2007

Designing an interactive visualization for intrusion detection systems with video game theory and technology

Shane William Paustian
Iowa State University

Recommended Citation

Paustian, Shane William, "Designing an interactive visualization for intrusion detection systems with video game theory and technology" (2007). *Retrospective Theses and Dissertations*. 15020.
https://lib.dr.iastate.edu/rtd/15020

Designing an interactive visualization for intrusion detection systems with video game theory and technology

by

Shane William Paustian

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Computer Engineering, Information Assurance

Program of Study Committee:
Doug Jacobson, Major Professor
Thomas Daniels
Steve Herrnstadt

Iowa State University

Ames, Iowa

2007

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

With an ever increasing number of attacks on networks that have an even more increasing amount of information being communicated across them, the old means of examining network data for intruders and malicious acts through text no longer works. Even with the help of filters and data aggregation there is too much for a person to read through and get a clear understanding of what is happen across a network, causing security officers to many times miss intrusions. With an overwhelming amount of false alerts from incorrectly setup Intrusion Detection Systems and not enough time to sift through them all, a new means of displaying and interacting with the network data presented by intrusion detection system is needed. That is why there has been an increase in research about how to create visualizations for networks that will allow someone to better understand what is happening across a network. Using previous research as well as a study of the theory and architecture used by the video game industry on interactive environments, it is possible to create an intuitive interactive visual environment of network data that will help network administrators more effectively understand their networks and where potential threats may lurk. Therefore, this proposed design attempts to help solve the problem of network communication comprehension.

CHAPTER 1. GENERAL INTRODUCTION

There is a growing problem in the technical community. With networks becoming larger and faster the amount of data that has to be analyzed for potential threats keeps growing and Intrusion Detection Systems are becoming harder and harder to use and set up properly. Regardless of training or experience, administrators must be able to rapidly understand the security state of their systems and networks, especially during a crisis. [17] This is why current research has focused on how to visualize network data in a way that can be easily understood. Graphical or visual presentations can not only describe data in different ways, but can also facilitate the comparison between different sets of data, stimulate scientific innovation, and even encourage theoretical insights.[13] With that in mind, the design proposed here means to use lessons from past works in the visual representation of network data as well as the ideas for interactive environments and visualization from the video gaming industry for a possible solution to the problem.

Problem Statement

According to Robert Erbacher, “the current network analysis problem is with too much information and inefficient analysis techniques.”[18] He is absolutely correct, current network Administrators and Network Security Personnel are still using text based analysis tools for their networks. Even with a Graphical User Interfaces for filtering, aggregating, and searching data logs, there is too much information to handle. This causes a great many problems when it comes to trying to recognize and compensate for an attack on one's network. Having so much information is meant to help experts find and prevent attackers form successfully intruding, but instead it causes most of the data to become almost useless to anyone reading it because it is absolutely overwhelming, making it impossible to find the patterns needed to make sense of what is happening. This prompted researchers to begin testing different ways of creating visualizations for network data. These visualizations are meant to allow a person to watch what a network is doing through the visualizations and try to detect patterns and/or tell tale signs of an attack or network anomaly.

This presents a whole new problem all together. These visualizations now need to be done in such a way that the user can intuitively understand the basics of how to use it, learn from what is seen, and understand what is happening across the network in order to properly ascertain the situation. With this there are four important problems that must be considered: (1) Positioning nodes, (2) Managing the links so they convey actual information, (3) Handling the scale of graphs with thousands or millions of nodes, and (4) Interacting with and navigating through large networks of information.[13] This requires a robust adaptive system that can keep up with the advancements in technology as well as the ever changing and evolving characteristics of any network. Therefore, what is needed is an interactive environment where the user can learn from different sets of queues while the system can learn from the user's choices and be able to adapt to improvements in technology.

Hypothesis

In approaching the problem of how to effectively create an interactive environment that can effectively demonstrate what is happening across a network with different levels of complexity in such a way that a user can learn from the environment and the environment can learn from the user there is one field that has not been entirely considered, the field of computer gaming. Computer gaming environments have been evolving graphically, audibly, and interactively for years. This entire field thrives off of the successful implementation of interactive visual environments since that is precisely what they are. Therefore, this thesis studies works of the past as well as gaming theory and design in order to design an interactive system for Intrusion Detection.

CHAPTER 2. BACKGROUND

In order to come up with an adequate design, three areas of study need to be researched and understood. First, since intrusion detection systems are the main source of information about the network and it's traffic one needs to know what they are, why they are used, how they operate and what problems they have. Second, the amount of research done on visualizations for network analysis can help in understanding what may work, what does not, and what needs to be seen. Third, in order to be able to use some of the work done in the gaming industry, an understanding of certain game theory is needed.

Intrusion Detection Systems

Intrusion detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of *intrusions*, defined as attempts to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a computer or network.[16] Intrusion detection systems (IDSs) are software or hardware systems that automate the process of monitoring the events occurring in a computer system or network, analyzing them for signs of security problems.[16] While this is rather straight forward, there are many different methods of intrusion detection that an intrusion detection system may use.

Why use an Intrusion Detection System

There are several compelling reasons to acquire and use Intrusion Detection Systems: To prevent problematic behaviors by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system, to detect attacks and other security violations that are not prevented by other security measures, to detect and deal with the preambles to attacks, to document the existing threat to an organization, to act as quality control for security design and administration, especially of large and complex enterprises, and to provide useful information about intrusions that do take place, allowing improved diagnosis, recovery, and correction of causative factors.[16] In other words, find out where and how attackers may attacked or may attack in the future in order to prevent them from succeeding at their goal.

Kinds of Intrusion Detection Systems

When considering the different kinds of intrusion detection systems, there are three fundamental components: Information sources, analysis, and response. Information sources are the different sources of information used to determine weather an intrusion has taken place. [16] Network based, host based, and application based IDSs (Intrusion Detection Systems) all have different information source components. Network based IDSs detect attacks by capturing an analyzing network packets, host based IDSs use information from an individual computer, and application based IDSs are subsets of host based IDSs that analyze the events within an application. Analysis is the part that organizes events from information sources and examines them through algorithms in order to decide if there is an intrusion or anomaly that needs attention. These include misuse detection and anomaly detection. Misuse detection uses detectors that look for patterns in individual or groups of events that represent an attack on the network or system. Anomaly detection uses defined 'normal' behavior parameters of a network or system and compares them to current activity to see if there are and anomalies. Response is the set of actions that are taken when there is an intrusion, there are two kinds, active and passive. Active responses are automatic behaviors that interrupt the attacker or collect additional information about the intrusion to use against the attacker, while passive responses instead try to inform the user in some way of the activity, they are things like automatic reports to the user or alerts, so that user can decide what to do and handle it.

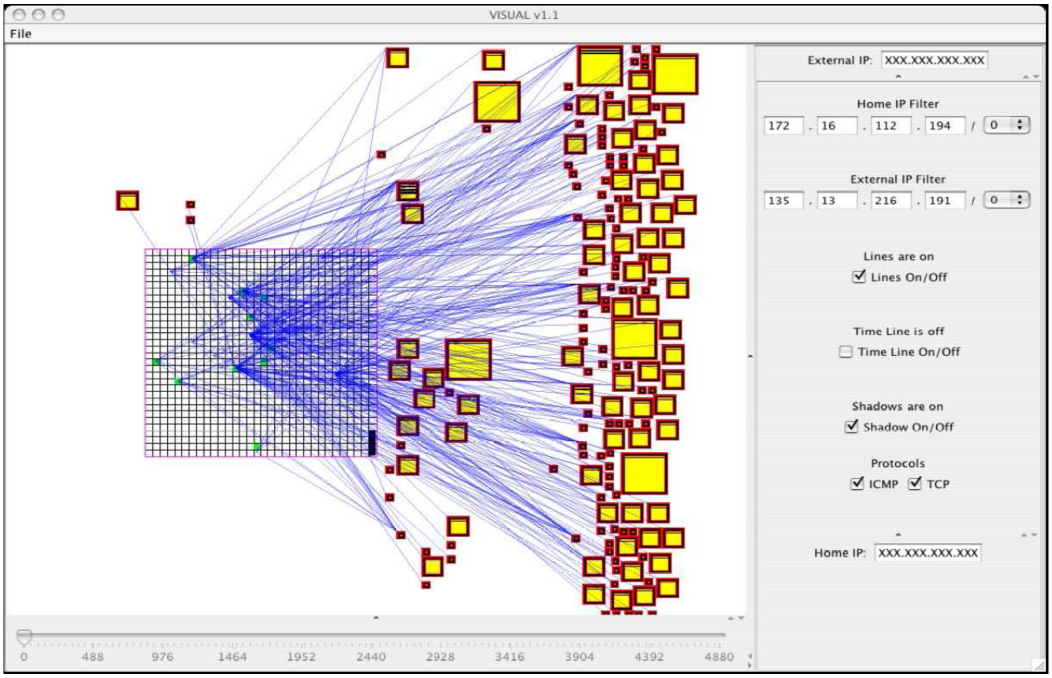
Problems with Intrusion Detection Systems

A major issue is the large number of false detections, especially for Network IDSs (Intrusion Detection Systems). There are two kinds of false detection, false positive and false negative. False positive judges normal traffic as an attack, while false negative fails to produce an alarm for a real attack. [9] This is most likely due to the IDS not being configured to suit the network. If the IDS is set to monitor the network too strictly then there will be many more false positive intrusion results, creating a lot of alerts making it impossible for anyone to tell when an alert is about a real problem or not. On the other side, if the IDS is not set to be strict enough then there will be a lot of false negative intrusion results, making it much easier for an attacker to sneak in unnoticed since the system will not alert to the intruders activity.

Previous Works

There is a lot of literature on trying to create successful data visualizations and network representations. Though not all literature sited here is about communication networks, those that are not still have interesting concepts that can contribute to study and understanding of network data. According to Shneiderman and Aris, there are four different common layout strategies in research literature on network layout. First, literature on network layout has been dominated by force-directed strategies because they produce elegant spreading of nodes and reasonable visibility of links. These digital forces have repulsion among nodes, opposed by the attraction of the links which makes the nodes spread out away from each other while allowing them to group where needed. A second common layout strategy utilizes geographical maps in which the node locations are fixed, like cities on a world map, allowing for association with location. The third common strategy uses a circular layout for nodes that produces an elegant presentation with crisscrossing lines through the center of a created circle. While another strategy is to use matrix-based representations instead of node-link diagrams. Such representations avoid some of the problems of node-link diagrams (especially with large graphs), such as node occlusion, edge crossings, and edges tunneling under nodes by having fixed places for nodes and links on the screen, but spatial characteristics such as finding nodes on a path and identifying clusters may become harder to perceive in the grid,. [2]

A matrix-grid style solution called VISUAL was created by Robert Ball, Blenn A. Fink, and Chris North in “Home-Centric Visualization of network traffic. For this scheme they interviewed people that used detection systems and had to pour over logs of network activity in order to find what priorities they should have in the development of VISUAL. They came up with four priorities: provide markers for external hosts that communicate with the home network, a set of home hosts, show which home network computers are communicating with the external computers, and show the rough amount of traffic each external host is responsible for during the observation time period. [17] In order to do this they came up with a grid that represented the home hosts and used boxes that represented the external hosts. With this they then linked communications between the external and home hosts with a line. (see Figure 1) This way the user can see the amount of connections between home and external hosts at a glance. Unfortunately, some users had difficulty using the time line to identify external computers that were described as “only communicating from time to time” with the home network.[17] This is understandable since all of the communication in between the grid and the different external hosts can make it hard to tell if a few hosts are only occasionally messaging a home host. Also, whenever one host had a lot of traffic, this traffic made it difficult to see any traffic connected to hosts that were located between the high traffic host and the node box sets, causing issues with heavy traffic networks.



External IP: XXX.XXX.XXX.XXX

Home IP Filter
172 . 16 . 112 . 194 / 0

External IP Filter
135 . 13 . 216 . 191 / 0

Lines are on
☑ Lines On/Off

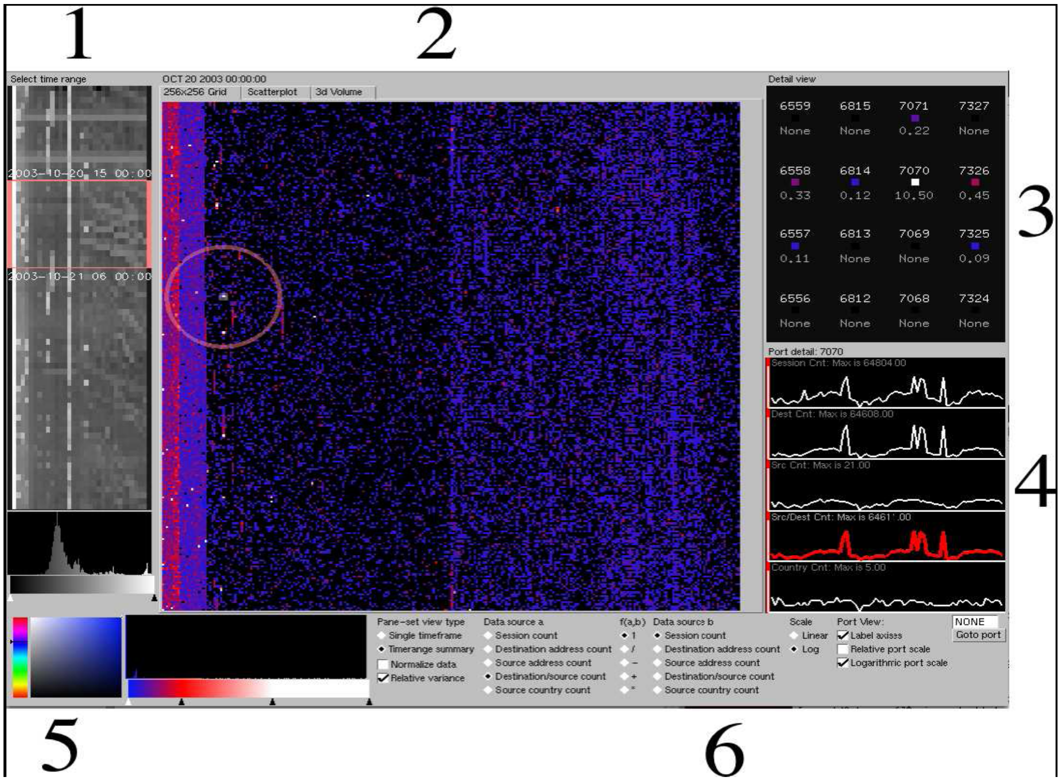
Time Line is off
☐ Time Line On/Off

Shadows are on
☑ Shadow On/Off

Protocols
☑ ICMP ☑ TCP

Home IP: XXX.XXX.XXX.XXX

Figure 1: VISUAL



6559 None	6815 None	7071 0.22	7327 None
6558 0.33	6814 0.12	7070 10.50	7326 0.45
6557 0.11	6813 None	7069 None	7325 0.09
6556 None	6812 None	7068 None	7324 None

Pane-set view type
○ Single timeframe
● Timerange summary
☐ Normalize data
☑ Relative variance

Data source a
○ Session count
○ Destination address count
○ Source address count
● Destination/source count
○ Source country count

f(a,b)
● 1
○ /
○ −
○ +
○ *

Data source b
● Session count
○ Destination address count
○ Source address count
○ Destination/source count
○ Source country count

Scale
○ Linear
● Log

Port View:
☑ Label axises
☐ Relative port scale
☑ Logarithmic port scale

NONE
Goto port

Figure 2: Interactive Visualization for Network and Port Scan Detection.

Another scheme that utilized a grid was designed by Chris Muelder, Kwan-Liu Ma, and Tony Bartoletti. For this visualization there were several components all displayed at once, it included a time line that is a showed the entire time rage available with the vertical axis of a rectangle corresponding to to time, with each row representing one unit, and the horizontal axis corresponding to the port range. Another component consisted of a grid that depicts the activity during a given time unit. It consists of a dot on a 256 × 256 grid for each of the 65,536 ports. The port number can be thought of as a two-byte number. Therefore, the x (horizontal) axis represents the high byte of the port number, and the y (vertical) axis represents the low byte of the port number. So each point corresponds to a particular port, and the color of each point is determined by the value of the current metric at the corresponding port.[6] (See Figure 2) Also included was a scatter plot in order to assist analysts compare metrics as an alternative to the grid visualization. The primary difference is that instead of laying the ports out by their numeric value, they are laid out according to the the values of two metric for the port.[6]. As is pointed out in the publication, this scheme is good at displaying the whole dataset without the need for several panels of information, but the problem is that ports of interest are very small and activity can easily be missed if not already anticipated.

A three dimensional visualization tool that utilizes grids in a different way is Cichlid. This tool uses 3D animated visualizations for differing data sets. Cichlid currently supports two types of graphs: 3-D bar charts, which are useful for displaying numeric quantities that are functions of two independent variables, and vertex/edge graphs, which are good for representing topology.[5] Unfortunately though, Cichlid uses very strict data set objects that have to be created from the network data before it can be used. The data collection code that abstracts the application-specific data into data set models is the responsibility of the server writer, since the toolkit itself has no knowledge of the application domain. [5] Another problem is that operations on these data set objects must be performed through method calls;

the user is not free to assign directly to any members of a data set. This can slow down and cause problem with analysis, but this it is strong in visual representation of data. (see Figures 3 and 4)

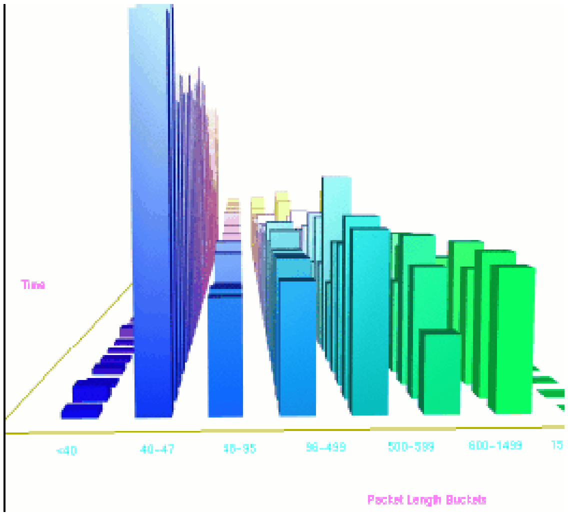


Figure 3: Cichild: Packet Length Distribution Over Time

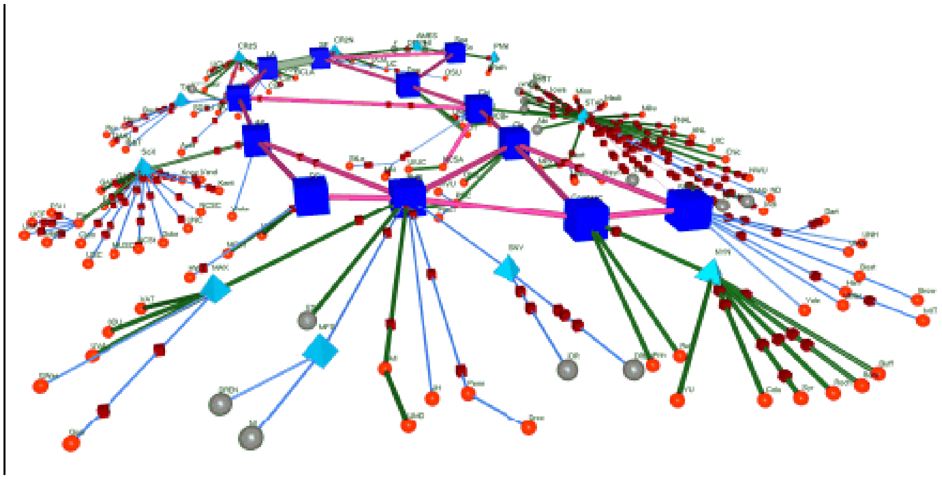


Figure 4: Cichild: Vertex Edge Example

Another three dimensional project was done by Adam Ronald Oline. Oline created three different visualizations for networks and ports. The first one was called Island. Island position trees on an island with each position that a tree could stand representing a port number. For the visualization the port numbers ran from the outside regions of the island starting with the lowest port numbers (0-1024) ringing the island and the rest continuing in a spiral pattern to the center where the largest port number would reside. Trees appeared on the island at their respective port positions whenever the port they represented make a connection or communication. The tree's shape was then dictated by the Destination and Source addresses. This innovative design makes the visual unique as well as functional, where it was able to visually show HTTP tunnel attacks and SYN flood attacks. Both attacks that require the use of many different ports on the victim machine.

The second visualization Oline came up with was Alert Plot. Alert Plot is a rectangular image with the horizontal values representing time and the vertical values representing the port numbers of a host. A small cube is drawn whenever an Alert occurs, these blocks are color coded according to the priority rating of the alert. Color coded lines lead to one or both of two grids that represent each host of a class B network and each class B network. Also, the Alerts can be toggled on or off. In many ways it seemed like a more functional version and combination of both previously mentioned schemes (VISUAL and Interactive Visualization for Network and Port Scan Detection), being able to show Port Sweep Probes and Repeated password attacks. Here the Third Dimension really helped, but it still suffers from some of the same problems that the previous schemes had. Lastly, Trends, is a 3D graph that uses the x coordinate for time, z coordinate for port number, and drew a bar along they y coordinate to represent the amount of data transmitted. The major improvement here, though, is that there are two overlapping bars. A red transparent bar and a opaque blue bar. The Red bar represents the average amount of data an a specific time checked every 12 hours, while the blue is the actual amount of data. This way one can see anomalies quickly and clearly in the large amount of data.[1]

While these existing network visualizations often seem impressive because of the colorful display of nodes richly connected with links. These visually engaging images enable users to estimate the network size while revealing important clusters. However, in most examples, the overlapped nodes prevent users from estimating cluster size and the crossed links make it impossible to follow connections, count node in-degree, or carry out other tasks. [2] Therefore, another scheme was developed. That is visualization by Semantic Substrates. Semantic Substrates have two principles: (1) layouts are based on *user-defined semantic substrates*, which are non-overlapping regions in which node placement is based on node attributes, (2) users interactively control link visibility to limit clutter and thus ensure comprehensibility of source and destination. This allows for automatic node placement by their attributes, so that the location conveys information, according to Ben Shneiderman and Aleks Aris.

One advantage of semantic substrates is that proportionally-sized regions would immediately give users some idea of the relative cardinality of each category. For example, in a food-web layout with five regions by mammals, reptiles, birds, fish, and insect groups, users would be able to see that there are many more insects than mammals or reptiles. A second advantage of semantic substrates is that users can quickly distinguish links that cross from one category (region) to another, for example, enabling users to see that reptiles eat insects and mammals, but insects do not eat reptiles or mammals. [2]

Of course semantic substrates are effective only if there is some categorical attribute or if a numerical attribute can be used to form categories. However a caution about semantic substrates is that they complicate node and link drawing by imposing an additional constraint on the layout. Though for some, the added utility of user control of link visibility may prove more advantageous. What is most interesting about this idea is that the grouping allows for a form of chunking. In doing this, if the groups are allowed to change, but because of their grouping stay relatively the same, it would help with spacial recognition since then a person can see that group first then consider the individual systems.

One other item of interest is Cheops. Cheops is an Open Source Network User Interface. It is designed to unify your network utilities. Cheops does for the network what a file manager does for your file system.[14] Cheops interface shows the different systems in a network and their connection to each other using icons for the systems and lines depicting the connection much like a network topology or design diagram. It also has certain feature for working with the network which include:

> Chunking: Multiple Pages: Organize your network into convenient pages so you can place relevant portions together, and quickly go to a specific area or specific network.
>
> Find: Quickly find hosts on a large network
>
> Mapping: Cheops can show you the routes taken to access areas of your network. (This feature is designed for larger networks, with routers, subnets, etc. If you only have a simple LAN where all your hosts are connected with hubs, then it'll just draw a bunch of lines between you and the other computers) This mapping not only makes hierarchy clearer, but can show unusual routing issues, like this unusual router triangle. Unfortunately, you have to place the machines yourself, but cheops handles the interconnections :)[14]

The reason this is added is that certain features like Find and Mapping can be very useful. If in a visualization a person finds something suspicious about a station on the network, being able to use a find function directly from the visualization would be very helpful. Also, being able to see a mapped path for electronic access can be helpful if a group of machines are compromised and attacking other systems, you would want to know immediately where it would be best to disconnect and quarantine the problem with the least amount of problems to the rest of the network especially if any types of mitigation would take large amounts of time.

Game Design and Technology

Unfortunately video games, with all that they can offer to the research community, are widely ignored. While a great deal of research has been directed towards discovering the features of non-leisure software that lead to maximal functionality and minimal negative user affect, very little research has looked at games. Furthermore, even less research has focused on bridging the gap between these two areas.[7]

Like productivity software, games contain menu systems for changing options, interfaces to communicate status information, and input and control devices. Furthermore, games also require that users develop a conceptual understanding of the rules of use, and often offer tutorials and help systems to aid players in learning basic skills.[11] Video games constitute a genre of software in which the user’s affective experience is paramount. If a game does not generate positive emotions in the user, it will not succeed – there is no other reason to play a game. [7] This is one of the major differences between a game and a non-leisure software application, driving games into further and further development of user experience. A part of this is that games must also be challenging. Challenge is something that applications are typically designed to minimize, which is appropriate, though with something like Intrusion Detection, the challenge is already there, the environment just needs to help immerse the user into the task of analyzing the data and in some way rewarding the user. Lastly, games constantly try to innovate by trying novel things, including interfaces, controls and cutting edge technology; applications, on the other hand, attempt to be consistent and typically change only incrementally from version to version.[11]

So how can game design be integrated into a visualization? First of all, many games provide minimal information to the user during actual game play[7]. That is, unless necessary the information is made so that it looks almost transparent. A person playing a game does not care about the underlying statistics and calculations needed to play, all he/she cares about is did I do it, did I hit the bad guy, did I pick up the item, etc. To the player it is all too easy and intuitive. The same is wanted for an IDS (Intrusion Detection System). The user does not care about the underlying data until he/she absolutely needs it. What he/she wants to know is, are the computers OK, is communication traffic clear, and is there anyone snooping around. Successful video games are able to provide players with needed information and control capabilities in an engaging and enjoyable fashion. Video games are effectively streamlined input-output systems, and to a player, a video game is little more than its interface [12]

Regarding playing to learn, the emphasis is on learning, which is to say that some content or skill should be the end result of game playing. As such, knowledge and skills are treated as effects or outcomes. In regard to learning to play, on the other hand, the emphasis is on the activity of playing. As such, learning might be regarded as an integrated part of mastering an activity, in this case, game play.[8] Of course, there is no reason why both can not happen one after another or simultaneously. Especially with an evolving entity like a network and the encompassing environment a game can provide, and since games give the player an opportunity to act in a multi modal and interactive environment, games foster active learning, or learning by doing. What is more, players encounter various signs and tasks in a multi modal environment where words, images, actions and sounds are intertwined with one another. [8]

Also, the idea that wherever possible users should be prevented from making errors is largely contradictory to the manner in which games promote flow and positive affect. When playing a game, part of the challenge for the user is the fact that mistakes must be avoided. Different from the errors that can be incurred during setup, these errors are what helps the player feel there is a challenge and that they are actually working toward a goal. During game play, the joy of success is dependent upon the possibility of failure[7] Now I am not promoting failure, but a system should try to promote learning from failure rather than just letting failure be failure. Though in a system that relies on the user for input in order to function properly, rather then an environment with set rights and wrongs, there is a need for more then just one user interacting in order to gain experience from not only the visual but the social environment between security officers as well. In this way when one person fails to catch a problem and another succeeds they can learn from each other. Nothing mentioned here is easy, especially when trying to tie in an application that does react, change, and work in the real world, away from digitally set constraints that a game can use, but there has to be a way.

CHAPTER 3. RECOMMENDED DESIGN

After careful consideration of the different game genres for a basis for this architecture, it was decided that in order to make the system interactive, usable, and informative, Real Time Strategy was the best choice. Games in Real time strategy have always had to compensate for lots of data constantly updating about multitudes of different objects and make it into a fun and easy to use experience. Therefore, in order to get all of the data acquired from a network successfully across to the user and make the interface intuitive and easily used, it should be developed much like a Real Time Strategy game.

A Real Time Strategy (RTS) game has a user construct and protect a base or home of systems used to build up forces to defend and attack opponents in an attempt to destroy their base (most common goal) without being destroyed. The only differences in this scenario and that of a network, is that the network is already built and it is always on the defensive. This lends itself to such an architecture in that all network systems can be treated like buildings for a base, and while the buildings can not move, they can accomplish different tasks. In an RTS, buildings are the main production units for creating forces, they always need to be protected. This is the same for our network since we need to protect the systems in order to ensure they can continue to produce for whatever organization the network belongs to. All of this makes it look like RTS and IDS were made for each other.

The Design

In designing a Graphical User Interface (GUI) for visualizing Network traffic data, there are many different considerations: how will the network traffic data be gathered, how will the data be stored, sorted, filtered, and aggregated, how will the data be shared, how will the data be represented, and how will the user be able to interact with the data? While the first two questions have been solved with the use of outside open source programs, the most difficult questions in the design still remain. Therefore they will be the main focus of this design and its discussion. It will start of with a discussion of what will be used to render the images for the user to see and interact with, what the user needs to do to make sure that they are the ones allowed to see the network data, how to become an allowed user, how the information will be displayed and why, different interactions and their associated windows that the user can use and why they exist, and an overview of the software used for the back end of the design and why it is used.

Torque Game Builder

In designing this Visual Interface, it was decided to make it using a 2D interface. Using a 2D interface makes it easier to create and update for future versions, it makes it easier to view all of the information at one time without getting confused our lost, objects do not accidentally hide other objects behind them, as can be the case in a 3D environment, and it takes less resources to do the rendering of the network. For this the Torque Game Buidlder game engine lends itself perfectly to this designs development. Torque Game Builder is build on C++ code and is compatible across many different platforms for computers, taking away many concerns of weather or not it will be able to work for most systems. Also, the Torques Game Builder environment is meant to allow those that have very little programming experience to create GUIs for gaming environments. One problem with the the builder is that it has so many options and controls that have to be configured that anyone new to the developing GUI can easily become confused, frustrated, and lost. Also, Garage Games the developers and sellers of the engine do have certain resources and tutorials for the engine but there are limits to the documentation. On the other hand, the Garage Games forums and community is full of people with experience in working with the engine and may have certain resources that can help speed up development. Another problem is that a license for the engine costs approximately $250 for a professional license with certain amounts of access to documentation and is the only way to get source code needed to make any additions to the engine.

Even with certain problems, Torque is the best choice since it's source, when obtained, is easily modified and organized very well, the scripting language for actually programming in torque is very similar to Virtual Basic and/or Flash and it's networking capabilities make it very easy to track and manage connections between the server and different clients. Open source engines available primarily work at creating 3D environments and do not lend themselves to creating Real Time Strategy (RTS) designs. Also, most open source technology for games are unreliable, constantly being changed in ways that could make the system unstable, and have lots of dead documentation (documentation that has become useless due to changes to the software). Also, with the use of torque, while 2D makes the most sense, if needed, this design is able to be converted to a 3D environment easily if Torque scripting is continued to be used for development, i.e. using the Torque Game Engine which is a 3D game engine that costs roughly the same as Torque Game Builder for a license and is also created by Garage Games, otherwise the entire visualization and all interaction for the environment will have to be completely redone.

User Authentication

One concern that came to mind during the research and planning for this design was that of user authentication. While an administrator should be able to tell if there is an outsider accessing and monitoring network data through the increased activity seen, it is more difficult for unauthorized access from those that are apart of the network or have broken into the administrators place of work to access through his/her machine. While some of these concerns seem unfounded it was decided that it would be better to add in some user authentication. Therefore, when a client system is started up, the user will be prompted to authenticate through a login screen. (Figure 6a) Here the user will have to enter a valid user name and password pair as well as make sure that the IP address to the Torque server is correct. This way there can be separate torque servers set up to collect data from different IDSs, if wanted, that can be accessed and displayed from a single machine. Unfortunately Torque does not allow for multiple clients to run on the same machine at the same time, but this gives the user added flexibility in their setup. Also, on the log in screen, if a user who has never used the system before wished to log in, they can click the new button and create an account with a user name and password in a new window that will appear. (Figure 6b) Though they can only successfully create an account if they either have a registered user confirm they are allowed on the system or get a temporary authorization code to create an account from an already registered user. Here it is assumed that security and network administrative personnel will be the only ones authorized, so they should be able to be trusted for this. Lastly there are Exit and Accept buttons for the screen, the Exit button closes the program and Accept submits the input user name and password pair to the server.

User Interface

This part of the design is the most complex and is the focus of this project. With a focus on how Real Time Strategy (RTS) games have created easy to use interfaces with easily viewed and interpreted environments this system attempts to give the user not only an overall view of the network and it's activities, but also allow the user to focus in on certain areas, single out traffic for one or more specific machines and view what ports are being accessed by or on a target system. This is so that not only can the user get an idea of where certain problems may be, but it allows the user to focus in on where those exact problems may be and narrow in on the data that will help discover what the cause for an anomaly is.

The Map

Like most Real Time Strategy (RTS) games there will be a map (Figures 6a and 6b) that can be navigated around with the mouse or arrow keys on the key board and have zoom in or out on the view of the map to a certain degree utilizing ether the '+' and '-' keys on the key board or a mouse wheel. This map will be the main source of visual information. On the map, network systems will be represented by Icons that can be customized and replaced for each individual system. This allows for the user to customize the systems in the network visually. By doing this the user will have a greater probability of remembering distinct facts about each system and what has happened to them since there will be the new association with the unique visual stimuli. Also, certain icons may be able to be earned for use through certain actions or accomplishments of the user or users of the system, examples include 1 week without compromise of network systems, most systems blocked, most hours logged on, etc. These icons may be included with certain badges or titles that will appear in the title bar of the screen. This and other potential gifts mentioned can help with the sense of accomplishment and give the system some form of positive reenforcement, especially if these rewards are made to be humorous in an attempt to incite their discussion amongst those that use the system.

As for the location of the network systems on the map, they will be arranged and organized by their network links. That is that each system that is connected to a certain router will be clustered together and a router will be shown, much like it is represented in the Cheops design. This will give the user a better feel for the network and the network paths that the information travels through. Also, the advantage of stationary locations is that those locations will better stick in the users memory allowing the user to better distinguish and understand patterns seen in the network over time, since there will always be the correct associations between the seen communication and their specific stations and locations. Also,

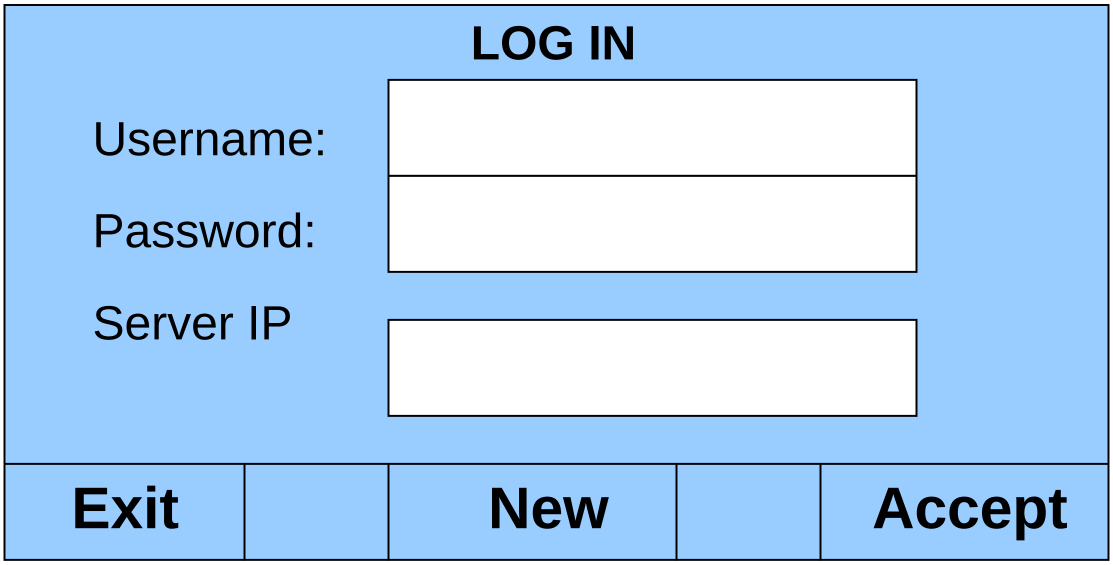


Figure 5a: Login Screen

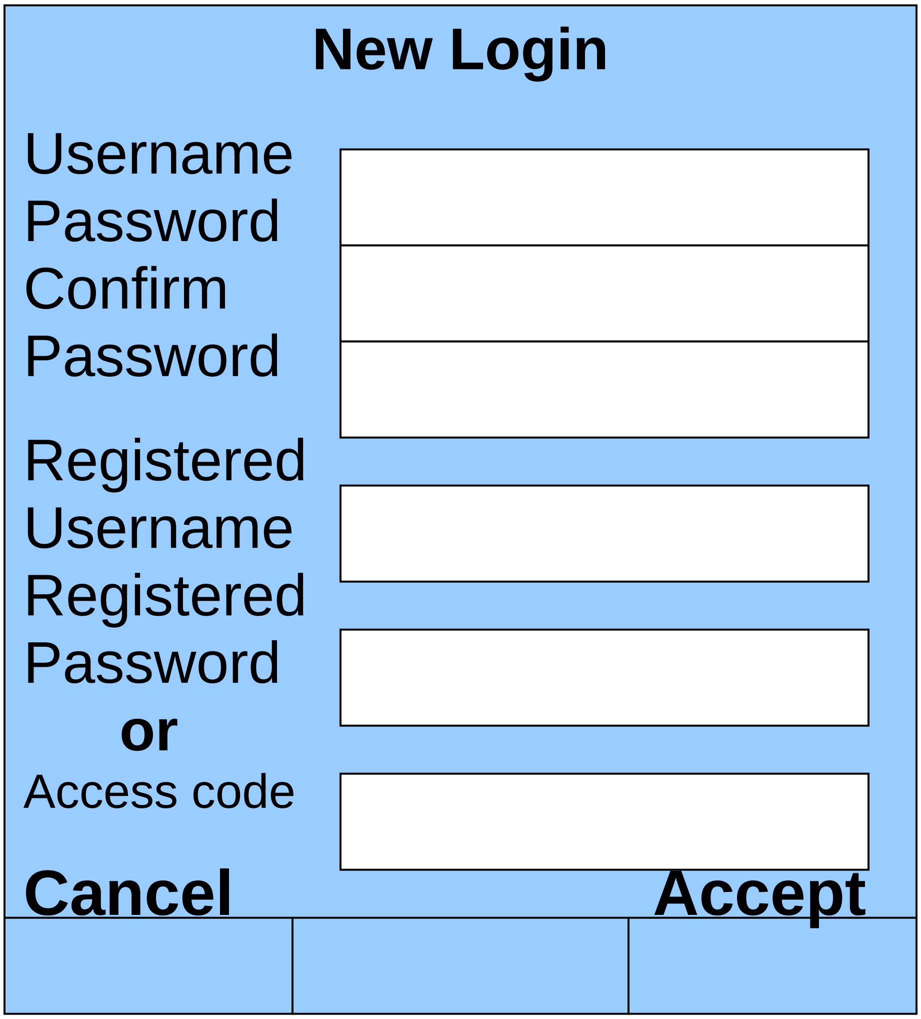


Figure 5b:

New Login

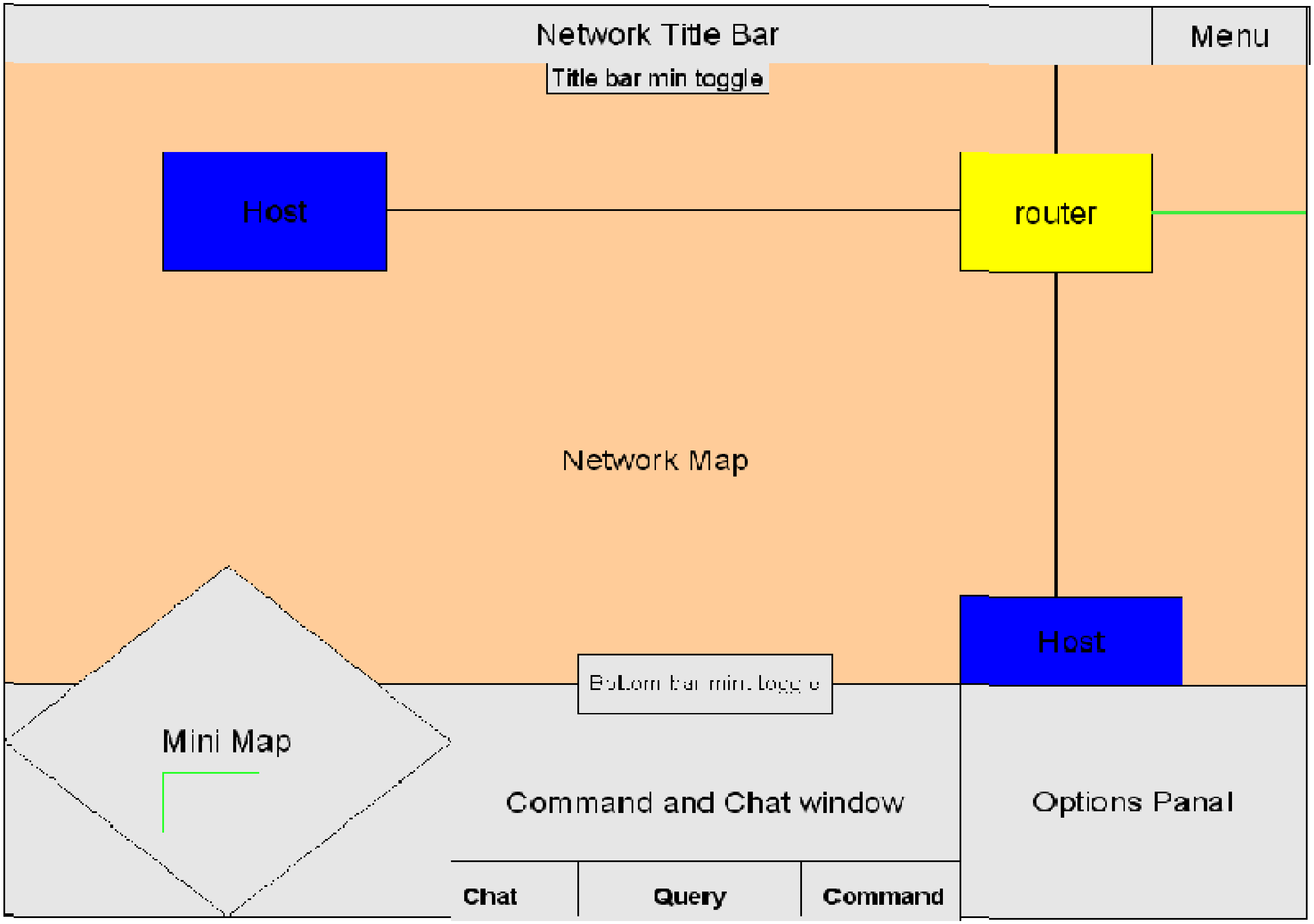


Figure 6a: Main Map

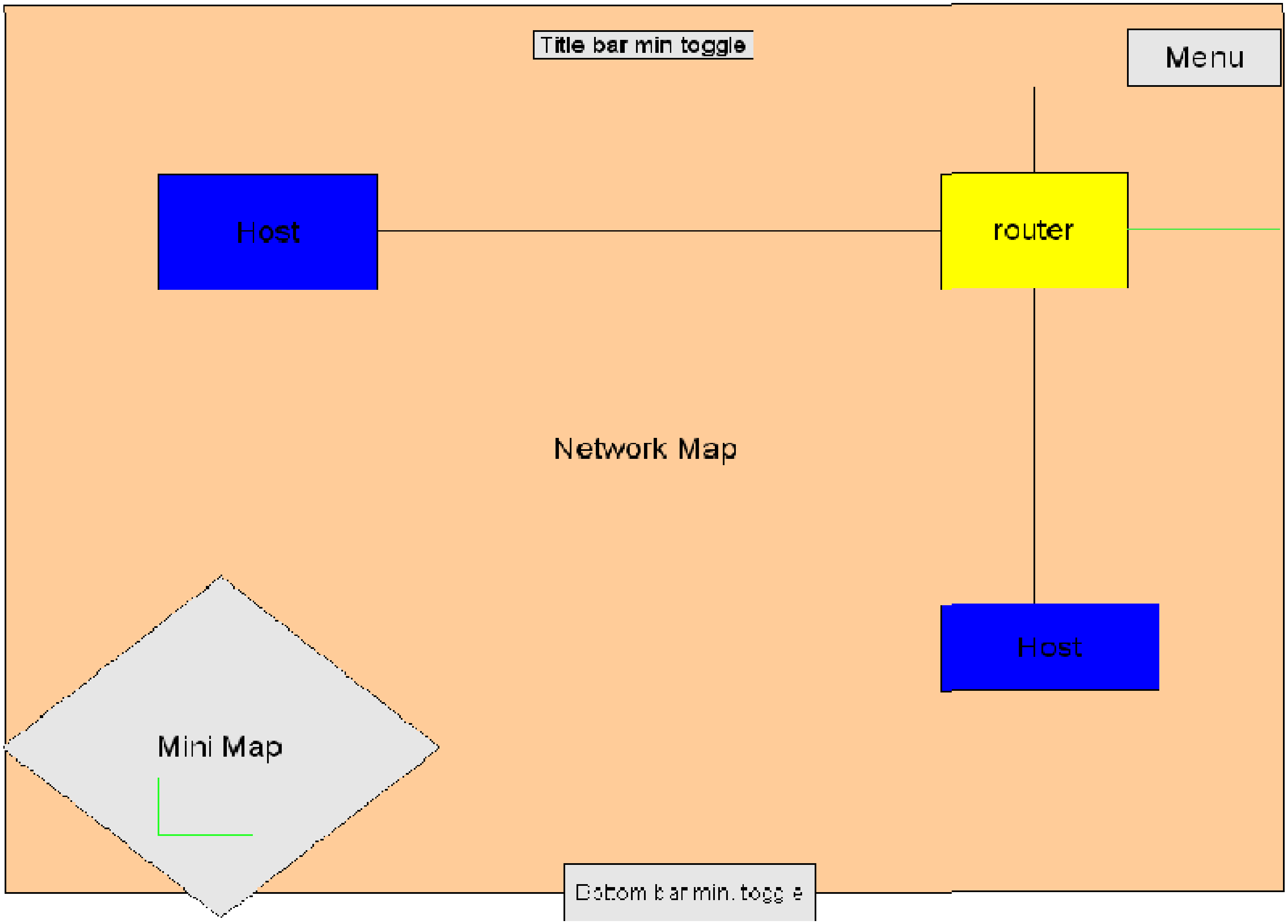


Figure 6b: Map without bars

since stations are no longer the only network objects being represented there will be different types of Icons for the different system types, like routers, servers, switches, etc.

In addition to the system icons being displayed on the map, all system icons will show two bars under them, these two bars represent the number of ports that are being used on the system. The more the ports there are being used the more the bar fills. Every bar will have a threshold that will be adjusted by their average activity for that time period. Therefore all systems will have a different threshold. If this threshold is breached the systems will alert the user. Also, the user can change this threshold for the system on his/her local client, and it will be stored on the SQL server under the users account so as not to effect the original data or the thresholds set by other users. This way every user can better customize the system to fit what they are looking for, giving different points of view and allowing for the old proverb, two heads ARE better than one. These bars give the user an idea of how busy a system is and helps in making decisions of weather or not there may be an intrusion and may require further investigation.

Also needed is a representation of the actual traffic flow. For this network connections will use two lines for each link along a path, one for inbound and one for outbound. A movement animation for each line that look much like chaser lights will be used to depict the speed of the of transmissions along a link, if possible. When the data is transmitting at high speeds then the animation is traveling at high speeds and if the link is slowed for some reason so is the animation. This can help to see if the speed of a line is being effected. These same lines will also show the amount of data traveling across them with the thickness of the line, if there is a large amount data traveling in the direction of the line on a link, then the this will cause the line to grow thicker. Also, if the connection is completely idle it will gray out until there is activity again. This is the same for system icons, if a system if found to have stopped transmitting at all across the network for 5 minutes, it will gray out to signify its lack of activity. Though, if a system is found to have crashed then it will become black, signifying there is a real problem.

Another element needed is that of external hosts. External hosts are represented by smaller icons connected to a link that leads from the network to the Internet. There are three distinct categories of external hosts: trusted, dangerous, unknown. As the number of accessing external hosts increases they may start to become crowded, so the hosts are then bunched together into one icon that can be looked at by the user to see what hosts are in the bunch. Bunches are formed according to IP address and group affiliation. Bunching also exists for larger networks where there are too many systems for the map. While bunches will be created by default if a network is too large for the main map, the use also has the option to create bunches at any time using systems on the network. These bunched systems must be connected in some way in order to be bunched together however. This is to prevent confusion and keep the system conformed to the designated design precedents. Also, bunching can be used for entire sections of a network or for smaller networks connected to the network. These bunches can be entered and the user can pull certain systems out of a bunch if needed. Also, bunches will use slightly different icons in order for the user to easily understand what they are quickly.

In order for the user to better gather information about the network activities, he/she needs to be able to quickly get information about a specific host or group of hosts. For this clicking on a icon will allow for several different actions for looking further into network activity. This clicking on a network system will cause different options to populate an options window in the bottom right corner of the screen. (Figure 8b) For hosts, the options panel in the bottom right of the screen will include an icon for highlighting flows of all connections to this system, highlighting flows to or from specific a port connection/s or number of port connections, highlighting flows to a specific host/s, allow sight of associated port numbers with each connection to a system, allows the user to open a System Alerts window for turning on and off different types of alerts, choose to set port threshold tolerance for the selected system or systems, select to save the system or systems to a specific group, select from a systems associated groups to select all systems in that group, dissociate as specific system or number of systems from a group, or designate a new icon and or name to the system. All of these option effect the user's account data and client only so as not to affect the system for other user's . When the option for highlighting flows of all connections is selected, then th paths to and from all connected systems will be highlighted. If one or more of those are in a bunch, such as external hosts, those systems will be pulled out of the bunch until another option or system is selected or the user clicks in any open space on the map, this is true for any highlighted flows selection that includes a bunched system. This allows the user to get a better idea of who is doing exactly what, whenever something occurs or looks like something will occur. The highlight flows of specific port/s opens the command window (Figure 9c) in the bottom center of the screen and allows the user to select the port/s. The same thing happens with the highlight flows of specific host/s. The toggle port numbers over connected systems option causes the ports that systems are connected to on this system to be visible and associated with them or not. Lastly the alert toggles allow the user to turn off certain alerts for this system from a window that opens and looks exactly like the Alerts window for the main menu that will be discussed later. (Figure 10c) This allows the user to turn off areas of false positives that he/she may have found, until they can be fixed for specific systems. For selecting a group of systems the user can select multiple hosts by clicking on more then one system while holding shift, using the select group option from a host for an already created group, or selecting an area of systems by enclosing them in a box shown when the user clicks, holds, and moves the cursor on the map. For these groups all the same options are available as with the host options, except for the group selections option. Also, after a group has been selected it can be hot keyed to a number by pressing ctr + [number] for temporary quick access to this group, the same can be done for individual systems and bunches. When one clicks a bunch, the options are the same as before with the addition of the option to enter the bunch. By doing this the map changes over to the systems of the network in the bunch as if it were a separate network itself. All systems outside the bunch that are in the network are portrayed as trusted external hosts unless designated differently by the user. Also, there is an added icon attached to the out leading link which if clicked on will have the option to exit the bunch to get back to the outer network map. Using this system of bunches allows for multiple levels of the network to be created. This way different users can customize the system to better suit what and where they are watching. For example, several users may be responsible for monitoring different parts of a large network, therefore they primarily only care about their sections and want to give them the most scrutiny. At the same time there is a user who supervises over all of them, here the user can Separate all the areas into different bunches to view the overall and can enter the specific bunches associated with location to monitor how someone's section of the network is doing. Lastly when one clicks an external system all the same options are available except the specific port/ports options is not available and there is an additional option to designate the system to a different external system group.

The map is the main and most important part of the visualization since it allows for the most direct interaction and represents the most data to the user. With all that can be done on the map, there are still many components that assist the map and the understanding of what is happening on the map that are needed.

The Mini Map

The mini map, (Figure 8) in the lower left corner, shows the entire map all the time. This is a snap shot of what is going on across the network and helps the user to navigate to all the different systems on the network quickly. The mini map has to always allow the user to see additional objects and locations that do not show on the screen because the camera (the users view) is not at that specific location and therefore is not currently shown on the main map.   The mini map will also show where the camera is located on the mini map to help the user become better oriented in what part of the network he/she is looking at. Also, flashing dots or circles will appear at locations where a system is the subject of an alert. This way the user can always have a good idea of where they are located in the game map and if there is anything important going on that they can not see. Using this scheme of exploring a larger map and having a mini map to help with navigation is helpful since it allows the user to explore the network and see chunks without having to watch the whole network all at once and suffer from information overload. Also, this allows there to many more icons for systems then otherwise possible, as can be seen when looking at an screen shot of a zoomed out view of a section of the overall map for a current RTS game “Roam Total War.” (Figure 7).

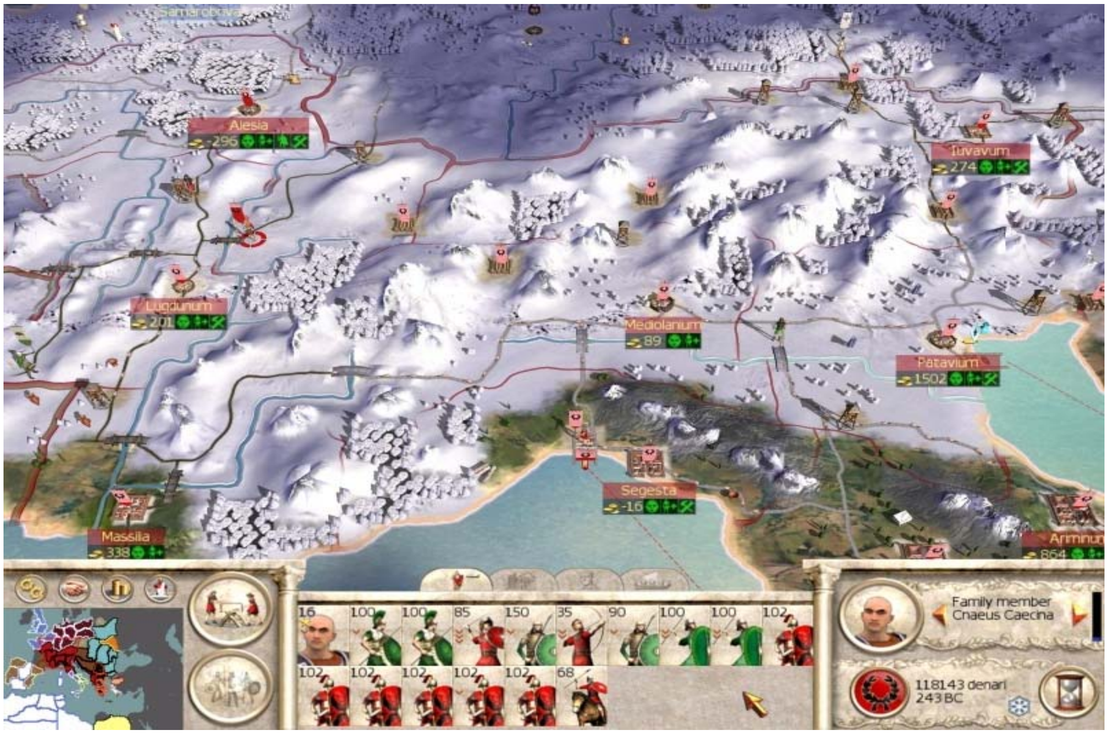


Figure 7: RTS map sample from Rome Total War

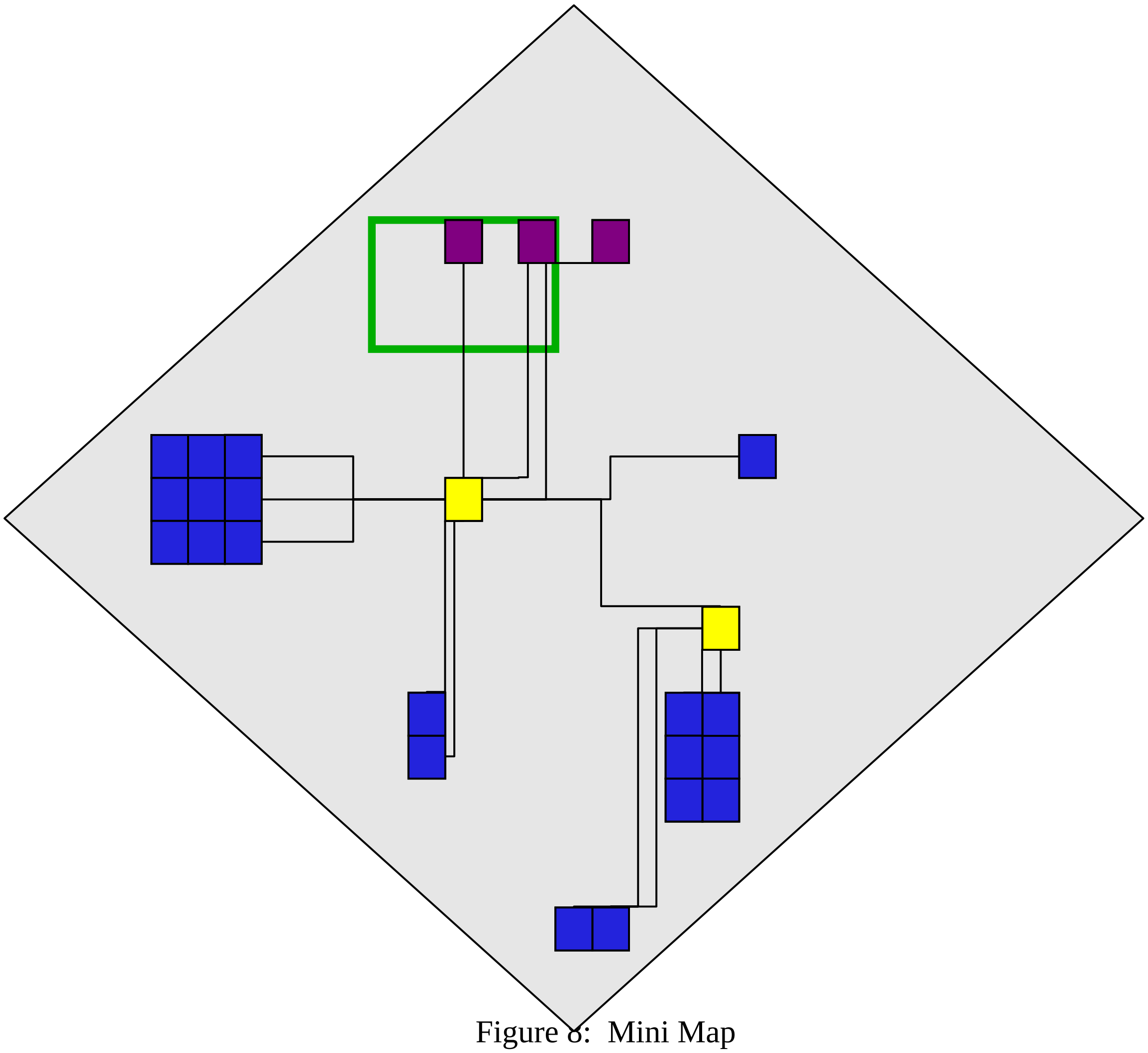


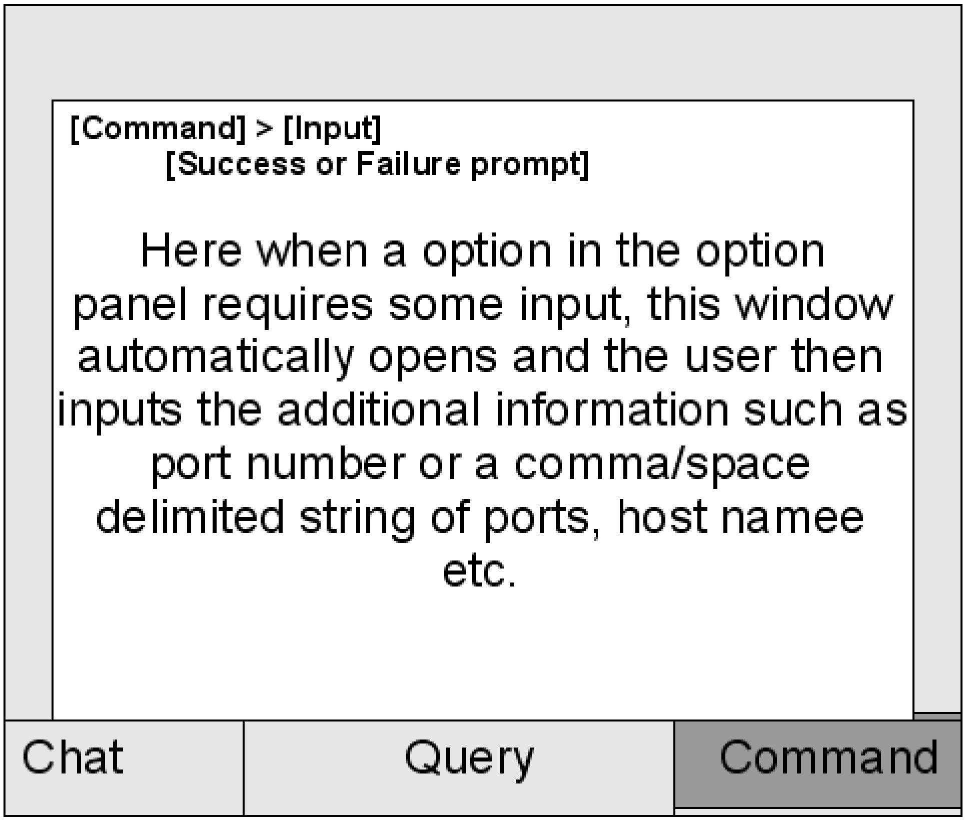
Figure 8: Mini Map

Options and Command Windows

As mentioned before another aspect that needs to be included is the option menu in the bottom right corner for object interaction. This menu is actually a panel that remains blank until a system is selected, then it is populated by a list of options that represent different actions that can be taken. These selections are discussed above in the sixth paragraph of section 3.3.1 The Map. The reason for this form of interaction is so that there is no distraction form or breaking away from the visual of the network environment. With this menu it is quick and easy to choose options for inquiring about more data. Also, using this window some options may need additional user input, therefore the window between the mini map and the options window, called the command window (Figure 9a) will display a prompt for text command, a list of options, or icons of actions depending on the options selected by the user in the options window. This relation between the windows allows for greater usability without sacrificing needed screen space for viewing network data. Just above the command window a toggle button for minimizing a lower bar containing the options and command windows is included, this way if the user needs the additional screen room, it is available. Keeping the environment clean and clear for the user is paramount when the user is expected to be absorbing large amounts of data.

The Chat and SQL Query Window

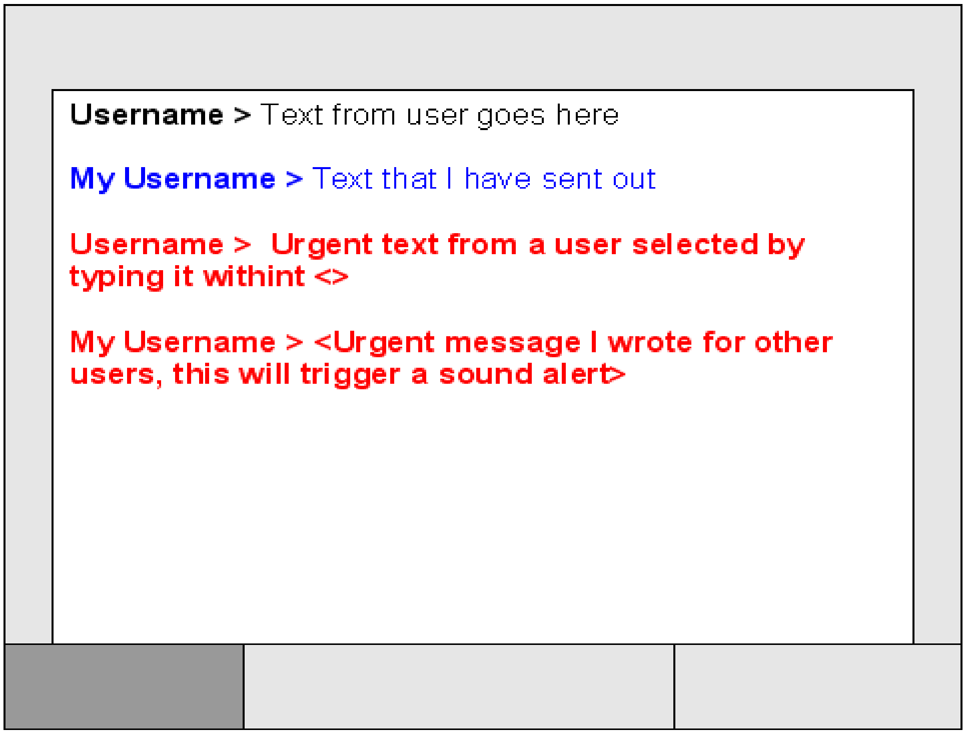
Also included in the design is a Chat and SQL Query Window (Figures 9b and 9c). This window is actually located in the exact same place as the Command Window and allows the user to either chat with other users connected to the same server or query the SQL database located on the server for specific information. The reason these windows occupy the same space as one another is that the user will not be able to use any two of them at the same time, because of this and for the sake of space for the main map, they overlap one another. In order to select the chat or query window the user chicks on a tab at the bottom of the window corresponding to the window required. The command menu will only be shown after the user selects a system and an option in the options window that requires use of the command window, otherwise either the chat or query window is shown, which ever was last selected. This allows the user to be able to quickly navigate between actions on the system chatting with others for advice, assistance, and entertainment, and searching for additional information from the network data directly while only having to work with the command window when needed. Though query window will require the user to know SQL commands in order to successfully query the server for data.



[Command] > [Input]
[Success or Failure prompt]

Here when a option in the option panel requires some input, this window automatically opens and the user then inputs the additional information such as port number or a comma/space delimited string of ports, host namee etc.

Figure 9a: Command Window



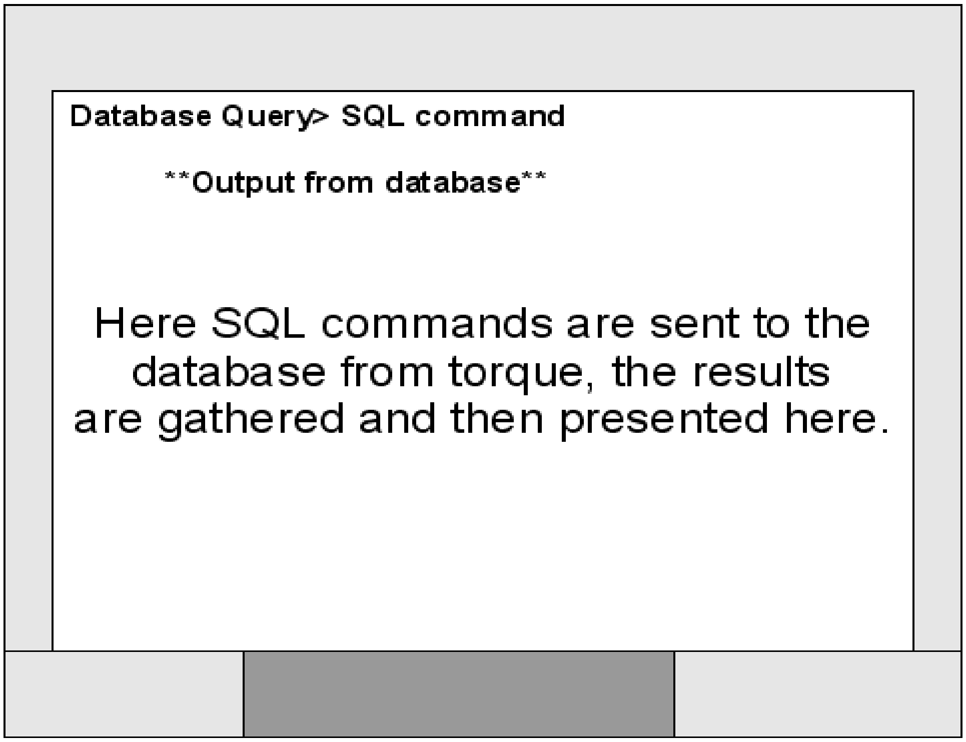
Username > Text from user goes here

My Username > Text that I have sent out

Username > Urgent text from a user selected by typing it withint <>

My Username > <Urgent message I wrote for other users, this will trigger a sound alert>

Figure 9b: Chat Window



Database Query> SQL command

****Output from database****

Here SQL commands are sent to the database from torque, the results are gathered and then presented here.

Figure 9c: SQL Query Window

The Menu

The interface must also have a menu button (Figure 10a) for user account specific operations to ease the users use of the system and make it more customized to him/her. The menu button, true to form for many RTS games, is located at the top right of the screen. This is to keep it out of the way while at the same time making it readily accessible. This menu will drop down and contain options for controls options, sound levels, going back to the log in screen, leaving the menu to go back to the map, editing network identification information, alert options, and exiting the system. The controls options button will open a new window that will allow the user to adjust the sensitivity of the mouse using a slide bar. (Figure 10b) The sound levels option opens a new window that uses slide bars for volume levels and a check box for muting sounds that include general sound effects and sound effects for different alerts. (Figure 10c) The Network Information option opens a window that allows the user to give a name to the network that will be stored and displayed a the top of the screen when the system is open, choose past stored network data according to time to view, and to give names to be associated with specific IP addresses for systems. (Figure 10d) This allows for the study of previous network data, and personal customization of the map to better improve the user's retention of and comprehension of information. The Alert option will allow the user to set volume settings or mute sound effects for different alert types, as well, as change their priority levels for his/her system. (Figure 10e) The Exit option will ask for confirmation and then close the client if confirmed. These options will have no effect on the information from the IDS directly since that is all stored in a SQL database with the torque server under the user's account, instead it will effect the environment that the data is begin presented in to fit the user.

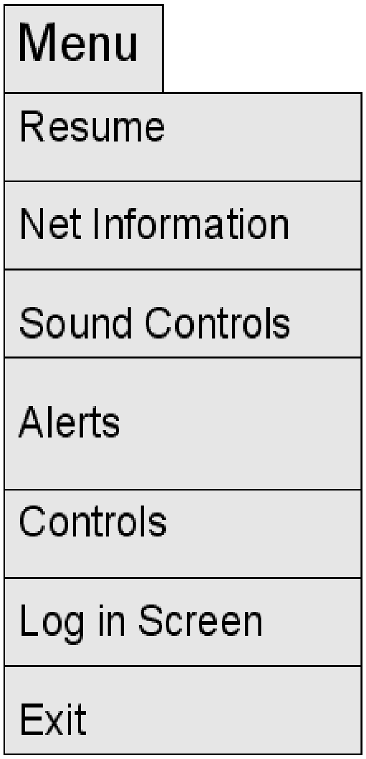


Figure 10a:

Main Menu

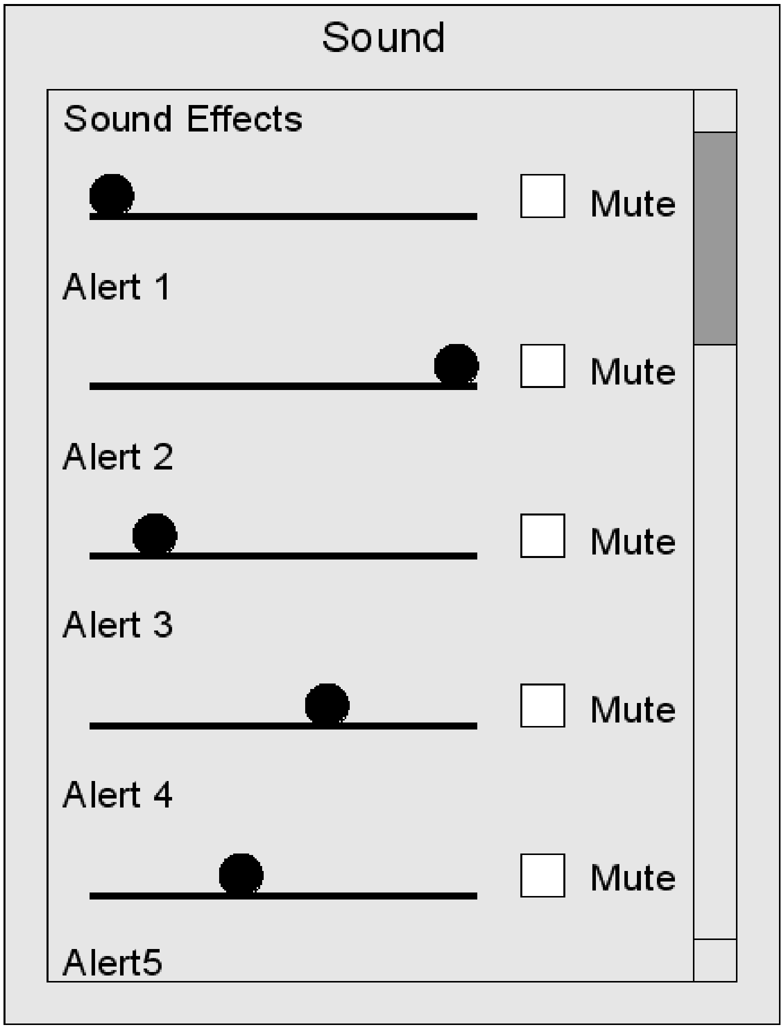


Figure 10b: Sound Controls Window

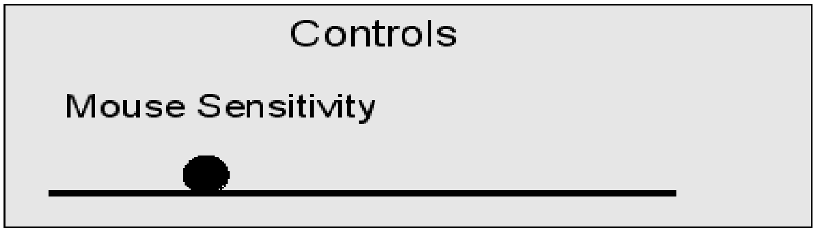


Figure 10c: Controls Window

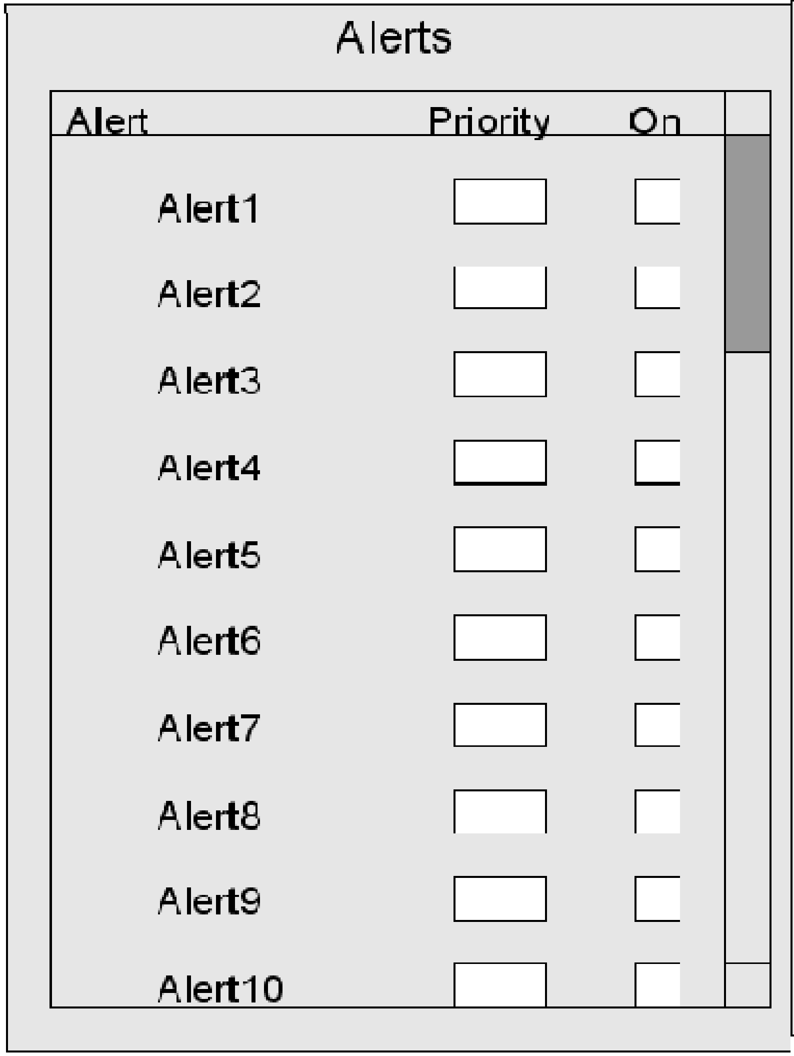


Figure 10e: Alert Settings Window

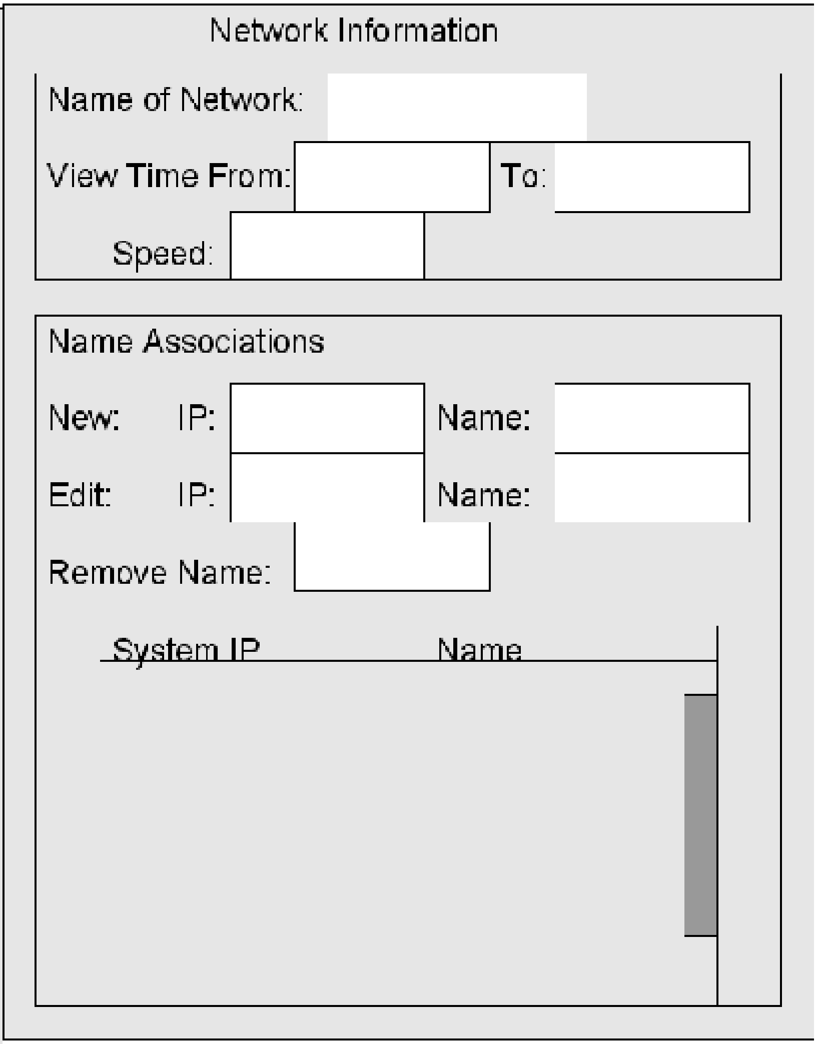


Figure 10d: Network Info. Window

Alerts

All alerts will have both an audible and visual component that will go off whenever that alert goes off, unless the user has designated otherwise through the different options available discussed above. The visual alert will be color coded Blue for good alerts, like Snort has just finished begin updated, Yellow for minor alerts, Orange for Moderate alerts, and Red for important alerts. These colored alerts will cause a ring shape to appear on the map where the affected system is located and a smaller circle will appear and flash on the mini map. Then the system or systems that are the subject of the alert will start flashing that color until the user clicks on an effected system. Also, a small icon box the color of the alert will appear on the far left side of the screen which if clicked will either cause the camera to go to the location of the effected system or cause the location to flash again on both the map and mini map. Audible alerts may be turned off in sound options in the 'Main menu' while the visual flash can only be turned off if the alert is turned off.

Back End Systems

In order to make the architecture as adaptable and scalable as possible a means of allowing for multiple Intrusion Detection Systems to share data about the network in such a way that it can all be collected correlated and distributed to all those in need of it for analysis is needed. Also, the back end systems were selected and set up so that they could still work separately if needed, as well as to accommodate future upgrades. Considering this and with some help, from others and other projects, the partnership of Snort, SQL, and Torque Game Builder was conceived. (See figure 11)

Snort Intrusion Detection System

Snort is an open source network intrusion prevention and detection system utilizing a rule-driven language, which combines the benefits of signature, protocol and anomaly based inspection methods. With millions of downloads to date, Snort is the most widely deployed intrusion detection and prevention technology worldwide and has become the de facto standard for the industry, as taken from the snort website.[19] Snort does all of the data collection for the project. Weather or not more then one System with snort is set up in order to cover a network does not matter as long as the systems can send the collected data to the SQL database the data can be aggregated, filtered and interpreted.

SQL Database

The next system in the back end design is SQL. SQL stands for "Structured Query Language". This language allows us to pose complex questions of a database. It also provides a means of creating databases. SQL is very widely used with many database products supporting SQL, this means that if you learn how to use SQL you can apply this knowledge to MS Access or SQL Server or to Oracle or Ingres, as well as countless other databases. [20] The SQL database is the central data storage for the project and will most likely be the MySQL version of SQL in order to accommodated Torque as well as the robust amount of data that needs to be handled from the IDS. Both the SQL and Torque server are located on the same machine locally and Torque will only query what it needs from the SQL database as needed. This will require additional system resources since both the SQL and Torque server will most likely be separate in nature, but will allow for greater flexibility within the system.

Torque Game Builder

Next is the back end software developed from Torque Game Builder. Torque Game Builder is an incredibly adaptive tool with it's own Client Server architecture. This allows

Torque to quickly and easily support multiple clients running the visualization software, also developed in Torque, simultaneously. This way the server can take care of the data with SQL, in fact the Torque server will be completely oblivious to the fact that the IDSs exist. Therefore if there is some experimental data or training data, etc. from another time or network, the system just needs it to be input into the SQL database for the Torque server to access and send to the client to display. This allows for a great amount of flexibility in development for testing and for research in different data sets.

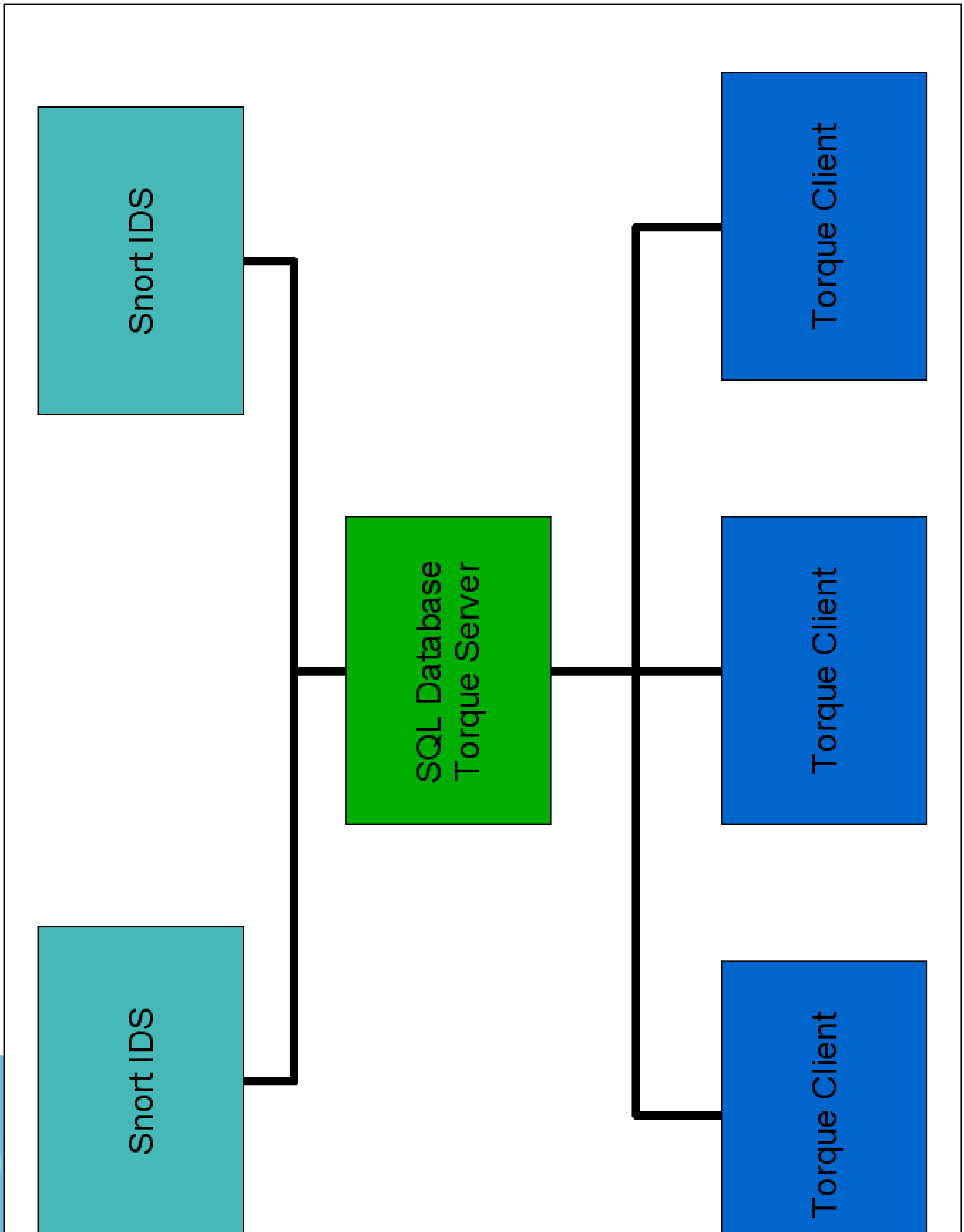


Figure 11: System Network

CHAPTER 4. SUMMARY AND DISCUSSION

The problem of creating a system that can create a functional, adaptive, understandable interactive visualization of network data is by far not an easy one. Continuing research in how to represent data in a visual form for more then just network traffic has yet to find a definitive answer, but few if any look at the potential that the gaming industry presents us. If the interactive and immersive environments of games could be utilized to show data, then a real solution could be much closer to completion then otherwise possible.

Conclusion

It is very important that network data become more comprehensible so that professionals who's job it is to keep a network safe can better find and prevent the intrusion of attacker that may wish to do harm not only to the network and it's systems but also to those people who trust their information to the owner's of the network. As it stands there is too much data and no way to interpret it all, and without a solution networks run the risk of becoming even more vulnerable as their intrusion detection systems become almost useless without the added input from their network administrators and security personnel. As Kofi et al. point out, the function of network intrusion visualization goes far beyond “illustration”. Network intrusion detection can help to improve communication about the data to third parties; it can help the analyst to better explore specific properties of certain network attacks, and it can facilitate the exploration of distributed network attacks [13] Thusly, with the help of lessons learned from past research in network visualization and the added experience of the game industry and their designs for interactive environments, the approach of using Real Time Strategy game design with network intrusion data to create an intuitive and understandable visualization for intrusion detection systems was conceived and is presented here as a possible solution.

Future Work

Now that a design based on Real Time Strategy game architecture has been established that should work for representing a networks communications in an interactive environment the next step is to consider the best types of images to represent each system icon, data links and flows, the background environment, and make considerations for the overall environment. While the design addressed how to make the system interactive and the basics of the visualizations, the imaging is needed for greater mental association and intuitive recognition of the visual environment. For example, buildings for systems and roads to show data flow are some imaging techniques, while the use of forests and rivers is may be a another possibility. Also, when it comes to the over all environment there are certain aspects such as the weather and lighting of the environment that by finding how to link certain information about the network to these environmental functions could make larger and more subtle changes more easily understood by the user. Also implementing the design and test it using collected network data is needed to see if the data representations show enough to make sound conclusions as to what is happening across the network. Additionally, work should be done in actually integrating scripts into the server for interactive environment, which would allow certain users to change settings for the intrusion detection system/s and do general tasks such as remote updates and installs on network systems, allowing the user to make the changes needed directly from the visualization and potentially making it an all around network administration tool.

REFERENCES CITED

[1] Adam Ronald Oline. “Exploring Three-Dimensional Visualization of Intrusion Detection System Alerts and Network Statistics.” M.S. Thesis in Information Assurance. Ames, Iowa. Iowa State University. 2005.

[2] Ben Shneiderman and Aleks Aris. "Network Visualization by Semantic Substrates." HCLI Online Technical Reports. 2006. Human-Computer Interaction Lab, University of Maryland. Dec. 2006. <http://hcil.cs.umd.edu/trs/2006-19/2006-19.pdf>.

[3] Blazej Kot, Burkhard Wuensche, John Grundy, John Hosking. "Information Visualization Utilizing 3D Computer Game Engines Case Study: A Source Code Comprehension Tool." Blazej Kot. Blazej Kot. July 2005. Department of Computer Science, The University of Auckland. Nov. 2006. <http://www.cs.cornell.edu/~bjk45/publications/papers/KotWuenscheEtAl_SourceCodeComprehension.pdf>.

[4] Bradley Huffaker, Evi Nemeth, k claffy . "Otter: A general-purpose network visualization tool." CAIDA. 2006. CAIDA: Cooperative Association for Internet Data Analysis. Nov. 2006. <http://www.caida.org/publications/papers/1999/otter/otter.html>.

[5] Brown J.A., McGregor A.J., Braun H-W. "Network Performance Visualization: Insight Through Animation." Proceedings PAM2000 Passive and Active Measurement Workshop, Hamilton, New Zealand, pp. 33-41, Apr. 2000

[6] Chris Muelder, Kwan-Liu Ma, and Tony Bartoletti. "Interactive Visualization for Network and Port Scan Detection." Kwan-Liu Ma. Kwan-Liu Ma. 2005. University of California, Davis Lawrence Livermore National Laboratory. Dec. 2006. <http://www.cs.ucdavis.edu/~ma/papers/raid05.pdf>.

[7] Daniel Johnson and Janet Wiles. "Effective Affective User Interface Design in Games." Daniel Johnson – PhD confirmation. 2001. University of Queensland, Australia. Nov. 2006. <http://www.itee.uq.edu.au/~uqdjohns/publications/Flow/CAHD2k1_Flow.pdf>.

[8] Hans Christian Arnseth. "Learning to Play or Playing to Learn - A Critical Account of the Models of Communication Informing Educational Research on Computer Gameplay." Game Studies. Dec. 2006. Institute for Educational Research, University of Oslo, Norway. Jan. 2007. <http://gamestudies.org/0601/articles/arnseth>.

[9] Hideki Doike and Kazuhiro Ohno. "SnortView: visualization system of snort logs." Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. Washington DC, USA. Pages 143-147. 2004.

[10] James Blustein, Chinig-Lung Fu, Daniel L. Silver. "Information visualization for an intrusion detection system." Proceedings of the sixteenth ACM conference on Hypertext and hypermedia. Salzburn, Austria. Pages: 278-279. 2005.

[11] John P. Davis, Keith Steury, and Randy Pagulayan. "A survey method for assessing perceptions of a game: The consumer playtest in game design." Game Studies. Oct. 2005. Microsoft Game Studios. Jan. 2007. <http://gamestudies.org/0501/davis_steury_pagulayan/>.

[12] Justin Richer and Jill L. Drury. "A Video Game-based Framework for Analyzing Human-robot Interaction: Characterizing Interface Design in Real-time Interactive Multimedia Applications." Proceeding of the 1st ACM SIGCHI/SIGART conference on Human-robot Interaction. Salt Lake City, Utah, USA. Pages 266-273. 2006.

[13] Kofi Nyarko, Tanya Capers, Craig Scott, Kemi Ladeji-Osias. "Network Intrusion Visualization with NIVA, an Intrusion Detection Visual Analyzer with Haptic Integration." IEEE Xplore. 2002. EVSC Laboratories, Morgan State University. Nov. 2006. <http://ieeexplore.ieee.org/iel5/7836/21555/00998969.pdf?arnumber=998969>.

[14] Cheops Network User Interface. Mark Spencer. 1999. Nov. 2006. <http://www.marko.net/cheops/index.shtml>.

[15] Markku Eskelinen. "The Gaming Situation." Game Studies. July 2001. Jan. 2007. <http://www.gamestudies.org/0101/eskelinen/>.

[16] Rebecca Bace and Peter Mell. "Intrusion Detection Systems." November 2001. NIST Special Publication on Intrusion Detection Systems. Infidel, Inc., Scotts Vallye, CA. .

[17] Robert Ball, Blenn A. Fink, and Chris North. "Home-centric visualization of network traffic for security administration." Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. Washington DC, USA. Pages: 55-64. 2004.

[18] Robert F. Erbacher. "Visual Behavior Characterization for Intrusion Detection in Large Scale Systems." Dr. Rob's Page. Dr. Robert F. Erbacher. Department of Computer Science, University of Albany. Nov. 2006. <http://www.cs.albany.edu/~erbacher/publications/SecurityVisPaper2-VIIP01color.pdf>.

[19] Snort. Sourcefire. April 2007. <http://www.snort.org/>.

[20] SQL Database Reference Material. April 2007. <http://www.sql.org/>.

[21] Torque Game Builder. 2000 Garage Games. Jan. 2007. <http://www.garagegames.com/products/torque/tgb/>.