Scaling and Visualizing Network Data to Facilitate in Intrusion Detection Tasks

A Thesis
Presented to
The Academic Faculty

by

Kulsoom Abdullah

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

School of Electrical and Computer Engineering
Georgia Institute of Technology
May 2006

Scaling and Visualizing Network Data to Facilitate in Intrusion Detection Tasks

Approved by:

Professor John A. Copeland
Electrical and Computer Engineering
Georgia Institute of Technology, Advisor

Professor John Stasko
College of Computing
Georgia Institute of Technology

Professor Henry Owen
Electrical and Computer Engineering
Georgia Institute of Technology

Professor James O Hamblen
Electrical and Computer Engineering
Georgia Institute of Technology

Professor Chuanyi Ji
Electrical and Computer Engineering
Georgia Institute of Technology

Date Approved: April 5, 2006

To my family and

in the memory of my father,

Syed Muhammad Abdullah Kakakhel

Now the words are over

and the pain they bring is gone.

Now you have gone to rest

in the arms of the Beloved.

-Mawlana Jalal ad-Din Muhammad Rumi

ACKNOWLEDGEMENTS

Without the help and blessings of Allah (swt), the completion of this dissertation and overcoming all the obstacles I have encountered along the way would not have been possible. It would also have not been possible without the help of friends, colleagues and those involved in my educational pursuits.

First, I want to thank my advisor, Dr. John A. Copeland, for patience, help and concern during my PhD career. He allowed me to have freedom and flexibility in my research choices and helped me every step of the way. Though the time it took me to finish was long, he remained consistently supportive regardless. He is truly one of a kind in character. From him, I learned how to approach research problems and make solutions apply to the real world as well.

I am appreciative and thankful for the support of my committee members: Dr. Chuanyi Ji, Dr. James Hamblen, Dr. Henry Owen, and Dr. John Stasko. Thank you for serving on my dissertation committee, giving me feedback and approval of my research goals to help me have a good thesis. I would particularly like to give a special thanks to Dr. Owen for providing feedback and support throughout my time at Georgia Tech and to Dr. John Stasko for providing encouragement and support for the information visualization aspects of IDS Rainstorm.

Thanks also goes to OIT who helped provide data for my research, specially Russell Clark and Jonathan Glass. Also to Bill Guilford, Peter Wan, Herbert Baines and the rest of the Internet Security staff for testing IDS Rainstorm and providing feedback.

The years in the Computer Systems Communication (CSC) lab were full of interesting times and discussions which kept me entertained. Thanks to the present and past members of this group for your friendship. Thanks to Dr. Raheem Beyah at Georgia State, an alumni

of CSC. Not only was he a colleague but he also advised us on our research. For helping me stay together and wade through the administrative sea of bureaucracy, thanks goes to Kathy Cheek. Regards to Brian Strickland for being patient in helping us setup the network in the lab and answering all my questions. I am grateful to Greg Conti for collaborating with me in network security visualization, also providing a lot of feedback and support.

Finally, without the support of my family and friends, I could not have completed this thesis. My parents, Syed and Ghazala Abdullah provided me with emotional and financial support that enabled me to continue with my education. They encouraged me to pursue all of my goals. I wish that my father could be around to see this day, for as going as far as I could in my studies was not just my want but his as well.

TABLE OF CONTENTS

DEDICATION . . . . . **iii**

ACKNOWLEDGEMENTS . . . . . **v**

LIST OF FIGURES . . . . . **x**

SUMMARY . . . . . **xii**

I INTRODUCTION **1**

1.1 Background 3

1.1.1 Data 3

1.1.1.1 Network Traffic 3

1.1.1.2 Intrusion Detection Alerts 4

1.2 Contribution 5

1.3 Thesis Organization 5

II RELATED WORK **6**

2.1 Visualization and Network Security 6

2.2 Visualizing Network Links and Layout 6

2.3 Network Statistics Visualization 8

2.4 Network Log Analysis 10

III DATA PROCESSING **12**

3.1 Scaling Data 12

3.1.1 Overall Graph Occlusion 12

3.1.2 IP Address and Port Number Scaling 14

3.1.3 Time 16

3.2 Data Parameters 17

3.2.1 Packet Capture and Header Fields 17

3.2.2 Alarm Generation Using the StealthWatch IDS 17

3.2.2.1 Alarm Parameters 18

IV VISUALIZATION SYSTEMS 21
4.1 Visualizing Port Activity 21
4.1.1 Design Goals 21
4.1.2 Methodology 22
4.1.2.1 Histogram Plots 22
4.1.2.2 Axis Parameters 22
4.2 Visualizing IDS Alerts 23
4.2.1 Design Goals 23
4.2.2 Main View 24
4.2.3 Zoom View 26
4.2.4 Other techniques 27
4.2.4.1 Glossing 27
4.2.4.2 Indexing 28
4.2.4.3 Filtering 28
4.2.4.4 Panning 28
V THREAT MODELS AND EXAMPLES 32
5.1 Port Statistics Visualization 32
5.1.1 Network Scanning and Mapping 32
5.1.2 Viruses, Worms and Trojans 33
5.1.3 Backdoors and Rootkits 33
5.1.4 Summary 35
5.2 IDS Alert Visualization 38
5.2.1 High Alarm Count 38
5.2.2 Exteral IP Connection Patterns 38
5.2.3 Dorm Activity 40
5.2.4 Worms, Viruses, Trojans 41
5.2.5 Botnet 42
5.2.6 Summary 43

VI USER STUDY 45
6.1 Background 45
6.2 Methodology 47
6.3 Results 47
VII CONTRIBUTIONS AND FUTURE WORK 50
7.1 Research Contributions 50
7.1.1 Port Activity 51
7.1.2 Alerts 51
7.2 Future Work 51
APPENDICES
APPENDIX A — VISUALIZATION RESOURCES 53
REFERENCES 56
VITA 60

LIST OF FIGURES

1 Scaling packet count using cube root for botnet traffic capture[1] 13

2 A system compromise by a botnet followed by the victim joining the botnet and transmitting traffic. Packet count every 15 minutes.[1] 13

3 Concept diagram of plotting technique. Port ranges are categorized and plotted on the graph 14

4 Collision of alarm alerts occuring in close time and IP space proximity . . . 14

5 Packet header fields parsed for statistics gathering 17

6 Design of the Basic Visualization and Representation. Each vertical axis represents IP addresses in sequential order. Each horizontal axis associated with the vertical axes represent one 24 hour period. 23

7 IDS RainStorm main view: The 8 vertical axes are shown that represent the 2.5 Class B IP addresses. The thicker horizontal lines between these axes show where each Class B starts. The other horizontal lines show the start and end of each department. Those addresses not in a department are either unallocated or reserved for special use by OIT and other departments. This screenshot shows an entire day's worth of real alarms generated. 24

8 User selecting area to zoom in. (Here we have artificially highlighted a region in green and magnified it to assist the reader.) The IP address shown is the one at the top position in the red box region (the last two octets are intentionally blurred for privacy). The result of the zoom is shown in Figure 9 25

9 Zoom of a cluster of alarms seen in the overview. Also shown is the alarm severity legend and internal and external IP axes layout. The selected subset of internal IP addresses are represented on the left vertical axis, and external IP addresses on the right vertical axis. 26

10 Concept diagram of the zoom view in IDS RainStorm. 27

11 Gloss of alarm with corresponding line/arrow and external IP highlighted. . 28

12 Screenshots of zooming and panning use to focus on a particular portion of a cluster of alarms. 29

13 Indexing through alarms with mouse button clicks. 30

14 Two screenshots showing panning movement in the zoom view with the coressponding red box movement. 31

15 Network scan from honeynet, packet count every 2 minutes. Pattern or ports probed over time is seen for this scan.[1] 32

16 Success of the Sasser worm and start of the worm traffic - count for every 30 minutes[1] 33

17 Sasser graph with sorted out ports: 445, 135, 139, 1026, 53 and 80 34

18 Sasser graph of the p400 range filtered for focus. In this case traffic is seen to be from port 445 34

19 Same packet count of Sasser using a smaller time scale of 5 minutes.[1] . . . 36

20 Scaling packet count using cube root for botnet traffic capture.[1] 37

21 A zoom view on time. This zoom is a double zoom view of Figure 9. Internal IP addresses are on the left and external IP addresses are on the right. 39

22 June 22nd overview. Clicking and dragging on the overview appears in the zoom view (shown in Figure 23) and animates the traversal down the IP space. The external IP axis is held constant. 40

23 Panning results of Figure 22 shown as two transitions. 41

24 Overview of April 27th alarms. (Two regions are artificially identified in green and magnified for easier viewing.) Here region 1 shows activity in a subset of campus dorm IP addresses, a cluster of activity for a machine in the dorm is outlined in region 2 and region 3 shows a cluster of activity in occurring over a small range of IP addresses for the entire day. 42

25 April 26th worm activity for a particular host located in the campus dorms. The left side is the zoom view of region 2 highlighted in Figure 24. The right side is a zoom of the left view that shows the activity from 3:00-6:00pm. 43

26 Bot activity shown for the same IP address space on April 26 (left) and April 27 (right). The activity time pattern for the two days is almost identical. 43

SUMMARY

As the trend of successful network attacks continue to rise, better forms of intrusion, detection and prevention are needed. This thesis addresses network traffic visualization techniques that aid administrators in recognizing attacks. A view of port statistics and Intrusion Detection System (IDS) alerts have been developed. Each help to address issues with analyzing large datasets involving networks. Due to the amount of traffic as well as the range of possible port numbers and IP addresses, scaling techniques are necessary.

Our approach to port statistics improves upon current techniques that lack effectiveness due to an overemphasis on flow, nodes, or assumed familiarity with the attack tool, causing either late reaction or missed detection. A port-based overview of network activity produces an improved representation for detecting and responding to malicious activity. We have found that presenting an overview using stacked histograms of aggregate port activity, combined with the ability to drill-down for finer details allows small, yet important details to be noticed and investigated without being obscured by large, usual traffic. We use Georgia Tech Honeynet traffic to test our design and show its effectiveness.

Another problem administrators face is the cumbersome amount of alarm data generated from IDS sensors. As a result, important details are often overlooked, and it is difficult to get an overall picture of what is occurring in the network by manually traversing textual alarm logs. We have designed a novel visualization to address this problem by showing alarm activity within a network. Alarm data is presented in an overview from which system administrators can get a general sense of network activity and easily detect anomalies. They additionally have the option of then zooming and drilling down for details. The information is presented with local network IP (Internet Protocol) addresses plotted over

multiple y-axes to represent the location of alarms. Time on the x-axis is used to show the pattern of the alarms, and variations in color encode the severity and amount of alarms. Based on our system administrator requirements study [9], this graphical layout addresses what system administrators need to see, is faster and easier than analyzing text logs, and uses visualization techniques to effectively scale and display the data. With this design, we have built a tool that effectively uses operational alarm log data generated on the Georgia Tech campus network.

For both of these systems, we describe the input data, the system design, and examples. Finally, we summarize potential future work.

CHAPTER I

INTRODUCTION

The expansion of network processing power and higher capacity network links have increased the amount of data and information that can be exchanged over the Internet. The Internet is the fastest growing information medium [33] in the world. Processing power is surpassed by traffic bandwidth speed, thereby making it easier for an attacker to get through without being noticed [26]. With these advances also comes complexity and vulnerabilities in networks. Tools and methods have been developed to deal with the influx of network attacks and misuse but they are not foolproof. Network security operators are overloaded with textual logs and interfaces using primitive graphs generated from these security appliances that prevent them from accurately determining significant problems. Quick and accurate action is needed to effectively combat security issues, but the amount of data and alerts make this impossible as often, many alerts are ignored. This decreases the value of security appliances and leads to many successful attacks.

There are continually new vulnerabilities available to target, and there is no sign that this trend will slow down. There are also ongoing "zero-day" exploits, those without a software patch, that are particularly difficult to stop. Intrusion detection systems were developed to analyze network traffic and alert human operators to security issues.

Research in network security using visualization shows great promise. At its core is the innovative use of information visualization techniques to assist those who need to analyze network data for anomalous behavior. According to Card, Mackinlay and Shneiderman, [12] information visualization is "the use of computer-supported, interactive visual representations of data to amplify cognition." Visualization tools are used to convey information

from a set of data. A properly designed visual representation, as opposed to a textual representation, allows one to understand a greater amount of data in shorter time. Researchers in psychology have shown that humans can process pictures, a parallel process, faster than text, a serial process. Images facilitate understanding and insights that one would not have made if that same data were presented in other textual ways. Images are also easier to remember because humans think and learn visually [45]. Network security visualization helps analysts by scaling and visualizing data and facilitating the identification of patterns in the network in order to make decisions accordingly. A prototype and tool are presented that deals with two types of data that is analyzed for detecting network security events. Both have been designed and use novel scaling methods to deal with processing the data and providing effective visualizations. The prototype visualizes network packet statistics and the tool visualizes alert log files over time.

Network attacks can be characterized by port activity. Though the amount of packets is tremendous, packets can be parsed online, where real time processing can be performed without having to wait for a flow to end to update statistics. The payload is not checked, but with so much traffic data, processing each packet would overburden a monitoring system. The prototype aids in intrusion detection by visualizing network packet header data over time. It is easy to understand, and allows those with minimal knowledge of network security to use it effectively. Scaling techniques have been developed to reduce occlusion and stacked histograms are used to efficiently visualize the data.

Intrusion detection systems were developed to analyze network traffic and alert human operators to security issues. Alarms are generated when set statistical thresholds have been reached, or when a certain sequence of events has occurred. While alarm logs are much smaller than network traffic capture logs, the amount generated is still large. Time wasted by analyzing these logs can effectively negate the value of the system. If real attacks are not stopped, they will generate many alarms [17]. As a result, logs can rapidly become filled with redundant information. This fact, together with the average amount of unique

alarms generated, can cause information overload and possibly hide the most significant attacks [25]. To address the problem of oppressively large alert logs, we introduce a tool that provides security system administrators with an informative, information-rich display and a convenient interface for monitoring network security alarms as well as researching details on a user-selected subset of those alarms.

1.1 Background

1.1.1 Data

Data that is analyzed and processed for network security comes in different levels of detail ranging from raw network packets to high level semantic information like IDS alerts [14]. Security appliances that generate this data can be IDS sensors, firewalls, servers, network sniffers and honeynets. Analyzing just one source of data is not enough to be fully aware of network state. Ideally, one should look at multiple types of data, as each data source can provide different information on the activity in the network.

Two systems presented use both lower-level data, network packet headers, and high-level data, IDS alarms. A background on these data inputs is given in the next sections.

1.1.1.1 Network Traffic

Of the data types, raw network packets are the largest amount. They contain the actual data and headers that traverse a network. To visualize port statistics, information needs to be extracted from packet headers. Implementing this on a general network will consume resources and not function efficiently. We chose to use Georgia Tech Honeynet traffic for this reason. Production networks contain large percentages of legitimate traffic, but Honeynets are different because they include primarily illegitimate or suspicious activity. Honeynet captures are being used initially because they give a good benchmark to test the effectiveness of the prototype. Analysis of Honeynet captures for forensic purposes have been time consuming but successful in detecting worms and in finding compromised machines on the Georgia Tech network [32]. Forensic analysis of an attack takes much of

the analyst's time. Any attack takes much longer to examine than it takes to occur (a few seconds vs. a few days). Our primary goal is to lessen the time of analysis. Specifically, we have made port activity easier to observe with our visual prototype, as opposed to reading the packet capture text which is how forensic analysis usually occurs.

1.1.1.2 Intrusion Detection Alerts

Though the amount of alerts is much lower than that of network packets, the number of alerts is still significant and cumbersome for network security administrators. User surveys showed that professional security analysts can deal with more alerts than the intermediate user (eg. network security students) but 230 alerts per hour is the threshold for them [14]. Of the 15 people surveyed who use the defacto IDS, Snort, 40% felt that there was a learning curve getting used to using the tool and the GUI that comes with Snort. With Snort setup on a network, Sixty percent felt that tuning the signatures was difficult and the signatures itself were not that effective in protecting the network from attacks. Forty four percent of these people thought that tuning the IDS as to reduce false alarms and successful attacks was difficult.

To design a tool to visualize IDS alerts, we interviewed Georgia Tech's Internet security administrators and found out how they use their current IDS tools, what they look for to determine an attack and what information is helpful. The average number of alerts on campus is 50,000. For design and testing purposes, we used alarms generated from one sensor that monitors the periphery of the campus network, which averages about 7,000 alarms per day. The administrators mentioned that they are overwhelmed with this amount and with the time it takes to go through the alert logs to determine if significant events are occurring and whether action is needed or not. We have designed the tool with this in mind and to reduce the time of analysis.

1.2 Contribution

Two systems have been developed to visualize two types of network data. In the process, techniques were developed to scale and process data to enable better visualizations that will be helpful to the users tasks. These methods could be used in other applications that require dealing with very large amount of data in addition to multivariable data and data with parameters of large ranges. Also, useful visualizations were designed for these data types that can contain a lot of information as well as represent good overview, with drill down techniques. Finally, a user study was performed to guide the design of the alarm visualization and then when the tool was developed, it was evaluated and feedback was given.

1.3 Thesis Organization

The remainder of the thesis is organized as follows: In chapter 2 we talk about related work in the field of network visualization and security network visualization. Chapter 3 describes the data-set in more detail, along with scaling the data and parameters associated with it, general and specific to the data we are using. Then details on the data parameters itself are given. The design of the systems and how they function are explained in chapter 4. Common threat models are presented in chapter 5 to show how the systems are effective for finding anomalous activity.

A detailed description of our user study performed to help design the IDS alarm visualization and surveys performed after the tool was used is discussed in chapter 6.

Finally chapter 7 gives the conclusion of the thesis along with potential future work.

CHAPTER II

RELATED WORK

2.1 Visualization and Network Security

Original work in network visualization was done as a map layout to understand network link behavior. Network layout methods have also been used in conjunction with network security. Other data sources visualized are network flow and packet statistics which will be discussed and finally visualization of network log files.

2.2 Visualizing Network Links and Layout

The work done in [11, 15] are representative of some earlier work done in using glyphs to represent nodes and lines to represent links. Overcrowding of the nodes and links is a problem with layouts. Authors of [15] used arcs lines and nodes on globe in a 3D view to help reduce clutter. Arc height was related to statistics enabling crucial links to be shown on the top so they were not hidden by other links. The authors of [11] allowed the data to be filtered, which reduced the amount of data on the layout. User interaction allowed variation of the symbol size, filtering of data presented, and using geographical maps. Recent work on Internet mapping has been done by CAIDA [4].

There is some recent work that uses network layout methods for security. The tool presented in [21] is used with the Hummer IDS and log files, uses glyphs and nodes where line type and circle attributes show information about the link and system respectively. A large circle, usually in the center, represents a server, while smaller nodes at a distance represent systems connecting to the server. Nodes outside are placed either randomly or according to force equations to avoid overlap. Colored links highlight unusual activity. A replay of events is possible with the VCR like GUI. Lines and nodes fade when the

connection ends. Haptics were used in [36] to visualize layout, and placed nodes according to equations from gravitational theory, electromagnetics, and fluid dynamics. Link color represented a severe attack.

Other work that represents IP addresses in a logical rather than topical fashion do not have the same occlusion issues. Some use parallel plots in which a vertical axis represents the entire IP addresses space from 0.0.0.0 to 255.255.255.255 where lines between them show connections. VisFlowConnect is different, in that it uses parallel plots to show the connections between the inside network to the outside network [46]. Lines that show links fade over time while line width shows traffic volume. Filtering can be done for protocol, port, IP address and traffic threshold. Possible signs of attack were seen by dissymmetry in the plot by connections where for e.g. an inside host connects to many outside subnets. Also unusually large amounts of traffic going back and forth are a possible indicator of malicious activity. Rumint [13] also uses parallel plots to show connections. It also gives users the option of mapping axes to other parameters, like port number. IP matrix [29], uses 2 matrix of 256 by 256 to represent IP space so each one can represent two octets of an P address. Local and internet level IP addresses are seen at the same time. The advantage is that there is more focus on local addresses as compared to the parallel plots where because the entire IP space is plotted, the scale of one class B will be very small. The disadvantage is it is limited to one class B and connections can not be shown. TNV [23] shows connections and also uses a time line and one can also view port connections. IP addresses are mapped on vertical axes and time on the horizontal, so time patterns can be seen but there is a limited amount of IP addresses that can be represented.

Some recent work, like [37], have different visuals, the alert plot which is a combination of 2D and 3D views. Two 256 by 256 matrix used to represent IP addresses as in [29] where each square represent 2 octets of an IP address. Lines from these are drawn to another set of matrixes to show connection to corresponding alarm glyphs. This has occlusion problems since all adresses are represented in this area and it is easy for lines to overlap. It does not

make efficient use of screen space but zooming and panning are implemented. Figuring out connections between IP addressees is harder to correlate with this scheme. Also, there is no representation of time to visualize time patterns, all information seen in the screen is for a specified time window. Filtering by priority helps with some of the occlusion issues and detail on demand is available by clicking on an alarm circle to get more textual details.

These tools and techniques are useful in seeing a visual map of a network but issues like occlusion, inefficient use of screen space and scalability are problems. The IDS alarm visualization tool we have developed improved on these techniques by focusing better on local IP addresses using multiple axes rather an a 2^{32} mapping, which scales better, is less occluded, makes better use of screen-space and uses a timeline for viewing time patterns. More is explained in the design section for IDS RainStorm (chapter 4) and in [9]

2.3 Network Statistics Visualization

Most systems use network flow data as the source of their visualizations. Therminator is used with Lancope's IDS, StealthWatch [7]. Therminator is a non-signature based real-time visual tool. StealthWatch is also non-signature based, does monitoring in real time and can be used for forensic analysis. Therminator is based on thermodynamic theories of energy, entropy and temperature [22] and keeps track of network state over time [18]. Network state is defined by the current setup of buckets and balls. Buckets are conversations that are defined by information from packets, such as if hosts or networks are transmitting data, the IP protocol and whether it is coming from a client or server. Balls represent the data that is transferred among buckets. Each time a packet arrives a new network state transpires and thermodynamic values are calculated and graphed. Any change in the symmetry of the graph means there is some anomalous behavior. The IDS also provides a detailed text event log. Flowscan [38] and NVisionIP [31] use Netflow data in its graph visualizations. Netflow was developed by Cisco for their routers and now is open source that other routers can use. It only collects flow information and not payload so the capture

is smaller than raw packet capture. The extra information that a regular packet header does not have are flow size, flow packet count, start and end of flow, and router IP (that collected the flow information). Some disadvantages are that the flows are reported after the termination of the flow so shorter flows will be reported first, according to the end time and independent of the starting time. Also it is possible this information can be spoofed and intercepted. Flowscan has an area plot of flows by amount of flows, the IP protocol (TCPin/out, MCAST in/out, UDP in/out, ICMP in/out, Total in/out), and direction (in or out of the network). Dissymmetry of in vs. out can indicate a problem such as in a DoS or port scan. NVisionIP parses flow information from Netflow. From the flow statistics gathered, NVisionIP implements three views which help to give an overall view of the network. In the galaxy view, an entire class B IP space is shown via a 256x256 grid matrix, where each point reps a unique class B IP address. Each point's color represents how many times that IP address was a source or destination of a flow. Further detail is seen in the small multiple view which shows a subset of the IP addresses and histograms of flow port count for each IP. Choosing one IP address to focus on gives the machine view where for that address a user sees the flow port count for well-known and dynamic ports. Some more recent work, as in [37], uses a 3D axis to show time, port numbers and number of bytes or packets. The port axis is not fixed; it is dynamically allocated according to activity. To avoid possible occlusion of port labels on the axis, the ports represented are limited to those that have a certain amount of activity. If a port scan occurs, such as large scale scan, e.g. web server, it would be seen rather than a basic host scan, because the amount of activity is greater for a bigger scan and those so those ports will be seen. Scaling is not used, and high values can skew results but it has an extra color bar that shows average amount over time in addition to the actual value for that time instant. IDgraphs [39] uses flow records and represents over time the following choices: number of unsuccessful attacks over time or unique IP and destination port pairs. To avoid overlapping pixels, luminance is used to show data density. PortVis [34] represents destination port as a 2 byte number mapped to an x and y axis

matrix. Pixel location represents a port number and color represents number of sessions. One can magnify on a particular area and select a port to see its histogram over time. One would notice something wrong if you notice the colored pixels in the matrix. If a set of contiguous ports have a lot of activity, and a port amongst them has suspicious activity, it is harder to know that one needs to focus on that. Time patterns are not represented as all data in one screen is shown for a window of time.

The advantages of plotting port statistics based on network header information is that a flow does not have to finish for it to be represented. For especially long flows, the information will not be presented in a timely manner. It uses scaling and grouping techniques to allow all information to be seen at once while not becoming occluded and maintain good use of screen space. More is in chapter 4 and [8].

2.4 Network Log Analysis

There have been several innovative approaches to log analysis. Representative of general log visualizations developed for security are Tudumi [44] and Mielog [43]. Tudumi shows server connection in a 3D visualization where lines represent connections and system nodes are placed on rings. Information from several logs are combined to form this layout. Use of 3D allows more network groups to be portrayed, but occlusion is a problem if the number of networks and nodes start to grow. Mielog shows log information in real time. Each line in the graph represents one line in the log file. This method allows an overall view of the log file along with straight-forward pattern observation.

Several tools have been developed to visualize and process Snort IDS [40] alarm log files. One is SnortView [28] where a matrix view is used to show IP address connections over time. Color is used to highlight user selected communication paths, and color is also used to encode the alarm severity (high, medium and low priority). Glyphs are used to encode network protocol type. Detailed information for the currently selected alarm is given at the bottom of the display. This tool is successful in combining multiple parameters,

visually representing them to assist analysts in finding anomalous behavior. However, the amount of information shown is limited to a subset of IP address ranges, time (4 hours), and number of attacks.

SnortSnarf [24] and ACID [16] display the Snort logs in a tabular format with limited visualization, thus there is little to distinguish between these and traditional log files. The only data processing performed are statistical analyses, but they are performed on a static log file where real time data viewing is not possible. RazorBack [10] provides a GUI interface for viewing alarms where alarm priority is represented by colored circles, and web browser reloading provides updated data. Again, this is not a significant improvement over alarm log files. What differentiates IDS Rainstorm from other tools is that it represents 2.5 class B IP address spaces (65,532 hosts $\times$ 2.5 = 163,830 total) on one display. Mapping alarms to pixels encodes a large amount of alarm data into one screen for a full 24-hour period. Continuation of their work is in [29], which focuses more on IP addresses patterns by using 2 matrixes of 256 by 256.

Our tool is able to show more information related to IDS alarms, represent more IP addresses with better scale and focus, show time patterns and implements zooming, panning and detail on demand compared to the other tools developed to visualize IDS alarms.

CHAPTER III

DATA PROCESSING

This chapter describes how network data needs to be processed before it is inputted to a visualization system, covering both the data scaling process and the specific data parameters the systems use. Because of the high amount of traffic and related data generated, processing is necessary to produce a meaningful representation.

3.1 Scaling Data

Data needs to be scaled both before and after it is used as an input for a visualization because of the tremendous volume of data and the extensive range of parameters. Scaling helps to avoid occlusion on the actual graph: without scaling, information might overlap because display space is limited. Without scaling, it is also difficult to represent parameters that have long ranges in methods that involve sampling, scaling axes, and semantic representation. This section describes the issues encountered in general and how they apply to the two tools under discussion.

3.1.1 Overall Graph Occlusion

Ideally, we want to fit all the information needed on the graph without overlap or disorder. Traditional information visualization methods suggest rearranging the data, allowing the user to tilt, zoom, scale, and pan it to achieve the most meaningful display.

High value quantities in histograms used for visualizing port statistics would either block or skew the relative scale, making it difficult to interpret results. One method of scaling data quantities involves using logarithms, but these tools use a cube root function because it scales better with the range of values, including zero. The cube root function

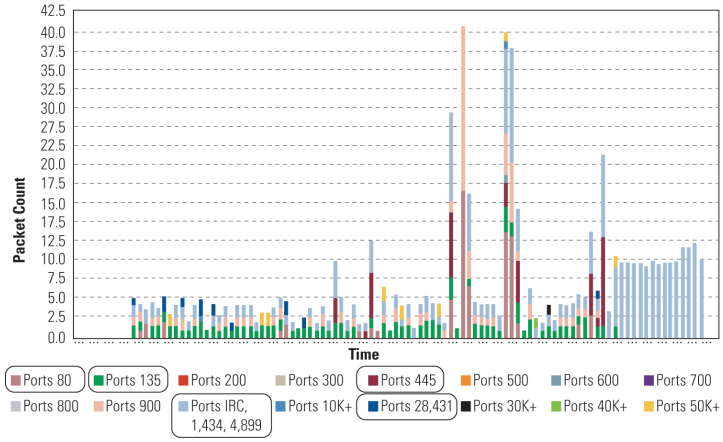

Figure 1: Scaling packet count using cube root for botnet traffic capture[1]

also allows mapping values between zero and one (see Figure 1)[1].

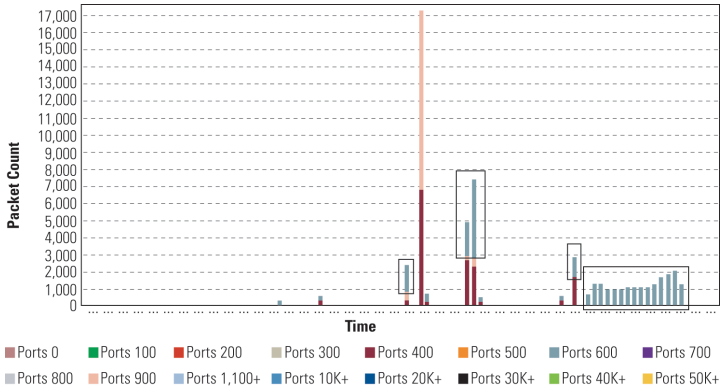

Figure 2: A system compromise by a botnet followed by the victim joining the botnet and transmitting traffic. Packet count every 15 minutes.[1]

One of the weaknesses of using the stacked histogram is that it may be difficult to evaluate data variations at the bottom of each column. The blue and green variables in Figure 3 demonstrate this problem: it is hard to discern the green and red variables. One solution is to use a popup window that displays measurement values and other details when the user moves the mouse over a portion of the graph.

For viewing alerts, IDS Rainstorm uses mouse-over popup displays using gloss boxes

[1]Graphic reprint permission granted by the Information Assurance Technology Analysis Center's (IATAC) IAnewsletter, Volume 9 Number 1, Summer 2006

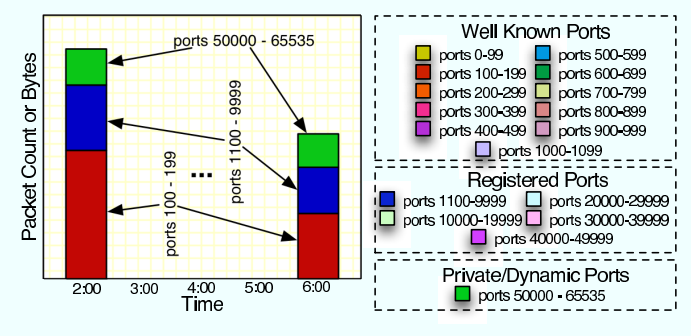

Figure 3: Concept diagram of plotting technique. Port ranges are categorized and plotted on the graph

to give detailed information for each alarm instance. The tool shows high-level information like alarm severity through color, time, and pixel location,which help to show more information in a limited area with details on demand using the glossing technique. Collision issues with this tool occur when alarms are close in time or IP space (one example in Figure 4. Two methods to deal with this are to show the alarm information with the highest priority first, and giving the user the option to click on a cluster of alarms to view information for other alarms clustered in this area.

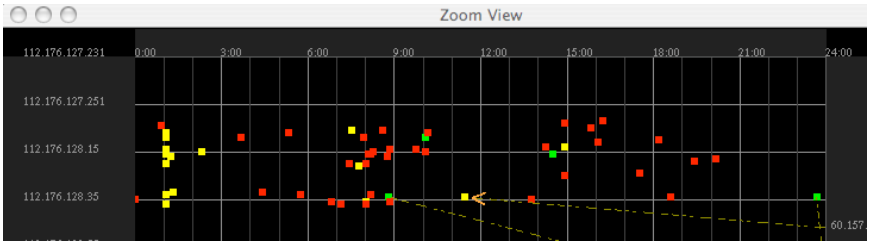

Figure 4: Collision of alarm alerts occuring in close time and IP space proximity

3.1.2 IP Address and Port Number Scaling

Scaling is necessary to deal with 65,536 possible port numbers and over 4 billion possible IP addresses (for the external network). Without scaling it would be impossible to display data in a meaningful way or avoid the loss of important details.

Common ports are based on what services a network has. For visualizing port statistics,

we divided the rest of the ports into ranges to provide less occlusion. A user can see the details for a port range by selecting the range on the graph. In our examples, using the Honeynet capture, we use the "Top Ten" [5] most probed ports as the known common ports. For both Windows and Unix systems, the well-known and commonly assigned port range is 0-1023 [2]. We want to focus more on these ports because we believe that a majority of vulnerabilities and services occur in this range and most attacks start within this range to gain access to the system. After separating the common ports, we place the remaining "well-known" ports into bins of 100 (see Figure 3) so that they do not cause an over-abundance of port segments that would be difficult to read on the graph. We chose 100 as providing the best view of the relatively high activity in this range (in our plot, 0-1099). The registered port range is 1024 through 49151 and can be used by any application; these ports are also temporarily assigned for a connection attempt to a server. Traffic can occur here, but not as much as through the well-known ports. We divide this port range into groups of 10,000 (in our plot, 1000-49,999). The private or dynamic ports are 49152 through 65535. Typically, no service should be assigned these ports. We include these ports in one range, since they are frequently used by Trojans and other malicious applications, but do not need to be displayed in as much detail as the lower port ranges [2]. A user can zoom in on a selected range to see a more detailed data display. Plotting the port range is a far more tractable problem when compared to visualizing the entire 32 bit external IP address range.

For alert visualization, we need to focus on where alarms occur on a network being monitored. Georgia Tech's network consists of 2.5 Class B IP addresses which is fewer than the total number of possible IP addresses, but still is a significant amount to scale for a visual representation. To deal with displaying these IP addresses without occlusion, we use multiple axes to extend the total axis length. This is described in more detail in section 4.2.

3.1.3 Time

Scaling time allows us to see different types of network activity at an appropriate level of granularity. For forensic analysis or archives, the user can adjust the time sample to display differing trends. For example, the time scale needed to display slow scans occurring over long periods of time (perhaps years) would not be useful for displaying activity related to viruses and worms quickly infiltrating a network. In real-time traffic analysis, the time window should be long enough to capture a representative sample of packets, but not so long that fluctuations in port activity are hard to recognize.

Honeynet traffic generates good illustrations for our visualization since all of the traffic is "abnormal" and because a specific attack has been collected in the captures we have used. Of the techniques mentioned, time scaling needs more experimentation than the others to be used effectively in the analysis of with regular network traffic. We want just the right granularity: if it is too small, we can mistake variances in normal traffic to be anomalous, and if it is too large, we will simply not see the variances that are anomalous. The other scaling techniques are applicable to both situations. For example, the Sasser capture did have other traffic in the background (representative of "normal" background traffic), and through IP, port, and quantity scaling we are able to see both that background traffic and the penetration and subsequent activity of the Sasser worm. With more development and testing, we would like to experiment with regular network traffic.

Most alarm logs are archived into separate days to display an initial view of 24 hours in a main screen, with the option of enabling a zoom view that allows zooming in on a specific time range. In the case of events that occur close together in time and IP space, the zoom view allows one to see the relevant data appropriately. For an active 24-hour day, the main view will not be able to show the entire range of alarm information. Without zooming, the user could miss important events or sequences of events. For days with average traffic, using 24 hours as the time range generally causes little occlusion. Testing the alarm visualization updating in real time would require a shorter time span, but we need

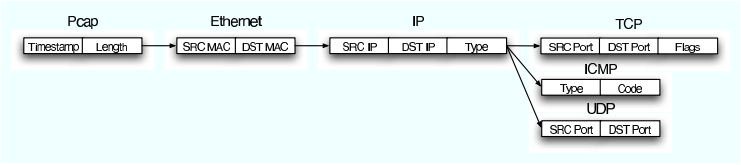

Figure 5: Packet header fields parsed for statistics gathering

to do more use testing to finalize our conclusions.

3.2 Data Parameters

This section describes the parameters used for port and alarm visualization and a discussion of what guides the choice of parameters and their display.

3.2.1 Packet Capture and Header Fields

Tcpdump and pcap [3] (and the functionally equivalent winpcap for Windows) are general-purpose packet capture libraries. "Pcap" collects any packet that it notices on the observed network segment, and adds a header to it. The pcap header includes the time the packet was received and the total size.

We carefully selected fields to maximize effectiveness and minimize processing overhead (Figure 5). Timestamps are important for calculating the data statistics over time. Port numbers allow us to track the activity on the ports and help to tie attacks to vulnerable services. Packet size information and Internet Protocol (IP) addresses give detailed flow information for the network.

3.2.2 Alarm Generation Using the StealthWatch IDS

The StealthWatch [47] anomaly-based IDS system is one of the security appliances defending the Georgia Tech campus. It monitors flow activity and bandwidth usage to detect anomalous behavior. StealthWatch does not need to know an attack's signature in advance to detect an attack. For our analysis, we were provided with StealthWatch IDS alarm logs generated from inbound and outbound Internet traffic on the perimeter of the Georgia Tech

network. While we used StealthWatch logs, it is important to note that IDS Rainstorm can be used for other IDS system alarm logs as well. An average of 7000 alarms is generated in one day from StealthWatch. We have access to the following parameters.

3.2.2.1 Alarm Parameters

The StealthWatch IDS contains the following alarm parameters that we use in our visualization tool:

- **Alarm type:** There are 33 alarm types, each identified by an integer value. System administrators usually judge the severity of an alarm and classify the alarms themselves if their definition differs from that of the IDS company's default definitions. Alarm types can be categorized by severity, and in the tool color is used to represent severity.

- **Time:** When each alarm is generated, the system records a unix timestamp with a resolution of one second starting from the start of the unix epoch. This information helps to determine the alarm's temporal position among the rest of the alarms and can help to display significant patterns or positions in a sequence of events.

- **IP Addresses:** The system records the internal LAN IP address that signaled the alarm and, if an external IP is associated with the alarm, the system records the external IP whether it is the victim or caused the alarm. Since the traffic we are monitoring is between the Internet and Georgia Tech, there will be no alarms caused by internal IP to internal IP traffic. The IP addresses are given as strings in dotted decimal notation.

Some types of alarm records also include port number, threshold value, concern index, and zone. We are currently working on a way to integrate this visually instead of just showing text in a detailed view.

Commonly occurring StealthWatch alarm definitions include the following:

- 61 *Host Max Flows*: Indicates that the host has had total active flows above some threshold in the last five minutes. This could be DoS, DDoS activity against the host, or the host sharing files on many connections. When activated, this alert also returns the number of new flows in the last five minutes.

- 66 *Watch Port Active*: Indicates that a port on the port watch list has become active. This alarm shows the external IP and the internal client IP. It also provides the protocol and service used.

- 68 *High Concern Index*: Indicates that the suspect IP has exceeded a limit for the accumulation of points based on flow anomalies called a Concern Index. Scanning activity and other nonconforming behavior generally cause this alarm.

- 77 *File Sharing*: Indicates that the suspect IP is transferring a large number of files.

- 79 *Touched*: Indicates that a high Concern Index host has exchanged data with an inside host. This may indicate a compromised host. The alarm details provide the protocol and service that triggered this alarm.

- 82 *Long Duration Flow*: Indicates an IP communication between an inside and outside host that exceeds the configured seconds required to qualify a flow as long duration. This alarm detects suspicious channels of communication such as spyware, remote desktop technologies, VPNs, IRC botnets, and other covert means of communication. It can also be triggered by legitimate use such as messenger programs, streaming media, and web-based email.

- 93 *Watch Host Active*: Indicates an IP communication between an inside and outside host that exceeds the configured seconds required to qualify a flow as long duration. This alarm detects suspicious channels of communication such as spyware, remote desktop technologies, VPNs, IRC botnets, and other covert means of communication.

It can also be triggered by legitimate use such as messenger programs, streaming media, and web-based email.

CHAPTER IV

VISUALIZATION SYSTEMS

This chapter will describe how the two visualization systems were designed and implemented. Additionally, the operation and features are described.

4.1 Visualizing Port Activity

4.1.1 Design Goals

Our goal is to get an overall view of what is happening on the network. In the case of plotting port activity, we want to separate and arrange it to see important details without occlusion. More specifically we wish to provide context first, followed by the user specified detail in a drill down fashion. Seeing data over time will help to notice any patterns or trends and is also helpful to see activity over a longer period of time. Our scheme shows aggregate port quantity, instead of the more common flow count per IP. Some design goals are as follows:

- We want this system to be useful for both forensic and real-time analysis.
- Making the tool lightweight and not CPU intensive is another goal. Only header information is stored, not the payload, dramatically reducing archival storage requirements.
- Reducing the time it takes to learn how to use and understand the tool is another motivation. This helps to reduce the amount of time for understanding the visual results and speeds reaction time for the person observing the visualization.

Using scaling techniques will reduce occlusion, and still show the overall view of the network. In addition to IDS, the human eye can catch a pattern of a possible attack which

will help to mitigate an attack or stop one. This tool does not need to run in conjunction with an IDS, since it only needs to parse raw network data. It can complement the function of the IDS by allowing human cognition and observation to assist in identifying attacks and making subsequent decisions.

We used packet capture data from the Honeynet [32] at Georgia Tech for our visualization examples. The Honeynet at GT is directly connected to the Internet and has no production software therefore any incoming traffic is suspicious. The machines have different operating systems hence they can be targeted by a wide variety of attacks.

4.1.2 Methodology

4.1.2.1 Histogram Plots

Histograms are easy to interpret and are good for visualizing a large data set because the data fits easily onto the plot due to stacking, the intra-bar, relative sizing is insightful, and comparisons with other bars on the chart are easily made based on their relative height [27]. Using multidimensional data to plot a relationship between two parameters over time can be visualized by using either a 3D chart or a 2D histogram. The disadvantage of a 3D view is in comparing patterns, widths and heights that are at various distances from the user, i.e., on the Z-axis, which can distort the perception [42]. Avoiding occlusion issues in 3D is difficult, and often requires user navigation. In a 2D stacked chart, it could be more difficult to focus on activity for a single port. To test the effectiveness of these techniques, we employed 2D visualizations against captures of real attacks.

4.1.2.2 Axis Parameters

The packet header fields can be categorized as ordinal (source and destination ports and IP addresses) and interval (packet count and total bytes). The horizontal axis is time and the vertical axis is used to display interval quantities. We chose to create stacked histograms mapping the horizontal axis to time and the vertical axis to interval values of packet count and total bytes. See Figure 3 for an example.

4.2 Visualizing IDS Alerts

4.2.1 Design Goals

Our system, IDS Rainstorm, provides a main view that presents an overall representation of all of Georgia Tech and a zoom view that provides more information on a user-selected range of IP addresses. Our goal was to design an overall view that conveys enough information so that an administrator can see network activity that needs immediate attention. Once alerted to patterns of suspicious network activity, administrators can retrieve specific details of particular alarms using the zoom view. Initially the tool was written using perl/tk and then ported to Java/OpenGL (JOGL). Screenshots of the tool will be a combination of both versions.

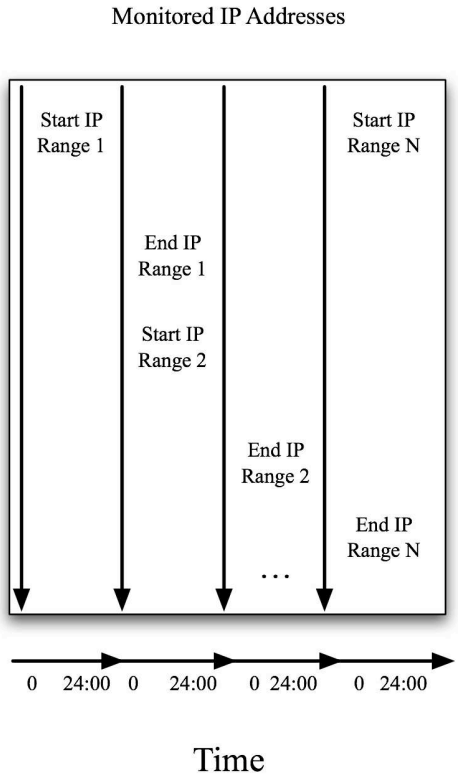

Figure 6: Design of the Basic Visualization and Representation. Each vertical axis represents IP addresses in sequential order. Each horizontal axis associated with the vertical axes represent one 24 hour period.

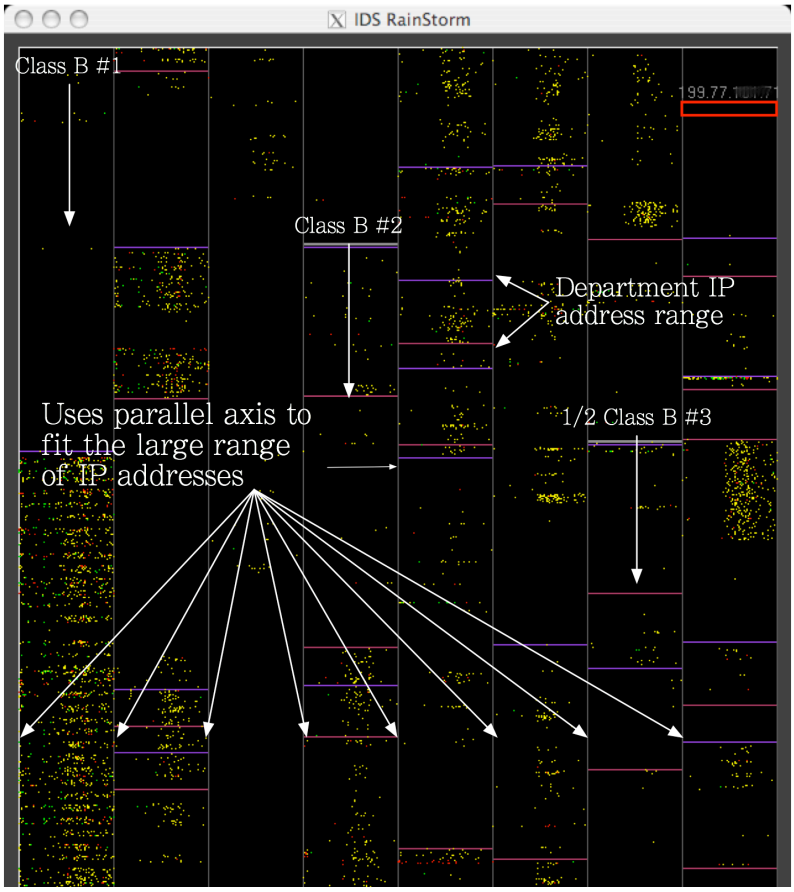

Figure 7: IDS RainStorm main view: The 8 vertical axes are shown that represent the 2.5 Class B IP addresses. The thicker horizontal lines between these axes show where each Class B starts. The other horizontal lines show the start and end of each department. Those addresses not in a department are either unallocated or reserved for special use by OIT and other departments. This screenshot shows an entire day's worth of real alarms generated.

4.2.2 Main View

Each of the views follows a general visualization technique developed to address this problem as shown in Figure 6. The visualization uses a set of rectangular regions that represent (top-to-bottom) the set of contiguous IP addresses, where 20 addresses are allocated to a row of pixels. Each column's horizontal width represents 24 hours of network activity. Individual colored dots in a row (IP addresses) represent total alarms for those 20 addresses at a particular point in time (horizontal position). The alarm with the most severity out of the 20 addresses will appear. In addition, the user has the option of configuring StealthWatch to correlate a series of low-priority events into a single higher priority alarm to reduce visual

clutter [14].

Color represents alarm severity where red is high concern, yellow is medium concern, and green is low concern. The IDS has default concern levels set, but a user can also modify these. Currently, the default colors are what are shown in the tool.

The parameter with the largest range values is the 2.5 Class B IP addresses. Since a way is needed to show an overview of all of them without cluttering the view, we applied a method used in the Tarantula tool [19] and the SeeSoft tool [20] for representing large source code files. Each represents a source line as a line of pixels, and then simply wraps around to the next column to continue the sequence of source lines.

Figure 7 illustrates the concept of using multiple Y-axes to present a larger range of points in addition to the idea of color-coding the severity of the alarm. Scaling time is not as much of a concern since its range is not as large. We use 24 hours for the range shown in detail in Figure 6. Both IP and time are aggregated onto their respective axis. Each pixel on the x axis represents 20 minutes, and each IP on the multiple y-axis represent approximately 20 IP addresses.

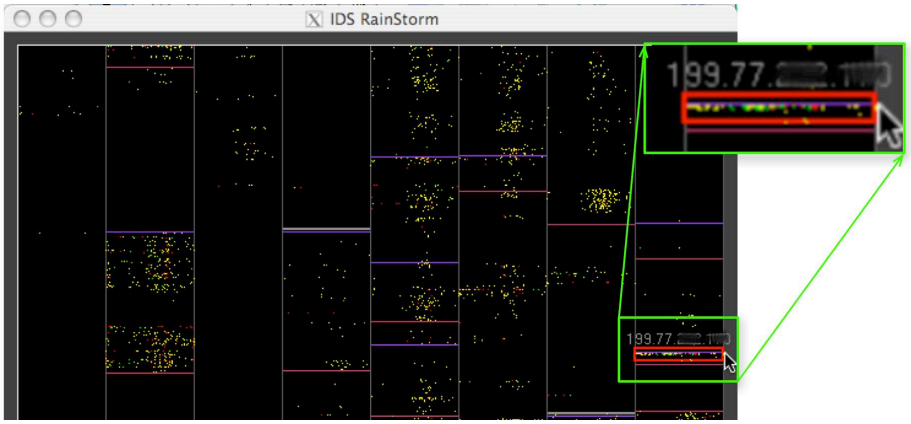

Figure 8: User selecting area to zoom in. (Here we have artificially highlighted a region in green and magnified it to assist the reader.) The IP address shown is the one at the top position in the red box region (the last two octets are intentionally blurred for privacy). The result of the zoom is shown in Figure 9

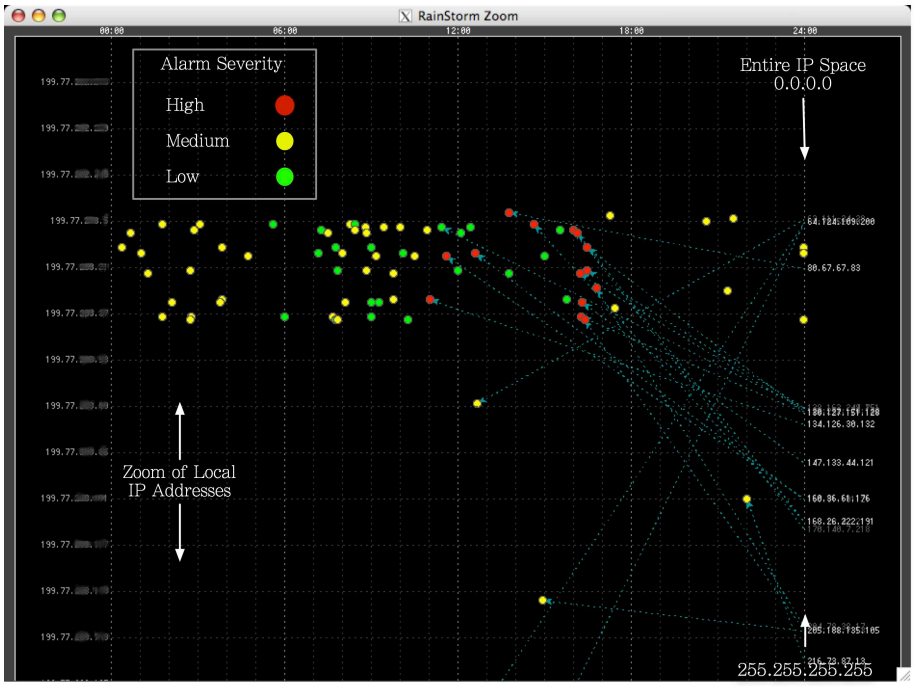

Figure 9: Zoom of a cluster of alarms seen in the overview. Also shown is the alarm severity legend and internal and external IP axes layout. The selected subset of internal IP addresses are represented on the left vertical axis, and external IP addresses on the right vertical axis.

4.2.3 Zoom View

As a user moves the mouse across the overview, a red box highlights the current cursor position as illustrated in Figure 8. This red box is an IP range selector. The IP address representing that top position is printed at the top of the box. When a user clicks on the overview, a secondary screen appears in a separate window with an enlarged view of the portion enclosed by the red box. The IP range contained in the red box are now printed in this view on the left. Labels are on the top horizontal axis to represent time within 24 hours. Alarms are seen as larger glyphs as seen in Figure 9. The zoom view also provides other information such as extra alarm detail for each alarm and any external IP address connections. Arrow are drawn to show connections between the external IP address and local IP address shown in Figure 9. The arrow originates from the attacker IP and points to the victim. One can zoom in and out of this zoom view by using the mouse wheel. A

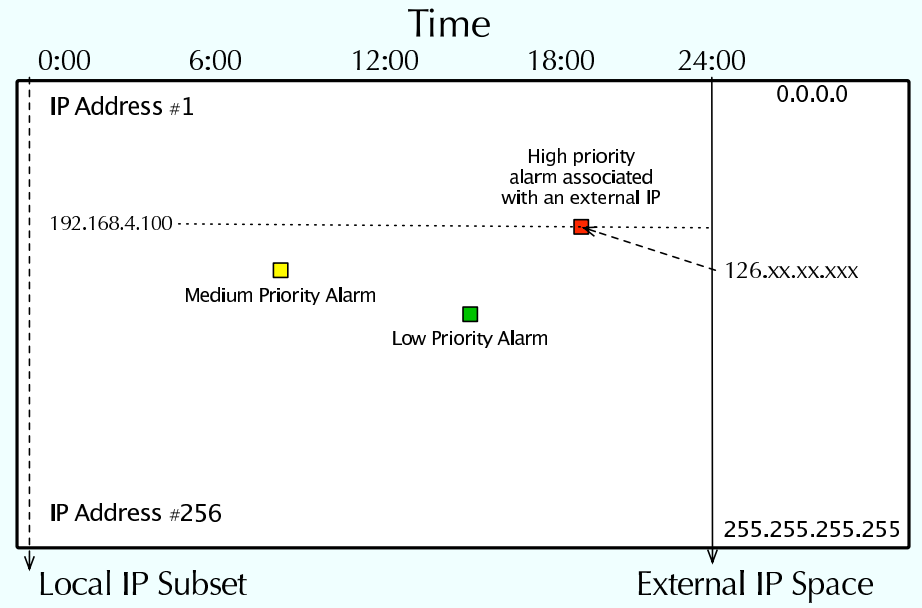

Figure 10: Concept diagram of the zoom view in IDS RainStorm.

concept diagram of the zoom layout is shown in Figure 10. The times labels in these zoom views are shown horizontally along the top of the graph. Zooming is helpful in reducing overlap when more than one alarm occurs for an IP address at the same time, and for addresses that are close together in position.

4.2.4 Other techniques

4.2.4.1 Glossing

Glossing happens when a user moves the mouse cursor over an icon or particular text, and expanded information is presented. The gloss disappears when the cursor is moved away. In the zoom view, when a user mouses over a particular alarm glyph, a pop-up gloss is shown that gives the alarm type, time, source and destination IP addresses. Also mousing over an external IP creates a gloss, highlights that address and highlights the line connecting the external IP address to the alarm glyph mapped on the graph. This is useful when multiple external IP addresses overlap in the same area on the left axis. Examples of these methods are shown in Figure 11.

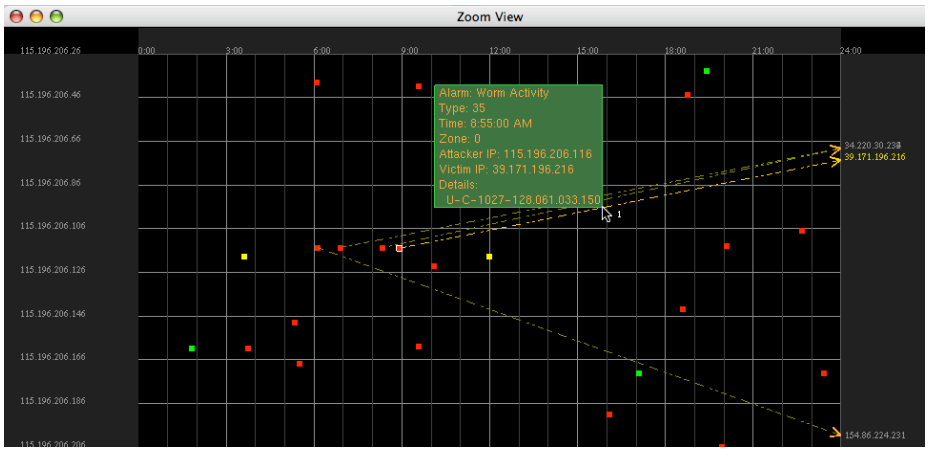

Figure 11: Gloss of alarm with corresponding line/arrow and external IP highlighted.

4.2.4.2 Indexing

When cluster of alarms are in the zoom view due to alarms occuring in close proximity, one can index through those alarms using right and left button mouse clicks. Figure 13 shows two zoomed screenshots indexing through 2 alarms that overlapped each other.

4.2.4.3 Filtering

IDS Rainstorm also includes simple filtering capabilities. In both the overview and zoom views, the user may filter on alarm severity, choosing to show only the high critical alarms (red), medium concern alarms (yellow), or the low concern alarms (green). This capability can help the user to focus on particular alarms for further analysis and to sort through multiple alarms that appear at the same time for a given set of IP addresses.

4.2.4.4 Panning

Panning results show in the zoom window and can be performed by moving up and down the IP column using the up and down arrows or clicking and dragging in the main view. The red box in the main view will move corresponding to what is seen in the zoom view. This is illustrated in Figure 14. As the user zooms in more, panning left and right can be done by using the right and left arrow keys. An example is shown in Figure 12.

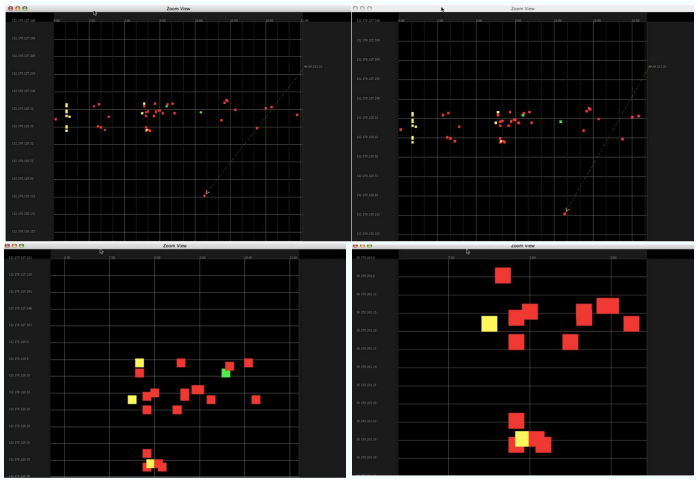

Figure 12: Screenshots of zooming and panning use to focus on a particular portion of a cluster of alarms.

Figure 13: Indexing through alarms with mouse button clicks.

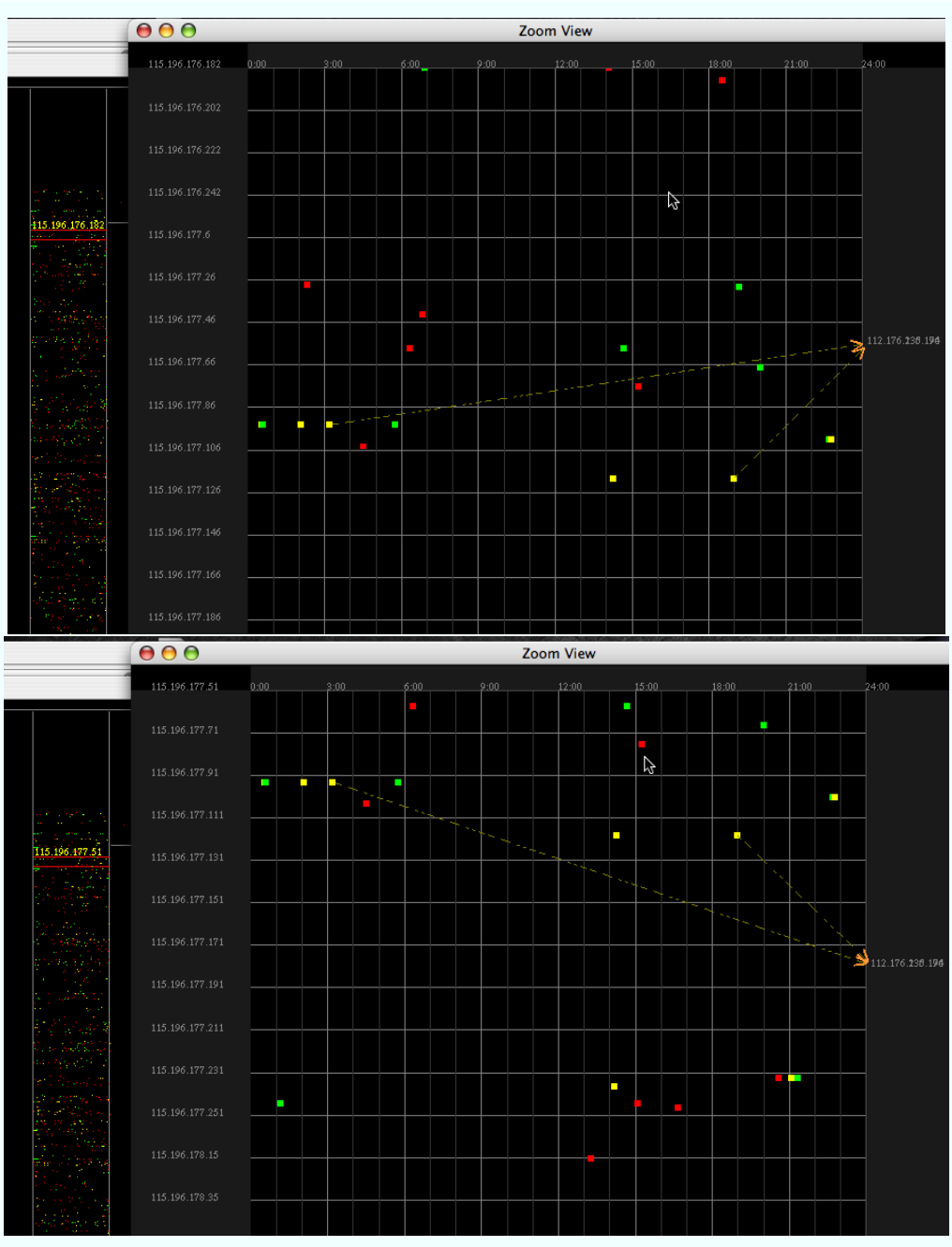

Figure 14: Two screenshots showing panning movement in the zoom view with the coressponding red box movement.

CHAPTER V

THREAT MODELS AND EXAMPLES

The following classes of attacks are the most common types that occur. To help demonstrate the effectiveness of the port visualization, packet captures of these attacks are visualized. Screenshots of days in which significant security events occured on campus are shown and explained to illustrate how one can use the IDS visualization to make discoveries.

5.1 Port Statistics Visualization

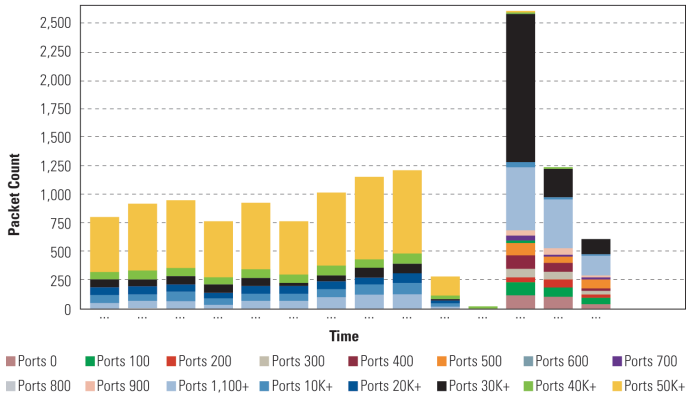

Figure 15: Network scan from honeynet, packet count every 2 minutes. Pattern or ports probed over time is seen for this scan.[1]

5.1.1 Network Scanning and Mapping

Scanning a network is commonly a precursor to an attack. A blueprint of the network can be made, finding active ports and IP addresses on the victim host. This is performed by sending probe packets to groups of desired IP addresses. If a response is received, then we know the respective ports and hosts are active. While less aggressive (or more subtle) naive scans are relatively easy to detect, scans on common ports are difficult to detect in

the midst of legitimate network traffic [41]. Typically, the packets are small, and few in number. Such scans are difficult to detect, particularly very slow scans. If non-existent hosts and unused ports are probed it is easier to detect because we expect little or no traffic to those destinations. Visually we will see values plotted for port ranges that have not shown up before as in Figure 15.

5.1.2 Viruses, Worms and Trojans

There are two primary motivations for those who create worms. One is for Distributed Denial of Service (DDoS) and the other for installing Simple Mail Transfer Protocol (SMTP) servers that receive connections on high ports and relay mail (outbound connection) to port 25 of the destination server, normally used for Spam. Since traditional methods of propagating Spam are being denied, worms are being used [35]. This allows a malicious sender to generate Spam from all attackers' compromised systems [30]. Infecting a system usually occurs by exploiting a vulnerability on a system. Once a system is infected, then it will scan for other machines and infect those in a similar way. A burst of activity will be seen on the vulnerable port. If penetration is successful we typically see odd behavior on a backdoor port which is used to scan for other hosts to infect, see Figure 16.

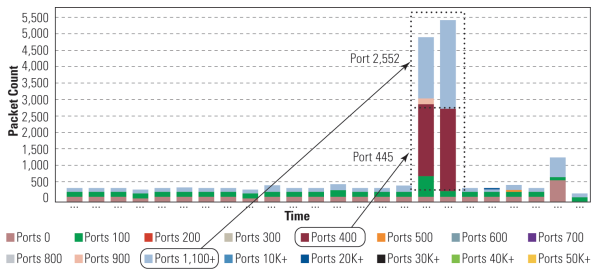

Figure 16: Success of the Sasser worm and start of the worm traffic - count for every 30 minutes[1]

5.1.3 Backdoors and Rootkits

A backdoor is installed after the successful takeover of a system to help maintain control. It opens a port on the system to send and receive traffic, thereby maintaining a hidden entry

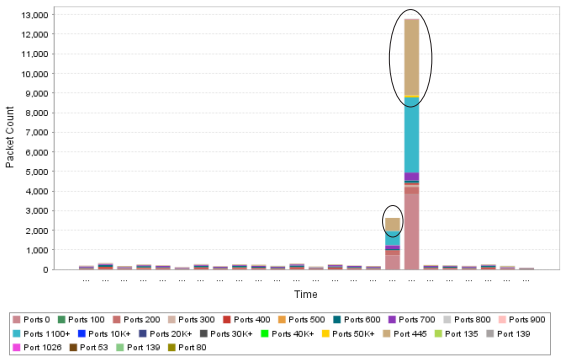

Figure 17: Sasser graph with sorted out ports: 445, 135, 139, 1026, 53 and 80

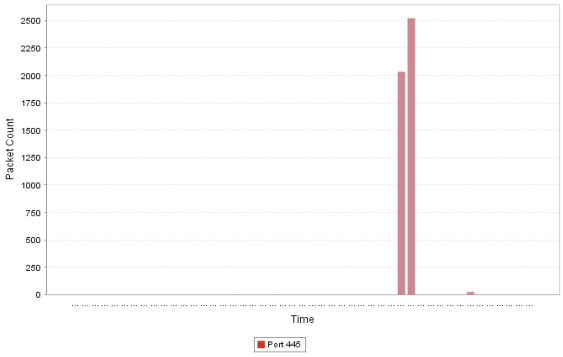

Figure 18: Sasser graph of the p400 range filtered for focus. In this case traffic is seen to be from port 445

point. Unusual activity on a normally active port or non-active port is readily apparent. Packet count and total size increase in a visualization if a significant amount of data is going in and out of the backdoor. Similar to backdoors, rootkits go one step further and replace existing application binaries instead of running as a new application like a backdoor does, e.g. a modified telnet program can be a rootkit. Another covert method to access a port is with IRC bots, which have been used for DDoS attacks. The binary for the bot is small and can easily be installed on someone's system without their knowledge. Commands can then be passed to the bot. Typically the bot is told to attack a given IP address. The servers and channels the bots are placed on can be difficult to find when the channel is set to hidden. Detecting unknown bot activity requires a check on the default IRC port 6667 or other chose ports [1]. In this case, there would be a rise of packets on the IRC port or other chosen ports used to transmit the bot traffic (see Figure 2).

5.1.4 Summary

This prototype helps to show the pattern and quantity of traffic on ports and systems where unknown attacks can be identified without a signature. Because of the general characteristics of how common attacks behave, this is a useful scheme. Observing port behavior is both a good indicator for a precursor of an attack or an actual attack. Our approach did not help to detect malicious behavior during the actual compromise on the local system (ie., when someone gained root access). Also, the prototype does not show network topology, layout and link information. Flow information is also not considered, just individual packet header data. Subsequently we get the port information instantaneously, and do not wait for a flow to finish, especially that of long flows.

The following graphs are of Sasser worm [32] and botnet [6] activity from the Honeynet. These results are used to illustrate the efficacy of the system, such as scaling, and overview/detail.

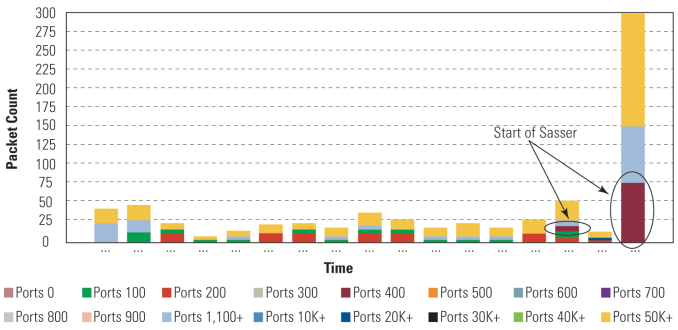

Figure 19: Same packet count of Sasser using a smaller time scale of 5 minutes.[1]

The first example considers the Sasser worm. Figure 16 shows normal probes and chatter that usually occur. The spikes indicate an increase of packet count for two port ranges. In the region labeled port 445, the port 400 range in red show the start of Sasser activity which had at that point successfully broke into the system. The tcpdump log showed that all of this was from port 445. In the region labeled port 2552, which represents ports 1,100-10,000, was a response port to port 445. Again, checking the log indicated that this was port 2552. This example shows that a user can be guided to notice activity in these port ranges, in the midst of usual traffic. Without having to look at the log, the user can focus on the port 400 range by filtering the graph to show only the port counts for 400 to 500. This is shown in figure 18, where we can see that this is port 445.

Ports that are commonly used on a network can be sorted out from the port ranges and plotted on its own. Doing this helps to both focus on the usual active ports for any abnormality and reduce the count from the port ranges. When the count is reduced, we can then focus on the rest of the ports which we do not usually expect traffic without being occluded by the high count of the active ports. We can see the same traffic capture of Sasser in figure 17 where ports from the top ten probed ports list are sorted out from the rest. At that time ports 445, 135, 139, 1026, 53 and 80 were getting probed on the Honeynet, which is why we choose to single these out for this illustration. The circled region shows where port 445 is plotted now. Compared to Figure 16, we can focus on port counts in the ranges minus the value of port 445 count obscuring the counts.

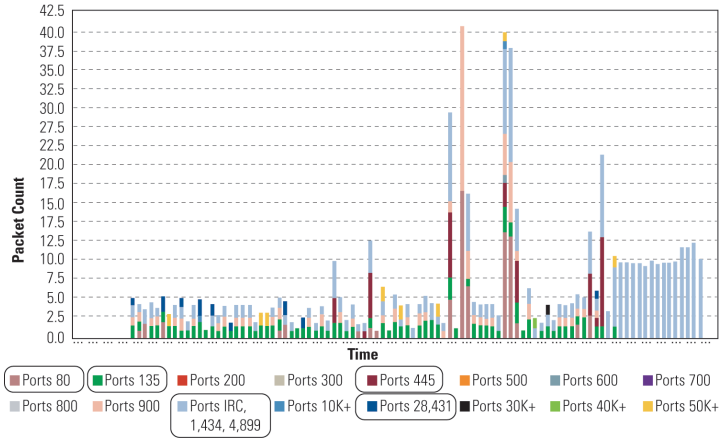

Figure 20: Scaling packet count using cube root for botnet traffic capture.[1]

In circled region of figure 19, we can see the start of p400 (port 445), to appear jagged, compared to the original graph count of every 30 minutes. A smaller time sample helps to notice behavior sooner, which is important for a regular network that needs to react quickly, but at the cost of a smaller picture. One example is the network scan that occurred in 30 minutes (figure 15) where we needed a smaller time scale to quickly notice this activity.

We can see the ports with smaller counts more clearly and its pattern over time using a cube root. In the botnet graph, because of the high count of bot packets skewing the scale (around 17,000), all of the other values are barely visible. In figure 20, we can see the other port activity now. Those were the vulnerabilities the bots were trying to exploit on the Honeynet system, until they were successful and installed the bot.

This has been kept as a prototype because viewing port information is not enough to detect all types of attacks. This can be used as part of a visual tool and the methods used to scale and view the data could be used scaling larger amounts of data with large scales of parameters.

5.2 IDS Alert Visualization

5.2.1 High Alarm Count

The first example deals with a cluster of alarms in one area of the graph. One case is illustrated in Figure 8, which shows a cluster of alarms over a full day selected and enclosed by the red box. This stands out compared to the rest of the graph, and the concentration is high for this range of IP addresses. When the user zooms in on this region, the resulting view appears as shown in Figure 9. The alarms can be seen more clearly but there is still some occlusion which has occurred because many alarms have been triggered for IP addresses located closer together in sequence. In the tool, two methods can be used to help with this problem. One is to zoom again, as is shown in Figure 21. Note that this has the same layout as in the original zoom view, but now we see alarms for 12 hours rather than the entire 24 hour period. This spreads the alarm glyph representations over a wider axis which reduces glyph occlusion. A second method for fighting occlusion is to filter by alarm level (color). In this example, the range of IP addresses are actually Akamai (content delivery) servers for GT's website content. Active servers generally trigger many alarms, unless the IDS is carefully tuned. Re-calibrating for hosts which trigger a high alarm count could hide alarm counts that occur on less active hosts; therefore the thresholds have not been modified. Though these alarms are generated by the IDS, human analysis and the visualization help to rule out these alarms as a serious problem because they have occurred on the logical IP space reserved for Akamai.

5.2.2 Exteral IP Connection Patterns

Another useful activity is to pan through the graph by clicking the mouse and dragging the IP range selector, or red box, through the overview. The resulting motion is shown in the zoom view. An example overview is shown in Figure 22 where the user clicks and drags on an area, and the corresponding zoom views are shown in Figure 23 to illustrate this technique. Time is constant while the internal IP addresses on the left vertical axis change

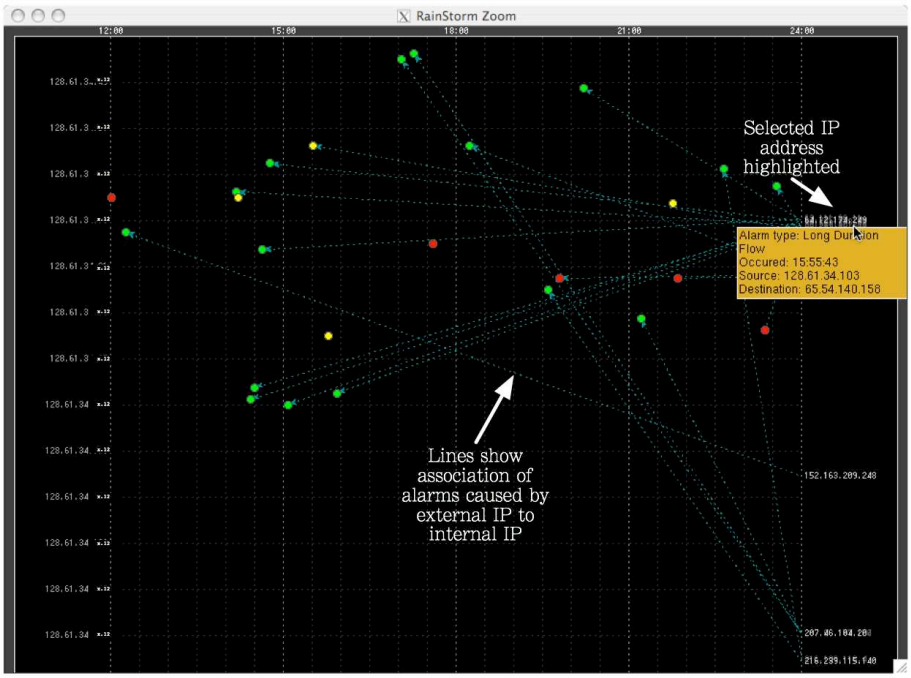

Figure 21: A zoom view on time. This zoom is a double zoom view of Figure 9. Internal IP addresses are on the left and external IP addresses are on the right.

sequentially. The external IP addresses on the right axis maintain the same 2^{32} bit mapping but as the user scrolls in the main view, the external IP addresses appear (and disappear) based on alarm activity associated with the changing/moving internal IP addresses. This activity allows traversal through the range of IP addresses to find detailed patterns. The external IP addresses remain constant through the panning, and this helps to find if there is some address or range of addresses trying to attack the network.

This technique in the tool is useful for when anomalous behavior could be targeting internal IPs that are spread across the logical space, like botnet and worm activity. The day of alarms shown in Figure 22 had a cluster of activity between 12:00 - 18:00 that happened consistently in certain portions of the IP address space. Most of these were *Long Duration Flow* alarms where a flow lasts longer than a set specified time amount. If the external IP is from an ISP (an individual user) then the long duration activity is suspicious. AOL, Hotmail, and Podcasting are examples of applications that can set off long duration alarms since the connection usually stays open until the user closes it. A user needs to be familiar

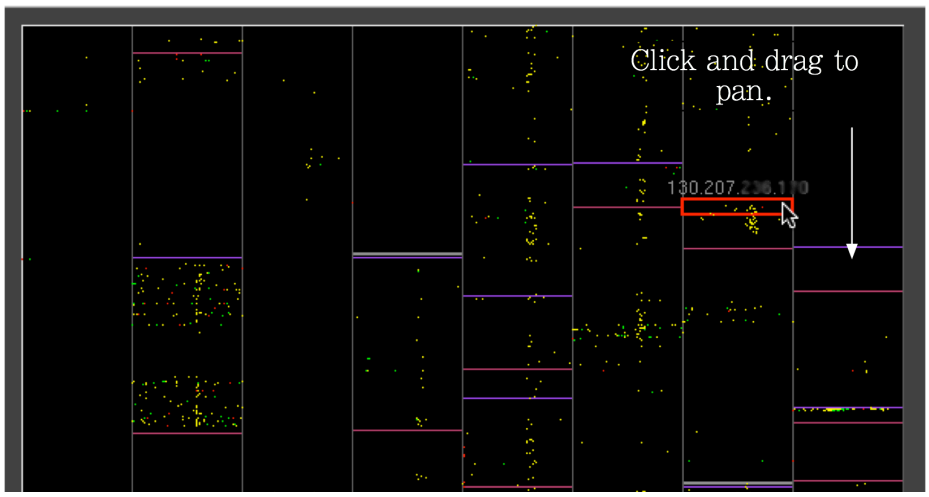

Figure 22: June 22nd overview. Clicking and dragging on the overview appears in the zoom view (shown in Figure 23) and animates the traversal down the IP space. The external IP axis is held constant.

with the local IP, such as what they can run, and if they are authorized to be a server or not. This coupled with the techniques demonstrated in this example will make determining patterns in which external IPs are connecting to the local network and whether an alarm is high concern, easier.

5.2.3 Dorm Activity

Figure 24 is an overview for alarms generated on April 26, 2005. Some patterns immediately noticed are that most alarms seem to occur in the last half of the day and for several of the IP ranges, similar patterns across them can be seen. One such range occurs on the left most column (region 1, Figure 24). These IP addresses are for the campus dorm residents. Upon taking a closer look, most of these alarms appear to be long duration flows. Most of the external IP addresses associated with these come from AOL instant messenger servers. Students are most likely starting their instant messenger programs later in the day after classes are over. If *Host Max Flows* alarms are seen here, then based upon our analyst's experience, that host is running a warez server or has a backdoor port. It is against campus policy to run a file sharing server, whether it is voluntary or not, hence that host's access to the Internet will be blocked and the student will be notified.

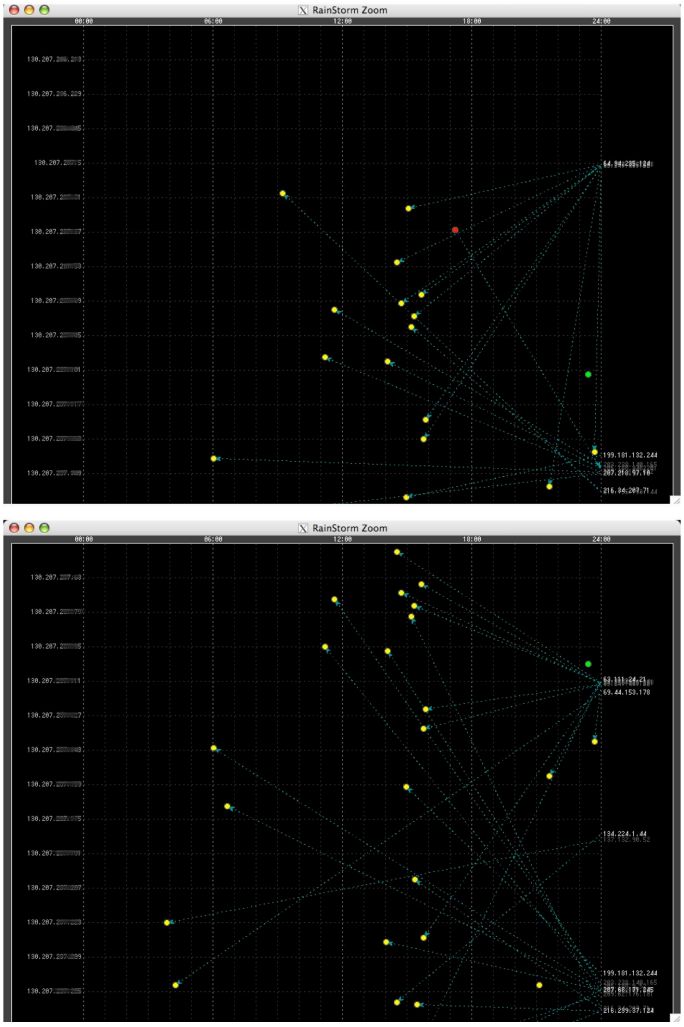

Figure 23: Panning results of Figure 22 shown as two transitions.

5.2.4 Worms, Viruses, Trojans

In this same figure, a cluster of red alarms at the bottom left can be seen in the midst of usual mid priority alarms in the dorm IP space (region 2 in Figure 24). Here one IP address was a source for *Watch Port Active* alarms from many external IPs as shown in Figure 25. A close-up of this activity showing the 3:00-6:00 pm time range can also be seen in Figure 25. A known exploitable port that had recently hit GT was added to the watch list and compromised on the host. The alarm pattern shows successful worm penetration as the host had consistent communication to various IP addresses.

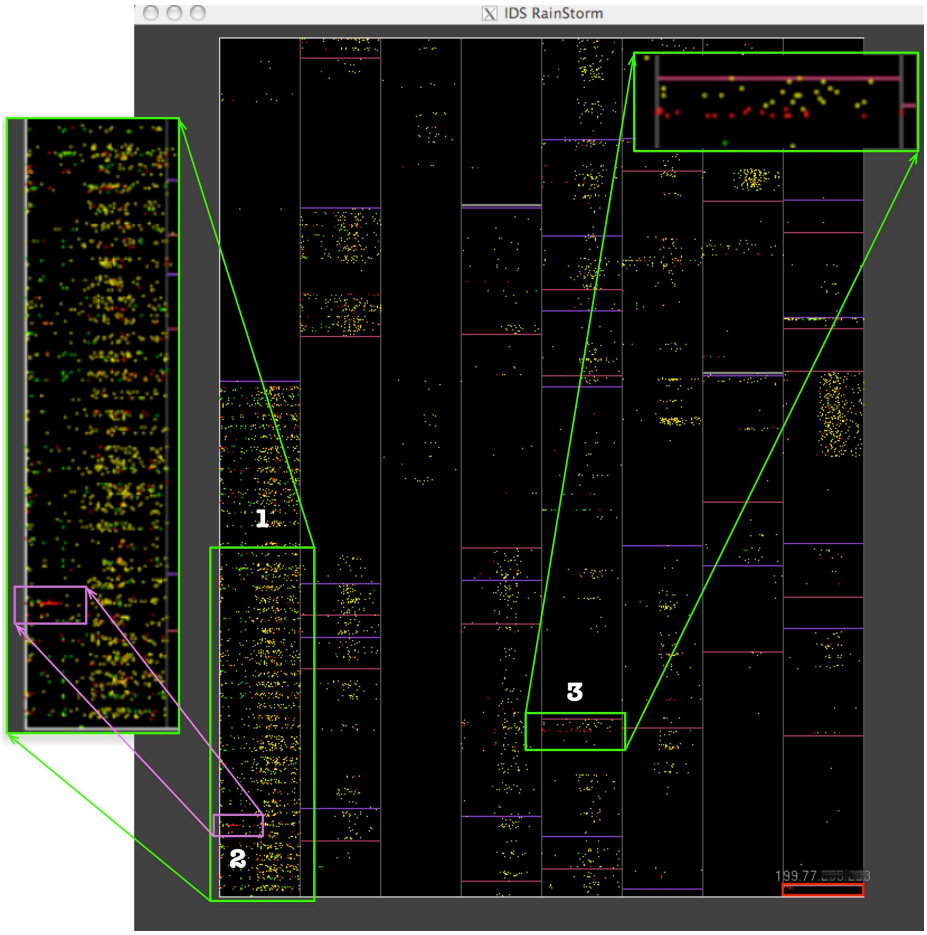

Figure 24: Overview of April 27th alarms. (Two regions are artificially identified in green and magnified for easier viewing.) Here region 1 shows activity in a subset of campus dorm IP addresses, a cluster of activity for a machine in the dorm is outlined in region 2 and region 3 shows a cluster of activity in occurring over a small range of IP addresses for the entire day.

5.2.5 Botnet

Also, on the same day, is another cluster of red alarms (region 3, Figure 24). These alarms are *Watch Host Active*. Some of these external hosts have made connections to other hosts on the local network previously and had bots installed on them. These bots were more active around midnight and in the figure we can see similar activity around midnight. The next day, for the same IPs, you can see almost the same time pattern of activity. The zooms for each consecutive day can be seen in Figure 26. We can conclude these IP addresses have become infected with a bot, that has a specific time pattern of activity. These examples

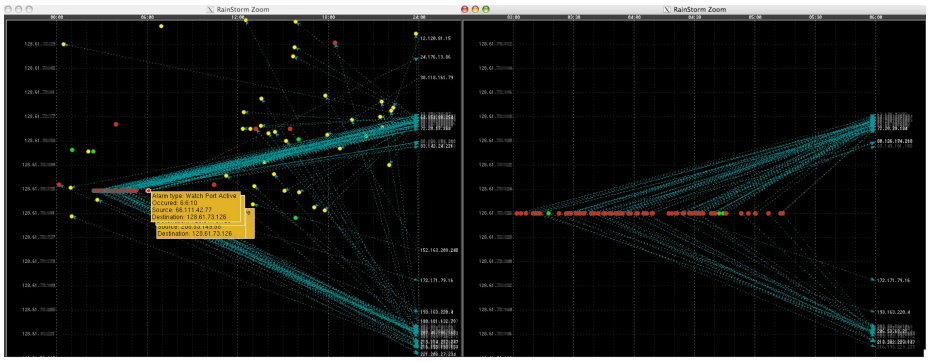

Figure 25: April 26th worm activity for a particular host located in the campus dorms. The left side is the zoom view of region 2 highlighted in Figure 24. The right side is a zoom of the left view that shows the activity from 3:00-6:00pm.

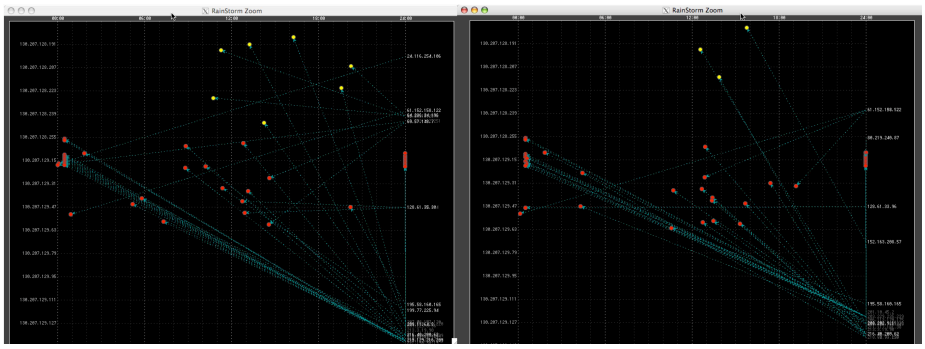

Figure 26: Bot activity shown for the same IP address space on April 26 (left) and April 27 (right). The activity time pattern for the two days is almost identical.

show how analysis is improved for casual occurring alarms (general dorm activity) and ones that were triggered due to anomalous behavior (botnet and worm case).

5.2.6 Summary

This tool, or visualization, is not designed to operate in isolation, but instead something to be used with other IDS tools. An IDS that checks for signatures can help with slow, stealthy activity, and an IDS that checks for anomalies can detect activity that deviates from a defined baseline. These methods are not enough, however, because network behavior is dynamic. The visualization enhances the view and adds another layer of analysis that allows us to notice activity that machines cannot. Other monitoring tools can optionally be

re-coded or re-calibrated according to insights gained from human observation. Nonetheless, the tool is only good as the data it receives; therefore, some problems can be difficult to find especially when false alarms are part of the data.

These examples show how analysis is improved for reoccurring alarms-due to general dorm activity-and for alarms that were triggered due to anomalous behavior (botnet and worm case). The visualization enhances the analysts' view of the logs and lets them more easily notice activity that machines cannot. Other monitoring tools can optionally be re-coded or recalibrated according to insights gained from human observation. Nonetheless, the tool is only as good as the data it receives; therefore, some problems can be difficult to find especially when false alarms are part of the data. The underlying IDS system generates this alarm data and, unfortunately, even today's best systems are prone to some degree of false alarms and are unable to detect all classes of anomalous activity. IDS RainStorm implements general techniques for displaying whatever alarms are generated by any IDS.

These visual images can give a system administrator a frame of reference of what a usual day looks like. If any day deviates from this image, then the system administrator may need to investigate further to find out if change is anomalous or not. Comparing a new view to a normal day's image is a much faster process than trying to do the same with text logs (the image of a day can be saved for later reference). This is significant for the amount of traffic that a large campus generates. This type of analysis also shows the advantage a human has over machine learning algorithms used to find anomalous activity. Situations and changes in the network can make changes to alarm patterns. Machine learning algorithms would have to be dynamic or constantly changed to accommodate for this.

CHAPTER VI

USER STUDY

This chapter describes the user study process that was used to design the tool and make subsequent additions. First, the network analysts' duties and the Georgia Tech network are described. Then how we used this information to design the tool is explained. Finally, the user evaluations of the tool and the results are given.

6.1 Background

We interviewed Office of Information Technology (OIT) system administrators at Georgia Tech, to learn about the stages in the alarm analysis process. Primarily, how alarms are monitored, the criteria that governs the next action, and the method in which alarms are analyzed. The requirements collected during these interviews guided the design of our visualization system.

OIT at Georgia Tech maintains the campus-wide network of computing resources. The organization also provides and maintains the Internet links coming in and out of campus, and is in charge of protecting the campus data. The security branch of OIT monitors the network and provides technical and educational support to the campus population.

In addition to OIT, each academic department operates its own internal network independently and keeps track of all operational hardware and software as well as user privileges and access management. The individual department system administrators, or Computer Support Representatives (CSRs), install patches, run virus protection programs, and check regularly for strong password compliance. They are the most familiar with their own departmental network. When security incidents become known to OIT, typically via their campus wide network of intrusion detection systems and firewalls, they will inform

the respective department and collaborate on problem resolution. If the CSR does not stop the problem or requires assistance, an OIT Information Security Specialist will investigate further. This is often performed at the center Network Operations Center by sniffing the traffic going to the particular host and examining the capture log. One exception is the student housing network (ResNet) where individual hosts are automatically quarantined from Internet use until the student fixes the problem or patches the vulnerability.

The Georgia Institute of Technology's total campus population is approximately 15,000 undergraduate and graduate students and approximately 5,000 staff and faculty. There are 69 individual departments spread over the campus with between 30,000-35,000 networked computers operational at any given time. The total amount of IP addresses allocated to Georgia Tech is equivalent to 2.5 Class B addresses. The connection from the campus to the Internet includes two OC-12's and one OC-48 with an average throughput of 600Mbps. On average, over four terabytes of data is processed each day. With the large size of the campus network, OIT's main concern is determining the location of high-priority alarms and effectively allocating limited human resources to resolve the problem.

In order to determine whether the alarms are significant or not, OIT analysts typically rely upon alarm count, alert severity and time of day. Browsing through text alarm log files is usually the method used. IDS tools come with visual components, but calibrating tools to filter and visualize alarm data is tedious. Therefore, administrators ultimately resort to text logs instead. An average of 50,000 alarms are generated from IDS sensors installed across the campus network each day. Currently, it takes a significant amount of the analysts time to sift through the alarms and determine which concerns are immediate, which need further analysis and which can be ignored, at least temporarily. The process of determining that an alarm was triggered due to a serious problem requires knowing what services the particular host provides, i.e., is this a department server, or a single user machine. Then deciding if immediate action needs to be taken depends what the location and function of the host is, for example, a major department server will compel immediate action while a

student's system in the dorm will not.

6.2 Methodology

Based on the information we received from the network analysts and going over the alarm parameters, the tool was designed. Location of alarm is one of the most important variables to help determine anomalous activity and knowing the source. To cover a network of this size, efficient use of screen space was important which is why multiple vertical axes are used. Time sequence is also given space on the horizontal axis while other information is visually encoded. Details were explained in section 4.2. Users who analyze IDS alarms for mid-size to large networks were surveyed after the first version of the tool was developed. This helped to design the extra functions developed since then. We also have received feedback based on the current version of the tool. As of this writing, more features have been added based on analyzing the feedback from the first version of the tool.

6.3 Results

A survey and questionnaire was used to get written feedback on the tool. Some of the questions were as follows:

- Did you still want to refer back to the text log for information while using this tool? If so, for what?
- What information was missing from the overall view that could have pointed you out to important events?
- What was the least useful or distracting feature and why?
- What was the best feature and how was it useful? Were there any insights or discoveries you made that you think were easier in this tool versus going through traditional methods (text logs, IDS graphical interface)?

They were asked to rate the following:

- Information represented was intuitive and made sense.
- Operation and performance of zooming and panning was good.
- It was useful to see the main screen while panning in the zoom view.
- Traversing overlapped lines and clustered alarm marks in the zoom view by clicking mouse buttons was useful and intuitive.
- Using the tool was faster in determining if alarm(s) are false or required further investigation compared to using the text logs.
- There was enough information in the overview to get a general idea of activity, identify alerts of interest and drill down for more detail.
- It was simple to use and to learn how to use this tool.
- This system has all the functions and capabilities I expect it to have.

All concurred that the current tool saves hours of time and that it was simple and intuitive to use. They found it helpful to see alarm event sequences and patterns using the time axis. Seeing what hosts were making connections and triggering alarms with lines showing those connections was also mentioned as an advantage. Additionally, all users thought the performance of the tool functions, such as zooming, panning, mouse-overs and filtering operated well and were easy to use. Some rarely feel the need to return to the alarm text log while some felt when they pinpoint an activity, they would look to the IDS console to get extra information.

The process of implementation has been an iterative one. In each stage, improvements can be made and going back to users for re-evaluation helps to reinforce the design as well. Features can be universally helpful, but some are unique to the type of network and

responsibilities of the analyst so one tool will not be ideal for everyone. Currently this tool is used by OIT to assist them in viewing IDS alarms and identifying and stopping behavior

CHAPTER VII

CONTRIBUTIONS AND FUTURE WORK

In this thesis new techniques were developed to aid in network security using information visualization. Research contributions have been made in the following:

- Network data scaling and processing
- Port activity visualization
- Useful visualization showing a larger amount of information than textual methods
- Scaling port numbers and IP address for maximum use of screen space without occlusion
- Performing and using user study results to design an IDS alarm visualization tool

7.1 Research Contributions

Chapter 1 explained motivations for using information visualization with network data for security purposes and background on the two data types we used. Chapter 2 describes some of the related state of the art in this growing area. Methods for scaling data and data parameters, data processing and a description of data parameters is described in chapter 3. Chapter 4 dsecribes the goals we have for the systems, how they were designed and explains how they function. We have worked with forensic data and IDS alarm logs showed these results. Chapter 5 describes threat models and case scenarios for the systems. Brief explanations of the general types of attacks that networks encounter were also given to explain how visualizing port activity would be helpful. Casual and anomalous events that occurred that were easier to find using IDS Rainstorm are shown as examples along with the

possible steps a network administrator would take in using the tool to find these. Chapter 6 discusses the user study that was performed to develop the design of IDS Rainstorm and the the feedback given after the tool was evaluated. Appendix A gives resources in network security visualization for those that are interested in this area.

7.1.1 Port Activity

The advantages of plotting port statistics based on network header information is that a flow does not have to finish for it to be represented. For especially long flows, the information will not be presented in a timely manner. It uses scaling and grouping techniques to allow all information to be seen at once while not becoming occluded and maintains good use of screen space.

7.1.2 Alerts

A user study was first performed to create the design and then feedback on the tool to evaluate the result. The feedback will also help for future work. This tool makes better use of screen space by representing more IP addresses by using multiple vertical axes than other tools and still show time patterns giving a visual map of the network. Zooming and panning are implemented to provide details on demand in addition to the main view which is useful for viewing even more information.

7.2 Future Work

Viewing more types of data are needed to get a full picture of network activity. More data parameters will also be helpful, but they must be visually encoded in a way that will not clutter up screen space and aid analysts in their tasks. When developing port activity visualization it showed that network activity findings were easier but it only shows one perspective and a subset of attack types. We also learned that going further with development would require user studies. Some of the techniques used, in addition to a user study, were

then used to develop a visualization for IDS alarms. A summary of the feedback given and other future work we realized during this process are given:

- Remap the two axes such that the entire internal IP address range is on the left and a small set of suspicious external IPs are on the right. For example, if a worm is targeting a network and the IPs affected are spread across the IP space of the network, then it is difficult to correlate the behavior. A subset of these external IPs that connect to the local network can be plotted on the right parallel axis and the entire local IP space condensed on the left axis. This will help to see what hosts are triggering alarms due to activity of the external IP address.

- Combine alert outputs from the other IDS systems used in the tool to review each system's output and help with false alarms.

- Find a way to show alarms repeated from other tools to help rule out false alarms and determine legitimate alarms.

- Integrate netflow information to correlate what hosts are making connections to the network along with the alarm data.

- Showing the country of the external IP and reducing occlusion when external IPs that are printed overlap.

The IDS alarms we used was from one sensor and not the total amount of alarms actually generated from all sensors. Enhanced filtering and improved querying capabilities such as on specific alarm(s), IP addresses, port number(s), and protocol will be required to compensate for the increased number of alarms. This will also be necessary if other IDS alerts are integrated along with netflow information.

APPENDIX A

VISUALIZATION RESOURCES

This section describes some of the excellent books in this area, along with technical information and other network visualization tools that have been developed. There are also good websites that give a lot of information, most of them made by people who are working diligently in this growing field. Hopefully this will be useful to those who are interested in visualization for network security.

A.1 Books

Edward Tufte is at the top of the list. His books and writings have given a good base of knowledge and inspiration. He also has seminars in different cities. His website has information on his works and his seminar schedule. It has a very active message board in which Edward Tufte himself participates and answers questions. People from various fields of work and study can benefit. `http://www.edwardtufte.com/tufte/index`

His list of books are:

- *The Visual Display of Quantitative Information*
- *Envisioning Information*
- *Visual Explanations: Images and Quantities, Evidence and Narrative*

There are two books used in the Information Visualization (CS 7450) class at Georgia Tech that explain the fundamentals. The first is the required text for the class and the other a suggested text.

- *Information Visualization* by Robert Spence

- *Information Visualization, Second Edition : Perception for Design* by Colin Ware

A.2 Tools

For data capture of raw network packets tcpdump or an equivalent packet capture library for your operating system is needed. `http://www.tcpdump.org/`

For IDS alarms, Snort is a good option as it is open source. `http://www.snort.org/`

Below are some tools developed for network security visualization:

Rumint by Gregory Conti visualize packet data using 7 different views. It runs on the Windows OS. `http://www.rumint.org/`

John Goodall's tool TNV visualizes activity using a matrix view, where x axis is time and y axis maps to IP addresses:

`http://userpages.umbc.edu/~jgood/research/tnv/`

NVisionIP show flow activity on a network, allowing a user to drill down for more details on particular host(s):

`http://security.ncsa.uiuc.edu/distribution/NVisionIPDownLoad.html`

VizFlowConnect visualizes flow connections between hosts using parallel axes:

`http://security.ncsa.uiuc.edu/distribution/VisFlowConnectDownLoad.html`

A.3 Programming Resources

JOGL, the OpenGL extensions for Java was used for the IDS alarm visualization tool. Java was chosen as we were familiar with this language and for its platform dependancy, and OpenGL for its performance and features. Ultimately, any programming language is fine, as long as one can implement and demonstrate ideas. There will be some start up time involved in getting used to OpenGL, but because of the capabilities of OpenGL, it is worth the effort. There are OpenGL extensions available for many other programming languages like perl, C, and python.

JOGL resources:

Homepage for the JOGL API project. Includes downloads to the JOGL libraries:

`https://jogl.dev.java.net/`

The Java Gaming website includes a forum on JOGL which includes helpful information and resources: `http://www.javagaming.org/forums/index.php`

Others have used Flash Macromedia, Java2D, and Perl/Tk to name a few. If one is creating a prototype for demonstration purposes, any language should work.

A.4 Website

The VizSEC (Visualization Security) Community Homepage has a mailing list and information on past and current workshops:

`http://www.projects.ncassr.org/sift/vizsec/`

REFERENCES

[1] "Netsys.com: The intelligent hacker's choice." DDoS article Available at: http://www.netsys.com/library/papers/ddos-ircbot.txt; accessed 28-March-2004. 5.1.3

[2] "Port numbers." Internet Assigned Numbers Authority. Available at: http://www.iana.org/assignments/port-numbers; accessed 28-May-2004. 3.1.2

[3] "Tcpdump/libpcap." Available at: http://www.tcpdump.org/; accessed 28-March-2004. 3.2.1

[4] "Visualization." Cooperative Association for Internet Data Analysis. Available at: http://www.caida.org/analysis/visualization; accessed 28-June-2003. 2.2

[5] "10 most probed ports." Distributed Intrusion Detection System, June 2004. Available at: http://www.dshield.org/topports.php; accessed 28-March-2004. 3.1.2

[6] "The honeynet project: Scan of the month," 2004. Available at: http://project.honeynet.org/scans/; accessed 28-March-2004. 5.1.4

[7] "Stealthwatch+therminator." Lancope, 2004. Available at: http://www.lancope.com/; accessed 03-March-2006. 2.3

[8] Abdullah, K., Lee, C., Conti, G., and Copeland, J. A., "Visualizing network data for intrusion detection," in *IEEE Information Awareness Workshop at West Point*, June 2005. 2.3

[9] Abdullah, K., Lee, C., Conti, G., Copeland, J., and Stasko, J., "Ids rainstorm: Visualizing ids alarms," in *IEEE Symposium on Information Visualization's Workshop on Visualization for Computer Security (VizSEC)*, pp. 1–10, 2005. (document), 2.2

[10] Alliance, I., "Razorback - snort network intrusion detection front-end." Available at: http://www.intersectalliance.com/projects/RazorBack/; accessed 03-May-2005. 2.4

[11] Becker, R., Eick, S., and Wilks, A., "Visualizing network data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 1, pp. 16–28, March 1995. 2.2

[12] Card, S. K., Mackinlay, J. D., and Shneiderman, B., *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999. 1

[13] Conti, G. and Abdullah, K., "Passive visual fingerprinting of network attack tools," in *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, (New York, NY, USA), pp. 45–54, ACM Press, 2004. 2.2

[14] Conti, G., Abdullah, K., Grizzard, J., Stasko, J., Copeland, J. A., Ahamad, M., Owen, H. L., and Lee, C., "Countering security information overload through alert and packet visualization," *IEEE Computer Graphics and Applications*, 2006. 1.1.1, 1.1.1.2, 4.2.2

[15] Cox, K. and Eick, S., "Case study: 3d displays of internet traffic," in *Proceedings of Information Visualization (INFOVIS)*, IEEE Computer Society, pp. 129–131, Oct. 1995. 2.2

[16] Danyliw, R., "Analysis console for intrusion databases (acid)." Available at: `http://www.andrew.cmu.edu/user/rdanyliw/snortacid.html`; accessed 03-May-2005. 2.4

[17] Debar, H. and Wespi, A., "Aggregation and correlation of intrusion detection alerts," in *Recent Advances in Intrusion Detection (RAID)*, pp. 85–103, Springer-Verlag, 2001. 1

[18] Donald, S. D. and McMillen, R. V., *Therminator2: Developing a Real Time Thermodynamic Based Patternless Intrusion Detection System*. PhD thesis, Naval Postgraduate School, Monterey, California, 2001. 2.3

[19] Eagan, J., Harrold, M. J., Jones, J. A., and Stasko, J., "Technical note: Visually encoding program test information to find faults in software," in *Proceedings of IEEE Information Visualization 2001*, (San Diego, CA), pp. 33–36, October 2001. 4.2.2

[20] Eick, S. G., Steffen, J. L., and Eric E. Sumner, J., "Seesoft-a tool for visualizing line oriented software statistics," *IEEE Transactions on Software Engineering*, vol. 18, no. 11, pp. 957–968, 1992. 4.2.2

[21] Erbacher, R., Walker, K., and Frincke, D., "Intrusion and misuse detection in large-scale systems," *Computer Graphics and Applications*, vol. 22, pp. 38–48, January/February 2002. 2.2

[22] Ford, D., "Application of thermodynamics to the reduction of data generated by a non-standard system," tech. rep., Department of Physics Naval Postgraduate School, Monterey, CA., Feb. 2004. Available at: `http://www.arxiv.org/abs/cond-mat/0402325`; accessed 03-May-2005. 2.3

[23] Goodall, J. R., Lutters, W. G., Rheingans, P., and Komlodi, A., "Preserving the big picture: Visual network traffic analysis with tnv," in *IEEE Symposium on Information Visualization's Workshop on Visualization for Computer Security (VizSEC)*, pp. 47–54, 2005. 2.2

[24] Hoagland, J. A. and Staniford, S., "Viewing ids alerts: Lessons from snortsnarf," in *Proceedings of 2001 DARPA Information Survivability Conference and Exposition (DISCEX 2001)*, pp. 12–14, 2001. 2.4

[25] Julisch, K., "Clustering intrusion detection alarms to support root cause analysis," in *ACM Transactions on Information and System Security*, vol. 6, ACM Press, November 2003. 1

[26] Jungck, P. and Shim, S. S. Y., "Issues in high-speed internet security," *Computer*, vol. 37, pp. 36–42, July 2004. 1

[27] Keim, D., Hao, M., and Dayal, U., "Hierarchical pixel bar charts," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, pp. 255–269, July-Sept. 2002. 4.1.2.1

[28] Koike, H. and Ohno, K., "Snortview: Visualization system of snort logs," in *VizSEC/DMSEC'04* (ACM, ed.), (Washington DC, USA), October 29 2004. 2.4

[29] Koike, H., Ohno, K., and Koizumi, K., "Visualizing cyber attacks using ip matrix," in *IEEE Symposium on Information Visualization's Workshop on Visualization for Computer Security (VizSEC)*, pp. 91–98, 2005. 2.2, 2.4

[30] Krawetz, N., "Anti-spam solutions and security, part 2." Security Focus. Available at: `http://www.securityfocus.com/infocus/1766`; accessed 09-March-2004. 5.1.2

[31] Lakkaraju, K., Yurcik, W., and Lee, A. J., "Nvisionip: netflow visualizations of system state for security situational awareness," in *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, (New York, NY, USA), pp. 65–72, ACM Press, 2004. 2.3

[32] Levine, J., LaBella, R., Owen, H., Contis, D., and Culver, B., "The use of honeynets to detect exploited systems across large enterprise networks," in *Proceedings of the IEEE Workshop on Information Assurance*, IEEE Systems, Man and Cybernetics Society, (West Point, NY), pp. 92–99, June 2003. 1.1.1.1, 4.1.1, 5.1.4

[33] Lyman, P., "How much information 2003," October 2003. Available at: http://www.sims.berkeley.edu/research/projects/how-much-info-2003; accessed 28-March-2004. 1

[34] McPherson, J., Ma, K.-L., Krystosek, P., Bartoletti, T., and Christensen, M., "Portvis: A tool for port-based detection of security events," in *VizSEC/DMSEC'04* (ACM, ed.), (Washington DC, USA), October 29 2004. 2.3

[35] Mimoso, M., "Experts ponder spam, worm-writing connection." Search Security, Nov. 2003. Available at: http://searchsecurity.techtarget.com/; accessed 03-March-2004. 5.1.2

[36] Nyarko, K., Capers, T., Scott, C., and Ladeji-Osias, K., "Network intrusion visualization with niva: an intrusion detection visual analyzer with haptic integration," in *10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS '02)*, pp. 277–284, March 2002. 2.2

[37] Oline, A. and Reiners, D., "Exploring three-dimensional visualization for intrusion detection," in *IEEE Symposium on Information Visualization's Workshop on Visualization for Computer Security (VizSEC)*, pp. 113–120, 2005. 2.2, 2.3

[38] Plonka, D., "Flowscan: A network traffic flow reporting and visualization tool," in *Proceedings of the USENIX Fourteenth System Administration Conference LISA XIV*, 2000. 2.3

[39] Ren, P., Gao, Y., Li, Z., Chen, Y., and Watson, B., "Idgraphs: Intrusion detection and analysis using histographs," in *IEEE Symposium on Information Visualization's Workshop on Visualization for Computer Security (VizSEC)*, pp. 39–46, 2005. 2.3

[40] Roesch, M., "Snort." Available at: `http://www.snort.org/`; accessed 03-May-2005. 2.4

[41] Skoudis, E., *Counter Hack: A Step-by-Step Guide to Computer Attacks and Effective Defenses*. New Jersey: Prentice Hall, 2001. 5.1.1

[42] Spence, R., *Information Visualization*. England: ACM Press, 2001. 4.1.2.1

[43] Takada, T. and Koike, H., "Mielog: A highly interactive visual log browser using information visualization and statistical analysis," in *Proceedings of LISA XVI Sixteenth Systems Administration Conference*, pp. 133–144, The USENIX Association, Nov. 2002. 2.4

[44] Takada, T. and Koike, H., "Tudumi: Information visualization system for monitoring and auditing computer logs," in *Proceedings of Information Visualization*, pp. 570–576, July 2002. Sixth International Conference. 2.4

[45] Ware, C., *Information Visualization: Perception for Design*. California: Academic Press, 2000. 1

[46] Yin, X., Yurcik, W., Li, Y., Lakkaraju, K., and Abad, C., "Visflowconnect: Providing security situational awareness by visualizing network traffic flows," in *Proceedings on the Workshop on Information Assurance (WIA04)*, 2004. held in conjunction with the 23rd IEEE International Performance Computing and Communications Conference (IPCCC 2004). 2.2

[47] ĺ, L., "Stealthwatch+therminator." Available at: `http://www.lancope.com/products/`; accessed 03-May-2005. 3.2.2

VITA

Kulsoom Abdullah received her Bachelor's in Computer Engineering at the University of Central Florida in 1998. She came to Georgia Institute of Technology and earned a Masters in Electrical and Computer Engineering in 2000. She started working with Dr. John A. Copeland as a graduate research assistant in the Communications Systems Center in 2002. Since then she has been working there in network security research. Her areas of focus for her PhD work are in network security visualization.