Visual Firewall: Real-time Network Security Monitor

Chris P. Lee [*] Georgia Tech CSC
Jason Trost[†] Georgia Tech CS Dept
Nicholas Gibbs[‡] Georgia Tech CS Dept
Raheem Beyah [§] Georgia Tech CSC
John A. Copeland [¶] Georgia Tech CSC

ABSTRACT

Networked systems still suffer from poor firewall configuration and monitoring. VisualFirewall seeks to aid in the configuration of firewalls and monitoring of networks by providing four simultaneous views that display varying levels of detail and time-scales as well as correctly visualizing firewall reactions to individual packets. The four implemented views: Real-Time Traffic, Visual Signature, Statistics, and IDS Alarm, provide the levels of detail and temporality that system administrators need to properly monitor their systems in a passive or an active manner. We have visualized several attacks, and we feel that even individuals unfamiliar with networking concepts can quickly distinguish between benign and malignant traffic patterns with a minimal amount of introduction.

CR Categories: C.2.0 [Computer-Communication Networks]: General—Security and Protection; H.3.1 [Information Storage and Retrieval]: Content Analysis and Indexing—Abstracting methods; H.5.2 [Information Interfaces and Presentation]: User Interfaces; I.3.8 [Computer Graphics]: Applications

Keywords: Network security, information visualization, user interfaces, firewall configuration, snort monitoring

1 INTRODUCTION

Network security has long been a concern of businesses as well as government agencies, which need to protect their intellectual property and sensitive information. The recent growth of computer attacks in the last decade has become more of a public concern, given the mass media reporting of worms, email viruses, and spam. Furthermore, network security has become a greater concern to the average person due to the impact of such attacks which generate large amounts of traffic (e.g., the Slammer and Blaster worms, Melissa virus, etc.).

Unfortunately, the current measures for securing networks fall short. Software patches are often never installed, installed late, or in some cases, take longer to download than the average survival time. SANS states that the current average survival time of an unpatched Windows XP box is 23 minutes [2]. The increase in this figure from the 15 minutes estimated a few months ago is due to Internet Service Providers (ISPs) starting to block activities on certain ports. Although virus scanners use heuristics to detect viruses similar to those which are already known, polymorphism may allow malicious code to elude detection. Our interviews with security specialists at Georgia Tech's Office of Information Technology confirm that firewalls commonly suffer from misconfiguration that often result in system compromises. Lastly, intrusion detection systems (IDSs) produce massive amounts of noise, require a large amount of complex configuration, and produce logs that are difficult to interpret, delaying any proactive response.

System logs are vital to the ability of a system administrator to assess the security of their networks. Logs contain information such as system accesses, IDS alarms, and summaries of network activity. Administrators must then read the logs, find any events that may pose a security risk, and finally perform the necessary tasks to correct the situation.

The shear volume of logs can quickly overwhelm the limited resources of the security staff. To process the logs, administrators need to read several thousand lines of terse messages that each take expertise to understand. Furthermore, high-level problems can be easily overlooked by focusing too much on the details of each log entry.

Since logs are often processed at regular intervals, if at all, attacks can be completed before the system administrator has a chance to stop them. For this reason, a real-time system is needed to augment the return on investment of reading log files. Ideally, this real-time system should allow administrators to assess the overall state of their networks at a glance.

System administrators and home users need tools to help them understand the state of their networks. Users need to be able to distinguish normal traffic from abnormal traffic and be able to filter through a large amount of IDS alarms. VisualFirewall aims to be the next innovation in visualization by presenting multiple views of the network state onto a single screen, which combine parallel comparisons along with various time-scales and network aspects.

In Section 1.1 of this paper, we describe the current state of network monitoring. The motivation for VisualFirewall and the implemented views are outlined in Section 2. The system architecture and design documentation are presented in Section 3. In Section 4, several attacks scenarios are analyzed. The conclusion of the paper is in Section 5, and an outline of future plans are in Section 6.

1.1 Related Work

Our work is unique from other security visualization tools in a number of ways. Our tool is one of the few that uses both firewall data and IDS alarm data. Most other visualizations use either raw packet dumps or IDS alert logs. Our visualization also utilizes network data to provide simultaneous representations of relevant information. This design allows users to see multiple representations of the network state, and makes attacking this visualization tool more difficult. The following are some visualization tools related to our work.

- VISUAL [3] uses a data source of only packet traces, specifically preprocessed PCAP files. The tool can be used for forensic analysis of packet data for a subnet consisting of less than 1000 hosts. Port scans and ping scans are easily recognizable as long as there is not a lot of other irrelevant traffic. This system is good for delineating general communication patterns, but not necessarily malicious activity, because it does not use any system logs or IDS alarms.

- Conti's [4] tool is used for real-time monitoring of network traffic. It uses parallel coordinate plots to show traffic patterns between various hosts on a network. This tool is designed to

[*]e-mail: chris@ece.gatech.edu
[†]e-mail: trost@cc.gatech.edu
[‡]e-mail: ngibbs@cc.gatech.edu
[§]e-mail: raheem.beyah@ece.gatech.edu
[¶]e-mail: john.copeland@ece.gatech.edu

Workshop on Visualization for Computer Security
October 26, Minneapolis, MN, USA
0-7803-9477-1/05/$20.00 ©2005 IEEE.

passively fingerprint network attack tools. Instead of attempting to fingerprint network attack tools, our Visual Signature view fingerprints the behavior of a host (traffic pattern) during and after a security incident, such as infection by a worm or a distributed denial of service (DDoS) attack.

- SecVis [9] is a visualization tool for real-time and forensic network data analysis. This tool displays packet capture data as a 3D parallel coordinate plot along with a dynamic scatter plot. Some network attacks are very apparent, but this tool does not take into account IDS data or system logs.

- SnortView [8] is tool that was developed specifically for analyzing Snort logs and syslog data. Its primary purpose is to use visualization to more effectively recognize false positives. It presents an updated view every two minutes and shows four hours worth of alert data. One slight limitation is that its user interface is in Japanese.

- PortVis [12] only analyzes high level summaries of packet data from a large network. Its primary focus is to detect large scale network security events. It provides multiple views of the same information to help correlate data and allow an operator to mentally shift between visualizations. The utility of this tool's multiple views is one of our motivating factors for presenting multiple views in our visualization software.

- Mielog [13] was made specifically for forensic analysis of system logs. It uses statistical analysis for classifying log entries and visualization techniques for displaying different characteristics of the logs. The main goal of this tool is to manually parse logs, not necessarily visualize their content.

- Erbacher's Hummer IDS Visualization [6, 7] uses a collection of logs and other network data from the Hummer IDS in order to represent network events between a monitored system and other hosts. Using real-time or forensic analysis, interacting hosts are visualized as a spoke and wheel diagram.

- The Spinning Cube of Potential Doom [10] represents Bro IDS alarms (which include every completed and attempted TCP connection) as colored dots in a 3D spinning cube. In this perspective, the X and Z axes represent local and global IP addresses, while the Y axis depicts port numbers. Network attacks have obvious visual illustrations; for example, port scans are displayed as linear lines.

- The Analysis Console for Intrusion Databases (ACID) [5] is devised for active analysis of Snort logs. ACID uses a web based interface to present alerts as charts and graphs in HTML. However, administrators must still peruse intrusion alerts in their native text format.

2 Motivation and Visualizations

Currently, there is a need for intuitive and effective network security visualization tools. Most intrusion detection systems and system monitors record alerts and notifications as text logs. Analyzing these logs can be monotonous and time consuming when done by hand. By presenting network security data graphically, visualization tools can reduce the time and burden of reviewing text logs.

Visualization takes advantage of the fact that humans have an outstanding capability to detect patterns and anomalies in the visual representation of abstract data [11]. This technique also transforms the task of analyzing network data from a perceptually serial process to a perceptually parallel process [7]. Consequently, these benefits can greatly reduce the time and effort spent in examining security logs.

The design of VisualFirewall is inspired by the needs of small business and home users to verify their firewall configurations and to passively monitor their network activity. The interface is designed to be clear and simple to use. Four visualizations of the network state are included, each working with the other to convey the multi-dimensionality of the data present on the network. This allows for traffic patterns to be distinguished from each other based different dimensions of the network data.

2.1 Real-Time Traffic View

The Real-Time Traffic view (Figure 1), uses glyphs to represent packets incoming and outgoing from the firewall. Motion is used as a parameter of these glyphs to show both the direction of the traffic and whether or not the traffic was rejected by the firewall. Color-coding was applied to mark streams of packets between the same hosts and size-coding was used to represent the data size of the packet. If a packet was dropped by a firewall, the corresponding glyph bounces off of the port axis to symbolize a packet rejection thus giving a sense of causality. Using motion to represent network activity has the effect of attracting attention when the network is active and being subtle when the network is quiescent. Time scaling is also allowed by altering the speed of the glyphs so that the viewer can see more data, although at the risk of occlusion.

This view shows packets flowing between the firewall (left axis) and foreign hosts (right axis). This view is especially useful for verification of firewall rules because accepted packets flow past the left axis, while rejected packets ricochet. The parallel-axis plot visually correlates the localhost port, foreign host IP address and port, as well as inbound and outbound packets.

For each connection or connection attempt, the localhost port is displayed on the left axis, while the foreign host IP address and port number are displayed on the right axis. This information, along with the associated traffic, is color coded based on the foreign IP address.

The position of the localhost ports on the left axis are defined by the cube root of the port number. We feel that the cube root scale provides a better graphical distribution of relevant ports when compared to the log base 2 scale. The pixel to port number ratio is greater for lower port numbers (especially for ports between 32 and 1024) and less for higher port numbers (ports greater than 1024). That is to say, lower ports are spread out among more pixels than their higher port counterparts.

Ports of interest on the localhost (at the top of the left axis) are visually separated from the rest of the ports in order to provide easy discernment of relevant traffic. These ports are typically used for open services, but could also be used to highlight known worm exploit-vector ports.

A packet is represented by a glyph in the form of circle or square. Circles indicate incoming packets and squares indicate outgoing packets. The size of the glyph is directly proportional to the packet size. The greater the size of the packet, the greater the size of the ball.

UDP traffic is delineated by glyphs with a white border, while TCP traffic has glyphs with no border. ICMP packets are represented by pie charts on the lower right hand side of the screen. The pie charts display ICMP type and code percentages for both incoming and outgoing ICMP traffic. This representation allows for quick analysis of suspicious traffic such as port scans. The pie chart legend is as follows:

- Echo / Echo Reply = red
- Net Unreachable = green
- Host Unreachable = blue
- Protocol Unreachable = yellow

- Port Unreachable = cyan
- Timeout = magenta
- All other types = white

In order to prevent information loss during a large volume of traffic from one or more foreign hosts, glyphs are evenly spaced from one another. In addition, the rate at which glyphs travel can be increased or decreased by pressing the *a* and *s* keys respectively.

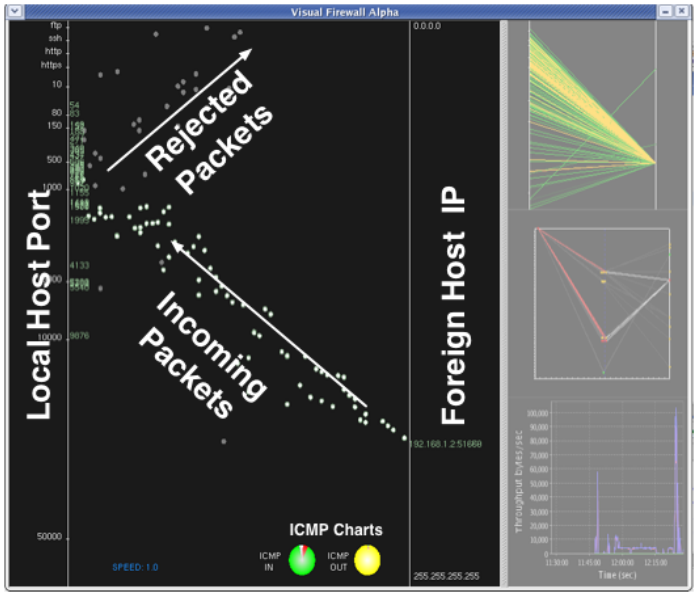

Figure 1: Real-Time Traffic View

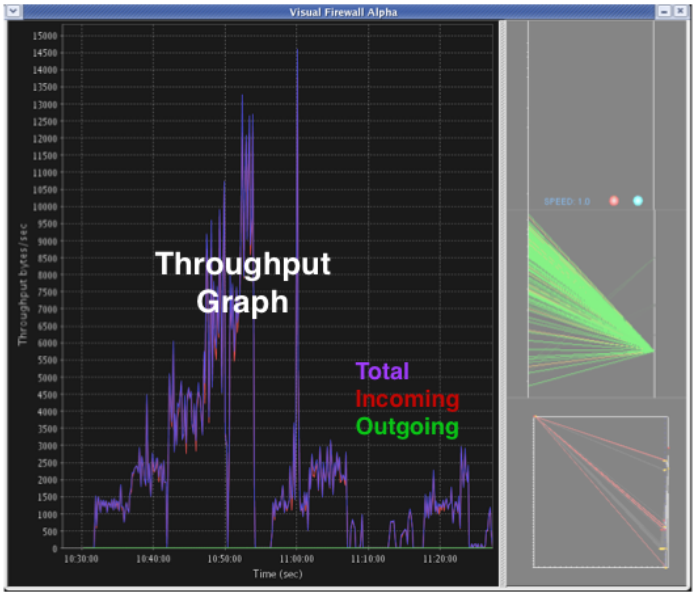

Figure 3: Statistics View

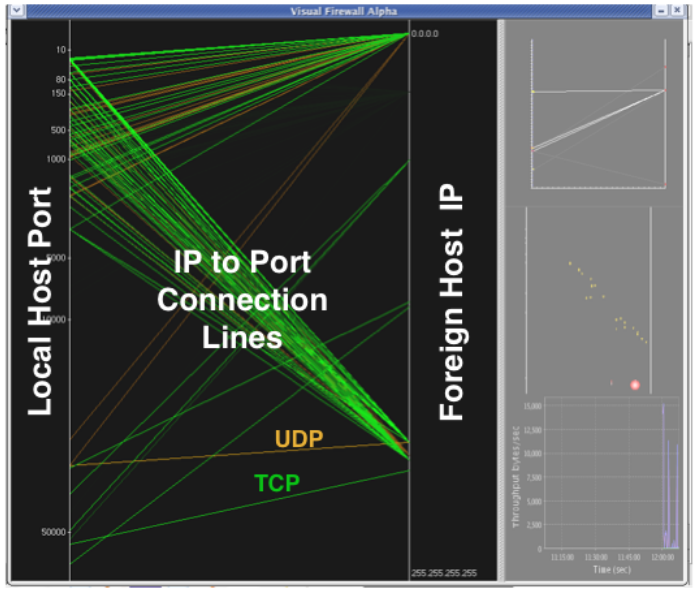

Figure 2: Visual Signature View

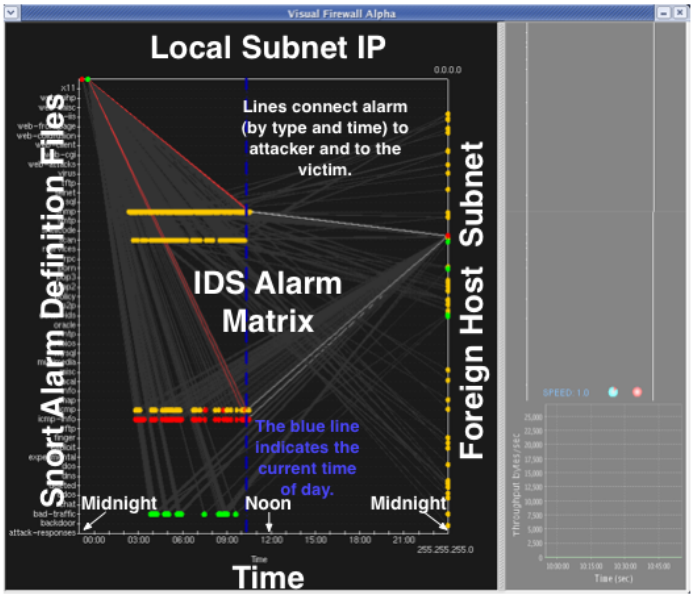

Figure 4: IDS Alarm View

2.2 Visual Signature View

The second visualization, the Visual Signature view (Figure 2), shows packet flows as lines on a parallel axis plot. The right axis

shows the global IP address space, whereas the left axis shows ports on the local machine, using a cube root scale. When packets are exchanged between the local host and a foreign host, a line is drawn from the local port to the foreign IP address. The line color represents the type of transport protocol that is being used. Green lines represent TCP packets and orange lines represent UDP packets. This view is especially helpful in recognizing attacks against the network. Incoming port scans and outgoing ping sweeps are obvious, in that they create unique visual signatures. To reduce confusion, older lines fade out after a prescribed period of time. The faded lines also help to give the user a sense of time; brighter lines correspond to newer packets, whereas dull lines correspond to older packets. Fading lines into the background color in the Visual Signature view uses the brightness-distance relationship to denote that the more transparent the line is, the more distant in time the event occurred. This allows the time variable to be displayed along with the port and IP dimensions of the data. To summarize, the following dimensions are represented in this plot:

- local port on the left axis
- foreign IP address on the right axis
- the protocol (TCP or UDP) by the color of the line
- Age of the packet by the brightness of the line

2.3 Statistics View

The third visualization, the Statistics view (Figure 3), illustrates the overall throughput of the network over time. It dynamically displays the throughput in bytes/sec on a line chart. Against the x-axis of time, network traffic throughput is shown as three lines: overall throughput (purple), incoming throughput (red), and outgoing throughput (green). As the throughput changes, the chart scales automatically. This auto-scaling provides a quick time reference for periods of increased network activity, such as large file transfers, port scans, or DoS attacks. The design of this visualization complements the others by showing the state of the network over a extended duration of time.

2.4 IDS Alarm View

The fourth visualization, the IDS Alarm view (Figure 4), displays IDS alerts in a quad-axis diagram. Colors are used to encode alarm severity and line transparency was used to represent the age of the event where the more faded the line, the older the event. Lines are used to map the multiple dimensions of the data to the local IP axis and the remote IP axis thus mapping the multiple IPs together as another dimensionality of our representation. The left axis lists the different categories of snort rules. The right axis represents all the possible subnets (0.0.0.0 - 255.255.255.0) where attacks originate. The bottom axis displays the time from 00:00 to 23:59. The top axis represents all the hosts on the local machine's subnet. These hosts represent the targets or victims of the triggered IDS alarm. IDS alarms are displayed as colored dots within the four axes. The position of the dot is determined by the rule category of the IDS alarm and the time at which the alarm was raised. A line is drawn from the attacking subnet to the dot and from the dot to the victim machine. To further aid the user in recognizing current alerts there is a constantly sliding, faint blue, vertical axis that indicates the current time. The color of the dots represents the severity of the alert. The possible colors are green, yellow, orange, and red, where green represents alerts with low severity and red represents alerts with extreme severity, as determined by the IDS.

This view is beneficial for quickly determining the types of attacks occurring on the network and the particular local machines affected. This functionality makes reviewing the IDS alert log a perceptually parallel process as opposed to a serial process.

To summarize, the following dimensions are displayed in this graph:

- type of IDS alarm on the left axis
- attacking subnet on the right axis
- time on the bottom axis
- victim machine on the local subnet on the top axis
- severity of the alert by the color of the dot

3 Architecture and Design

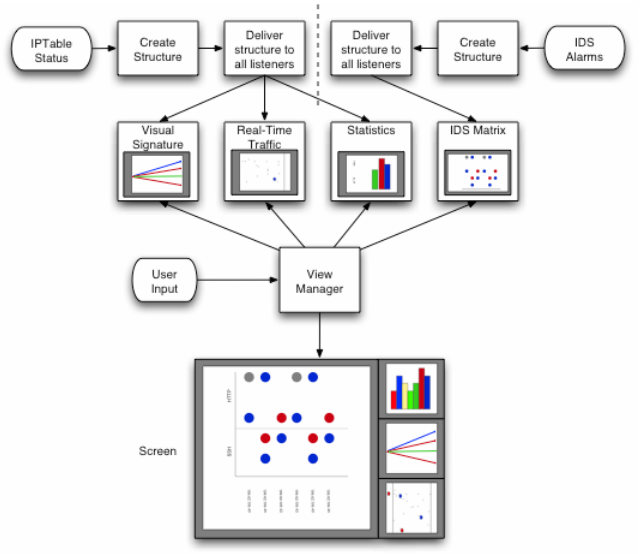

Figure 5: System Architecture

This software is implemented in Java. JOGL (Java bindings for OpenGL) and JFreeChart were used to create the visualizations. Java was chosen in order to make the tool as portable as possible, and allow for the easy addition of modular extensions in future development. Figure 5 shows the basic flow of information from the network data to the visualization.

VisualFirewall uses an event driven architecture based on the Model View Controller (MVC) paradigm. Two data sources, IDS alerts and firewall packet events, are continually updated as network events occur in order to produce Java event objects that represent such network activity. These event objects are created and dispatched to listener objects. The listener objects then use the events to update their internal state accordingly. The View Manager handles user input and maintains a consistent layout for the on-screen windows. The View Manager switches the main and side panel windows by creating a permutation array, swapping entries upon mouse-click, and then redrawing the windows on the screen. This procedure also accomplishes the task of adjusting the positions and the sizes of the views.

We chose Snort as our IDS because of its popularity and ease of installation and use. Custom built parsers handled the reading and translation of Snort logs as well as iptables logs (for Linux) and ipfw logs (for Mac OS X). We configured both iptables and ipfw to log every packet with an accept or deny flag. Since both the firewall and Snort log files can quickly become quite large, we use UNIX

named pipes to have the firewall (through syslog) and snort feed information to our program.

4 Monitoring and Attack Scenarios

To show the effectiveness of the chosen views at quickly describing certain traffic patterns, we took screen shots of the VisualFirewall interface after running several attacks or downloading files. The traffic patterns we present in this paper are: TCP and UDP port scans using Nmap, a simulated UDP worm, a simulated UDP DDoS attack, and a BitTorrent ISO download. These key examples show how the multiple views work collaboratively to convey the nature of the activity and help differentiate similar traffic patterns.

4.1 TCP Port Scan

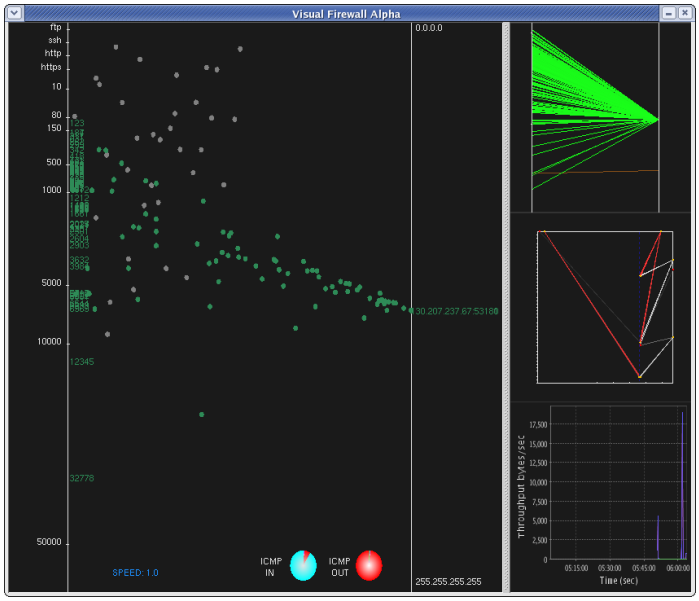

Figure 6: TCP Nmap Scan

In Figure 6, the Real-Time Traffic view shows TCP packets from an attacker hitting various ports on the local firewall. A majority of the packets are being rejected by the firewall (represented by the grey round balls at the angle of reflection) and the Visual Signature view on the right side shows the port scanning pattern that is easily recognized as a Nmap port scan [4]. The Statistics view shows a marked increase in incoming traffic, and thus total traffic. The IDS Alarm view displays the resulting IDS alerts from this attack.

4.2 UDP Port Scan

A UDP port scan will show much the same pattern as the TCP port scan, but will have orange lines in the Visual Signature view to represent UDP traffic as seen in Figure 7. In the Real-Time Traffic view, the packets are surrounded with a white border to represent UDP traffic. The statistics view increases in incoming and total throughput just like in the previous example. The IDS Alarm view shows alerts generated affecting one host and originating from one subnet. This is what is expected from a port scan.

4.3 UDP Worm Attack

We wrote a Perl script that sends shellcode to port 1434 of random hosts all over the entire IP Address space. We ran this script on

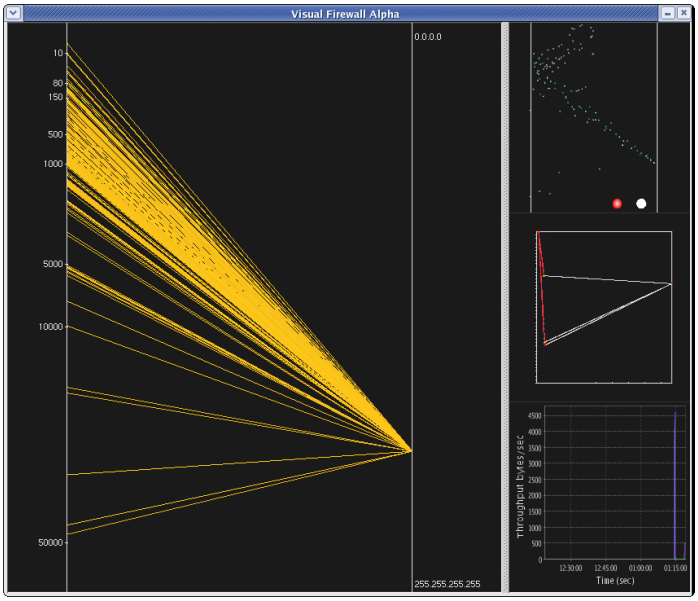

Figure 7: UDP Nmap Scan

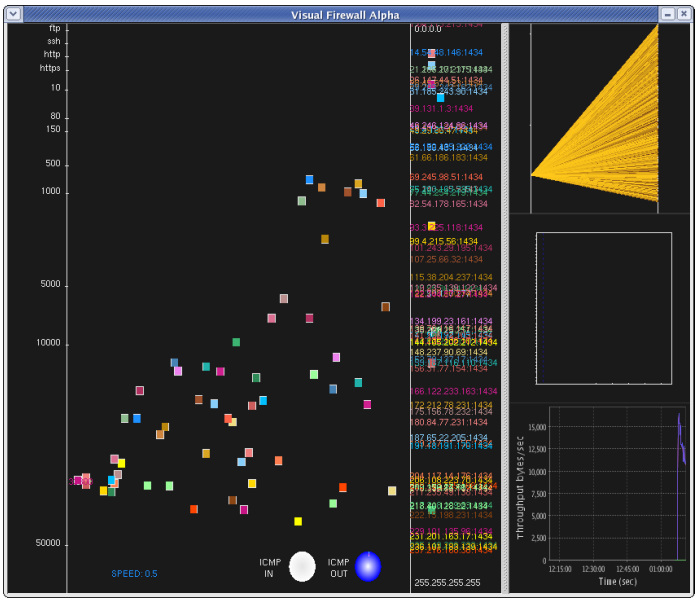

Figure 8: Fictitious UDP Worm Attack

the monitored host (on an isolated network) to simulate the attack pattern of the Slammer worm (Figure 8). The outbound packets are represented by squares moving from the left axis to right axis, so it is easy to tell that this is an outbound attack. Also, the Statistics view shows the outgoing traffic throughput to be very close to the total throughput, further illustrating the outbound nature of the attack. The intrusion detection system was not configured to catch this particular worm; thus there are no IDS alerts displayed in the IDS Alarm view. However, in this case the IDS was not needed to recognize the attack because the Visual Signature and Real-Time Traffic views clearly convey the malicious activity.

4.4 UDP DDoS

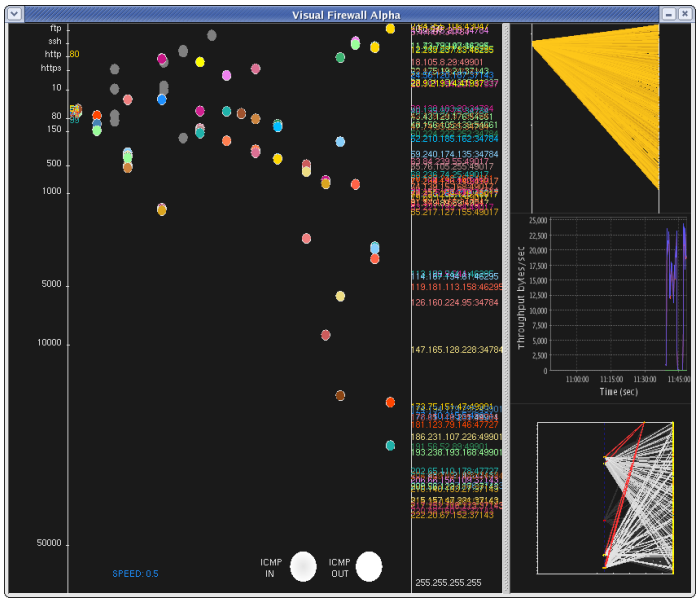

Figure 9: Visual Signature of UDP DDoS

A very similar looking attack in the Visual Signature view is the UDP-based DDoS. The Visual Signature view in Figure 9 has the same cone-like shape as the UDP worm, but the Real-Time Traffic view clearly shows that the traffic is inbound by using round glyphs moving from the right axis to the left axis. Furthermore the Statistics view in Figure 10 shows the incoming traffic throughput to be much higher than expected (in fact close to saturation). In both of these figures it should also be apparent from the IDS Alarm view that there are many IDS alerts originating from many different Internet subnets. This is an example of how the multiple views work together to provide an accurate depiction of the state of the network. If only one of these views were provided, an administrator could mistakenly think that his machine was infected with a worm.

4.5 BitTorrent Traffic

Although BitTorrent traffic is generally not an attack, it is very useful to be able to easily distinguish between it and worm attacks, which could have a similar pattern (multiple external hosts to a few local ports). In this case, the Visual Signature view presents a similar visualization of the two activities. However, the Real-Time view would allow the user to clearly distinguish between the two activities. In addition, the Snort view would present different alarms for the different traffic. Using these three views collaboratively enables the user to quickly discern between these different traffic patterns.

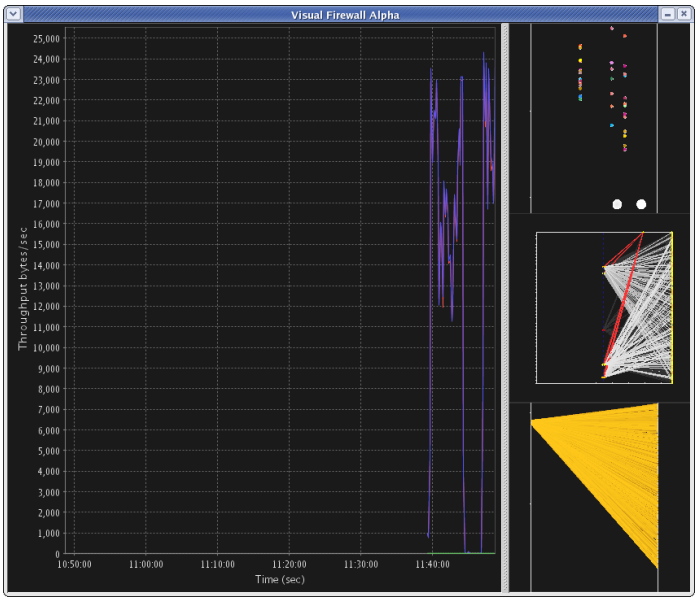

Figure 10: Traffic Statistics of UDP DDoS

This benefit of integrated views, can be utilized to differentiate between similar traffic patterns that vary in some distinct way because one view can provide the key indicators lacking in the other views. For this reason we feel that having multiple simultaneous views makes our tool harder to attack.

In the last set of traffic, a BitTorrent session was started to retrieve a large audio book file. BitTorrent is a file sharing protocol that tries to solve the leeching downloads problem by having downloaders also share with others blocks of the file they have completed [1]. In Figure 11, there are several active flows with large circles (incoming packets) flowing through the firewall and mostly small squares (outgoing packets) flowing out to the destination hosts. There were approximately seven peers actively sharing the ISO and only one peer requesting blocks from our testing host. This led to large packets inbound, a large number of small acknowledgment packets outbound, and a few large data packets going outbound. As the download progresses, there were more peers requesting completed blocks and thus a greater number of maximally sized outbound packets. After the entire download is finished (Figure 12), all the BitTorrent traffic has maximally sized outbound packets and only acknowledgment-sized inbound packets. The Visual Signature view shows lines for the connections with each peer, but does not give a representation of the amount of traffic over each line. This information is available on the Real-Time Traffic view and the Statistics view. In the Statistics view, there is a distinct spike in throughput during the BitTorrent session. The IDS was configured to flag P2P traffic, and the IDS Alarm view shows this with several low severity alerts being raised. Visualizing BitTorrent traffic can allow network administrators to enforce a no file-sharing policy.

5 Conclusion

VisualFirewall is a unique tool for monitoring firewall operation, IDS alarms, and overall network security. Each of the four separate views provides specific details about network traffic, packet flow, throughput and suspicious activity. The four perspectives combine to form one coherent illustration of the network state. The value of VisualFirewall is clear not only to experienced administrators but also to novice users. An administrator can immediately grasp the state of the network without having sift through several text

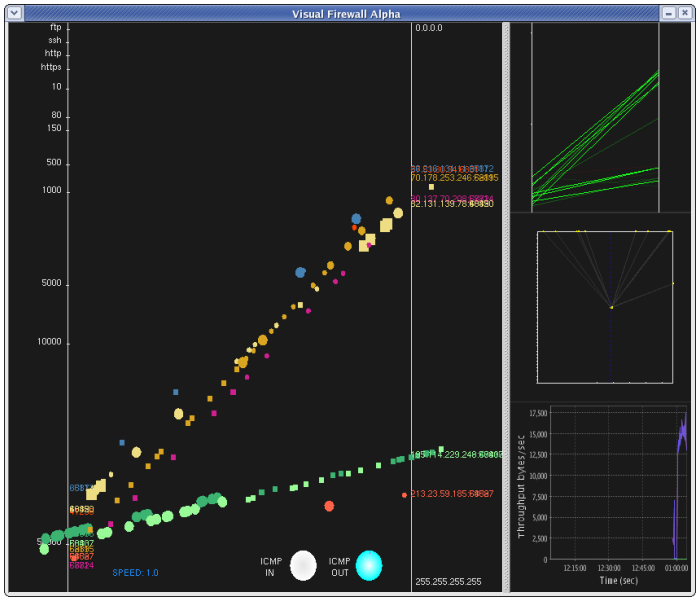

Figure 11: A typical BitTorrent session in the download only stage

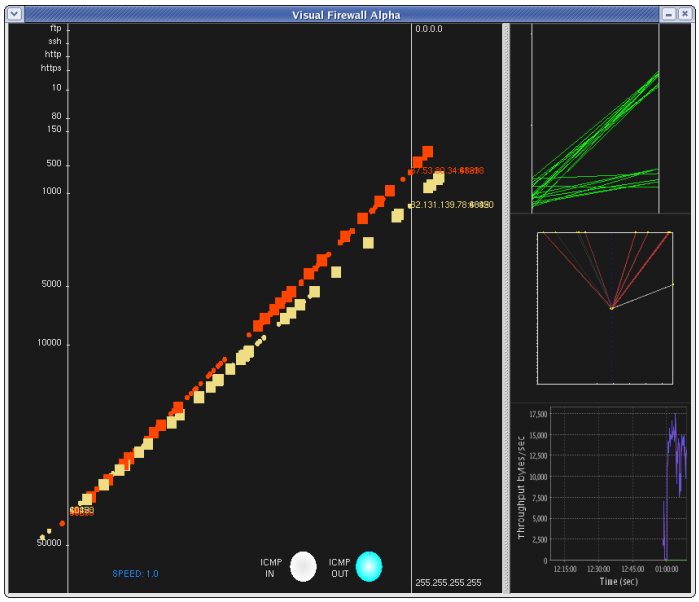

Figure 12: A typical BitTorrent session in the download only stage

logs. With minimal training, a novice user will be able to easily distinguish normal from abnormal traffic.

6 Future Work

The major shortcoming of our current tool is scalability regarding large networks. For larger networks we plan to visualize flows and aggregate IDS alarms from various sensors. For the Real-Time Traffic view, instead of presenting each packet as a ball, flows would be used to signal the creation of new connections. Another axis would be added on the left side to denote the internal network. When a new flow is created, a ball with a flow number would travel from the originating host, through the firewall (if accepted), and on to the destination host. Meters would be used on both the left-most and right-most axes to represent the amount of traffic relative to the corresponding hosts. Likewise, the Visual Signature view would use flow information to draw the lines; the thickness of the lines would represent the amount of data transferred between the firewall and the external host. The Statistics view would also have extra lines plotted for accepted and denied packet throughputs.

The second most important enhancement is to use the current views in a forensics mode that can replay firewall and Snort logs. A forensics mode would allow an administrator to review previous traffic. The administrator could interactively select a time point to begin replaying and then examine the network activity again in various speeds: normal, slower than real-time, or faster than real-time.

Thirdly, the IDS Alarm and Real-Time Traffic views should allow for filtering and zooming. Filtering in the IDS Alarm view would be accomplished by selecting a severity level, one through five, allowing only the selected severity levels to be displayed. Filtering in the Real-Time Traffic view would be based on packet type and packet size, allowing known "good" traffic to be removed during real-time or forensic analysis. Zooming would use a fish-eye styled zoom technique in order to focus in and pan across the alarms. Additional information would be made available when a user clicks on an alarm, permitting them to see all the information relating to that alarm.

Next, further user tests with users of varying networking expertise need to be conducted to identify how well they understand network activities by using the VisualFirewall tool. Specifically, we want to show that the use of coordinated views (the four presentations chosen) help users to quickly identify normal versus malignant traffic patterns. The test results may indicate a need to change the user interface with the addition of other widgets, such as dialog boxes, legends, or tool tips.

Finally, we plan to integrate the direct control of firewall rules with the VisualFirewall interface. The user would therefore be able to dynamically, through our interface, open and close ports on the firewall, kill ongoing flows, and block external IP addresses.

7 Acknowledgments

We offer our gratitude to those who have made this research possible and enjoyable. First, we would like to give thanks to Greg Conti for his guidance and patience. We also thank Peter Wan, who shared his practical experience as a senior network security administrator. Lastly, we thank Kulsoom Abdullah for her help with visualization techniques and countless reviews.

References

[1] The official bittorrent home page. BitTorrent, May 2005. `http://www.bittorrent.com/`.

[2] Survival time history. SANS Internet Storm Center, June 2005. `http://isc.sans.org/survivalhistory.php`.

[3] Robert Ball, Glenn A. Fink, and Chris North. Home-centric visualization of network traffic for security administration. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 55–64, New York, NY, USA, 2004. ACM Press.

[4] Gregory Conti and Kulsoom Abdullah. 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*, pages 45–54, New York, NY, USA, 2004. ACM Press.

[5] R. Danyliw. Analysis console for intrusion databases (acid).

[6] R. Erbacher. Intrusion behavior detection through visualization. In *Proceedings of the IEEE Systems, Man and Cybernetics Conference*, Crystal City, Virginia, October 2003.

[7] R. Erbacher, K. Walker, and D. Frincke. Intrusion and misuse detection in large-scale systems. *Computer Graphics and Applications*, 22(1):38–48, January/February 2002.

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

[9] Sven Krasser, Greg Conti, Julian Grizzard, Jeffrey Gribshaw, and Henry Owen. Real-time and forensic network data analysis using animated and coordinated visualization. In *2005 IEEE Workshop on Information Assurance*. IEEE Press, 2005.

[10] Stephen Lau. The spinning cube of potential doom. *Commun. ACM*, 47(6):25–26, 2004.

[11] Kwan-Liu Ma. Visualization for security. *SIGGRAPH Comput. Graph.*, 38(4):4–6, 2004.

[12] Jonathan McPherson, Kwan-Liu Ma, Paul Krystosk, Tony Bartoletti, and Marvin Christensen. Portvis: a tool for port-based detection of security events. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 73–81, New York, NY, USA, 2004. ACM Press.

[13] T. Takada and H. 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.