect anomalies, Figure 2.1 [1]. Internet protocol (IP) addresses are plotted along the y-axis of the tool's interface, the x-axis presents the analyst with a measure of time allowing them to observe variation in alarm numbers and severity. IDS Rainstorm plots an alarm after receiving 20 alerts and selects the color based upon criticality to the network state. Red is critical, yellow is average concern and green is low concern. IDS Rainstorm provides a high level overview of all alarms on the network and allows the administrator to zoom in on a select IP range. Performing a mouse-over presents the administrator with the type, time, source and destination IP of the alert. IDS Rainstorm allows administrators to apply filters, such as only display critical alarms in order to focus their defense vector.
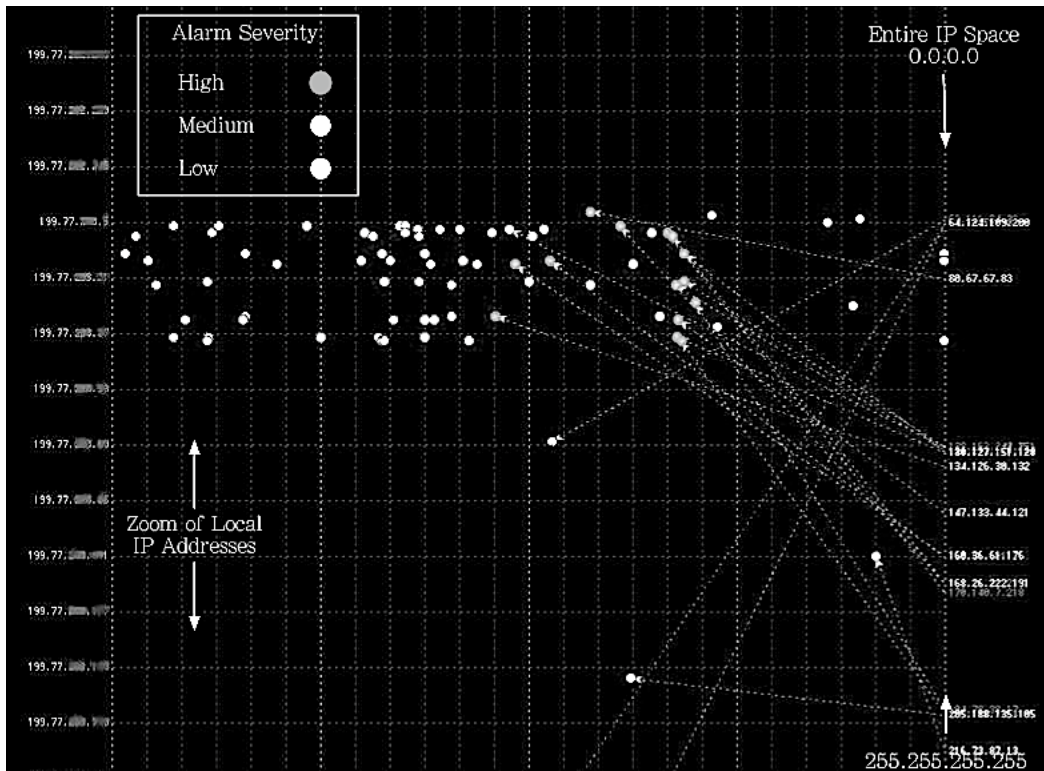

Figure 2.1: IDS Rainstorm

IDS Rainstorm provides an fairly intuitive interface that allows the user to drill down to a finer grained level of detail. A limitation of the tool is that it does not draw the user to a particular part of the visualization automatically for example by highlighting large clusters of critical alerts. IDS Rainstorm probably would be difficult for a layman to utilize. This tool relies on the network administrators experience and understanding of the network in order to be effective. The authors themselves note that the tool in its current form is useful as a forensics tool but that there is a learning curve attached to implementing it as a live network monitoring tool [1]. The use of color to denote the severity of the alarms is a useful feature and one that was incorporated into the design of the MNDI designed for this research.

2.1.3 VizAlert.

VizAlert was designed to visualizes disparate network logs. The goal is to provide a holistic view of the network to improve situational awareness, see Figure 2.2.

The visualization maps network nodes alerts from routers, switches, host etc. based upon the type of alert, when the alert happened and where in the network the alert occurred [16]. Where the alert occurs is an important attribute because it allows the visualization system to correlate disparate events to a specific node [16]. The visualization interface shows the local network topology in a center ring that is encircled with various alert types, that are mapped to specific IPs in the center ring. The alert ring stacks from the inside and builds out based upon the time of the event. VizSec allows the administrator zoom into a single node or zoom out until the entire network is visible. As the administrator zooms out the network is abstracted into clusters of subnets.

VizAlerts ability to visualize multiple network logs is a benefit to administrators attempting to correlate alerts in a large network. Unfortunately the tool was only tested using a single IDS alert so it is unclear how much benefit is gained from

13

Figure 2.2: vizAlert

visualizing multiple logs for quick correlation. VizAlert has intuitive interface making it possible for a novice administrator user to understand the information presented. However, since it does not focus the analyst attention to critical areas automatically it requires the experience of a seasoned administrator in order to be of value to an organization. Additionally, the tool does not provide any actionable options so it is better suited as a forensics tool.

2.1.4   Network Intrusion Management Benefiting from Learned Expertise (NIMBLE).

NIMBLE is a visualization and recommendation tool. NIMBLE correlates alert logs from an input file, creates a model for each alert, matches each alert model against historical models and offers a recommend course of action [24]. The administrator may choose between viewing the data visually or in tabular form[24]. NIMBLE visual display is called the interactive incident diagram (IID). The IID visualization

14

is a graph that represents each node on the network as a card, a line connecting nodes represents IDS alerts between the two nodes [24] see Figure 2.3. The card is displayed in a simple table format that contains relevant information such as IP address, importance of the asset, operating system (OS), and administrative data.

By default NIMBLE displays the source cards of the alert on the left side of the display and the destination cards on the right. If the card is both the recipient and originator of an alert it is placed in the middle of the view. The administrator is able to manipulate the card view manually by clicking and dragging the cards. NIMBLE also allows the user to zoom in and out viewing the cards as either a group of nodes or a single node depending on the zoom level.

A justification panel on right side of the screen contains a list of possible explanations of what triggered the IDS alert and how it was categorized. If the administrator selects the explanation panel the visualization places a orange shading over the effected nodes. The darker the orange shading the higher degree of confidence the learning algorithm has in its explanation of the incident. Users of the visualization returned 31% correct responses without justification, but when justification was provided correct responses increased to 36% [24].

NIMBLE's visualization is crude but effective for demonstrating the desired capabilities. Of the tools reviewed NIMBLE was the only one attempting to predetermine the category of the alert. The justification of the alert classification has the potential to introduce automation bias but in its current form it show some benefit. Automation bias for this research is defined as errors of omission and commission [31]. Omission errors are made when the administrator is specifically prompted to address an event. Commission errors occur when the automated system recommends an action that contradicts the administrator's training but they still implement the action.
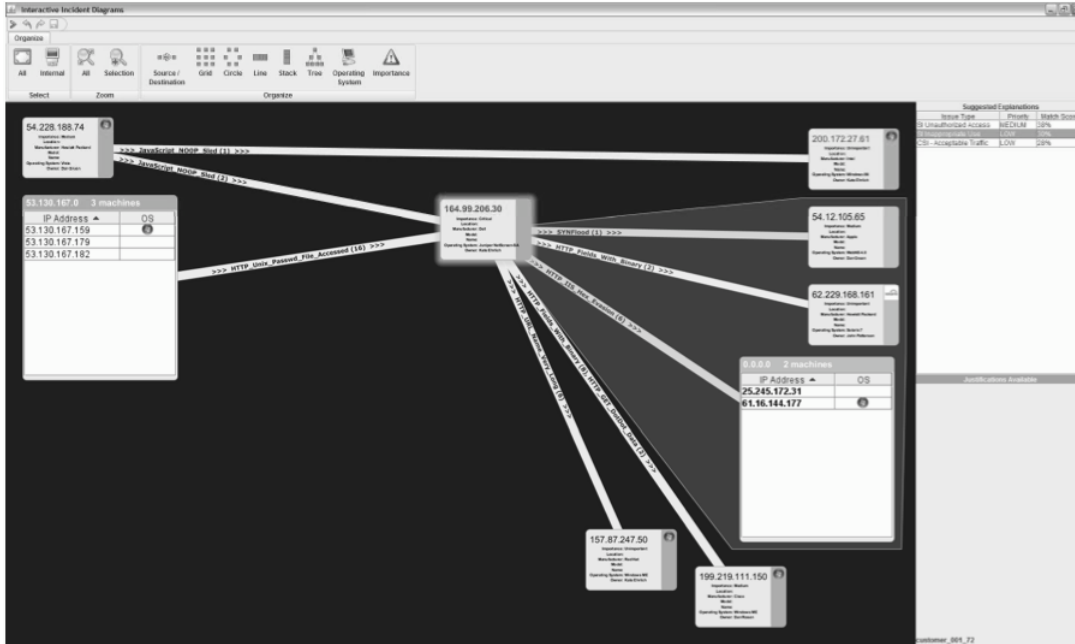
Figure 2.3: NIMBLE

2.2   Visualization System Design

Implementing a visualization that takes advantage of the interfaces ability to present data in an intuitive manner has many benefits to cyber defense. For example maintaining situational awareness and making time critical decision from anywhere network connections are available is beneficial to network administrators.

Implementing a mobile visualization takes advantage of cutting edge tablet technology to present network administrators with a cognitively relevant touch-based interface. The mobile environment comes with its own set of challenges for example the initial display's screen real-estate is limited. Therefore, it is imperative that the interface directs the administrator's attention to the most critical task and presents them in a clear, concise manner. Data presentation is vital to the utility of a mobile interface.

2.2.1 Mobile Interface Design.

Mobile interface design comes with a unique set of challenges. Mobile devices are often used in public areas so the user may have less cognitive ability to apply to understanding the interface [5]. Minimizing the amount of time needed for a user to reorient themselves is key to overcoming this challenge. Mobile interfaces must also address limitations in hardware and screen real-estate [5, 12]. Several methods exist for dealing with limited screen space such as 3D visualization and scrolling mechanisms. However, this gives rise to another challenge, preventing users from getting lost while navigating a small screen's graphical space [5]. Environmental challenges such as the amount of lighting can make deciphering complicated graphics difficult [5]. Mobile interface designer must be mindful of the complexity of graphics and reliance on sound effect in their interface designs to ensure understandability is not lost in less than ideal environmental conditions.

Chittaro [5] provides a checklist of items to consider when designing a mobile interface.

Mapping: how the data will be mapped to the interface

Selection: presenting the most relevant data to the user

Presentation: the effectiveness of the visualization to utilize the screen space

Human factors: considering the user has potentially limited cognitive ability, due to distraction, is the interface easy to interpret

Evaluation: perform rigorous user testing of the interface

These guidelines allow developers of mobile interfaces to keep their end product in perspective, thus increase the potential for delivery of an appealing interface:

2.3 Automated Assistance

Another method to help increase the effectiveness of network operators is automation of alerts. Addressing DoD's challenge specifically Moitra states that in today's cyberwarfare environment we must counter and mitigate as much of the adversary activity within our networks automatically whenever possible. Automation will free up the finite defender forces to focus on the critical threats to our key cyber terrain. [21].

Automating tasks to resolve time critical tasks to avert disaster is not a new concept and has been implemented with some measure of success in the aviation industry and power industry among others. Jones states that fully automatic and fully manual systems are both extremes and rarely found in real-world implementation. Instead what is found is a combination of human and technology or semi-automatic systems [13]. To achieve an effective semi-automatic system Endsley states the following must be analyzed, to realize the benefits of automating cognitive tasks the potential negative effects must be avoided by discovering a method of keeping the operator actively involved in the decision-making loop while simultaneously reducing the load associated with doing everything manually [6].

III.   Methodology

3.1   Introduction

This chapter discusses the methodology used to measure performance of test subjects during cyber attacks on a network using the Mobile Network Defense Interface (MNDI) developed for this research or alternatively a desktop network management configuration. First, the problem is discussed and defined. Second, the goals and objectives of the experiment are presented. Third, the system description, services, workload and metrics for evaluation are presented. Finally, a description of the techniques used to evaluate the data is presented.

3.2   Problem Definition

Effective network administration requires experience and familiarity with the network. The number of available and fully qualified administrators is often in short supply. Additionally, current network monitoring systems alone are insufficient to ensure the protection of computer networks. The combination of monitoring systems and administrator involvement promises the best solution for protecting the network. Administrators need the ability to quickly identify the current attack situation so they can react quickly to alerts and maintain a healthy operational state for their network.

3.3   Goals and Hypothesis

The hypothesis of this research is that by using state-of-the-art mobile technology an administrator can attend to time critical alerts more quickly and accurately when compared to the current norm, that is, fixed terminal network administration interface.

This research aims to significantly increase the performance of veteran network administrators, as well as novice administrators during a cyber attack. The intuitive interface and visual information presentation allows the novice administrator to perform within a reasonable level of experienced network administrators using MNDI. MNDI presents the administrator with the minimum amount of information needed to make quick, accurate decisions. Reducing the information presented prevents the administrator from getting overwhelmed by extraneous information and thus reduces the time to make accurate decisions. This thesis will show performance enhancement occurs despite not having any previous experience with the current network configuration or the visualization interface.

3.4 Approach

The approach used during this experiment was to evaluate 19 volunteer administrator's effectiveness at defending a network during a cyber attack. Test subjects consisted of volunteers that have a minimum of a novice level of network administration knowledge. The test subject's network administration skill and experience is determined by a standardized self-assessment and quiz, see Appendix A. No knowledge of the current network configuration was required prior to testing.

Each test subject was randomly assigned a network interface device and attempted to protect the network during a series of random cyber attacks, occurring at random time intervals. Test subjects received either MNDI or the desktop configuration as their network interface during the test.

MNDI was developed for this research. The desktop interface is a set of open source software that provides an similar set of capabilities. Both systems were set up with the same level of expertise in network administration and cyberspace defense tactics. For automation purposes both systems use the same expert system (ES)

to resolve alerts with impact levels that are below the level of automation (LOA) threshold.

3.4.1 Network Interfaces.

MNDI is an iPhone Operating System (iOS) application that provides the user with a visual representation of the network state and visual scripting objects. MNDI was written in Objective C using Apple's X-Code software development kit environment 4.2 and is installed an iPad 2 with 32 GB of storage. The graphical user interface includes the networks node state, physical links and link saturation. Table 3.1.

The desktop network interface is comprised of freely available network administration software and custom software need for this research, that provide a comparable capability to MNDI. Snort is used as the intrusion detection system, Spiceworks is used for the network health information and Basic Analysis and Security Engine (BASE) is used to visualize the alerts. The desktop system is running Spiceworks 6.1.01074 and BASE 1.4.5 operating on a Dell Precision with a 2.4 gigahertz (GHz) Intel i7 eight core processor with eight gigabytes (GB) of memory and 700 GB of storage.

The virtual network was created in VWare's ESXi 5.0 and is operating on a Dell Power Edge T710 with two Intel Xeon X5690 3.5 GHz, 196 GB of memory and two terabytes of storage.

3.4.2 Spiceworks.

Spiceworks is free network management software designed for networks with up to 1,000 devices. It contains network inventory, network monitoring, network configuration management, active directory management, bandwidth monitoring and many other features [33]. Spiceworks is the provided to the test subjects primarily for

Table 3.1: Nodal Health Information

IP Address	the IP address assigned to the node on the network
Operating System	OS with version running on the node
CPU utilization	percentage of CPU cycles in use
RAM	amount of available RAM in megabytes on the node
RAM utilized	amount of RAM in megabytes in use
Saturation	percentage of the network pipe in use
Packet loss	the number of packets lost at the destination

network status and health information even though it is capable of other functionality

Figure 3.1.
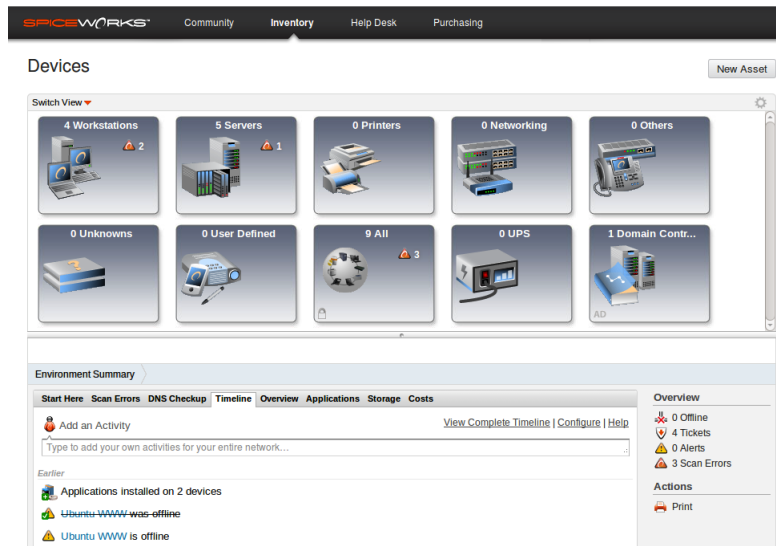
Figure 3.1: Spiceworks

3.4.3   BASE.

BASE provides a web front-end to query and analyze alerts from Snort [2]. This

capability is provided to give desktop users an overview of all alerts on the network.

BASE is comparable to MNDI's alert information which appears in both the action view and the history log. Figure 3.2.

Figure 3.2: Basic Analysis and Security Engine

Figure 3.2.

3.4.4 Scripting Interface.

A custom-scripting interface was developed for this research to allow the user to execute a set of scripts provided for the experiment. To resolve the alert the test subject had to enter the proper Snort identification number from BASE and the correct script to resolve the attack Figure 3.3. The scripting interface and provided script are comparable to the capability of MNDI's action view, minus the visualization of multiple scripting objects and the ES's recommendation.

Figure 3.3: Scripting Window

3.4.5 Alert Resolution Window.

A custom-built alert resolution webpage was built for this research to provide the test subject with a visualization of resolved alerts. The webpage displays all alerts addressed by the user or the ES during the course of the experiment Figure 3.4. This capability is comparable to the history log view provided by MNDI.

Event ID	Signature Name	Source IP	Dest IP	Impact	Start Time	End Time
45689	Possible TCP SYN Flood DoS	192.168.251.150	192.168.1.37	5	Tue Jan 29 13:20:12 EST 2013	Tue Jan 29 13:20:12 EST 2013
45693	Possible TCP SYN Flood DoS	192.168.1.180	192.168.1.20	4	Tue Jan 29 16:19:34 EST 2013	Tue Jan 29 16:19:34 EST 2013
45697	SQL injection attempt	192.168.1.133	192.168.1.37	5	Tue Jan 29 16:19:35 EST 2013	Tue Jan 29 16:19:35 EST 2013
45701	Possible TCP SYN Flood DoS	192.168.1.157	192.168.1.10	4	Tue Jan 29 16:19:40 EST 2013	Tue Jan 29 16:19:40 EST 2013
45705	Possible TCP SYN Flood DoS	192.168.1.159	192.168.1.10	4	Tue Jan 29 16:19:52 EST 2013	Tue Jan 29 16:19:52 EST 2013
45709	ET SCAN Potential FTP Brute-Force attempt	192.168.1.37	192.168.251.174	5	Tue Jan 29 16:20:13 EST 2013	Tue Jan 29 16:20:45 EST 2013
45717	POP3 Mailbox overflow attempt	192.168.251.192	192.168.1.5	3	Tue Jan 29 16:20:25 EST 2013	Tue Jan 29 16:20:25 EST 2013
45721	Possible TCP SYN Flood DoS	204.109.190.191	192.168.1.37	5	Tue Jan 29 16:20:32 EST 2013	Tue Jan 29 16:20:32 EST 2013

Figure 3.4: Alert Resolution Window

3.4.6 Expert System (ES).

The alert generated during this experiment are from known attacks, with known resolutions, an ES was created to assist MNDI users with these alerts. The ES augments MNDI's action view capability and enhance MNDI user's experience by offering multiple actions that will resolve the alert.

One of the actions presented by the ES is ignore. The test subject is not obligated to execute any of the actions provided by the ES and is always given the ignore action. This allows user-initiated action on the affected node giving them the ability to take actions they feel are appropriate to address the alert. The ignore action also allows the test subject to quickly disregard false positives.

The ES is a truth table that takes a known alert and offers actions that resolve the alert. All the action presented will resolve the alert. However, the action is not always proportionate to the effects of the attack. Some of the actions that resolve the attack could have catastrophic effects to the usability of the network.

24

The ES was used by both network interfaces during the experiment to automatically resolve alerts with impacts above the threshold of the current LOA. The performance of the ES during automation was not under test during this experiment and that performance is not evaluated.

3.4.7   Network Modeler.

The network modeler gathers network state information from network mapper (Nmap), Snort and a custom Java program that reports node states. MNDI relies on this information to visualize the network in a human understandable manner. The desktop only visualizes the alert information from the network modeler.

3.5   System Boundaries

The system under test (SUT) consists of an iOS application designed to allow administration of a network under cyber attack. The SUT is installed on an Apple iPad 2 with 32 GB of storage. The performance of the SUT is compared to a suite of freeware network administration tools combined with custom tools that provide a similar capability to the SUT; the suite of tools are installed on a desktop PC.

The SUT polls a web server on the virtualized network for health status, potential nodal actions, alerts and potential user actions. The desktop polls the web server for alert information but uses the capabilities of Spiceworks to gather network health information. Both systems use the same automated system. The automated system uses an ES to automatically resolve alerts that have an impact above the current LOA threshold. Any attacks that are resolved by the automated system are posted in a resolved event log. For MNDI the automatically resolved events appear in the history log. For the desktop users the automatically resolved events appear on the resolved events webpage. The LOA is randomly generated and manually set prior to each test case for both network interfaces.

Each test case is comprised of a series of randomly generated cyber attacks of varying complexity and duration. Alerts not automatically resolved are presented to the user for their resolution.

The test network consists of four clients, three servers, switch, firewall and a blackhat client. The three servers on the test network are the Ubuntu server 12 (web server), Ubuntu server 12 file transfer protocol (FTP) and Windows Server 2008 (Mail and Domain Server). The firewall is running PfSense. The blackhat machine is running backtrack 5 r2. See Figure 3.5 for the configuration of the test network.

The test network contains a modeler that is primarily responsible for providing network health information, alerts and actions to the SUT. While availability of the modeler is important to the functionality of the SUT, evaluation of the performance of the network modeler is out of the scope of this test.
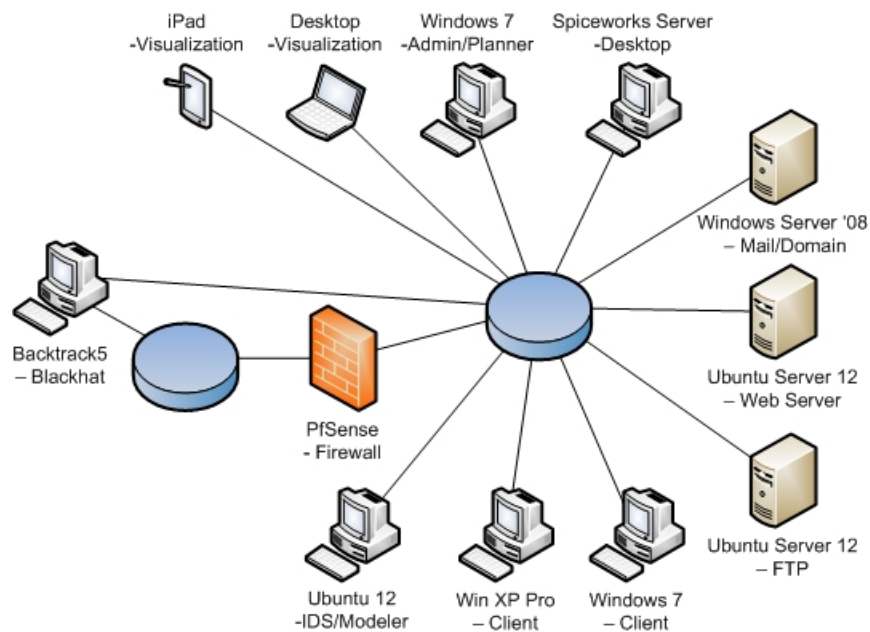
Figure 3.5: Test Environment

Table 3.2: Alert, Expert System Response and Augmenting Actions

Snort Alert	Possible Actions	ES	Augment
ET SCAN Potential FTP Brute-Force attempt	Block IP, Shutdown host, Disable Service, Reset password, Ignore	Block IP	Reset Password
Possible TCP SYN Flood DoS (IP within the subnet)	Ignore and shutdown host	Ignore	N/A
TCP SYN Flood DoS (external IP)	Block IP, Shutdown host and Ignore	Block IP	N/A
POP3 Mailbox overflow attempt (external IP)	Block IP, Disable Mailbox, Clear Mailbox, Ignore	Block IP	Clear Mailbox
POP3 Mailbox overflow attempt (IP within the subnet)	Disable Mailbox, Clear Mailbox and Ignore	Disable Mailbox	Clear Mailbox
SQL injection attempt	Patch, Disable Service, Shutdown host and Ignore	Patch	N/A
ET NETBIOS Microsoft Windows NETAPI Stack Overflow Inbound MS08-067	Reboot host, Disable Service, Reset password, Shutdown host, Patch and Ignore	Reboot host	Patch

3.6   System Desgin

3.6.1   Main view.

The main view of MNDI provides a visual overview of the network topology. The data for the network is contained in an extensible markup language (XML) file created by the network modeler. The SUT reads the XML file from a web server every five to thirty seconds depending on the operation being performed. All network nodes and the links are visualized on the main view. A simple spacing function is used to visualize the nodes around their primary switch. Each node is drawn to a grid point on interface to maintain an organized look to the network. The grid was created by dividing the entire pixel space into grid points, when a node is dragged by the user a calculation is performed to move it to the nearest grid point, see Figure 3.6.
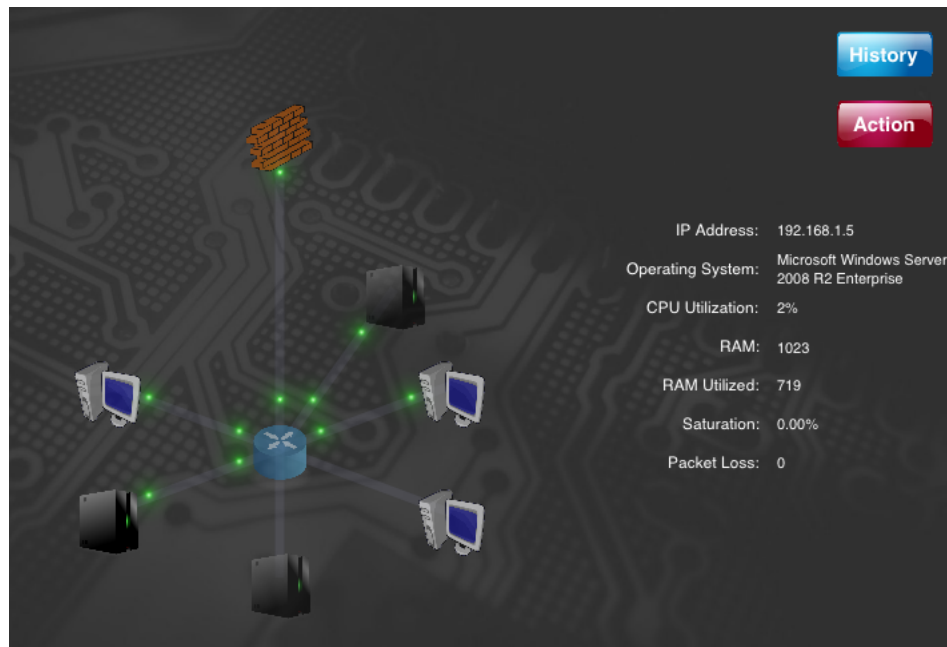

Figure 3.6: Main View

3.6.2   Nodes.

Each node on the network is represented by a common graphical representation for its type. The types of nodes on the network are clients, servers, routers/switch and firewalls. The switch that is visualized is presented to give the test subject a visual understanding of the logical network configuration. During the experiment the switch does not come under attack and the user may not perform any actions on it.

3.6.3   Node Information.

A single tap on a node will highlight the selected node and present health information about the node. The information presented is Internet protocol (IP) address, operating system (OS), centrel processing unit (CPU) utilization, random access memory (RAM), RAM used, saturation and packet loss. See table Table 3.1 for a description of the node information categories. This information is updated approximately every 30 seconds by downloading the XML file published by the network modeler and evaluating whether there is a change in network health information. If a change is detected the status information is updated in the network health view. Changes in network status may result in a change in the visualization. For example, a network link that was previously experiencing very little network pipe saturation is visualized with fast moving green particles on the link. However, if that link were to become suddenly saturated because of significant increase in traffic the particle speed would slow down and the color transition from green to yellow to red.

3.6.4   Node Actions.

Tapping the action button displays all available actions on the selected node. Available actions are displayed in a single column. The actions available depend on the node select. The firewall's single action is block IP address. For the client and server nodes; disable user, enable user, shutdown host, reset host, reset password, disable service and enable service are available. Tapping on an action button, such

as disable user, displays the available options for the action or N/A if there are no additional options available. Tapping an option and then tapping on the select option button prepares the script for submission. At this point the test subject may either select submit and execute the selected actions or press reset to begin over. Each column on the event view may contain an action to be taken with a maximum of three actions being executed per submission, see Figure 3.7. Any actions that are submitted are transformed into actionable scripts and sent to the network for execution.
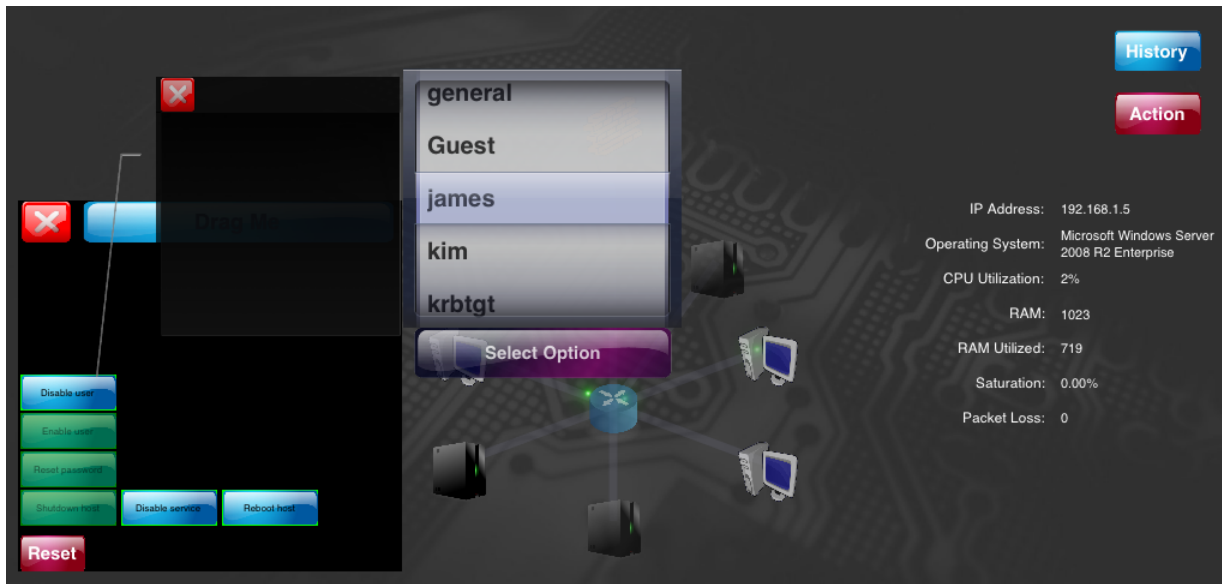

Figure 3.7: User Initiated Action

3.6.5 Links.

The links between the nodes represents the network traffic with colored particles of varying speed. Green indicates low network traffic, yellow indicates moderate network traffic and red represents heavy network traffic. Additionally, the particles become slower as the network saturation increases (red particles move very slow, green

particles move quickly). The equation used to calculate the percentage of bandwidth free is:

$$B_a = 100 - ((t/B_t) * 100) \tag{3.1}$$

$$t = \text{throughput (mb)}$$

$$B_t = \text{Total bandwidth (mb)}$$

The result of the above equation are used to set the color and the speed of the particles that traverse the links shown.

Table 3.3: Link Saturation

Percent Available	Speed	Color
75-100	4-5	green
25-75	2-3	yellow
less than 25	1	red

Saturation PercentBandwidthAvailable = 75-100% free equates to a speed of 4-5 (green) PercentBandwidthAvailable = 25-50% free equates to a speed of 2-3 (yellow) PercentBandwidthAvailable = less than 25% free equates to a speed of 1 (red)

3.6.6 Event View.

When an alert is received by the network controller an event view is presented to the test subject. The event view shows the name of the alert that was received as well as the target node, see Figure 3.8. Single tapping on the action button opens a table with all the received alert information, an ID number, name of the alert, source IP, destination IP, impact and a timestamp. The user is presented with a list of actions,

Figure 3.8: Alert Received

all of which will reasonably resolve the event. An ES then determines the single best action to resolve the alert. The button in the column that is highlighted is the ESs recommend course of action.

The ES was designed to attribute a single action to a known alert. In the case of the SUT the known alert is from the Snort Intrusion Detection Service. The ES's recommendation for each alert is an accepted and proportionate response given the network and the impact the attack may have on network operation and health.

For the test subject to maximize their score an additional action or actions may be required. Each column on the event view may contain an action to be taken with a maximum of three actions being executed per submission.
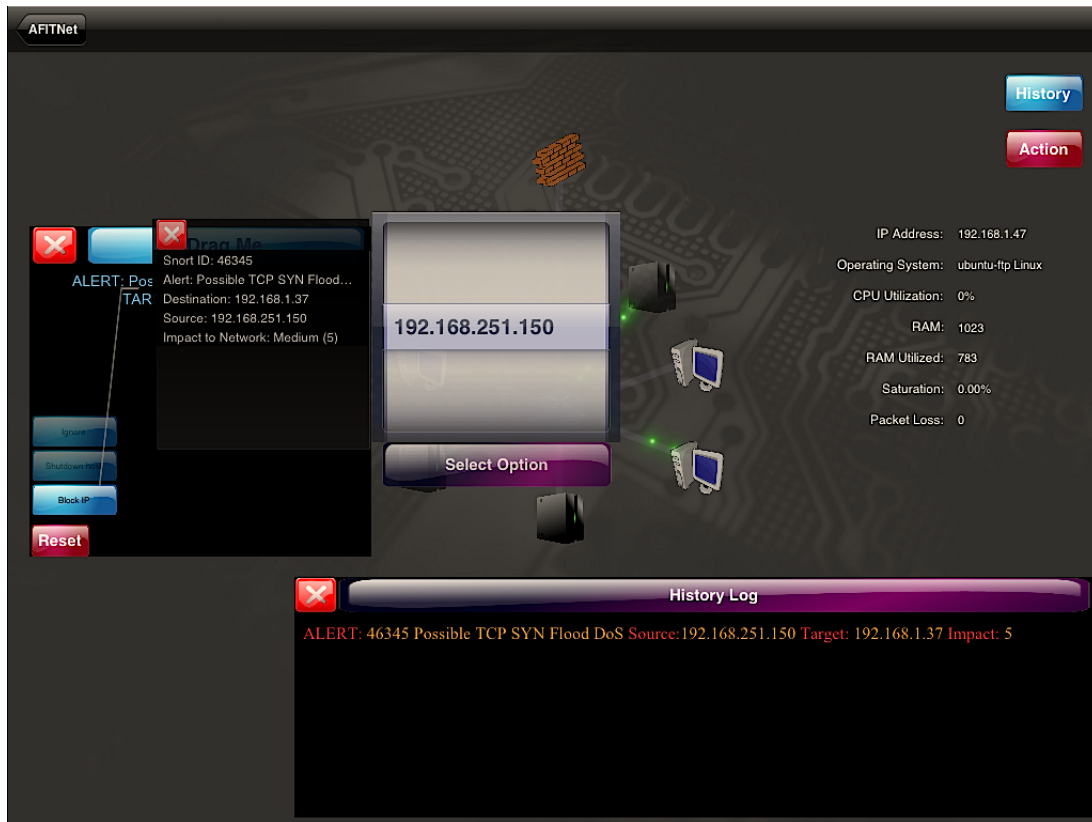
Figure 3.9: Action View

3.6.7 Action View.

When a node is selected, the test subject may bring up all available actions on the node by tapping the Action button. A stack of buttons will appear in a separate view with actions such as Disable user, Shutdown host, Block IP, etc. Depending on the action there may be associated options such as various user names or IP address.

3.6.8 Option View.

By tapping on the action a list of options are presented. The options for each action are presented in a picker view and the test subject must select one option and tap the Select Option button. Once an option is selected and the submit button is

tapped the script is executed. Each column on the event view may contain an action to be taken with a maximum of three actions being executed per submission.

Figure 3.10: History Viewt

3.6.9 History View.

History View shows the test subjects the last action(s) taken, resolved events and alerts received on the network. The incoming alert contains the unique Snort identification number, the common name for the alert, the impact to the network and the time it was received. If the impact of the attack is greater or equal to the level of automation then the alert is resolved by the ES. In the history view any action that is taken by the automated system reports the Snort ID the action was taken against, the action taken and the source of the attack. If the impact level versus level of automation warrants a user event the event view becomes immediately visible. Once the user selects and submits the action or actions that they wish to execute the history log will reflect which action was taken, which node was affected, time submitted and the associated Snort ID.

The history view is color coded for quick visual identification of events that have occurred. Alerts are posted in red text with orange variable values. Automated responses are posted in white with grey variable values. User actions are presented in

34

dark green with light green variable values. Resolved events are posted in blue with grey variable values.

3.7 Workload

The workloads for the system are normal network traffic between nodes and cyber attacks. Normal network traffic is defined as routine traffic between nodes on the network. The system receives a random number of cyber attacks between 7-25 occurring at random offsets, over a single scenario, each test case consists of four scenarios. Each attack is classified by the modeler based upon the perceived impact to the current network configuration. If the impact level is higher than the current automation level the alert is automatically resolved, otherwise it is presented to the user for action.

When an alert is received for the users actions they are presented with scripted actions to resolve the alert. Prior to the user taking action, the ES determines which actions would reasonably resolve the alert. The user is presented with a list of viable actions to resolve the alert, the single best action according to the ES is highlighted. The other actions presented to the user would reasonably resolve the event but may not be a proportionate response or simply augmenting actions to the best course of action. An example of an non-proportionate response would be shutting down the mail server because of a mail blitz attack against a single user's mail box. An augmenting action is one that adds value to the single best action. For example, during an FTP brute force password attempt, the single best action is to identify and block the offending IP address, a good augmenting action would be to also reset the effected user's password.

Finally, alert resolution is achieved by the submission of the network configuration changes for implementation.

3.8   Performance Metric

The following parameters may affect the performance of the SUT:

Time to resolve attacks, elapsed time from when the action plan is presented to the user and when they press the submit plan button.

Percentage of alert resolved, as alerts continue to be in an unresolved state, they will continue to queue for the user's action.

Additional user actions, the amount of actions the user is taking outside of the actions required to resolve the incoming alerts.

The response variable is the performance of the user on the platform they were assigned.

3.9   Parameters

The following parameters exist in the SUT:

- Cyber attack scenarios and test cases: each test subject will complete four scenarios with four test cases at varying levels of automation with varying number of cyber attacks.

- Types of attacks each test subject will be required to resolve a random set of eight different attacks Table 3.5.

- Network modeler: MNDI extracts its network health data and alert data from a network modeler. This data is then transformed and visualized on MNDI.

- Assigned platform: each test subject is randomly assigned a network management platform.

- Network management experience: the ability of test subjects to administer a network based upon a self assessment.

- LOA: effects which events the user is required to resolve versus which events the automated system will address with the single best action.

3.10 Factors

- Assigned platform: each test subject is randomly assigned a network management platform. Level one is the iPad 2 and level two is the desktop. The test subject were randomly assigned a test platform.

- Network management experience: the skill level of the test subjects based upon a self assessment. From the mean test scores, subject were split as closely into thirds as possible to maximize the sample space, Table 3.4.

Table 3.4: Network Management Skill

Level	Designation	Score Range
1	Novice	$score =< 50$
2	Intermediate	$65 > score > 50$
3	Expert	$score >= 65$

- LOA effects which events the user is required to resolve versus which events the automated system will address with the single best action. The LOA is randomly set at the beginning of each test case until all four (4) desired automation level are tested. The LOA is set on a scale of one (1) to eight (8), where one (1) is full automation and eight (8) is no automation (factor levels 1, 5, 6,8).

- Cyber attack scenarios and test cases: Each Test Case consists of four scenarios with 7-25 attacks which initiate from 0-10 seconds apart. Depending on the

type of attack this may generate one to many alerts. Between each scenario is a two to five minutes pause in attacks. The LOA may be set between one and eight corresponding to the impact levels of the attack. An automation level of one represents full automation and an automation level of eight represents no automation. For the purposes of the experiment four levels of automation where selected and each scenario is performed at a randomly selected level of automation. The levels of automation the test subject experienced are; full automation of alerts, 75% automation of alerts, less than 1% automation of alerts, and no automation of alerts. For the attacks against the SUT these four levels of automation offered the best sampling of automated assistance. The test subject was informed of the automation level at the beginning of each test case.

3.11 Evaluation Technique

Test subjects are evaluated on their ability to properly recognize and respond to a cyber attack visualized on a MNDI or desktop configuration that is providing the state of the network while under a cyber attack.

To analyze MNDI's effectiveness, the test subjects selected must have a moderate level of network experience. Moderate network experience in this instance is defined as the ability to distinguish common network components and software such as routers, servers, firewalls, intrusion detection systems, ports, etc, and a basic understanding of the role of each component. It is important that the user have some networking experience to mitigate the amount of bias introduced by the ES that is recommending a course of action. The moderate amount of networking experience allows the user to conduct a basic research to ensure that both the identified issue and the recommended course of action appear valid. The test subjects level of network

38

administration expertise was determined by a self assessment and quiz, see Appendix A. The test subjects were asked to assess themselves on several question in the following categories: operating systems, software, programming languages, protocols, networking concepts and computer security. They were asked to rate themselves as either no experience, beginner, intermediate, advanced or expert. A numerical value of zero for no experience, to four for expert, was assigned to each response. The self assessment coupled with their score on a four question quiz determined their overall score. The overall score was translated into a skill level based up the mean value of the self assessment scores.

3.12 Experimental Design

The experiment each lasted approximately 115 minutes and was structured as displayed in Table 3.6 First, the test subject receives a briefing describing the theses being evaluated and an overview of the system they would be using. Next, the user was given hands on training, including a demonstration of what an attack would look like on their system. The test subjects were informed that during automated actions the ES takes the single best action to resolve the event but that augmenting actions may be required to achieve the best solution to the attack. Finally, the user begins the scenario.

Table 3.5: Attack Description

Attack Name	Impact	Description
SMB exploit	7	Relays an SMB authentication request to another host, gains access to an authenticated SMB session if successful. [23]
FTP Password Attempt	5	Systematic brute force password attempt against user's FTP accounts
External Denial of Service	5	Packet flood from an external internet protocol address. Intended to deny the user access to the resource.
SQL injection	5	Database is provided with malformed data and the victim machine builds a SQL statement using string concatenation [11]
Internal Denial of Service	4	Packet flood from an internal internet protocol address within the subnet. Intended to deny the user access to the resource.
Internal Mail Blitz	3	Repeatedly sends an email to a particular address. In many instances, the messages will be large and constructed from meaningless data in an effort to consume additional system and network resources.[34]
External Mail Blitz	3	Repeatedly sends an email messages to a particular address. In many instances, the messages will be large and constructed from meaningless data in an effort to consume additional system and network resources.[34]

Table 3.6: Experiment Schedule

Event	Description	Time (minutes)
Training and self assessment	Overview of system capability and self assessment of network management experience Appendix A	15
Alert Demonstration	Single attack sent to the network demonstrating how the test subject's device will alert them and how to resolve the alert	5
Test Case 1	Random level of automation, attack types and attack duration	20
Break		5
Test Case 2	Random level of automation, attack types and attack duration	20
Break		5
Test Case 3	Random level of automation, attack types and attack duration	20
Break		5
Test Case 4	Random level of automation, attack types and attack duration	20
Test Subject Assessment of Interface	Test subjects perception of the usefulness of the interface, Appendix B	5
Total Time		120

IV.   Results and Analysis

4.1   Introduction

This chapter presents the results and analysis of the experiments. The workload that the test subjects experienced during the experiments is presented in a stacked bar graph. Alerts correctly resolved and average times between actions were used to evaluate test subject's performance. The two metrics are used to compare the performance of Mobile Network Defense Interface (MNDI) users versus desktop users. Each metric is presented in a boxplot followed by analysis and the calculation of the confidence interval.

4.2   Workload

Figures Figure 4.1, Figure 4.2 and Figure 4.3 show the workload that a user experienced during a test scenario. Workload is divided into three categories low, medium and high. A low workload is less than 50 alerts, a medium workload is greater than 50 and less than 100 alerts and a high workload is greater than 100 alerts. Each stacked bar graph compares the MNDI to the desktop configuration. The black bar represents the number of unresolved alerts and the white bar represents the number of correctly resolved alerts. The absence of a black bar indicates that the user correctly resolved 100% of the alerts.

The low workload figure is comprised almost entirely of level of automation (LOA) five scenarios, where the medium and high workloads are a mixture of LOA's six and eight. At low workloads the advantages of the expert system (ES) and actions provided by the MNDI clearly effects the performance of the test subject. As the test subjects becomes increasingly overwhelmed by the workload the benefit of MNDI's features are still apparent but less drastic.
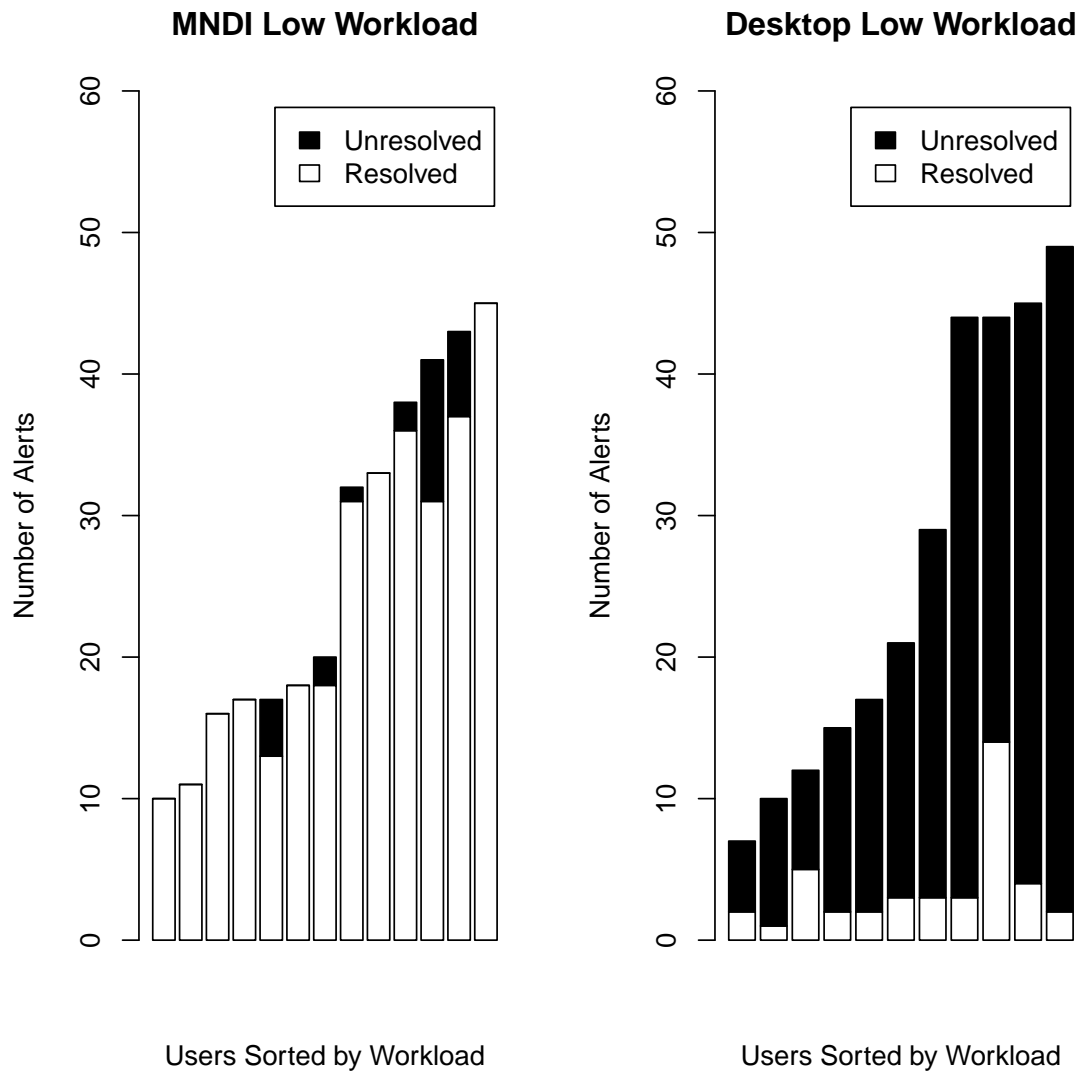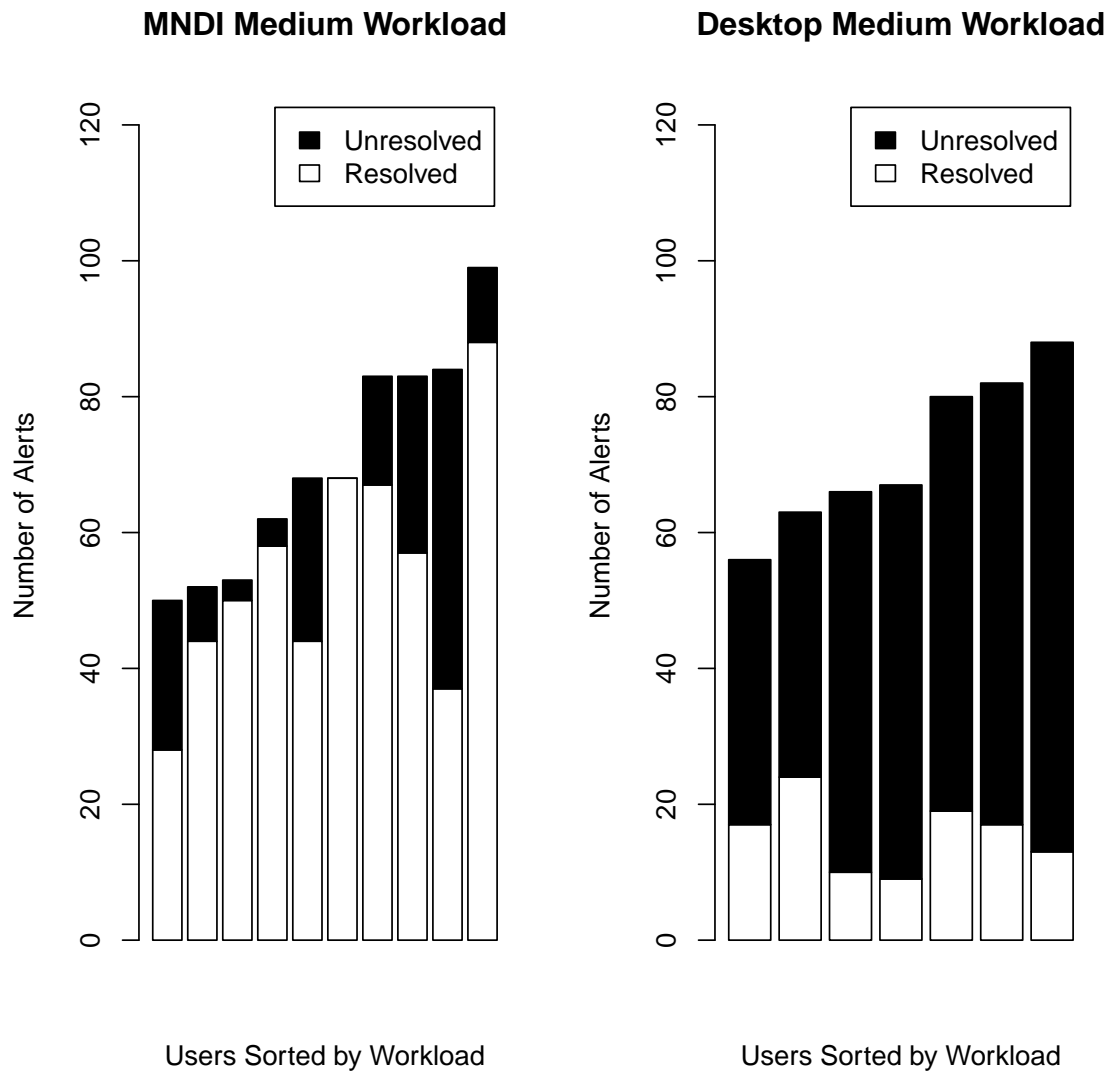
Figure 4.1: Low Workload MNDI vs Desktop

MNDI Medium Workload

Desktop Medium Workload

Figure 4.2: Medium Workload MNDI vs Desktop

MNDI High Workload　　　　　　**Desktop High Workload**
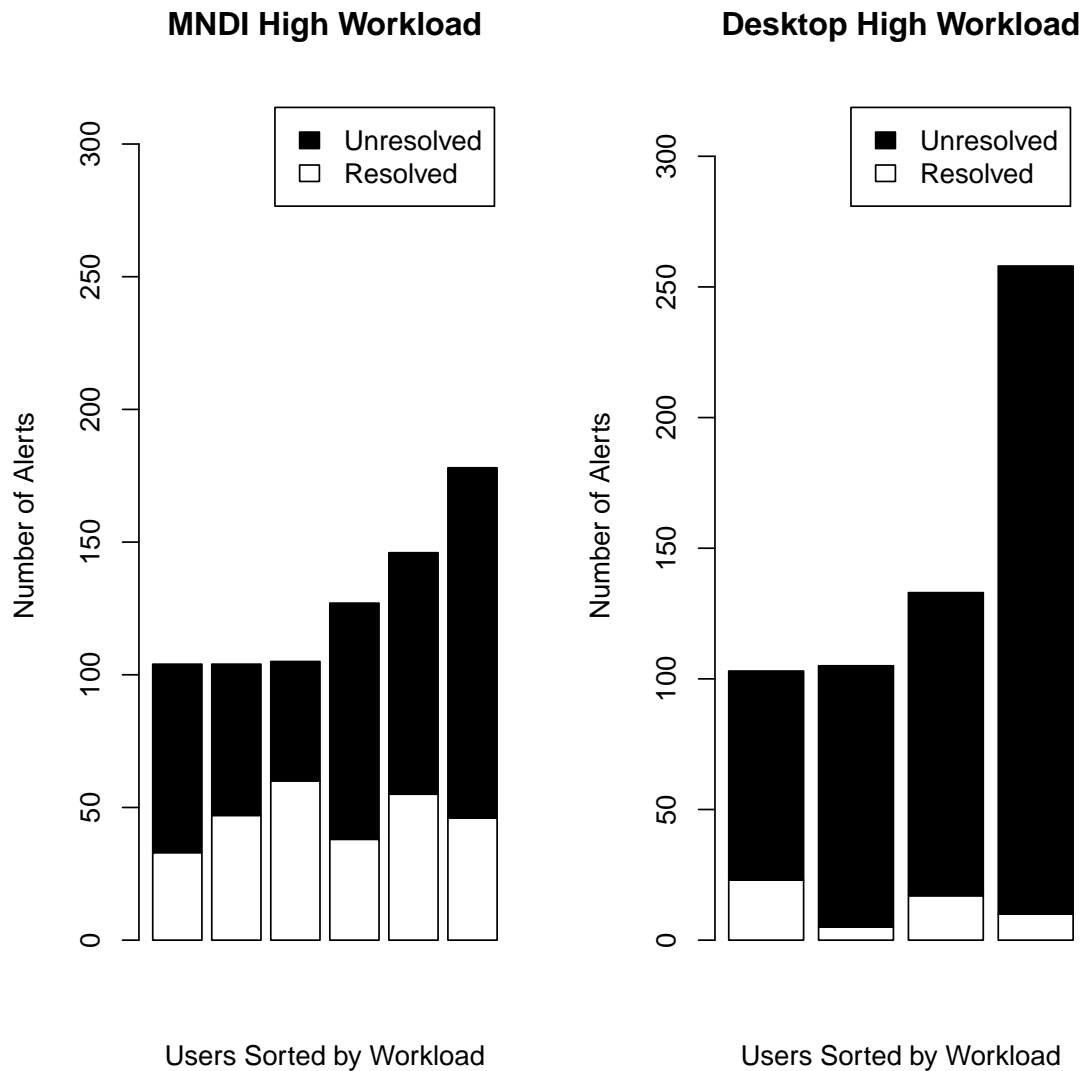

Figure 4.3: High Workload MNDI vs Desktop

4.3 Overview of statistical methods

The following is a description of the two statistical methods used to analyze data.

The boxplot visualizes the distribution of the data set without any assumptions
of statistical distributions. The boxplot is therefore a non-parametric method of

statistical analysis. The dark middle line indicates the median of the data set, the upper half of the box is the 75th percentile, the lower half of the box is the 25th percentile, the bar attached to the dotted line represents the maximum value on the top and the minimum value on the bottom. Any outliers that exist in the data set are show as a dot [20].

The Wilcoxon rank sum test is a non-parametric statistical test that compares two related samples on a single sample to assess whether their populations mean ranks are different. The null hypothesis Wilcoxon rank sum test is that the median difference between the pairs is zero. The Wilcoxon rank sum test assumes:

1. Paired values from the same population

2. Pairs are randomly selected

3. Independence

4. Measured on an interval scale

5. Normality is not assumed

4.4 User Performance LOA Five (75% Automated Alerts)

The percent of alerts resolved is calculated by dividing the number of correct actions taken during the test case by the number of alerts received on network during the test case. A high percentage of resolved alerts is desirable.

The boxplot shows the percentage of user alerts resolved on MNDI versus the desktop Figure 4.4. The boxplot shows user performance at automation level five. Automation level five is approximately 75% automation of network alerts. The upper bound of each boxplot represents the top performers and the lower bound the weak performers.

46

At automation level five MNDI outperforms the desktop configuration. MNDI had a median performance of 94.7% versus the desktop's 16%. MNDI's lower quartile performance was 83.2% versus the desktop's 11.3%. Finally, MNDI's minimum was 50%, the outlying data point on the boxplot, versus the desktop's 8.33%.

The Wilcoxon rank sum test was used to test if the two distributions are from the same population. The null hypothesis is that the mean difference is zero, thus the two populations are identical.

The data was tested at a 0.05 significance level. Performing the Wilcoxon rank sum test produces a p value of 0.0003089, rejecting the null hypothesis and thus showing that the data is not from identical populations. The confidence interval is 59.375 to 86.111 on the median difference, summarized in Table 4.1 and Table 4.2.
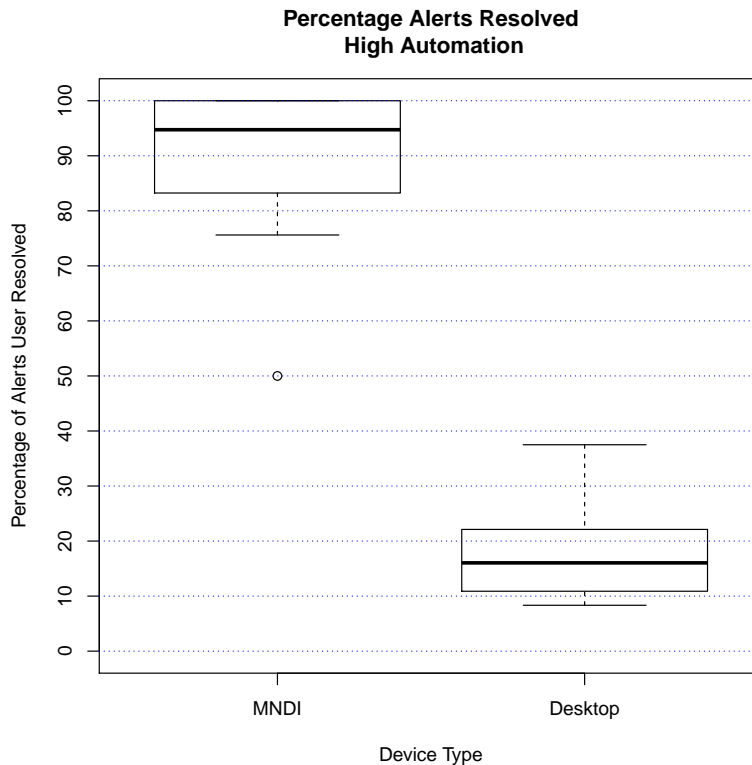

Figure 4.4: User Percent Alerts Resolved Automation Level Five

4.4.1 Average Time Between User Actions LOA five.

Time between actions is defined as the time between submitting actions to the network for implementation, see Figure 4.5. Test subjects average time between actions was calculated by taking the difference between action times and dividing it by the total number of actions taken. A low average time between actions is desirable since this provides an indication that the user is interacting with the system.

The boxplot shows the MNDI's distribution of performance is smaller than the desktop, showing an overall better performance. MNDI had a median performance of 50.5 seconds versus the desktop's 84.8 seconds. MNDI's lower quartile, which shows the top 25% of performers from the median, was 34.8 seconds versus the desktop's 59.4 seconds. The median and lower quartile's are different showing that the top 25% of each device differs. The upper quartile of MNDI and the desktop's lower quartile overlap slightly.

The data sample is tested at a 0.05 significance level using the Wilcoxon rank sum test. The calculated p value is 0.01574, which rejects the null hypothesis and therefore the data sets are not from the same population. The confidence interval is 5.92 to 87.0 on the median difference, summarized in Table 4.1 and Table 4.2.
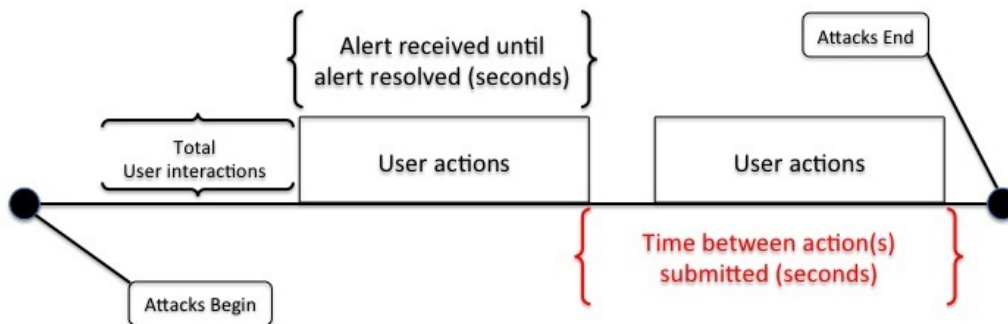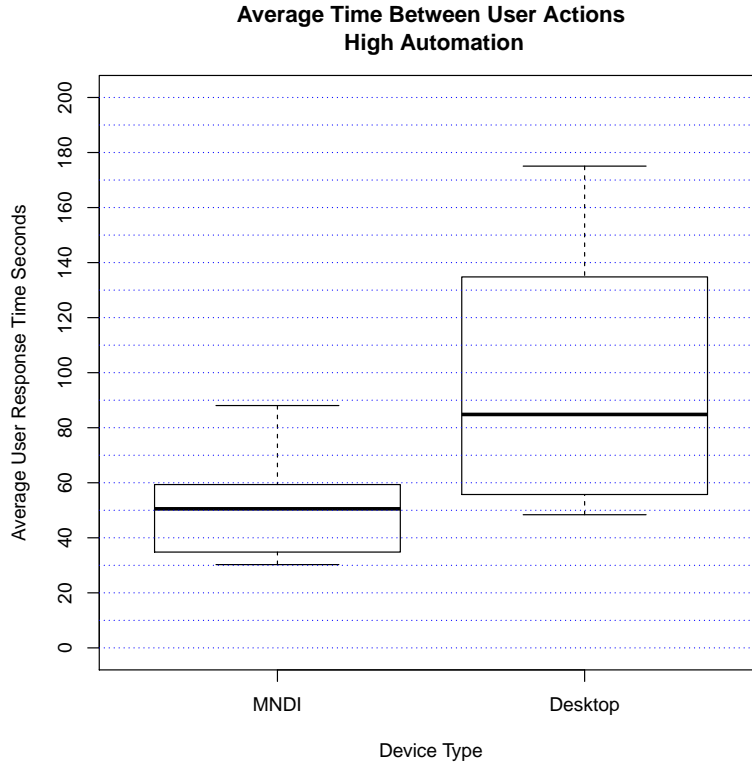

Figure 4.5: Alert/Action Timeline

Average Time Between User Actions
High Automation

Figure 4.6: User Average Time Between Actions Automation Level Five

4.4.2 Analysis of automation level five.

When the automation level is set at five the user is only responsible for addressing approximately 25% of the alerts, on MNDI this may be completed rather quickly. This means the test subject may reach the break period at the end of attack scenario with few or no alerts to address. The break lasts two to six minutes and if the test subject remains idle this entire time it may appear they are not interacting with the system; if the test subjects had chosen to perform user initiated actions during this break it would have lowered the average action time for MNDI users. Conversely, on the desktop implementation the test subject receives all the alerts for the scenario simultaneously and since they take longer per alert, the pause between scenarios does not have any noticeable impact on their performance.

49

4.5 User Performance Level of Automation Six (less than 1% Automated Alerts)

The boxplot, Figure 4.7, shows the percentage of user alerts resolved on MNDI versus the desktop. This boxplot is at an automation level of six. During automation level six, less than 1% of network alerts are automatically resolved. The upper bound of each boxplot represents the top performers and the lower bound the weakest performers.

MNDI had a median performance of 68.9% versus the desktop's 21.4%. MNDI's lower quartile performance was 53% versus the desktop's 18.3%. Finally, MNDI's minimum was 29.9% versus the desktop's 8.26%.

The Wilcoxon rank sum test is used to demonstrate that the two samples are not from an identical population. The data will be tested at a 0.05 significance level to see if the two samples are from identical distributions. Performing the Wilcoxon sum rank test produces a p value of 0.001332, rejecting the null hypothesis and thus showing that the data is not from identical sets. The 95% confidence interval is 20.973 to 73.213 on the median difference, summarized in Table 4.1 and Table 4.2.
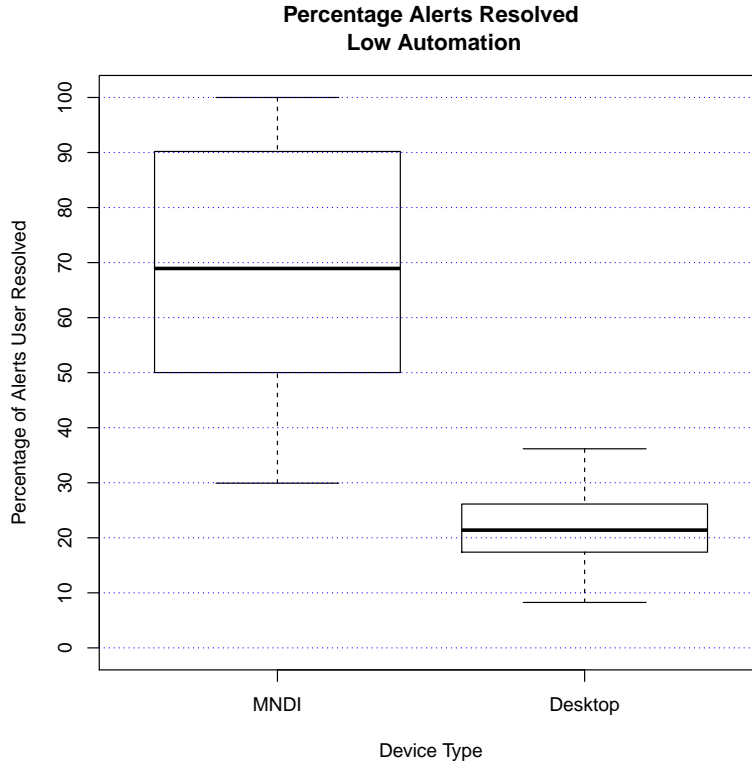
Figure 4.7: User Percent Alerts Resolved Automation Level Six

4.5.1 Analysis.

A significant change occurs in the performance of MNDI users between LOA 5 and LOA 6. This may be attributed to the increase in workload. Median performance of MNDI user's between LOA 5 and LOA 6 decreases by 25.8%; however, the workload has increased by 39.6%. The user is now responsible for the resolution of 99% of the alerts versus 25%.

Desktop users receive a slight increase in median performance between LOA five and LOA six from 16% to 21.4% which is an increase of 5.4%. The slight increase may be due to the lack of automation. All test subjects were briefed on the current automation level and that any alert with an impact above the automation threshold would be automatically resolved. Even though the LOA was known and the impact

level for each alert visible several desktop interface test subjects reported difficulty in correlating which alerts were automated or targeted for automated. When no automated action occurred desktop user could address the alerts at their pace without worrying about duplicating the efforts of the automated system.

4.5.2   Average Time Between User Actions LOA Six.

MNDI had a median performance of 24.4 seconds versus the desktop's 48.2 seconds. MNDI's lower quartile, which shows the top 25% of performers from the median, is 18.3 seconds versus the desktop's 43.9 seconds. While the median and lower quartile's are different showing that the top 25% of each devices user differ, the upper quartile of MNDI and the lower quartile of the desktop interface overlap slightly.

The data sample was tested at a significance level using the Wilcoxon signed-rank test with a null hypothesis is that the median difference between the pairs is zero. The calculated p value of 0.000666, the null hypothesis is rejected and therefore the data sets are not from the same population. The 95% confidence interval is 14.1 to 36.19 on the median difference, summarized in Table 4.1 and Table 4.2.
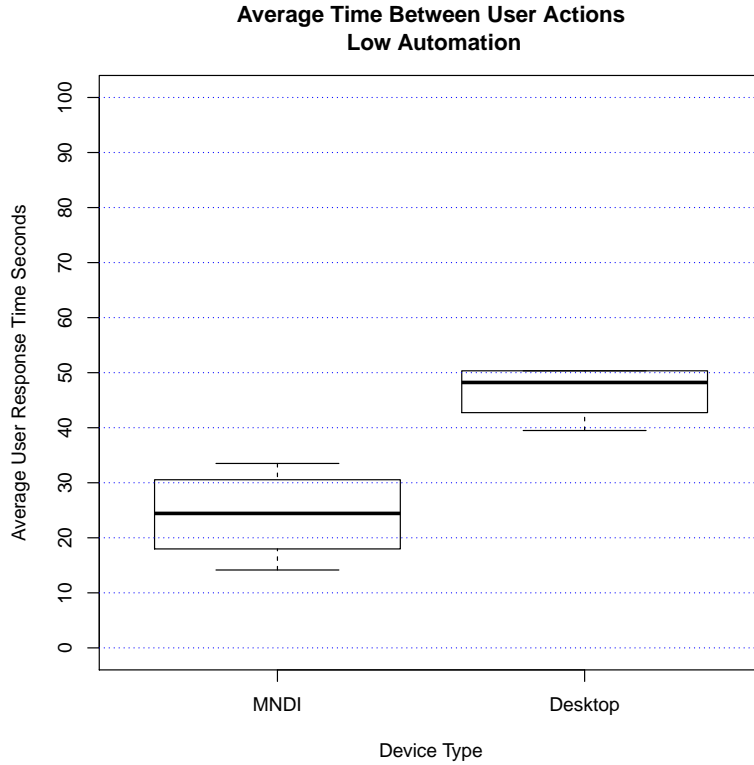
Figure 4.8: User Average Time Between Actions Automation Level Six

4.5.3   Analysis.

During this automation setting both systems are automating less than 1% of the received alerts and become effectively fully manual. With the increase in user workload the advantage of presenting the user with each alert begins to show itself in the boxplot. Not only are MNDI users at automation level six on average resolving 46.7% more alerts but they also are resolving the alerts correctly on average 33.4 seconds faster than user's on the desktop system.
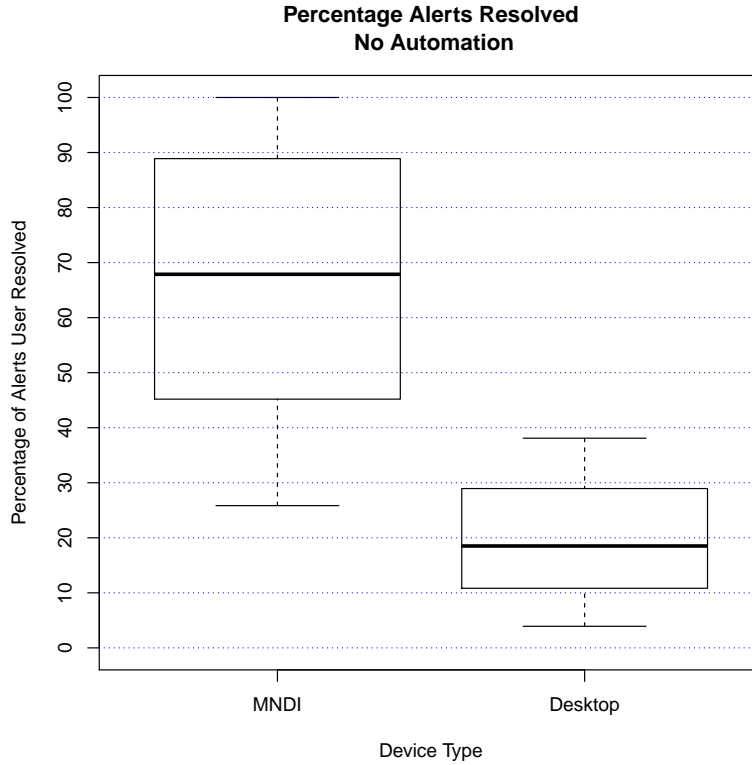
Percentage Alerts Resolved
No Automation

Figure 4.9: User Percent Alerts Resolved Automation Level Eight

4.6 User Performance Level of Automation Eight (Zero Automated Alerts)

The boxplot shows the percentage of user alerts resolved on MNDI versus the desktop. This plot is at an automation level of eight. Automation level eight is no automation of network alerts. The upper bound of each boxplot represents the top performers and the lower bound represents the weakest performers.

At automation level eight MNDI had a median performance of 67.9% versus the desktop's 18.5%. MNDI's lower quartile performance was 50.1% versus the desktop's 11.8%. Finally, MNDI's minimum was 25.8% (the outlying data point) versus the desktop's 3.92%.

The Wilcoxon sum rank test is used to demonstrate that the two data samples are not from an identical population.

The data was tested at a 0.05 significance level to see if the two data sets are from identical distributions. Performing the Wilcoxon rank sum test produces a p value of 0.0005485, rejecting the null hypothesis and thus the data is not from identical populations. The confidence interval was 21.96 to 70.82 on the median difference, summarized in Table 4.1 and Table 4.2.

4.6.1 Analysis.

The performance between automation level eight and six are similar, this is expected since the workload is nearly identical. The MNDI's users outperformed desktop users by 49.4% median difference.

4.6.2 Average Time Between User Actions LOA eight.

MNDI had a median performance of 19.1 seconds versus the desktop's 50.1 seconds. MNDI's lower quartile, which shows the top 25% of performers from the median, is 15 seconds versus the desktop's 40.8 seconds.

The data samples were tested at a 0.05 significance level using the Wilcoxon rank sum test with the null hypothesis that the median difference between the pairs is zero. The calculated p value of 0.00008227, rejecting the null hypothesis and thus showing the data sets are not from the same population. The confidence interval is 19.91 to 39.8 on the median difference, summarized in Table 4.1 and Table 4.2.
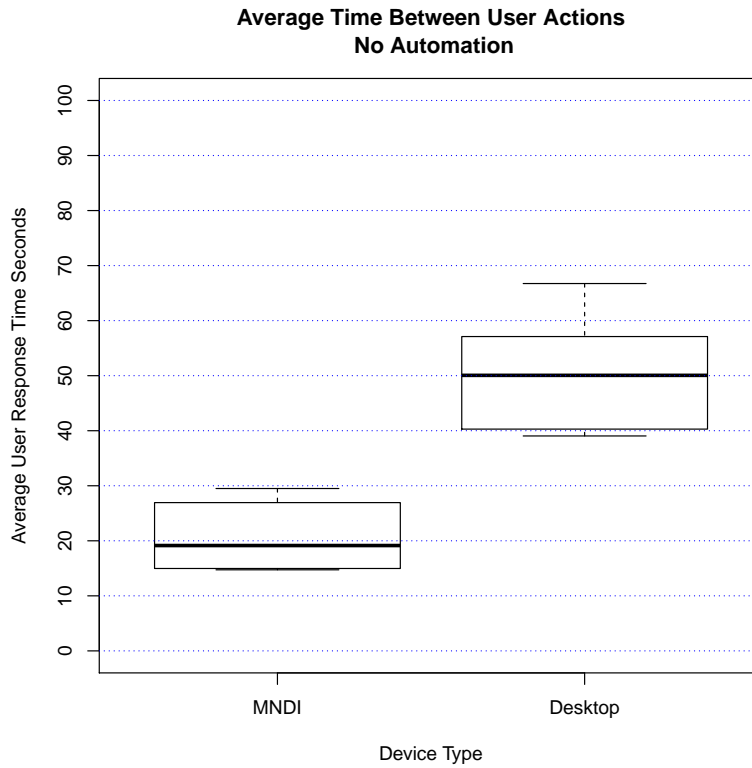
Figure 4.10: User Average Time Between Actions Automation Level Eight

4.6.3 Analysis.

The average number of seconds between user actions on LOA eight was similar to LOA six, which is expected given the nearly identical workload. On average the MNDI test subjects performed actions 31 seconds faster than users of the desktop interface.

4.7   Summary of Statistical Data

Table 4.1: User Percent Alerts Resolved and Average Time Between Actions

LOA	P-Value	Reject H_o	95% Confidence Interval
5	0.0003	Yes	59.38 to 86.11
6	0.0013	Yes	20.97 to 73.21
8	0.0005	Yes	21.96 to 70.82

Table 4.2: User Average Time Between Actions

LOA	P-Value	Reject H_o	95% Confidence Interval
5	0.01574	Yes	5.92 to 87.0
6	0.000666	Yes	14.1 to 36.19
8	0.0005	Yes	19.91 to 39.8

4.8   Self Assessment

Each test subject completed a self evaluation of their network administration abilities prior to the experiment, see Table 4.3 for a summary of scores. The test subjects were asked to evaluate themselves on their skills with Operating Systems, Software/Tools, Programming Languages/Concepts, Protocols, Networking Concepts and Computer Security Concepts, (Appendix A).

Each question required the test subject to rate themselves as either no experience, beginner, intermediate, advanced or expert. A point value of zero (no experience)

through four (expert) was assigned to each question the test subject answered. Test subjects also completed a simple four question quiz with each question scored as a one if correct and a zero if incorrect, (Appendix B). The self assessment score and quiz were totaled to calculate their overall score. The test subjects on each network interface were of similar skill on the desktop and MNDI with an average score of 59.25 on the desktop and 52.64 for MNDI. Based upon the average scores the test subjects were designated novice if there score was below 50, intermediate if their score was greater than 50 but less than or equal to 65 and expert if their score was above 65.

The following figures show the mean alert percentage of correctly resolved events by novice (Figure 4.11), intermediate (Figure 4.12) and expert (Figure 4.13) users on MNDI versus the desktop. Finally, a boxplot comparing the mean alert percentage of correctly resolved events of novice MNDI users versus expert desktop users is presented in Figure 4.14. Test subjects on MNDI outperformed the desktop test subjects in correct alert resolution percentage and average time between actions. A comparison of novice, intermediate and expert users on all three levels of automation shows that MNDI outperformed desktop test subjects. A comparison of expert desktop users versus novice MNDI users shows that novice user on the MNDI outperformed expert desktop administrators. Unfortunately, the small sample size does not allow this research to demonstrate statistical significance.

Table 4.3: Test Score and Skill Designation

User	Platform	Score	Skill Level	User	Platform	Score	Skill Level
1	MNDI	32	Novice	1	Desktop	30	Novice
2	MNDI	33	Novice	2	Desktop	38	Novice
3	MNDI	38	Novice	3	Desktop	38	Novice
4	MNDI	41	Novice	4	Desktop	61	Intermediate
5	MNDI	52	Intermediate	5	Desktop	65	Intermediate
6	MNDI	56	Intermediate	6	Desktop	79	Expert
7	MNDI	60	Intermediate	7	Desktop	80	Expert
8	MNDI	60	Intermediate	8	Desktop	83	Expert
9	MNDI	64	Intermediate				
10	MNDI	65	Expert				
11	MNDI	64	Expert				

4.8.1   Performance of Novice Users.

Table 4.4: Novice User P-Value and Confidence Interval

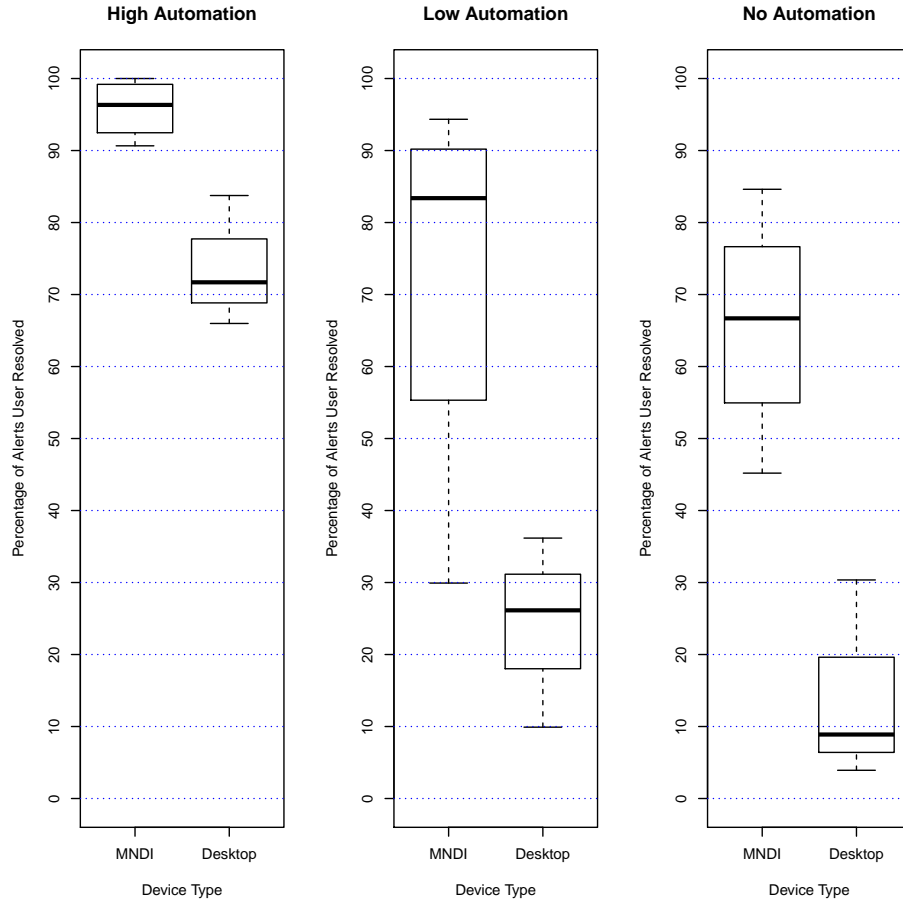| LOA	P-Value	Reject H_o	95% Confidence Interval
5	0.05714	Yes	6.9 to 34.02
6	0.11	No	N/A
8	0.05714	Yes	14.83 to 80.69

Figure 4.11: Novice User Percentage Alerts Resolved

Table 4.5: Intermediate User P-Value and Confidence Interval

LOA	P-Value	Reject H_o	95% Confidence Interval
5	0.07864	No	N/A
6	0.2	No	N/A
8	0.2667	No	N/A
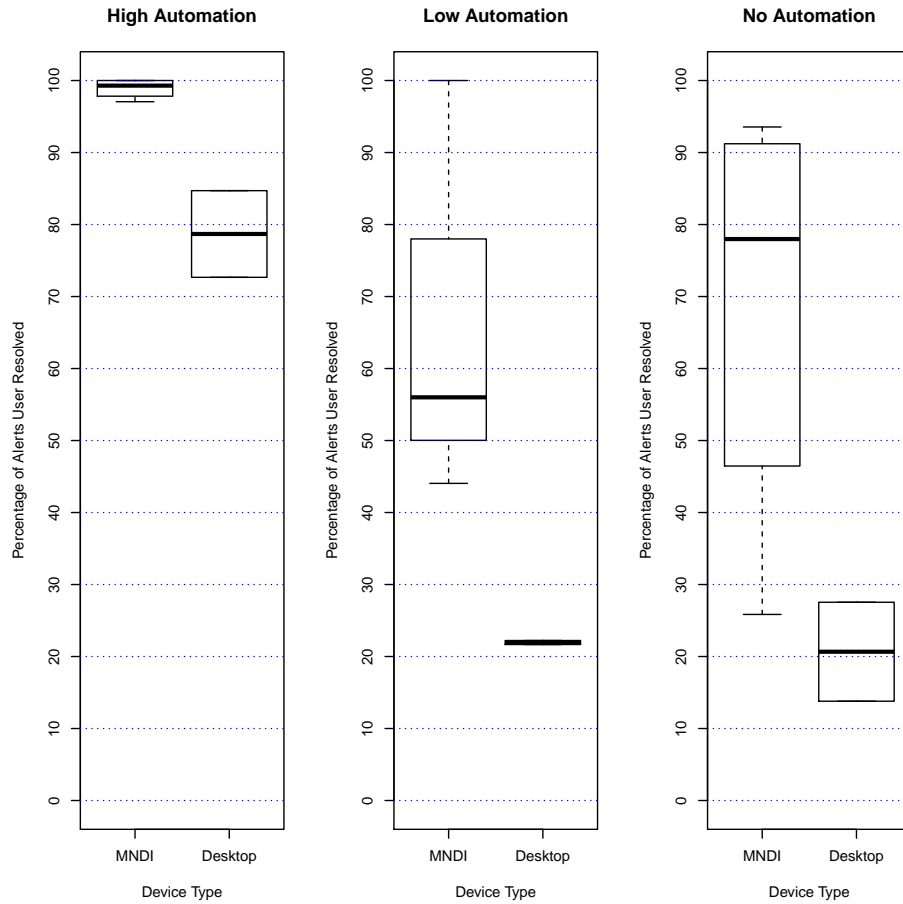
Figure 4.12: Intermediate User Percentage Alerts Resolved

Table 4.6: Expert User P-Value and Confidence Interval

LOA	P-Value	Reject H_o	95% Confidence Interval
5	0.37	No	N/A
6	1.0	No	N/A
8	0.4	No	N/A

61
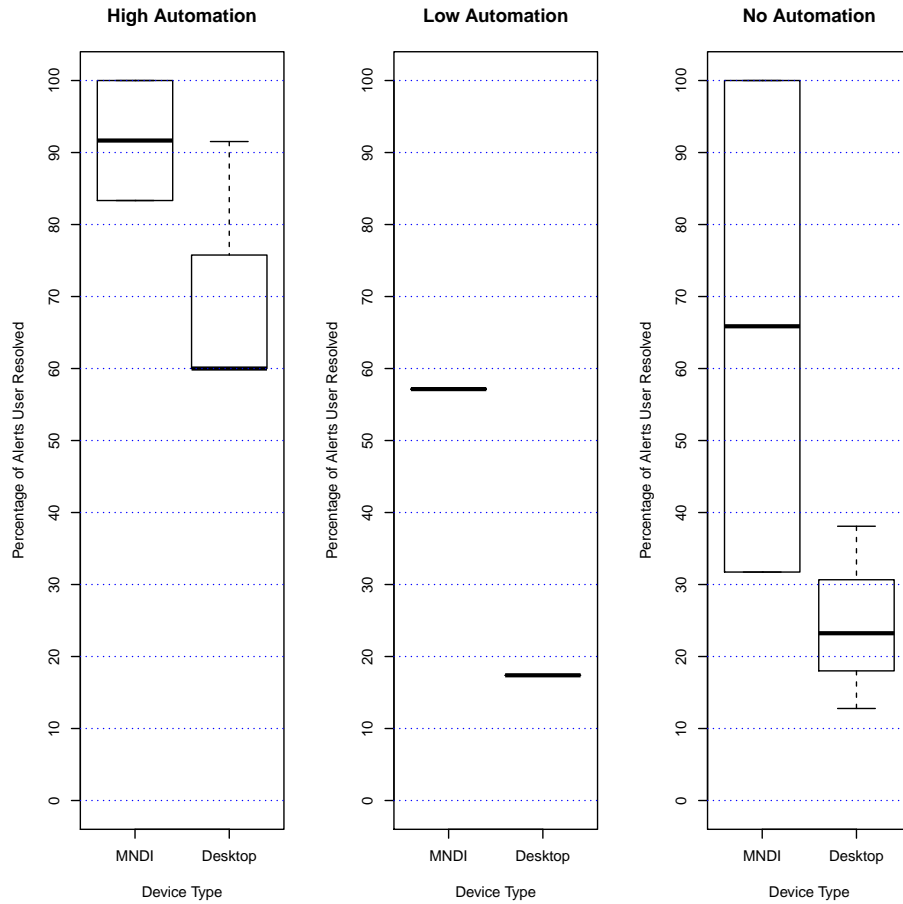
Figure 4.13: Expert User Percentage Alerts Resolved

Table 4.7: Expert Desktop Vs. Novice Mobile User P-Value and Confidence Interval

LOA	P-Value	Reject H_o	95% Confidence Interval
5	0.11	No	N/A
6	0.4	No	N/A
8	0.05714	Yes	7.1 to 71.83
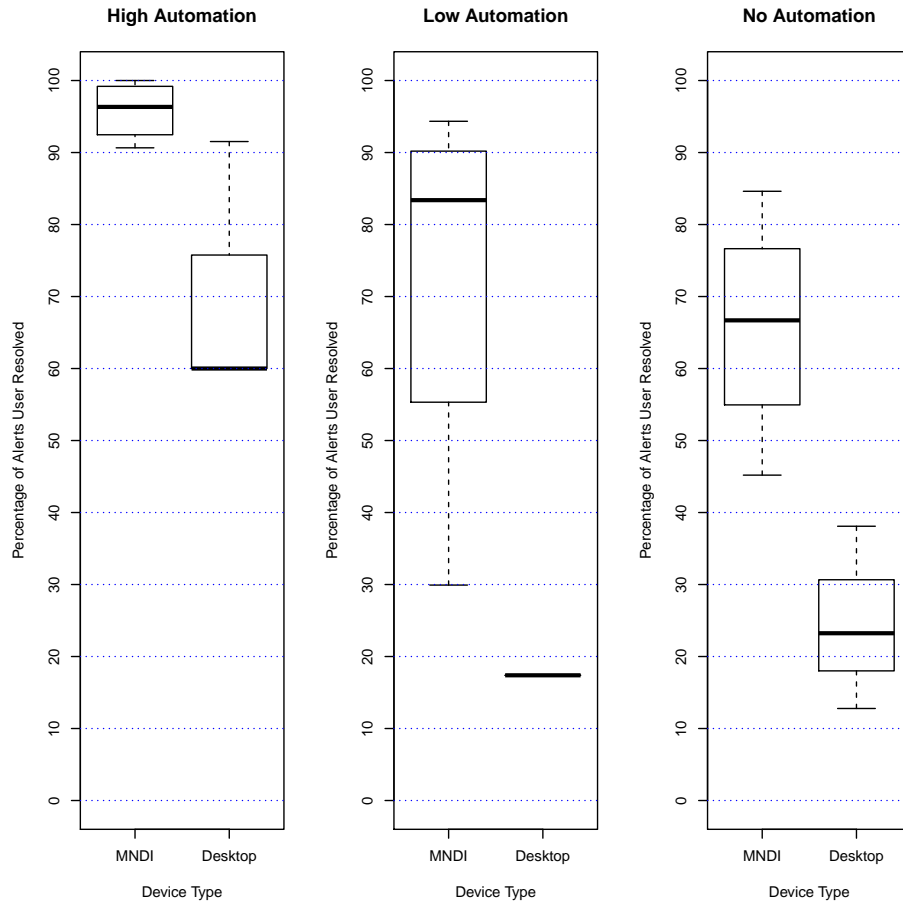
Figure 4.14: Novice Mobile Controller Vs. Expert Desktop Percent Alert Resolved

V.   Conclusion

5.1   Introduction

In the constantly changing landscape of cyber defense network administrators must be able to gain situational awareness of the network quickly. Enabling them to make time-critical policy and configuration changes. Visualization interfaces have accomplished the goal of allowing administrators to gain high-level information more quickly than traditional textual logs. However, most visualization interfaces do not allow the administrators to perform actions, instead administrators require another interface to execute policy and configuration changes. Current visualization also trends toward becoming an all-encompassing data source, seeking to replicate every textual log visually. As a result most visualization interfaces recreate the problem that they sought to abate: information overload. The Mobile Network Defense Interface (MNDI) provides a capability that offers a mobile, touch interface that allows the administrator to take action on the network in real time.

MNDI was designed to provide administrators with the minimum appropriate amount of information to quickly make a correct decision. The MNDI shows promise as an interface to augment current network defense systems. Although the advantages of the mobility were not tested, the MNDI could be used anywhere a network connection is available. The MNDI provides the user with a intuitive graphical representation of the network that allows them to quickly understand the status of the network. The visualization of the possible actions provides an intuitive means to address alerts with multiple possible solution that helped novice administrators perform at a comparable level when compared to more experienced administrators.

5.2 Test Results

During the experiment network administrators of varying self assessed skill levels defended a network of virtual machines against randomly generated cyber attacks, occurring at random intervals. Each administrator was randomly assigned either MNDI or the desktop interface. During each test case the assigned interface was set to a specific level of automation (LOA). Depending on the LOA up to 75% of the alerts were automatically resolved.

Test subjects that used MNDI correctly resolved more alerts and performed actions quicker than users of the desktop interface. MNDI showed a significant advantage over the desktop configuration at all skill levels and all levels of automation. Outperforming the desktop configuration in percentage of alerts correctly resolved and average time between actions. Using MNDI, novice, intermediate and expert test subjects outperformed their peers that were assigned the desktop interface. Finally, surprisingly novice mobile controller users outperformed expert users on the desktop system.

5.3 Goals

The goal of this experiment is to demonstrate that providing network administrators with visualized event objects in a touch-based, intuitive, mobile interface improves their ability to effectively defend their network. By visualizing the environment the MNDI significantly reduced the data a network administrator was required to process to maintain situational awareness of the network status. Providing a list of actions from an expert system (ES) allowed the test subjects to react quickly and accurately. The visualization of actions on MNDI provided a distinct advantage over the typical manual input of scripts onto a command line.

5.4   Future Work

Scaling MNDI to support an enterprise level network is an important step. Maintaining the appropriate level of information for timely, accurate action will become an increasingly difficult problem as the size and number of networks increases. Visualizing the network down to the nodal level may no longer be appropriate. Creating a method to abstract the nodes into larger cluster is important for any interface based on MNDI that aims to visualize multiple networks. Furthermore, the scope of the actions may have to be adjusted as well where actions are not performed at the nodal level but perhaps on the subnet.

MNDI did not provide a mechanism for sorting the alerts it received or controlling which alerts were presented to the user. In the current system the user receives the alerts as they enter the network model. This is not desirable. The sorting feature is a necessity for any interface based on MNDI. Providing the ability to sort by impact, type, identification number or other variables could allow the administrator the ability focus their abilities in a way that is comfortable to them.

Allowing the user to customize the level of automation to their preference is worth exploring. Many test subjects expressed feeling overwhelmed or under pressure when the automated system did not take any actions. Allowing the user to customize the automation level by target nodes, type of alert or impact level to the network would allow them address the alerts at a rate they are comfortable with. Allowing the customization of alerts received for action with custom automation levels would allow the administrator to maintain a workload that is appropriate to their skill set.

The cyber attacks used in the experiment were not very sophisticated and did not follow any kind of attacker methodology. Test subjects were briefed that the ES would execute an action that would resolve the alert but that additional actions may be required to completely resolve the effects of the attack.

Adjusting the complexity of the attacks so that the user would see the value in performing additional actions would add a valuable data point to analyze the situational awareness gained from the mobile visualization. During the experiment, few users chose actions in addition to those taken by the automation system. Automation bias describes a tendency for users to accept the action taken by an automated system even if there is evidence that the action taken was incorrect. Both MNDI and desktop users suffered from automation bias throughout the experiment. Prior to the experiment all users were informed that automated actions would sometimes require additional action from the user to provide the best total solution. Despite the knowledge that the automated action may not provide the best total solution, most users chose not to take additional actions.

Appendix A: Computer Network Proficiency Self Assessment

Subject ID: _____

Computer Network Proficiency Assessment
Please Rate your Experience

Please **check the box** of the applicable letter symbol to indicate your experience for each concept or tool.

- **N** = No Experience (Never used it, never seen it, and/or can't define it)
- **B** = Beginner (Can define the term, General knowledge, Can use concept or tool in limited way)
- **I** = Intermediate (Familiar with concept or tool, Used frequently)
- **A** = Advanced (Possess specialized knowledge, Could instruct others on the subject)
- **E** = Expert (Top 10% of individuals in the Computer Science/Engineering field)

1. Operating Systems

	N	B	I	A	E
a) Windows	☐	☐	☐	☐	☐
b) Linux	☐	☐	☐	☐	☐

2. Software/Tools

	N	B	I	A	E
a) Nmap/Zenmap	☐	☐	☐	☐	☐
b) Packet Capture\Network Sniffer	☐	☐	☐	☐	☐
c) Cain	☐	☐	☐	☐	☐
d) A web proxy (e.g., Burp)	☐	☐	☐	☐	☐
e) Metasploit	☐	☐	☐	☐	☐
f) Network Vulnerability Scanners	☐	☐	☐	☐	☐

3. Programming Languages/Concepts

	N	B	I	A	E
a) Scripting Languages	☐	☐	☐	☐	☐

4. Protocols

	N	B	I	A	E
a) Ethernet	☐	☐	☐	☐	☐
b) IP/TCP/UDP	☐	☐	☐	☐	☐
c) FTP/SFTP	☐	☐	☐	☐	☐
d) Telnet/SSH	☐	☐	☐	☐	☐
e) ARP	☐	☐	☐	☐	☐
f) ICMP	☐	☐	☐	☐	☐
g) DNS/DHCP	☐	☐	☐	☐	☐
h) HTTP/HTTPS	☐	☐	☐	☐	☐

5. Networking Concepts

	N	B	I	A	E
a) Client-Server Model	☐	☐	☐	☐	☐
b) IP Addresses	☐	☐	☐	☐	☐
c) Network Topology	☐	☐	☐	☐	☐

6. Computer Security Concepts

	N	B	I	A	E
a) Encryption	☐	☐	☐	☐	☐
b) Access Control	☐	☐	☐	☐	☐
c) Buffer Overflows	☐	☐	☐	☐	☐
d) Denial of Service	☐	☐	☐	☐	☐
e) Arp cache poisoning	☐	☐	☐	☐	☐
f) Linux password representation	☐	☐	☐	☐	☐
g) Windows password representation	☐	☐	☐	☐	☐
h) Intrusion Detection System	☐	☐	☐	☐	☐

Appendix B: Computer Network Proficiency Quiz

Subject ID: _____

Question:

Q1: What is the most appropriate action to take during a non-distributed Denial of Service Attack?
    a) Block the attacking IP at the Firewall
    b) Disconnect the Internet connection
    c) Block all incoming traffic
    d) Shutdown effected computers

Q2: What is the difference between a Router and Switch
    a) A Router will only send traffic to the specified nodes where a Switch will broadcast all traffic to all nodes
    b) A Router will allow you to connect to the internet where a Switch will not
    c) A Router will send traffic between subnets where a Switch will not
    d) A Router will broadcast traffic to the entire network while a Switch will only broadcast to a subnet

Q3: If your network is being actively scanned, which of the listed tools would detect the scan?
    a) Anti-Virus
    b) Nmap
    c) Firewall
    d) Intrusion Detection System

Q4: What is FTP used for in a computer network?
    a) Request a webpage from a server
    b) Copy a file across the network
    c) Send e-mail traffic to the email server
    d) Establish a two-way connection to another node

Bibliography

[1] Abdullah, K., C. Lee, G. Conti, J.A. Copeland, and J. Stasko. "IDS rainstorm: Visualizing ids alarms". *IEEE Workshop on Visualization for Computer Security*, volume 2005, 01–10. Citeseer, 2005.

[2] Analysis, Base and Security Engine, 2008. URL http://bases.secureideas.net/about.php.

[3] Batsell, S.G., N.S. Rao, and M. Shankar. "Distributed intrusion detection and attack containment for organizational cyber security", 2005. URL http://www.ioc.ornl.gov/projects/documents/containment.pdf.

[4] Bishop, M. "What is computer security?" *Security & Privacy, IEEE*, 1(1):67–69, 2003.

[5] Chittaro, L. "Visualizing information on mobile devices". *Computer*, 39(3):40–45, 2006.

[6] Endsley, M.R. and E.O. Kiris. "The out-of-the-loop performance problem and level of control in automation". *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(2):381–394, 1995.

[7] Fabian, M.A.R.J.Z. and M.A. Terzis. "A multifaceted approach to understanding the botnet phenomenon". *Proceedings of the 2006 ACM SIGCOMM Internet Measurement Conference (IMC)*, volume 2006. 2006.

[8] Foresti, S., J. Agutter, Y. Livnat, S. Moon, and R. Erbacher. "Visual correlation of network alerts". *Computer Graphics and Applications, IEEE*, 26(2):48–59, 2006.

[9] Goodall, J.R. "Visualization is better! a comparative evaluation". *Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on*, 57–68. IEEE, 2009.

[10] Henry, M.S.S., V.S. Kumar, and D. Campus. "Current Status of Mobile Internet Protocol Version 4 and its security issues". 2012.

[11] Howard, M., D. LeBlanc, and J. Viega. *24 deadly sins of software security: programming flaws and how to fix them*. McGraw-Hill, Inc., 2009.

[12] Huang, K.Y. "Challenges in human-computer interaction design for mobile devices". *Proceedings of the World Congress on Engineering and Computer Science*, volume 1, 236–241. 2009.

[13] Jones, R.E.T., E.S. Connors, and M.R. Endsley. "A framework for representing agent and human situation awareness". *Cognitive Methods in Situation Awareness and Decision Support (CogSIMA), 2011 IEEE First International Multi-Disciplinary Conference on*, 226–233. IEEE, 2011.

[14] Kai, H., Q. Zhengwei, and L. Bo. "Network anomaly detection based on statistical approach and time series analysis". *Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on*, 205–211. IEEE, 2009.

[15] Lipson, H.F. "Tracking and tracing cyber-attacks: Technical challenges and global policy issues". 2002.

[16] Livnat, Y., J. Agutter, S. Moon, R.F. Erbacher, and S. Foresti. "A visualization paradigm for network intrusion detection". *Information Assurance Workshop, 2005. IAW'05. Proceedings from the Sixth Annual IEEE SMC*, 92–99. IEEE, 2005.

[17] Manikopoulos, C. and S. Papavassiliou. "Network intrusion and fault detection: a statistical anomaly approach". *Communications Magazine, IEEE*, 40(10):76–82, 2002.

[18] Mansmann, F., F. Fischer, D.A. Keim, and S.C. North. "Visual support for analyzing network traffic and intrusion detection events using TreeMap and graph representations". *Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology*, 3. ACM, 2009.

[19] Marty, R. *Applied security visualization.* Addison-Wesley, 2009.

[20] McGill, Robert, John W Tukey, and Wayne A Larsen. "Variations of box plots". *The American Statistician*, 32(1):12–16, 1978.

[21] Moitra, S.D. "Evaluating the Benefits of Network Security Systems". *Security Automation*, 42, 2011.

[22] Musman, S., A. Temin, M. Tanner, D. Fox, and B. Pridemore. "Evaluating the impact of cyber attacks on missions". *Proceedings of the 5th International Conference on Information Warfare and Security*, 446–456. 2010.

[23] Rapid7. "Microsoft Windows SMB Relay Code Execution", 2012. URL http://www.metasploit.com/modules/exploit/windows/smb/smb_relay.

[24] Rasmussen, J., K. Ehrlich, S. Ross, S. Kirk, D. Gruen, and J. Patterson. "Nimble cybersecurity incident management through visualization and defensible recommendations". *Proceedings of the Seventh International Symposium on Visualization for Cyber Security*, 102–113. ACM, 2010.

[25] Raulerson, E. L. *Modeling Cyber Situational Awareness Through Data Fusion.* Master's thesis, Air Force Institute of Technology, March 2013.

[26] Richardson, R. "CSI computer crime and security survey". *Computer Security Institute*, 1:1–30, 2008.

[27] Saydjari, O.S. "Cyber defense: art to science". *Communications of the ACM*, 47(3):52–57, 2004.

[28] Sharma, A. "Cyber Wars: A Paradigm Shift from Means to Ends". *Strategic Analysis*, 34(1):62–73, 2010.

[29] Shiravi, H., A. Shiravi, and A. Ghorbani. "A survey of visualization systems for network security". *Visualization and Computer Graphics, IEEE Transactions on*, (99):1–1, 2011.

[30] Singh, B. "Network security and management". *Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on*, 1–6. IEEE, 2010.

[31] Skitka, Linda J, Kathleen L Mosier, and Mark Burdick. "Does automation bias decision-making?" *International Journal of Human-Computer Studies*, 51(5):991–1006, 1999.

[32] Skoudis, Ed and Tom Liston. *Counter hack reloaded, second edition: a step-by-step guide to computer attacks and effective defenses.* Prentice Hall Press, Upper Saddle River, NJ, USA, second edition, 2005. ISBN 9780131481046.

[33] Spiceworks, 2013. URL http://www.spiceworks.com/about.

[34] University, Carnegie Mellon, 2002. URL http://www.cert.org/tech_tips/email_bombing_spamming.html.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704–0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
21–03–2013	Master's Thesis	Oct 2010–Mar 2012

4. TITLE AND SUBTITLE

Mobile Network Defense Interface for Cyber Defense and Situational Awareness

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Hannan, James C., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENG-13-M-21

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Dr. Richard Fedors AETC AFRL/RISF 525 Brooks Road Rome, NY 13441-4505 (315)330-3608 richard.fedors@rl.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

This work is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Today's computer networks are under constant attack. In order to deal with this constant threat, network administrators rely on intrusion detection and prevention services (IDS) (IPS). Most IDS and IPS implement static rule sets to automatically alert administrators and resolve intrusions. Network administrators face a difficult challenge, identifying attacks against a vast number of benign network transactions. Also after a threat is identified making even the smallest policy change to the security software potentially has far-reaching and unanticipated consequences. Finally, because the administrator is primarily responding to alerts they may lose situational awareness of the network.

During this research a MNDI was created that visualized a live network under cyber attack. MNDI allowed test subjects to take actions and make configuration changes in real time on the network. The interface was designed to take advantage of state-of-the-art touch technology engaging the network administrator in the defense of the network. MNDI increased administrator's ability to make time-sensitive decision quickly and accurately on their network.

MNDI was tested against a set of open source network administration tool implemented on a desktop system. Both systems used an automated system that polled an ES to resolve zero to 75% of the alerts. The amount of alerts resolved is referred to as level of automation (LOA). During the experiment MNDI outperformed the desktop configuration at all LOAs. The test results showed a statistical difference between the percentage of alerts correctly resolved and the time between actions on MNDI versus desktop test subjects.

15. SUBJECT TERMS

cyberspace, visualization, mobile communications, computer network management

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Maj Kennard R. Laviers(ENG)
U	U	U	UU	85	19b. TELEPHONE NUMBER *(include area code)* (937) 255-3636 x0000 kennard.laviers@afit.edu

Standard Form 298 (Rev. 8–98)
Prescribed by ANSI Std. Z39.18