IDS RainStorm: Visualizing IDS Alarms

Kulsoom Abdullah* Georgia Tech
Chris Lee† Georgia Tech
Gregory Conti‡ Georgia Tech
John A. Copeland§ Georgia Tech
John Stasko¶ Georgia Tech

ABSTRACT

The massive amount of alarm data generated from intrusion detection systems is cumbersome for network system administrators to analyze. Often, important details are 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 where system administrators can get a general sense of network activity and easily detect anomalies. They then have the option of 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, 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. The motivation and background of our design is presented along with examples that illustrate its usefulness.

CR Categories: C.2.0 [Computer-Communication Networks]: General—Security and Protection C.2.3 [Computer-Communication Networks]: Network Operations—Network Monitoring H.5.2 [Information Systems]: Information Interfaces and Presentation—User Interfaces

Keywords: IDS alarms, alert visualization, log visualization, alarm visualization, network monitoring, network security information visualization

1 INTRODUCTION

Network attacks are persistent and growing. 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. Despite this reduction, time wasted by analyzing these logs can effectively negate the value of the system.

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. Faster understanding of a greater amount of data and better insights are some of the benefits. 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.

If real attacks are not stopped, they will generate many alarms [3]. 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 [7]. We believe visualization will allow human analysts to effectively complement machine learning algorithms and traditional text log file analysis and significantly improve detection of for anomalous activity. Additionally, a visualization can represent information more densely than text, shortening analysis time. 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.

2 RELATED WORK

Visualization is becoming an increasingly effective tool to aid network security and there have been several innovative approaches to log analysis. Representative of general log visualizations developed for security are Tudumi [10] and Mielog [11]. 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 [9] alarm log files. One is SnortView [8] 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, but the amount of information shown is limited to a subset of IP address ranges, time (4 hours) and number of attacks. Snort-Snarf [6] and ACID [2] 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 [1] 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 our tool, IDS Rainstorm, is that 2.5 class B IP address spaces (65,532 hosts $\times$ 2.5 = 163,830 total) can be represented successfully on one display. Using pixels to represent alarms encodes a large amount of alarm data into one screen for a full day (24 hours).

*e-mail: kulsoom@ece.gatech.edu
†e-mail: chris@ece.gatech.edu
‡e-mail: conti@cc.gatech.edu
§e-mail: copeland@ece.gatech.edu
¶e-mail: stasko@cc.gatech.edu

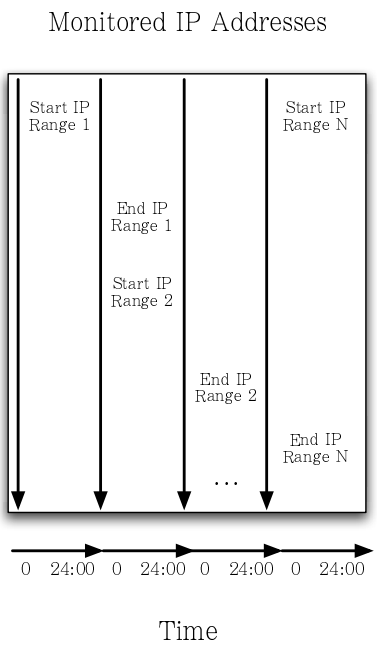

Figure 1: Design of the Basic Visualization and Representation. Each vertical axis represents IP addresses in sequential order. Each horizontal axis associated with the vertical axes represent one 24 hour period.

3 MOTIVATION

We interviewed Office of Information Technology (OIT) system administrators at Georgia Tech, to find out how they deal with and monitor alarms, how they go about the process, what they look for, and how they analyze possible intrusions. The requirements collected during these interviews guided the design of our visualization system.

The Office of Information Technology (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 Technologys 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.

4 SYSTEM DESIGN

4.1 Data Generation Using the StealthWatch IDS

The Stealthwatch [12] anomaly based IDS system is one of the security appliances used to defend the Georgia Tech campus. It monitors flow activity and bandwith usage to detect anomalous behavior. Signatures known a priori are not needed to detect an attack. For our analysis, we were provided with Stealth Watch 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 are generated in one day from StealthWatch. The parameters we have access to are explained below.

Table 1: Sample alarm log file format. This includes the unix time stamp when the alarm was generated, the alarm type code, and the IP address or addresses involved. IP addresses are represented here as 32 bit integers. Alarms that involve an external IP address are given, otherwise a 0 value is given.

Unix Time	Alarm Type	Internal IP Address	External IP Address
1112481973	68	11124819XX	0
1112482140	66	11238117XX	2151501364
1112482200	66	33437668XX	0

4.2 Alarm Parameters

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

- **Alarm type:** Each of the 33 alarm types are defined by an integer value. System administrators usually judge severity

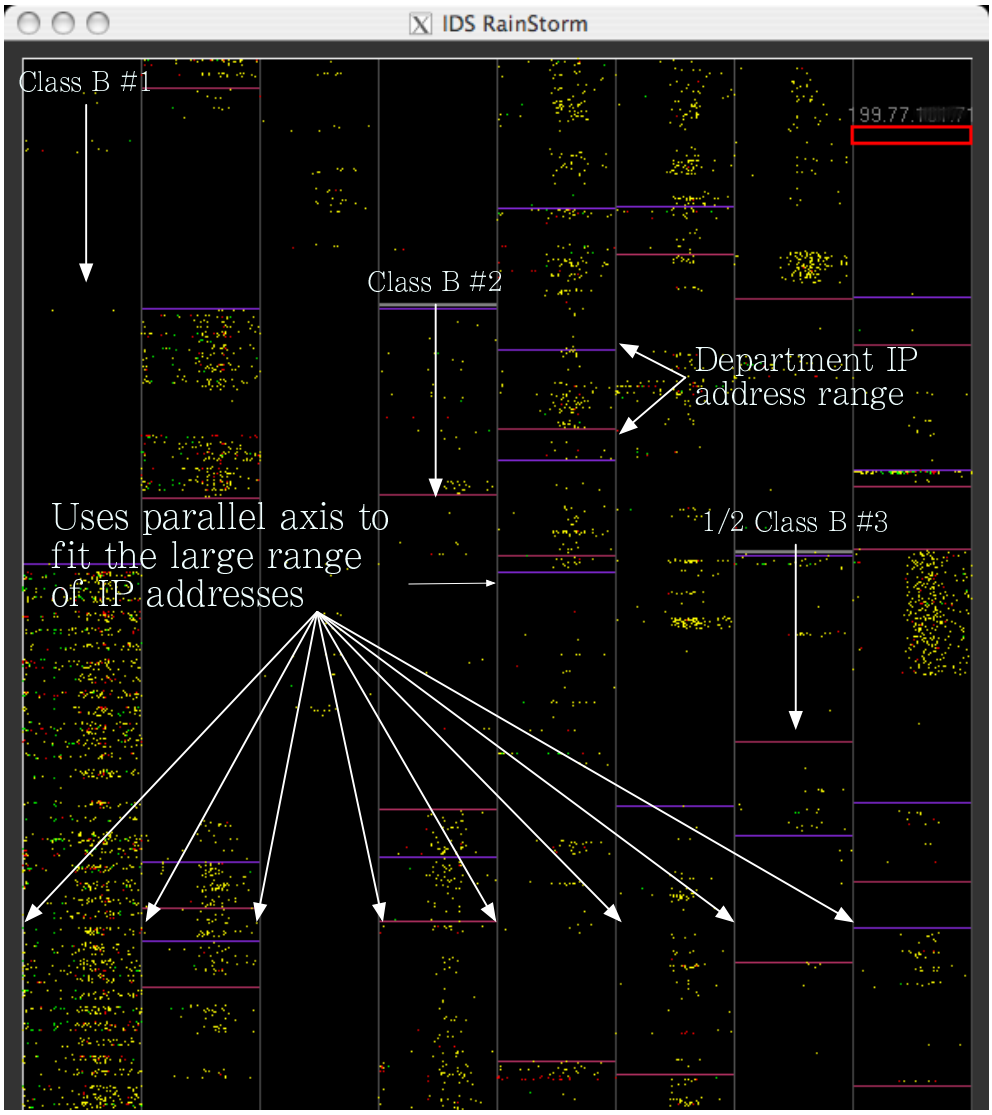

Figure 2: IDS RainStorm main view: The 8 vertical axes are shown that represent the 2.5 Class B IP addresses. The thicker horizontal lines between these axes show where each Class B starts. The other horizontal lines show the start and end of each department. Those addresses not in a department are either unallocated or reserved for special use by OIT and other departments. This screenshot shows an entire day's worth of real alarms generated.

of the alarm and classify the alarms themselves if their definition differs from that of the IDS company's default definitions. Also available in a separate definition file is a detailed description of what the alarm means and what activities could have triggered the alarm. The definitions in the file were used to give alarm information in the tool.

- **Time:** When each alarm is generated, a unix timestamp with a resolution of one second starting at the unix epoch is recorded. This information helps to determine its temporal position among the rest of the alarms and can help to find significant patterns or position in a sequence of events.

- **IP Addresses:** The victim internal IP address of the alarm is given, and if an external IP is associated with the alarm, then the external IP is given as well. The IP addresses are given as 32 bit integers.

Port number is also included with some types of alarms. We are currently working on a way to integrate this visually instead of just showing a number in a detail view. An example of part of an alarm log file is shown in Table 1. Commonly occurring StealthWatch alarm definitions are given in the list below:

- 61 *Host Max Flows*: Indicates that the host has had total active flows above some threshold in the last 5 minutes. This could be possible DoS, DDoS activity against the host, or the host is sharing files on many connections. When activated this alert also returns the number of new flows in the last 5 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 an index that counts the times a host has triggered an

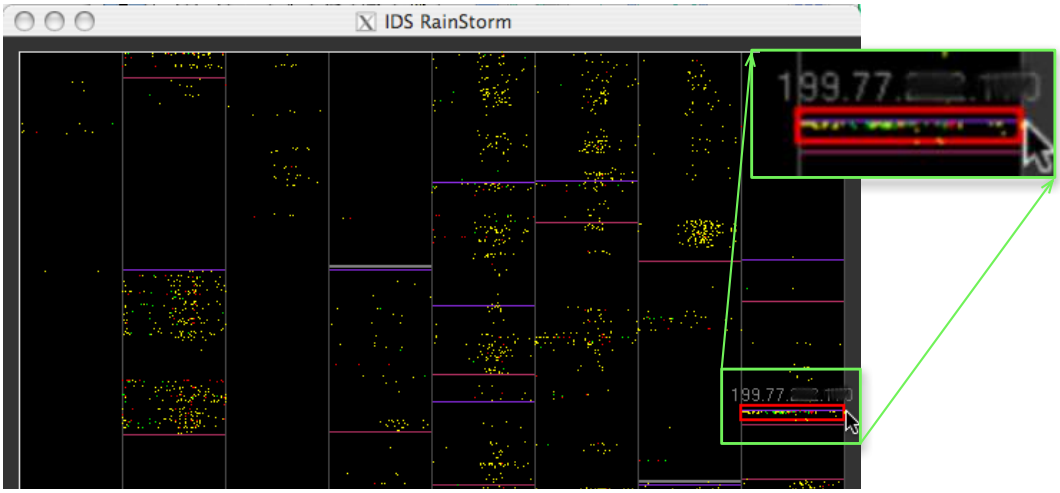

Figure 3: User selecting area to zoom in. (Here we have artificially highlighted a region in green and magnified it to assist the reader.) The IP address shown is the one at the top position in the red box region (the last two octets are intentionally blurred for privacy). The result of the zoom is shown in Figure 4

alarm, also called a Concern Index. Scanning activity and other nonconforming behavior generally causes 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 that a host on a user specified watch list has been active. This alarm is triggered anytime that the watch listed host transmits a packet or any outside host on the watch list receives a packet.

4.3 Visualization System

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. The overall view was designed to convey 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.

4.3.1 Visual Representation and Main View

Each of the views follows a general visualization technique developed to address this problem as shown in Figure 1. 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. 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 [4] and the SeeSoft tool [5] 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 2 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 1. 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.

4.3.2 Zoom View

As a user moves the mouse across the overview, a red box highlights the current cursor position as illustrated in Figure 3. 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 right. Labels are on the top horizontal axis to represent time within 24 hours. Alarms are seen as larger glyphs as seen in Figure 4. The zoom view also provides other information such as extra alarm detail for each alarm and any external IP address connections. Lines are drawn to show connections between the external attack IP address and local IP address shown in Figure 4. We implemented an extra zoom function within the zoom view. This happens when the mouse is double clicked on the zoom view, and within this same window, the layout is redrawn. This will allow time scale zoom by

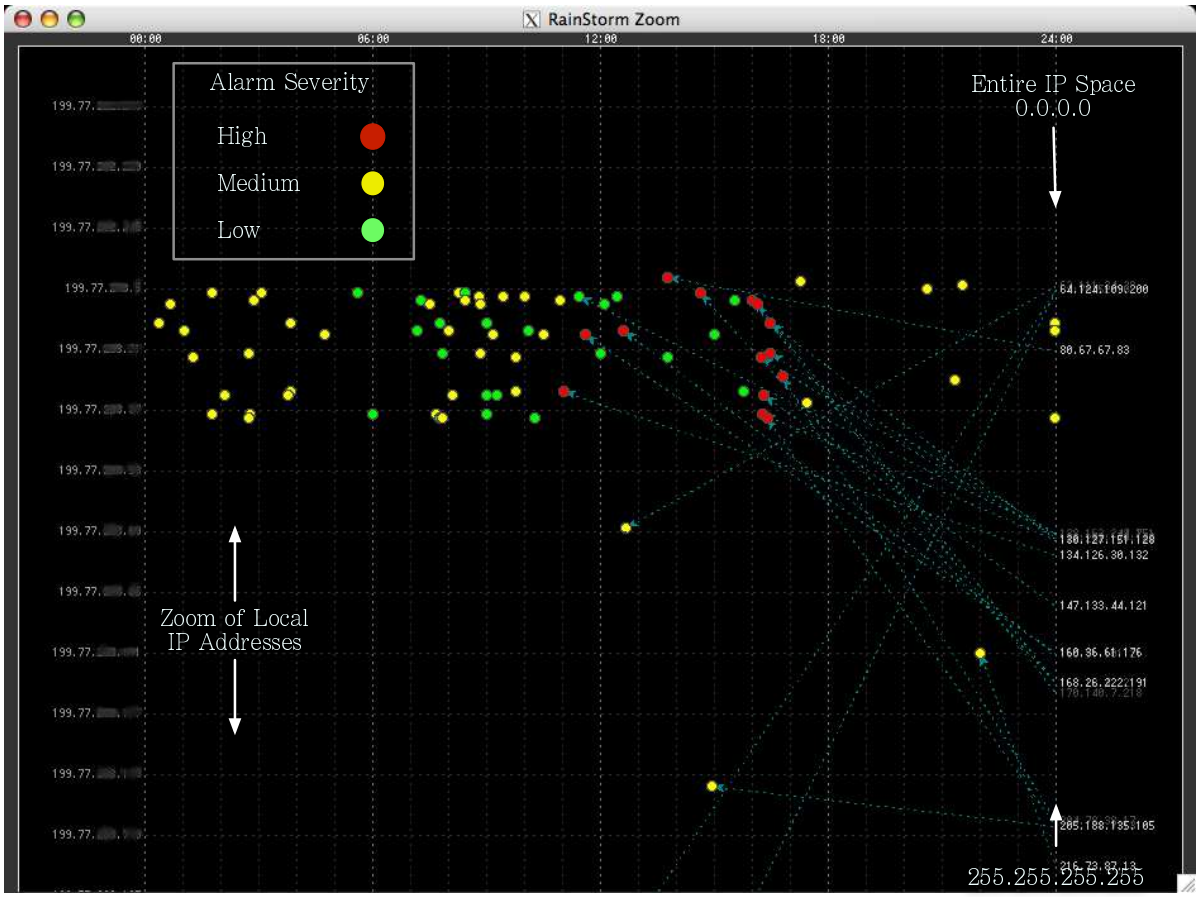

Figure 4: Zoom of a cluster of alarms seen in the overview. Also shown is the alarm severity legend and internal and external IP axes layout. The selected subset of internal IP addresses are represented on the left vertical axis, and external IP addresses on the right vertical axis.

a factor of 2x as shown in Figure 5. 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.

4.3.3 Other techniques

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 5.

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.

5 Example Usage Scenarios

The first example deals with a cluster of alarms in one area of the graph. One case is illustrated in Figure 3, 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 4. 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 5. 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

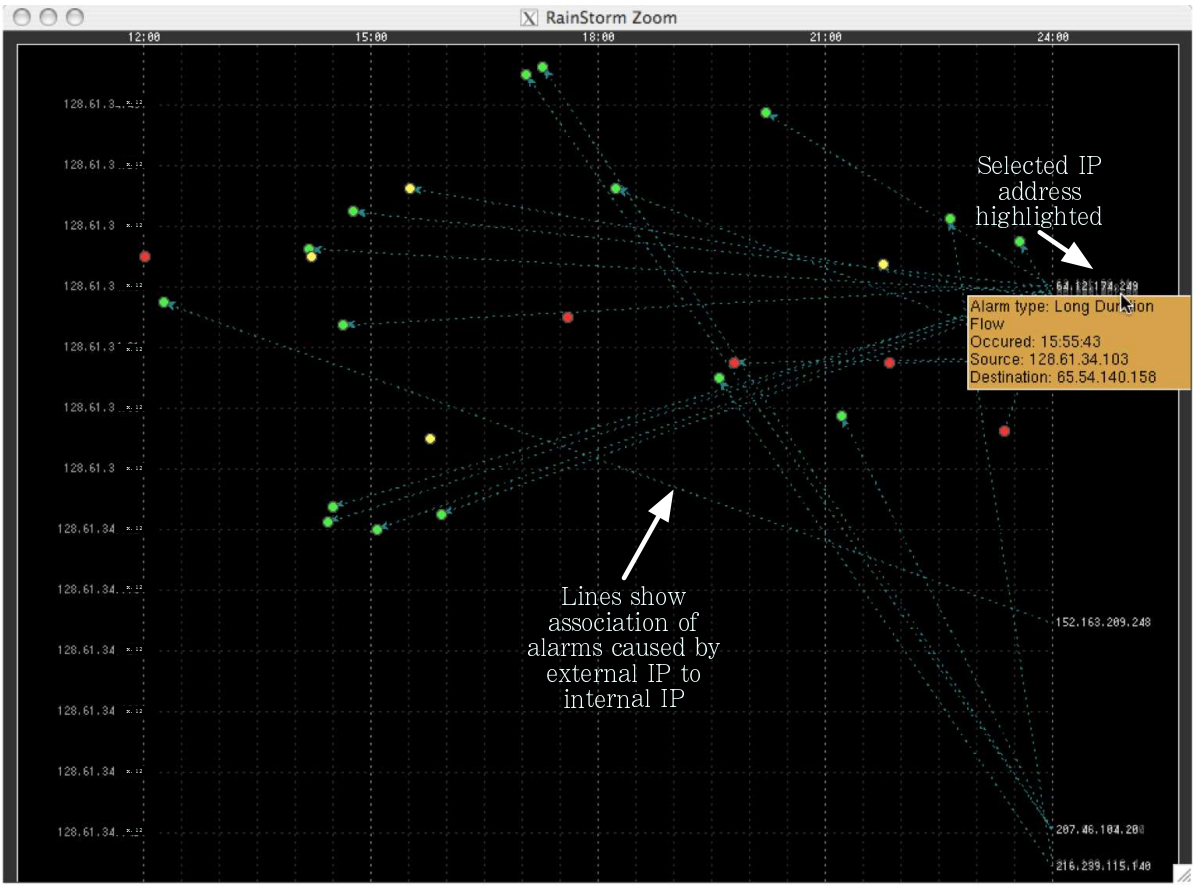

Figure 5: A zoom view on time. This zoom is a 2× zoom view of Figure 4. Internal IP addresses are on the left and external IP addresses are on the right.

rule out these alarms as a serious problem because they have occurred on the logical IP space reserved for Akamai.

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 6 where the user clicks and drags on an area, and the corresponding zoom views are shown in Figure 7 to illustrate this technique. Time is constant while the internal IP addresses on the left vertical axis change 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 6 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 (see Table 1). 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 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.

Figure 8 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 8). 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.

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 8). Here one IP address was a source for *Watch Port Active* alarms from many external IPs as shown in Figure 9. A close-up of this activity showing the 3:00-6:00 pm time range can also be seen in Figure 9. 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

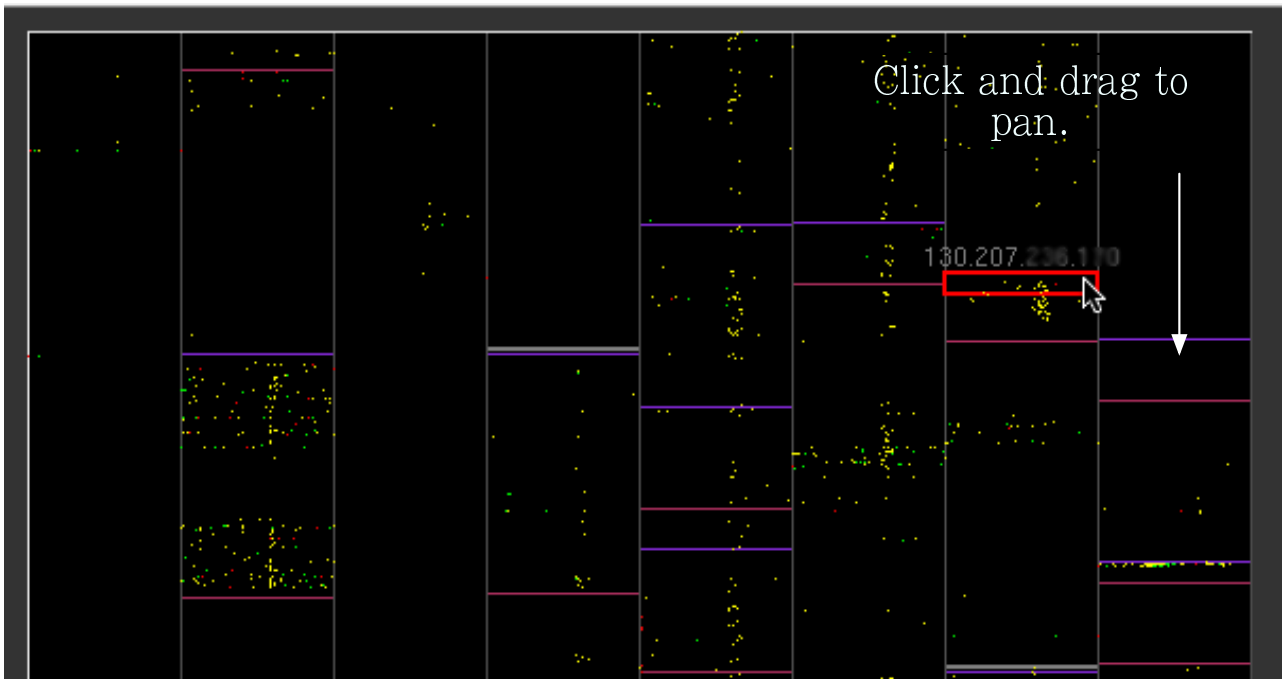

Figure 6: June 22nd overview. Clicking and dragging on the overview appears in the zoom view (shown in Figure 7) and animates the traversal down the IP space. The external IP axis is held constant.

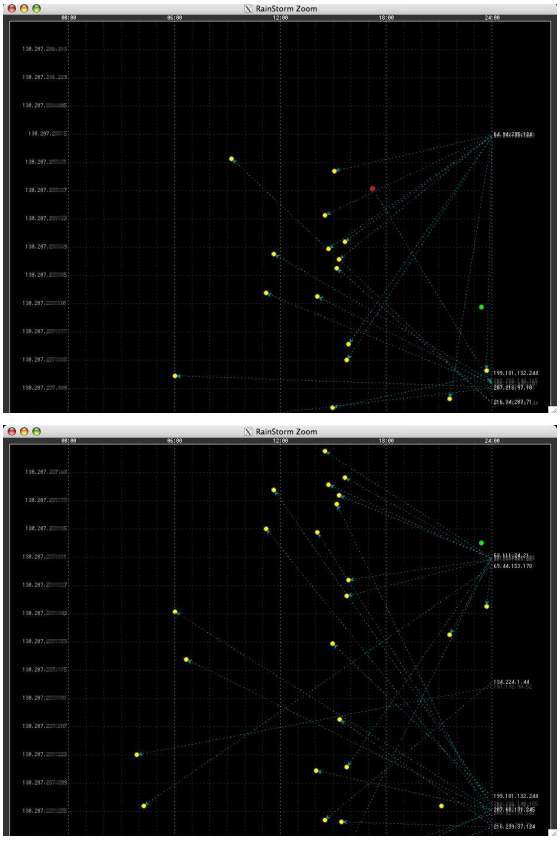

Figure 7: Panning results of Figure 6 shown as two transitions.

as the host had consistent communication to various IP addresses.

Also, on the same day, is another cluster of red alarms (region 3, Figure 8). 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 10. We can conclude these IP addresses have become infected with a bot, that has a specific time pattern of activity. These examples 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).

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 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

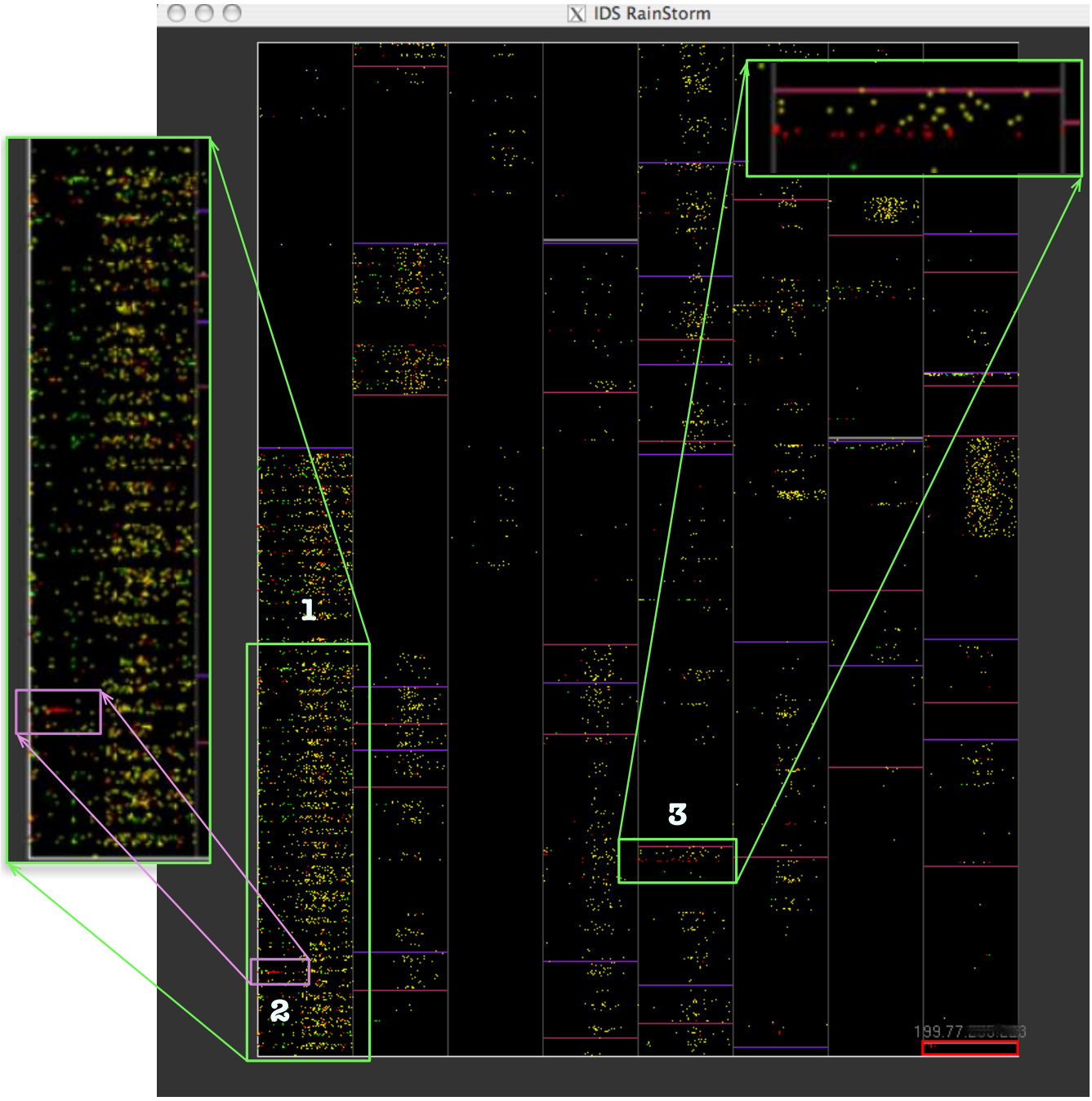

Figure 8: Overview of April 27th alarms. (Two regions are artificially identified in green and magnified for easier viewing.) Here region 1 shows activity in a subset of campus dorm IP addresses, a cluster of activity for a machine in the dorm is outlined in region 2 and region 3 shows a cluster of activity in occurring over a small range of IP addresses for the entire day.

would have to be dynamic or constantly changed to accommodate for this.

6 Conclusions and Future Work

Currently, IDS Rainstorm is useful for visualizing IDS alarms on a large network, observing time patterns, knowing location (local and external IPs) and severity. Our analysis of the requirements and tasks of the system administrators of Georgia Tech's network identified that these capabilities would be helpful. The tool presently can be used for forensic analysis, but we also would like to implement real-time analysis for live monitoring of the network. In order for the tool to be used on the network, system administrators will have to learn how to use it, how to interpret the display and what the visual patterns mean. People are generally good at these tasks and we are optimistic that system administrators will grasp these concepts quickly.

When we demonstrated a prototype version of IDS Rainstorm to system administrators, they seemed receptive to the idea and gave positive feedback. Some of the suggestions they made were the

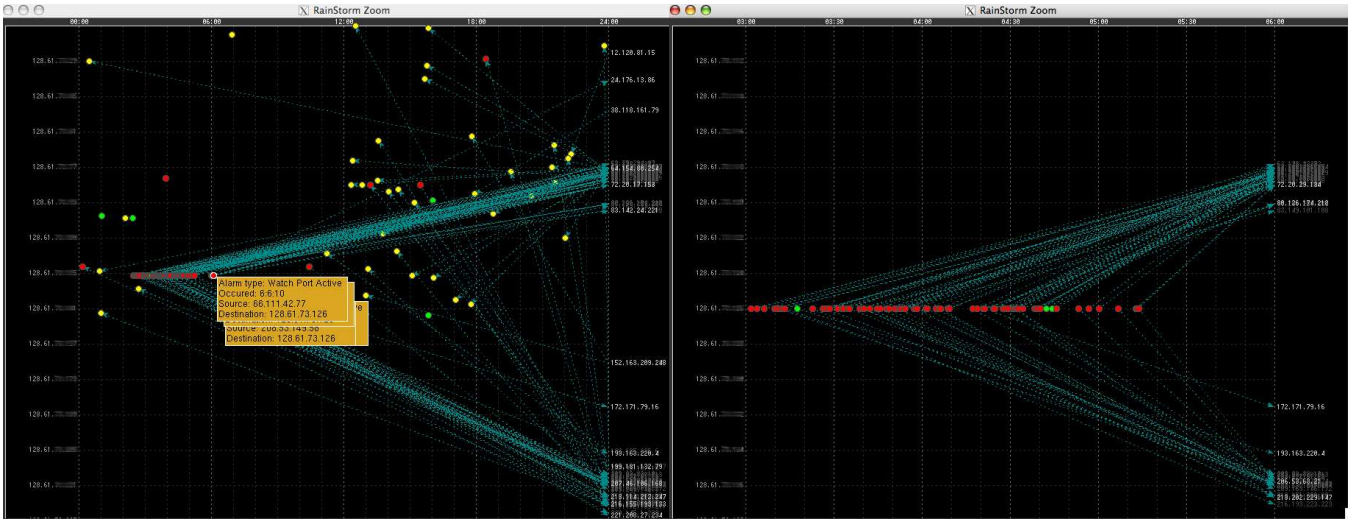

Figure 9: April 26th worm activity for a particular host located in the campus dorms. The left side is the zoom view of region 2 highlighted in Figure 8. The right side is a zoom of the left view that shows the activity from 3:00-6:00pm.

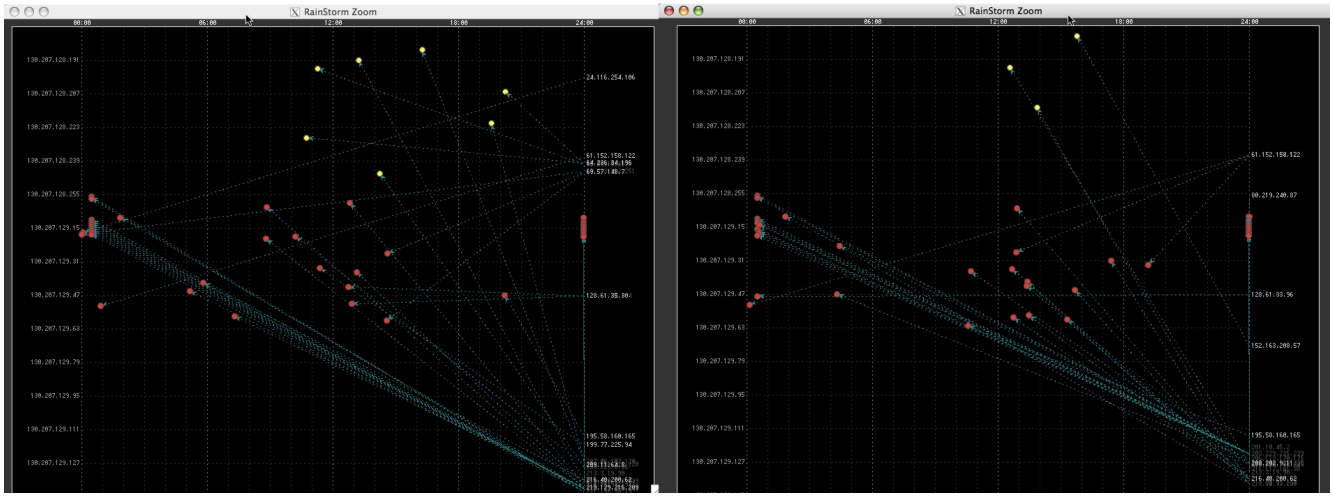

Figure 10: Bot activity shown for the same IP address space on April 26 (left) and April 27 (right). The activity time pattern for the two days is almost identical.

following:

- 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 harder to correlate the behavior. A subset of these external IPs that are connecting 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, i.e., who is connecting to whom in one zoomed view.
- 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.

An additional information parameter we would like to present in the design is associating ports with the alarm, or protocol. We plan to show external host information in the zoomed view which can help determine which alarms are important based on where the host is and whether it is an individual or a valid service (like in the example shown in Figures 6 and 7 and explained in section 5). This can be helpful in reducing false alarms and knowing if the alarm is a real threat. In addition to the glossing technique we used to show alarm type, we would like to visually encode the alarm type along with its respective classification. In addition to the color filtering, further query functions can be helpful for a more detailed search, such as on specific alarm or IP addresses, to save from having to traverse the graph manually. We would like to expand on these implementations by adding more filter and query options that are useful for analysis and real-time monitoring. Finally, our zooming action is discrete and jarring. Implementing a smooth, animated zoom will help the viewer maintain context into the overview better.

7 ACKNOWLEDGMENTS

We would like to thank the OIT department at Georgia Tech providing access to the alarm logs and for providing feedback about the system design.

REFERENCES

[1] InterSect Alliance. Razorback - snort network intrusion detection front-end. URL: `http://www.intersectalliance.com/projects/RazorBack/`.

[2] R. Danyliw. Analysis console for intrusion databases (acid). URL: `http://www.andrew.cmu.edu/user/rdanyliw/snortacid.html`.

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

[4] James Eagan, Mary J. Harrold, James A. Jones, and John Stasko. Technical note: Visually encoding program test information to find faults in software. In *Proceedings of IEEE Information Visualization 2001*, pages 33–36, San Diego, CA, October 2001.

[5] Stephen G. Eick, Joseph L. Steffen, and Jr. Eric E. Sumner. Seesoft-a tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering*, 18(11):957–968, 1992.

[6] James A. Hoagland and Stuart Staniford. Viewing ids alerts: Lessons from snortsnarf. In *Proceedings of 2001 DARPA Information Survivability Conference and Exposition (DISCEX 2001)*, pages 12–14, 2001.

[7] Klaus Julisch. Clustering intrusion detection alarms to support root cause analysis. In *ACM Transactions on Information and System Security*, volume 6. ACM Press, November 2003.

[8] Hideki Koike and Kazuhiro Ohno. Snortview: Visualization system of snort logs. In ACM, editor, *VizSEC/DMSEC'04*, Washington DC, USA, October 29 2004.

[9] M. Roesch. Snort. URL: `http://www.snort.org/`.

[10] Tetsuji Takada and Hideki Koike. Mielog: A highly interactive visual log browser using information visualization and statistical analysis. In *Proceedings of LISA XVI Sixteenth Systems Administration Conference*, pages 133–144. The USENIX Association, Nov. 2002.

[11] Tetsuji Takada and Hideki Koike. Tudumi: Information visualization system for monitoring and auditing computer logs. In *Proceedings of Information Visualization*, pages 570–576, July 2002. Sixth International Conference.

[12] Lancope . Stealthwatch+therminator. URL: `http://www.lancope.com/products/`.