Massachusetts Institute of Technology
Department of Electrical Engineering
and Computer Science

Master of Engineering Thesis Proposal

Title: FireViz: A Personal Network Firewall Visualizing Tool

Submitted by: Nidhi Sharma
143 Albany Street, 134A
Cambridge, MA 02139

Date of Submission: December 13, 2004
Expected Date of Completion: May, 2005
Laboratory: Computer Science and Artificial Intelligence Laboratory

Abstract:
This proposal outlines the development of a personal firewall visualizing tool called FireViz. FireViz leverages the human perceptual capabilities by employing various visual analytic techniques to depict network activity in real time. FireViz creates network usage profiles to depict anomalous behavior so that users may update their firewall security policies to provide greater computer security. The primary goal of FireViz is to educate typical computer users of the security threats their computers face, when connected to a network, through the use of effective visual cues.
Contents

1 Introduction 3
   1.1 The Need for User Education 3
   1.2 The Need for Visualization and Usability Engineering 4
   1.3 FireViz 4

2 Related Work 5
   2.1 Network Traffic Analysis 6
   2.2 Network Traffic Visualization 6

3 Design 7
   3.1 Design Overview 7
   3.2 Usability Engineering 10

4 Implementation 11
   4.1 Overview 11
   4.2 Evaluation 11

5 Schedule 11

6 Conclusion 12
1 Introduction

The Internet is playing an increasingly important role in business, education and communication. While the Internet is a powerful means to establish connections to other remote hosts and effectively share useful data, it also serves as a medium to quickly and widely spread malicious data. Any host connected to a network is liable to be compromised by various means. Given this hostile internet environment, it is imperative that every computer connected to the internet be protected by appropriate means. While organizations have a greater incentive in securing their hosts and networks, ordinary home users are generally unaware of both the risks and the measures for preventing attacks. However, few personal security tools focus on educating users about the threats they are exposed to.

This proposal describes a personal network visualization tool called FireViz. FireViz is designed to provide real-time visualization about the network processes on a given host and reveal potential threats and holes in the firewall’s security policies.

1.1 The Need for User Education

Almost every computer connected to a network is constantly scanned for various security vulnerabilities. Worms and viruses are self-replicating programs that scan the network for vulnerable hosts and infect them. However, unlike viruses, worms need no user initiation, such as opening a file, and hence spread very quickly and easily. Attacks launched by viruses and worms, among others, quickly compromise the integrity of the affected host and spread to many others. Over the last few years, both the intensity and frequency of such network-based attacks have increased rapidly. By the end of the year 2004 alone, internet users will have been confronted by an estimated 100,000 forms of malicious attacks [1]. Not just the frequency and intensity of these attacks alone but even the shrinking time lag between when the vulnerabilities are announced and are exploited is a cause for major concern. A study of network worms launched over the last 24 months shows the time lag shrinking from 330 days for the Nimda Worm in 2002 [2, 3], to 16 days for the Sasser in April 2004 [4, 5]. According to the Gartner research [6], this pattern will only get worse. The study projects a 25% increase over the next several years in 'Zero Day' attacks, which exploit software vulnerabilities that have no known fixes [7]. Given these statistics, network users have to stay current with the existing security mechanisms that are already struggling to keep up with the sophistication of attackers today. As a result, the tolerance for any laxity in maintaining computer security is diminishing quickly. Statistics report that the average survival time for an unprotected computer has fallen from 40 minutes in 2003 to a mere 20 minutes in 2004 [8].

Many commercial developers are promoting tools to monitor and protect individual computers. Personal network firewalls such as Zone Alarm [9], Kerio [10] and Sygate [11] succeed in detecting and blocking numerous unfriendly network probes. However, few personal security applications focus on informing the user of the extent or nature of these threats. This is particularly evident when examining the user interfaces of these firewalls. Zone Alarm, for example, allows users to grant internet access to programs on application granularity. Once applications are deemed as trusted, the user is not given any further feedback on their network activity. This eliminates all feedback for users about potential attacks launched through these trusted applications. Zone Alarm does, however, provide users with information on port scanning activities. However, this information is provided to the users in very intrusive ways. As a result, many users turn off feedback from port scans, further reducing the information they may receive.

Security is a complicated and an important aspect of using computers today. It is therefore essential for all users to understand the possible vulnerabilities their computers are exposed to.
Given current trends, awareness is irreplaceable for survival. To this end, many network security applications have singularly failed to educate the very users they hope to protect. This lack of awareness also provides negative feedback by reducing users’ motivation to run personal security software at all times. It is no surprise then that a majority of internet users today fail to appreciate the reality of the security threats they are exposed to. In a study conducted by the NCSA, more than a third of the users surveyed said that they had a greater chance of winning the lottery than being hit by malicious code [12]. This is not just characteristic of novice users. Many sophisticated computer users are also unaware of the sheer volume of such threats. At the SC03 conference in 2003, many expert computer users were surprised at the number of malicious attempts at the conference’s high bandwidth network [13]. All these statistics suggest that there is immediate need for computer users to be educated of potential threats to their computers.

1.2 The Need for Visualization and Usability Engineering

While many personal security applications do not provide any feedback or visual cues about network activity, they do maintain activity logs to a certain extent. These logs can, in principle, be used to find unexpected or anomalous network behavior that may be potentially harmful. The analysis of these logs can either be automated or conducted manually by an expert user. Automatic analysis is based on statistical modeling and machine learning. However such data mining techniques minimally engage human interaction and visualization and are likely to miss important features in the data. Manual inspection on the other hand is extremely tedious and both approaches are unreasonable to be expected from typical network users. Most importantly, these logs lack the situational context about what the user was doing at the time of the network activity. Such situational context is often more important than the nominal connection information to detect anomalous or unexpected activity.

The human perceptual processor is capable of very fast visual processing; therefore, information depicted visually is easier to process and make out interesting patterns and features [14]. Consequently, tools that use visualization can leverage this human capability to enable users to easily discern anomalous network events. FireViz is a personal network firewall analyzer that uses information visualization to take advantage of the huge bandwidth of the human vision sensory system. It provides quick, real-time network information for users and uses visual cues to aid in detecting anomalous activities. The use of visual cues is especially useful when the users are unaware of what information to particularly look for, which is typical of network activity monitoring. FireViz works in conjunction with the users’ personal firewall and provides a more intuitive window to the network on which the firewall acts.

1.3 FireViz

FireViz is an attempt at solving both these problems - educating users about network threats and doing so effectively. FireViz provides a peripheral window that runs alongside a personal network firewall and on which all activity is displayed. It detects all network activity in real-time and visually displays the information highlighting the most crucial pieces (such as host application, remote location and the ‘expectedness’ (the notion of expectedness is explained in detail in the design overview section of this proposal) of that connection), while allowing the user to analyze any subset of the events more closely. This way, FireViz creates a model of the network weather and behavior for the user.

FireViz is based on a simple philosophy - You cannot protect when you can’t see. FireViz is used in conjunction with tools that enforce security policies - such as personal firewalls. However, no
security system is perfect and hence there is constant need to update not just the policy-enforcing tools but also the policies themselves. However, users cannot be expected to update their security policies unless they know what the security policies should be. Furthermore, users cannot know what the security policies should be unless they have an understanding of which aspects of network behavior are expected and which ones are potential threats. Additionally, most users, when using the network, do not have security as a primary goal. This insight forms the basis of FireViz’s emphasis on the visualization of network security at a low cost to the user.

FireViz is designed to be useful to even novice computer users. It aims to achieve the following main goals:

- **Educate users about their network conditions.** The primary goal of FireViz is to educate users to keep up their defense mechanisms at all times by providing them with a more concrete network security model.

- **Provide users with immediate feedback on all network activity.** This is to allow users to gain a deeper understanding of the network on which their computer is connected and develop a model of expected behavior. Such model is complemented by the situational context around each network activity. Such knowledge is helpful in detecting potentially malicious behavior.

- **Employ effective visual techniques to display important information.** This will allow even novice users to easily discern exceptional events and examine them more closely. This goal translates into the following subgoals:
  
  - Since network activity is very frequent, all information must be provided in the most non-intrusive fashion as possible.
  
  - The cost on the user to retrieve this information should be minimal. This means that costly operations such as a complete context switch of users’ attention and current application should be avoided.
  
  - Since displaying all relevant information related to a specific connection is infeasible (since displaying more information also requires more space), the most easily available information should highlight only the most interesting features of the connection. All other information should be easily accessible if the user chooses to retrieve it.
  
  - The visualization should be selectively biased towards exceptional events or anomalies instead of routine, safe actions.
  
  - The visualization should effectively summarize the network weather for a certain time window. This will allow a user to get an overview of the network conditions even if she has been away from the computer.

My thesis researches a novel way to visualize network activity. I believe that more effective visualization techniques will help achieve increased network awareness for users at a low learning cost and improved system security.

2 Related Work

FireViz’s design objectives intersect two important research areas - network traffic analysis and network traffic visualization, both of which have a substantial body of current research.
2.1 Network Traffic Analysis

Many users secure their computers using software mechanisms such as anti-viruses and personal network firewalls. Commercial firewalls such as ZoneAlarm [9], Kerio [10], Sygate [11] and Norton [15] make sure that the doors providing entry to the computer are not left wide open. Such doors may include vulnerable or buggy applications running on the computer that listen for network data. Firewalls isolate the host computers by intercepting each packet of data, incoming or outgoing, and selectively allowing some packets to continue, based on the security policies. The challenge for firewalls is to maintain accessibility while maintaining security. FireViz relies on both the system firewall and its own network scanning module to monitor network activity on the host it is running.

Tools such as Netstat [16] and TCPView [17] provide the state of network activity on the host at any given point. FireViz uses such data for profiling expected network behavior and expose any anomalies as potential holes in the firewall’s security policy. However such tools may not be used to display activity that may already have been blocked by the firewall.

FireViz maps the expectedness of allowed network accesses over time by monitoring observable features of network traffic such as the frequency of such accesses and the level of trust the firewall has for the access. By creating this model of expectedness, FireViz attempts to create a profile of what constitutes typical network activity. This profile is then compared to each new activity and the user can easily discern any deviation from the regular profile. This idea is similar to Anomaly detection systems (ADS) and Intrusion Detection Systems (IDS) such as Symantec Advantage [18] and Cisco Trellis [19]. Anderson et al. [20] describe such ADS systems in greater detail. However, FireViz uses network firewall data. Firewalls often merely implement security rules and do little intrusion detection. Thus, unlike visualization tools that are based on IDS data [13], FireViz’s profiles are constructed from data with no false positives.

The various ADSs and tools such as NetFlows [21] record information on unidirectional end-to-end transactions, aggregating packets into larger flows of data. However these tools best operate on whole network systems rather than focusing on individual hosts on the network. Erbacher [22] describes visualizations of collections of individual transactions on a single machine. A similar tool, NVisionIP [23] spans multiple levels of network abstractions including the entire network, a subnet or a single machine. As evident from this sampling, various tools exist to monitor network traffic on various levels of abstraction.

2.2 Network Traffic Visualization

The use of information visualization to display network traffic is an idea that is being widely experimented with. Information visualization mechanisms such as parallel coordinates and self-organizing maps [24, 25], have been specifically designed for this purpose. PortVis [26], which uses port-based detection of security activities, uses a visualization system that depicts the network traffic by choosing axes for important features of the connection data and creating cells in a grid which represent the network activity at that point. SeeNet [27] uses a colored grid, where each point represents the amount of traffic between the hosts represented by the x and y coordinates of the point. NVisionIP [28] uses a graphical matrix representation to show relationships between events on a network. VisFlowConnect [29] uses a parallel axes representation to display network traffic both within and between domains. However, all of these tools are designed to facilitate anomaly detection in whole network systems intended to be used by expert system administrators. FireViz on the other hand, is intended to be used by any users, regardless of their proficiency level and without any special training.

The Spinning Cube of Potential Doom [?] provides an animated display of network traffic within a 3D cube that users can spin at will. The cube is intended to be used by novice users as well. However,
unlike the cube and other network visualization tools, FireViz is a real-time display - it provides a display of the activity as it happens, without the user having to explicitly request the information or switch context to the visualization tool. Additionally, many of these tools provide network level information and finding host-specific information is relatively harder. Network Eye [30] attempts to provide a more balanced Host/Network picture by preserving the context when displaying the whole network at once and showing the interactions within the hosts and their programs. It is nevertheless still meant to be used by network administrators to detect potential threats.

One of the goals of FireViz is to provide the user with the situational context along with network traffic. This goal has inspired us to provide animated logging features that basically allow the user to replay the moment of the said activity. VisFlowConnect [29] employs animation to replay events recorded in the data logs as they occurred. Teoh et al. [31] describe a focus + context radial layout to manage screen real estate by showing snapshots of activity in adjacent time periods in a circle around a larger focal image. Such a layout could be used to display a specific feature of network activity during consecutive time periods.

3 Design

This section describes the design of FireViz.

3.1 Design Overview

FireViz aims to achieve the above goals by employing a careful design. FireViz uses a number of visual variables [32] and redundant cues to meet the design requirements. To make the interface non intrusive, most information is displayed by transient elements such as labels and lines, instead of modal dialog boxes mandating user input. Furthermore, these elements are drawn on the user’s desktop instead of a separate FireViz application window. Each activity element is accompanied by a small label that shows the application that caused/received the activity, the remote host and a subjective frequency measure (such as ”First Access”, ”Rare”, ”Occasional” or ”Frequent”) [33]. This label appears for a few seconds, around the periphery of the screen, so to receive this information, the user need only to glance at the spot where the label appears. Also, the transient elements are replaced by a small lump at the same location. The lumps remain on the location for a few hours and can be clicked at any time to retrieve more detailed information. Additional cues such as the shape and color of the activity gives information about whether the connection involved the current process or a background process and whether the connection was successful.
Figure 1 presents a snapshot of FireViz detecting network activity from Internet Explorer. As soon as the connection was detected, FireViz displayed a line from the application window to a label along the periphery of the screen. The label itself contains the hostname of the remote host and a subjective connection frequency measure ('Rare' in this case). This display stays on the users screen for all of five minutes and is replaced by the green lump alone. The green color signifies that the connection was allowed by the firewall. The green lump serves as a vestige to the said connection and can be clicked on to receive further information as shown in Figure 2.
The transient display in Figure 1 very compactly encapsulates the remote host, the connection frequency, the firewall action (Allowed or Blocked) and the application that made the connection. Thus FireViz takes advantage of the situational context to compactly highlight the most interesting features of the connection. In case the application window was hidden or non-existent, FireViz would instead have a transient, flashing green circle along with the application icon.

Another important aspect of the visualization in FireViz is the network mapping around the periphery of the user’s screen. FireViz creates an image of the network traffic involving the host by mapping the level of expectedness along the screen periphery. The expectedness is measured in terms of the frequency of the connection and the trust policy of the firewall (whether or not the connection is deemed "trusted" by the firewall). Each attempted (successful or unsuccessful, incoming or outgoing) connection to the machine is depicted, as described above, at the point which correlates with the expectation of that connection. As a certain network action is repeated multiple times, its expectation increases with the frequency. However, the most expected connections are those that are frequently made by trusted applications. Thus mere frequency does not make connections completely expected.

An expectation-based profile, such as the one described above, in combination with some situational context can help users identify extraordinary events. For example, if a user checks her email, the connection to the mail server will likely be expected (since not only would the email client be a trusted application, email is also checked frequently). However, if one of the email messages had a web bug embedded in it which made a connection to some unidentified server, this connection will be unexpected (since the frequency of connections to the unidentified server from the user’s computer will likely be low, even though the mail client is a trusted application) and can successfully draw the user’s attention. On closer analysis of this connection, the user may realize that the connection was indeed unexpected and unsafe and consequently change the security policy in her firewall to disallow connection that said remote host.

The location of the label along the periphery of the screen is dictated by the expectation of the connection. The expectation is a quantitative value which is a function of the frequency of the
connection and the trust associated to it by the firewall. The expectation increases from the lower right corner of the screen to the top left corner of the screen in a continuous fashion. This mapping is translated into four nominal expectation values which are displayed in the labels. Thus if an unexpected network event occurred while the user was working on trusted applications, she could easily make out the unexpected event.

As FireViz collects more data about the system’s network patterns, expected (i.e., trusted and frequent) activity slowly starts to fade away until it is no longer visible. This allows the user to focus her attention only on more interesting events, such as an untrusted application (e.g. spyware) making lots of connections.

Finally, FireViz will employ an animated logging scheme to provide situational context in its logs. It will store not just the connection state for each network activity but also a screenshot of the computer at the time of the said activity. Thus, a user viewing the log can easily perceive (in the above example) that the mail client made a connection to the unknown server while checking a certain email message. No information is lost in the logs and hence the user can still make the appropriate security policy changes.

An important consideration in the design of FireViz is to make the interface non-intrusive and helpful at the same time. Given the scenario where users do not have security as their primary goal and are unsure of what to look for, the interface has to be extremely easy to learn and use. Thus FireViz uses a simple expectation-based mapping that is easy to learn and provides an easy means to retrieve information that the user chooses. The following section describes the usability engineering process of FireViz in greater detail.

### 3.2 Usability Engineering

The development process of the user interface for FireViz is following the spiral model [34] of iterative design [35]. As a first step, I conducted a user analysis and task analysis [36] to establish the user characteristics and identify the most commonly performed tasks. This stage involved informal interviews with various people in the target user population to assess their familiarity with personal firewalls and understand their security priorities. I identified two broad categories of users - novice computer users and expert computer users. FireViz is designed specifically keeping in mind the capabilities and concerns of novice users. This decision influenced many design features of FireViz. For instance, this lead us to incorporate minimal or no initial setup and configuration, carefully speaking the users’ language where appropriate and to provide an easily learnable profiling algorithm.

After conducting the user analysis, I established some crucial tasks that the users would perform through FireViz. Once the tasks were identified, I started brainstorming about the user interface keeping in mind the design requirements outlined above. The implementation process started with developing low fidelity prototypes such as design sketches and paper prototypes and performing user tests. The next step involved the development of a horizontal computer prototype that incorporated the feedback from the user tests of the previous prototypes. The computer prototype was then heuristically evaluated by four users. Heuristic evaluation [37] revealed several issues with the computer prototype that were then rectified in the successive computer prototype. The next step is to user test the latest computer prototype while continuing to add more functionality on the backend. Finally, a fully operational version of FireViz will be distributed to users to run and test for extended periods of time to assess each aspect of FireViz’ user interface. Users will be asked to evaluate this version of FireViz on various dimensions of usability such as learnability, memorability and subjective satisfaction both through quantitative measures (such as time taken to perform a task) and qualitative feedback.
4 Implementation

4.1 Overview

FireViz is implemented using the C# .NET technology. It is designed to work on the Windows XP operating system with a personal firewall. The choice of firewall will depend on the availability of an exported API to extract the requisite information, or at the minimum the source for the firewall. Certain open source firewalls for Windows, such as NetDefender, Fiseclab and Agnitum Outpost are available on the internet.

The implementation of FireViz is structured around various independent modules that interact with each other. The following is a high level listing of FireViz’s modules -

- **Network Monitoring Module**
  This module is responsible for detecting network activities. This is done in two parts. An independent thread scans for newly established connections and another scans the firewall logs for any blocked accesses. The network module detects new activities and notifies the View module and the logging module for further processing.

- **Logging Module**
  This module maintains long term state on all network connections detected by the network module. The Logging module uses this information to calculate the expectedness of the connection.

- **Display Attributes Module**
  This module uses data from the logging module and the firewall to calculate the display attributes for the view module. Display attributes such as the location and intensity of visible controls are calculated. Factors such as high frequency and firewall trust are considered here to produce attributes that highlight extraordinary behavior. For instance, a new, unexpected connection would be more prominent than a more expected connection.

- **View Module**
  The view module receives information from the network module and the attribute module to show the appropriate transient activity on the user’s screen. It also provides means for user input. It is also responsible for placing an application icon for FireViz in the system tray.

4.2 Evaluation

The backend for FireViz will be tested to ensure that it works accurately. The detection module will be tested to make sure that detected connections match the results from other similar tools such as TCPView [17] and that all firewall logged events are detected as well. The logging and attribute modules can be tested by checking that connection state is persisted and restored correctly across multiple sessions of running FireViz. Finally, the user interface will also be evaluated through user tests and heuristic evaluations, as outlined in the usability section of this paper.

5 Schedule

The following proposed schedule outlines the major milestones in development of FireViz, including design and evaluation iterations and technology transfer.

- **September 2004** - Brainstorming ideas and collecting related research; user analysis
October 2004 - Low fidelity paper prototype testing
November-December 2004 - Development and testing of hi-fidelity prototypes
January-February 2005 - Second iteration of fully functional FireViz version
March 2005 - Extensive user testing
April-May 2005 - Documentation, technology transfer and publication of research results

6 Conclusion

Even with the presence of personal security mechanisms such as personal firewalls, many malicious activities go unnoticed and unabated. Using existing knowledge about every network connection combined with a network behavior profile and proper use of visual cues, it is easy for even unsophisticated users to detect potentially malicious network activity. This proposal describes the design of a network firewall visualizing tool that can be used by typical network users to gain a deeper understanding of security vulnerabilities and leverage their protection mechanisms to gain a much higher level of security.

References

[1] Study: Consumers take cyberattacks lightly
[2] Nimda worm strikes Net, email
   http://securityresponse.symantec.com/avcenter/venc/data/w32.nimda.a@mm.html
   http://www.microsoft.com/security/incident/sasser.mspx
   http://securityresponse.symantec.com/avcenter/venc/data/w32.sasser.worm.html
   http://www3.gartner.com/research/focus_areas/asset_48267.jsp
#Research
[7] InfoSec 2003: 'Zero-day' attacks seen as growing threat
   http://www.computerworld.com/securitytopics/security/story/0,10801,88109,00.html?SKC=news88109
[8] Study: Unpatched PCs compromised in 20 minutes
[9] ZoneAlarm Security Suite
  http://www.zonelabs.com

[10] Kerio Personal Firewall
  http://www.kerio.com/us/kpf_home.html


[12] AOL/NCSA Online Safety Study


[15] Norton Personal Firewall
  http://www.symantec.com/sabu/nis/npf/

[16] Microsoft Windows XP - Netstat

[17] Sysinternals TCPView
  http://www.sysinternals.com/ntw2k/source/tcpview.shtml


[19] Cisco Trellis IDS


[23] W. Yurcik, J. Barlow, K. Lakkaraju, and J. Rosendale. A prototype tool for visual data mining of network traffic for intrusion detection. 3rd *IEEE International Conference on Data Mining (ICDM), Workshop on Data Mining for Computer Security (DMSEC)*, 2003


