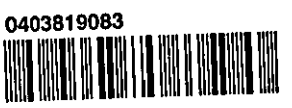


Visualising Network Security Attacks with Multiple 3D Visualisation and False Alert Classification

By
Shahrulniza Musa

A Doctoral Thesis Submitted in partial fulfilment of the requirements for the award of the degree Doctor of Philosophy of Loughborough University

(April 2008)

Acknowledgements

First and foremost, I am deeply indebted to my supervisor, Professor David Parish. Without his support and vision, I would not be able to complete this thesis. I appreciate him for giving me the flexibility in doing this project. I would like also to thank him for sponsoring my papers presentation at international conference.

I would also like to thank Dr. Mark Withall for introducing me to the 'Python' and the 'visualisation' worlds. Not to forget also other researchers in the HSN lab especially to Yaaqob, Kostas, Akthar and Dr. Shiru De-Silva.

I would also like to thank MARA and University Kuala Lumpur for sponsoring my study in UK. Without their sponsorship, this dream may not come through.

Finally, I am eternally grateful to my dear wife for her love and sacrifices. I thank her for the countless supports during the past years.

List of Publications

Conference Papers

[1] Musa,S.; Parish, D.J. "Visualising Communication Network Security Attacks" in 11th. International Conference on Information Visualisation (IV07), Zurich, Switzerland, IEEE Computer Society Press, pp. 726-733 4-6 July 2007 – Paper

[2] Musa,S.; Parish, D.J. "Integrating a False Alert Classifier with Network Security Data Visualisation" in 11th. International Conference on Information Visualisation (IV07), Zurich, Switzerland, 4-6 July 2007 – Poster

[3] Musa,S.; Parish, D.J. "Using Time Series 3D AlertGraph and False Alert Classification to Analyse Snort Alerts" Vizsec 2008 Workshop on Visualisation for Computer Security, MIT Cambridge, MA USA, Springer LNCS, pp. 169-180 15 September , 2008 – Paper

Abstract

Increasing numbers of alerts produced by network intrusion detection systems (NIDS) have burdened the job of security analysts especially in identifying and responding to them. The tasks of exploring and analysing large quantities of communication network security data are also difficult. This thesis studied the application of visualisation in combination with alerts classifier to make the exploring and understanding of network security alerts data faster and easier. The prototype software, NSAViz, has been developed to visualise and to provide an intuitive presentation of the network security alerts data using interactive 3D visuals with an integration of a false alert classifier. The needs analysis of this prototype was based on the suggested needs of network security analyst's tasks as seen in the literatures. The prototype software incorporates various projections of the alert data in 3D displays. The overview was plotted in a 3D plot named as "time series 3D AlertGraph" which was an extension of the 2D histographs into 3D. The 3D AlertGraph was effectively summarised the alerts data and gave the overview of the network security status. Filtering, drill-down and playback of the alerts at variable speed were incorporated to strengthen the analysis. Real-time visual observation was also included.

To identify true alerts from all alerts represents the main task of the network security analyst. This prototype software was integrated with a false alert classifier using a classification tree based on C4.5 classification algorithm to classify the alerts into true and false. Users can add new samples and edit the existing classifier training sample. The classifier performance was measured using k-fold cross-validation technique. The results showed the classifier was able to remove noise in the visualisation, thus making the pattern of the true alerts to emerge. It also highlighted the true alerts in the visualisation.

Finally, a user evaluation was conducted to find the usability problems in the tool and to measure its effectiveness. The feedbacks showed the tools had successfully helped the task of the security analyst and increased the security

awareness in their supervised network. From this research, the task of exploring and analysing a large amount of network security data becomes easier and the true attacks can be identified using the prototype visualisation tools. Visualisation techniques and false alert classification are helpful in exploring and analysing network security data.

Keywords

Network Security Visualization, Visualization for Cybersecurity, Visualization for Computer Security, Information Visualization, Intrusion Detection Alert Visualization, False Alert Classification, Human Computer Interaction

Used Acronyms / Abbreviations

ACK	Acknowledgment
CGI	Common Gateway Interface
CIDR	Classless InterDomain Routing
DOS	Denial of Service Attacks
DPORT	Destination Port
DSTIP	Destination IP
HCI	Human Computer Interaction
HIDS	Host Intrusion detection system
ICMP	Internet Control Message Protocol
IDS	Intrusion detection system
IP	Internet Protocol
ISP	Internet Service Provider
NIDS	Network Intrusion detection system
RIP	Routing Information Protocol
SRCIP	Source IP
SYN	Synchronize
SYN-ACK	Synchronize- Acknowledgment
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

Table of Contents

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Introduction

Internet security threats are fast-growing. According to the Computing Technology Industry Association (CompTIA), the number of organisations reporting they have suffered internet-based attacks increased significantly for the third consecutive year [1]. Almost every day, there will be network attacks or the spreading of viruses or worms in the Internet. Symantec's Threat Team reported in their biannual report on Internet Threats that secret networks were recruiting more than 30,000 personal computers each day to spread spam and viruses in the first 6-month of 2004 [2].

To detect these threats, network administrators normally rely on an intrusion detection system [3]. The Intrusion detection system (IDS) is a system that checks network traffic and warns the network administrator of possible intrusions and network attacks. The alerting is usually undertaken using e-mail messages and log files. However, the IDS has also some drawbacks. For example, they are known to produce many alerts and at the same time high rates of false alarms [4, 5].

The numerous alerts produced by all intrusion detection systems have increased the tasks of network security analysts. Network security analysts need to analyse these alerts and identify the true alerts among them. Considering the massive volume and the "textual form" of these alerts, to perform security analyst tasks will need a lot of time and skill.

Burdened with many alerts and logs to analyse, the network security analysts or network administrators have to identify the true attacks as early as possible. Early response to true alerts and not wasting time on false ones may protect a company from losses caused by malicious attacks. Using visualisation to display the alerts in a creative way can help the security analysts or network administrators to understand, explore, and respond to the numerous alerts faster.

1.2 Security Visualisation

Giving a visual perspective of network security data has been the aim of many researchers in this area. The success of visualisation in fields such as medicine, weather and atmospheric study, archaeology, chemistry, financial analysis and airport security motivates further research in this area. Security visualisation is an area of research drawn from information visualisation. According to Card, Mackinlay and Shneiderman [6], information visualisation is *"the use of computer-supported, interactive, visual representations of data to amplify cognition"*.

Visualisation tools are a way to transfer information from a set of data through images. The visual perspective of data allows users to understand great amounts of data in a shorter time [7]. The network security data includes Tcpdump traffic files, network event log files and event logs produced by an intrusion detection system. Thus, security visualisation is an area where researchers try to give meaningful visual images from seas of network security data.

The main objective of security visualisation is to make exploring huge security related logs or network traffic faster. It is to view abnormal patterns of network traffic that are related to intrusions. With visualisation, the network security analyst can explore the information or specific properties of a network attack better [8]. According to Erbacher [9], visualisation techniques allow the network administrator to identify the behaviour of users connecting to a network with bad motives.

1.3 Advantages of the Visual Method

The advantages of using a visual image for huge and large data are indisputable. An old proverb says that *"a picture says more than a thousand words"*. Yin [10] said that human mind has fast visual processing in comparison to data mining powers.

Visualisation also offers a method for seeing the unseen. It creates visual metaphors so that information is represented as visual symbols. Visual metaphors also allow efficient knowledge extraction and help one to gain more insights in comparison to the information presented in textual form [11].

Ball [12] suggested that to have an effective visualisation, *"one should take advantage of the parallel and pre-attentive nature of visual-spatial cognitive modality"*. Researchers in psychology have shown that humans can process pictures, which is a parallel process, faster than text, which is a serial process. Also, it is easier for humans to remember as humans think and learn visually [13].

With fast-growing threats in the Internet and huge amounts of data to be analysed, the critical missions of network security analysts and network administrators are to identify threats faster and to react quickly to them [14]. A visualisation tool can be an effective way to explore vast amounts of network security data.

1.4 Statement of the Problem

Increasing numbers of alerts produced by network intrusion detection systems (NIDS) have increased the job of security analysts to identify the true alert among them. In a large-scale network, there may be millions of alerts or alarms each day. These alerts are normally presented in "text form" and most of them are false positive alerts [5]. Considering the massive volume of alerts received, it is impossible for a human being to deal with all of them manually. This becomes even worse as there is a lack of experts to identify the true alerts among the thousands of alerts received.

This thesis researches the use of visualisation to represent network security attack data for effective monitoring, exploration and analysis. To prove this, prototype visualisation software has been developed to visualise the network security attack data effectively. A classification tree algorithm has been

studied based on the C4.5 classification algorithm with visualisation techniques to identify true alerts. Lastly, a user evaluation has been conducted to evaluate the effectiveness and usability problems of this prototype.

The prototype visualisation software visualised the communication network security data with various displays in 3D allowing filtering, drill-down and animation to give visual insight of the data. The displays were a 3D AlertGraph, scatter plot, parallel coordinates plot, timeline view, plane view and geographical view. In the visualisation software, data such as alerts and network hosts were displayed as objects that gave abstract information regarding its value. Glyphs, lines, colour-coding and visual positioning were normally used to give abstract information regarding the attacks.

In this research, network security data produced by Snort [15] was used as the source of the data and the prototype was designed to use data specifically from Snort. To use data from different types of IDS is another topic of research which includes data fusion. The scope of this research is limited to the study and analysis of the alerts produced by Snort IDS using visualisation and classification techniques to uncover the network security threats.

To solve the research problem, answers to the following questions was sought:

1. How should network security information be visualised to aid user analysis of network attacks?
2. Does the proposed visualisation tool effectively inform user regarding the security status of the supervised network?
3. Is the visualisation tool scalable to receive huge amounts of network security data and representing them?
4. Is the user able to recognise the patterns related to network security attacks?
5. Is interactivity with the visualisation tool sufficiently addressed i.e drill down, IP address etc.?
6. Does the visualisation tool inform users the host being attacked and

the probable attackers?

1.5 Thesis Contributions

This thesis proposes solutions for the problems outlined in the previous section, and provides the following contributions.

- Designing novel 3D visuals of network security alerts for better viewing and understanding of the data. The designs were the time series 3D AlertGraph, the alerts scatter plot view and the alerts timeline view.
- Using a classification tree algorithm based on C4.5 classification algorithm with visualisation to help users to identify true alerts.
- Using animation, real-time monitoring, interaction, user-friendly graphical user interface, drill-down and filter with the visualisation to get better insight of the network security alerts data.

1.6 Thesis Outline

This thesis is organised into seven chapters. The first chapter introduces the research problems, gives a statement of the problem and the research conducted. The scope of this research is also explained.

The second chapter introduces background knowledge in information visualisation, visualisation techniques, network security data, and related work by other researchers in network security visualisation.

The third chapter explains the need analysis, the fulfilling needs, the visualisation design and the details of the visualisation features. The justification of the design is further discussed.

The fourth chapter explains the use of a classification tree algorithm in the visualisation prototype. The choice, method, performance and accuracy of the classifier are discussed.

The fifth chapter considers on the simulation framework and the dataset used. Then, some of the results from the attack examples and the patterns of the known attacks are shown.

The sixth chapter explains the current practice of user evaluation. The evaluation method and result are then explained. Conclusions and the suggested improvements are discussed.

The last chapter layouts the conclusions, research contributions and suggestions for future work.

Chapter 2: Network Security and Information Visualisation

This chapter gives background information regarding network security and information visualisation. The objectives and the common techniques in network attack are explained. Then, the use of information visualisation in network security to unveil network attacks is discussed.

2.1 Understanding Network Attacks and Threats

Security threats can be in various forms and may have diverse effects. The attackers target might be to cause destruction, to interrupt services, to compromise and to steal information or resources, and to tamper or to alter information. The attackers, also known as hackers, create malicious programs and methods to accomplish their goals. Hackers will exploit weaknesses and vulnerabilities in the network transport protocols and in the application software to compromise a computer system. The hacker may also spend some time gathering essential information regarding a computer network before launching an attack.

In information gathering, the hackers normally use the following techniques:

- Port Scanner: A port scanner is a program that automatically detects any open port in a remote system. Port scanning is not an attack but is analogous to a thief looking for unlocked doors or windows of a house [16]. It is an attempt to identify any port that is opened which the attacker can access into the machine. Some of the available scanners are TCP scan, SYN scan and UDP scan; each of them having different scanning techniques.

- Packet Sniffing: A sniffer is software that captures 'packets' that travel along a network [17]. The sniffer software puts the network interface device into a 'promiscuous' mode and captures all network traffic packets. By looking into the content of the packets, hackers can reveal valuable information such as user names, passwords, addresses and any information passing through the

internet.

In the following sections, typical hacker techniques in launching attacks are explained. The objective is to give an understanding of how the hackers exploit the weaknesses of the internet protocol that make the network vulnerable.

2.1.1 Denial of Service Attack

A denial of service attack is a security breach of a computer system that does not usually result in the theft of information or other security losses. Denial of service attacks (DOS) will lead to the unavailability of services of the targeted machine. The targeted machine such as an internet service provider (ISP) system, online banking system, e-commence system or any web hosting server will be out of service. It is done by exploiting the targeted machine resources at the maximum rate possible and therefore crashing the system [18]. There are varieties of known DOS attack. Some of the famous DOS attacks are explained below.

2.1.1.1 Flooding Attack

The Flooding Attack can be categorised into SYN flooding and UDP flooding. SYN flooding exploits the TCP Three-Way Handshake mechanism [19]. The three-way handshake stages are sending SYN, receiving SYN-ACK and sending ACK. The attacker simply sends a large number of SYNs without responding to the receiver ACK. Thus, the targeted machine will have stacks of awaiting TCP connections. As half-open connections use system resources, the targeted machine will be saturated and will reject any new connections.

UDP flooding is a DOS attack that exploits the unconnected mode of the UDP protocol. It creates a UDP packet storm to a single machine or to a group of machines in a network. Such attacks will increase network congestion, consume the whole network bandwidth and leaving only a tiny part or none for the TCP protocol. Therefore, all new connections will be rejected.

2.1.1.2 Smurf Attack

The Smurf Attack exploits the Internet Control Message Protocol (ICMP), which enables users to send an echo packet to a remote host to check whether it is alive or not. A smurf program builds a network packet that appears to originate from a false address (IP spoofing) [20]. The packet contains an ICMP ping message that is addressed to an IP broadcast address, meaning all IP addresses in a given network. The echo responses to the ping message are then sent back to the "victim" address. Many pings and resultant echoes will flood the network and make it unusable for real traffic. A smurf attack is also known as a simple Distributed Denial of Service (DDOS) attack.

2.1.2 IP Spoofing

IP Spoofing is a technique used by hackers to gain access to a computer system using a faked IP address [19]. The technique consists of an intruder sending a message to a victim computer with a forged IP address that appears to be from a trusted host. In this case, the intruder needs to modify the packet header by changing the source IP address of the packet with a trusted IP address host.

2.1.3 Routing Attack

The Routing Attack is an attack that uses the Routing Information Protocol (RIP) in the router [21]. RIP distributes routing information within networks, such as shortest-paths, and advertising routes out from the local network. The intruder forges a RIP packet, and claims his host has the fastest path out of the network. All packets sent out from that network would then be routed through his host, where the packets could be modified or examined. The intruder can also use RIP to send all traffic to a particular host to be sent to the attacker's machine instead.

2.1.4 Malicious Code

Malicious code is a program that damages the computer system. It is available either in the form of viruses, Trojan horses or worms. A computer virus is a malicious executable code that is attached to another executable program [22]. When the program, which the virus is attached to, is executed, the virus program will be secretly executed and will perform its destructive actions such as deleting files and causing the system to be unusable. Some viruses do not perform any malicious activity but instead spread themselves to other systems.

A Trojan horse is a malicious program that disguises itself as a legitimate program [22]. The victim may receive an e-mail believed to be from a legitimate source with a pretended useful program attached. When the victim opens the attached program, it will instead install the virus on to the computer. The effect of a Trojan horse is similar to a virus and sometimes; it will install a backdoor program[1] that the hackers can use to gain access to the computer remotely.

A worm is similar to a virus. The different between a worm and a virus is that the worm can spread itself without any help from humans [22]. As the main feature of a worm is the ability to reproduce itself, the major effect of worms is the burst of memory or network bandwidth used. This will cause web servers, network servers, and individual computers to stop responding.

2.1.5 Spam

Spam is unsolicited e-mails or junk e-mails in the internet. The effect will be high consumption of the network bandwidth and will then congest the network traffic. Other effects from spamming can be identified as lost productivity, maintenance, resources and sometimes fraud.

[1] A backdoor program, also called a trapdoor, is a way of gaining access to a program, online service or an entire computer system.

2.2 Intrusion Detection Systems

The objective of this section is to give an overview of an intrusion detection system. An Intrusion detection system (IDS) is a system that monitors network traffic for suspicious network activities and produces alerts to the network administrator. The IDS is analogous to a burglar alarm installed in a house or theft alarm installed in a car. The purpose is to defend a system by alerting the network administrator that the system has been compromised or attacked. The network administrator then should respond to the alert and make the necessary action to resolve the problem. The alerting may be in the form of e-mail messages and at the same time, the IDS will produce a series of log files for recording and investigation purposes.

Basically there are two types of monitoring strategies for IDS which are the network intrusion detection system (NIDS), and the host intrusion detection system (HIDS) [23]. NIDS monitors all inbound and outbound traffic from all devices in a specific network. NIDS should be placed at a strategic point to monitor all traffic. On the other hand, HIDS is a stand-alone system running individually on a workstation and monitors traffic to and from its own device only.

IDS detect intrusion and abnormal activities either by misuse detection (signature detection) or by anomaly detection [23]. Signature based IDS checks network packets, and compares them against a signature database of known threats such as in antivirus software. In an anomaly-based IDS, the IDS will check the packet and will then compare it against a baseline. A baseline is a constructed base of various parameters of a normal network activity, such as bandwidth, protocol, port and devices used.

Some IDS only detect intrusions and alert the network administrator. They do not block the intrusions but rely on humans to take the necessary action, based on the information they reveal. This is known as passive IDS. Conversely, reactive IDS is IDS that reports intrusions to the network administrator and proactively blocks them.

There are many IDS available commercially and as open source software. There are also some IDS systems especially built for research. This research used an open source IDS, Snort, to produce the intrusion alerts for the visualisation prototype.

Snort is an open source network intrusion detection system which uses a rule-driven language that combines the benefits of signature, protocol and anomaly-based inspection methods. It is able to perform packet logging and real-time traffic analysis on IP networks. Snort claims it is the most widely used intrusion detection technology worldwide, and has become the de facto standard for the intrusion detection industry [15].

2.3 Information Visualisation

2.3.1 Visualisation Terminology

According to Webster's dictionary, visualisation is "*the act or process of interpreting in visual terms or of putting into visible form*". In the visualisation field, there are terminologies such as scientific visualisation, data visualisation and information visualisation. Currently, there is no clear consensus on boundaries that distinguish them, as they share common goals, methods and techniques. In general, scientific visualisation deals with data that has a natural geometric structure (for example, MRI data, wind flows). On the other hand, data and information visualisation are slightly more general terms which handle more abstract data structures such as databases, graphs and include non scientific data, such as financial, marketing and business data. Information visualisation also has been described as the use of advanced computer graphics to present complex data, either to provide an overview, or to search for specific patterns [24]. The term "information visualisation" is suitable for describing this research, as this research focuses on visualising complex network security data for better understanding of that information.

2.3.2 Visualisation Framework

According to Kennedy [24], a basic visualisation framework consists of four components which are the user component, the database component, the visualisation component and the interaction component. The user component describes the user description of the tasks or the needs analysis. The database component represents the actual data in the database and its structures. The visualisation component presents the data mapping and physical layout on the screen. The interaction component addresses the interaction between the user and the data presentation itself. All the four components become the main factors in deciding the style and the complexity of the visualisation.

2.3.3 Effective Visualisation

In developing a visualisation tool, the major challenge is to produce meaningful images from a set of data. Keller [25] suggested three main points that should be carefully thought trough in order to develop an effective visualisation tool. They are

- Identify the visualisation goal. This is identifying the meaning that was sought from the data before an image is constructed.
- Remove the mental roadblock. This means that one's should regard data as nothing more than a set of numbers that was wanted to be visualised.
- Decide between data and phenomena. Data representation shows data values independently of the phenomenon. The viewer needs to deduce the relationship between the data value and the phenomenon.

In the contextual-cue technique, the data values are related to the phenomenon and add meaning to the visualisation. If the phenomenon is the focus of the visualisation, then the choice of colour, shape, texture or setting should create the viewer's experience with the phenomenon [25]. As an example, a red colour will stand out compared to blue, with a black background colour. So, using a red colour to attract viewer attention is recommended [13] if

a dark background colour is used. This is also known as the colour chromatic contrast illusion characteristic. Colour, greyscale and black-and-white are suitable to represent data properties such as values and classification [25].

The use of lines can also suggest a relationship or connectivity [25]. Different line colour, thickness or style (dotted or dashed) can be used in order to indicate distinctions.

A visualisation is considered effective if the visual image that represents the data can be performed "effectively" and "correctly" [26]. The term "correctly" means the visual image reflects the actual quantities and relationships of the actual data, whereas "effectively" means the maximum information the user can perceive in a minimal time.

2.4 Network Security Visualisation Techniques

From the author survey of literatures, there are mainly three types of dataset used in network security visualisation systems. They are

- alerts and logs from IDS,
- raw packets from network traffic collected using Tcpdump or Ethereal [27] programs or router accounting tools such as NetFlow[2] [28], or
- security logs of web server.

Using these datasets, various visualisation tools with different approaches have been developed in the research community to uncover security events. Their techniques are summarised in the following section.

2.4.1 2D and 3D Scatter Plot

A scatter plot is a chart that uses Cartesian coordinates to display values

[2] Netflow is an auditing software that records network traffic flow in the router. It was introduced by CISCO in their routers but later become standard in other commercial routers.

of two or three variables depending on the scatter plot dimensions. The data are displayed as a collection of points, each having one coordinate on each axis. Scatter plots represent the association between the variables but not necessarily a cause and effect relationship. In network security visualisation, researchers adopt this technique but add some new features to improve the diagram. The new features are plotting the points using different colours to show value associated to them, adding lines between two points to show connectivity, and adding animation to show the temporal data attributes of the events.

Abdullah [7] built a visualisation system that presented alarm data from the StealthWatch [29] IDS. In that visualisation, system administrators can get a summary of network activities and from that detect anomalous activities. The visualisation is based on a 2D scatter plot where the vertical axis represents internal IP addresses and the horizontal axis represents a 24 hour timeline. Alarms with different colour-coding will be located according to their IP address and instance of the event. The alarm colour-coding is to show the severity of the attack. The main display is divided into eight columns which allow the display of 163,830 hosts (2.5 times class B IP address space, i.e. 2.5x65,532 hosts). The visualisation includes zoom, filter and glossing (like a tool tip) functions to help users explore the alarms.

"The spinning Cube of potential doom" software developed by Stephen Lau [30] adopted a 3D scatter plot represented within a cube to visualise the connection logs from Bro[3] [31] IDS. The software plots coloured dots using the internal IP address, the port address and all the possible IP address values on each axis. The different colour dots in the cube represent the connection status and mainly focus on unaccomplished connection. The user interacts with the display by zooming in, zooming out and swirling. The software replays the recorded Bro connection log by displaying the successful connections in grey and incomplete connection attempts in rainbow colours. The successful

[3] Bro is an open source, Unix-based Network Intrusion Detection System (NIDS) that passively monitors network traffic and looks for suspicious activity.

connections are coloured in grey to give less visual focus, while the incomplete connections are mapped in rainbow colours to give more attraction. The visualisation relies on the coloured dots to represent connections and assumes that unsuccessful connections are intrusion attempts.

NVisionIP tool [3] is also based on a 2D scatter plot by plotting machines as dots according to their host IP address and their subnet values. The objectives are to give status information accurately and concisely with more details about specific machines, and to be able to process different sources of input. The tool includes filter, drill down and zooming to aid the network security analysts performing their tasks. In drill down, the information is presented in a histogram graph.

McPherson et al. [32] developed a port-based detection tool for security events, known as PortVis that used raw packet header information. The tool plots coloured dots that represent port activities according to port number value and time. The main display is based on a 2D scatter plot, and the tool includes five other displays simultaneously. Selecting an area in the main display will show the port numbers used. Then, selecting a port number will show the port activity in a histogram graph. Using this selection technique, the user is allowed to view the same data with different levels of details. Another two displays are for colouring and visualisation settings.

Ren and others [33] designed an interactive visualisation system, IDGraph by plotting the aggregated number of unsuccessful TCP connections on the vertical axis versus the time ordered sequence on the horizontal axis of a 2D plot. Then the brightness of the mapped point was changed according to the density of the data at each pixel. The mapped data was either the pair of source IP-destination port, source IP-destination IP or destination IP-destination Port. This technique was named the histographs technique and was derived from the information mural visualisation technique [34]. In IDGraph, the darker is the point, the higher is the density of the data at that point. By clicking at that point, the traffic traces underneath it will be revealed, such as the list of source IP address and destination port pairs (see the bottom picture in Figure 2-1). This

system has successfully detected varieties of attacks and anomalies such as a port scanning, a worm outbreak and a stealthy TCPSYN flooding.

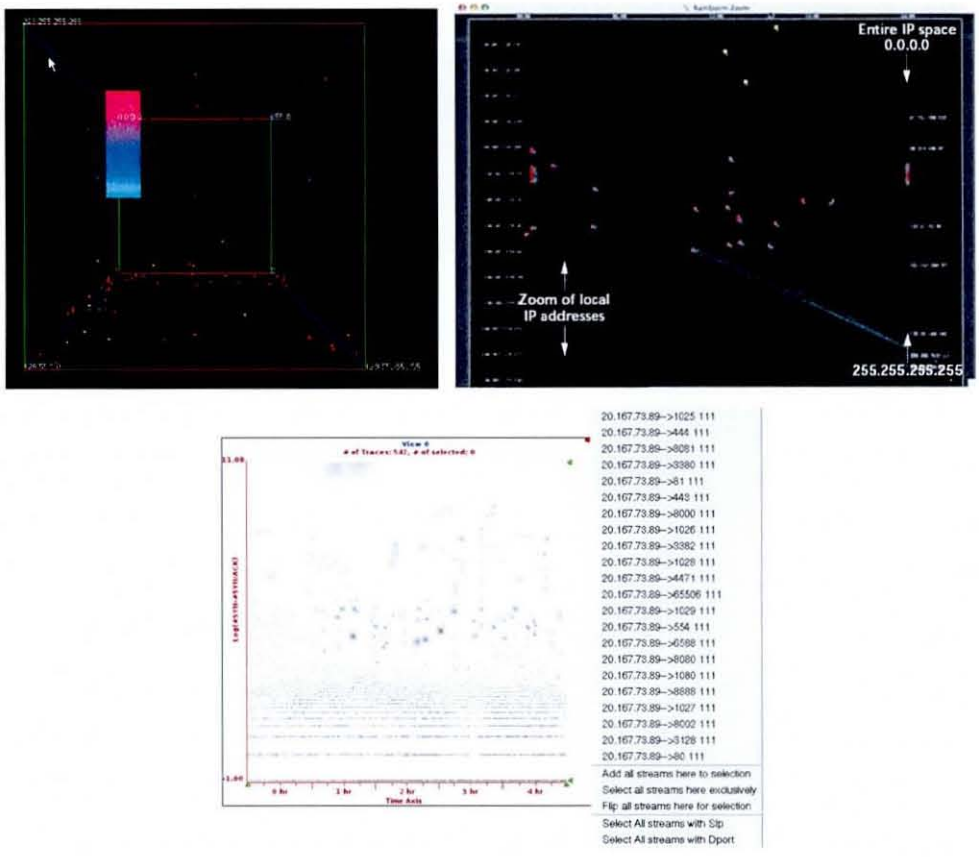


Figure 2-1: Examples of network security visualisation tools using 2D and 3D scatter plots. The picture on the upper left shows port scan attacks and the picture on the upper right shows zooming features in IDS Rainstorm. Furthermore, the bottom picture shows the IDGraph that applied the histograph techniques.

2.4.2 Parallel Coordinates Plot

Another diagram used by researchers is a parallel coordinates plot. A parallel coordinates plot is a technique for plotting multivariate data. In the parallel coordinates plot, a set of parallel axes are drawn for each variable. Then, for a given row of data, a line is drawn to connect the value of that row on each corresponding axis.

VisFlowConnect is prototype software developed by Xiaoxin Yin [10], that

uses connection logs of Netflow to visualise network traffic patterns. The design was based on the parallel coordinates plot which displays NetFlow records as links between two machines or domains. The axes are the originating host of the network traffic, the internal domain and the destination host of the outgoing traffic (see the picture on the left in Figure 2-2). The line thickness and darker colour tone suggest the amount of traffic. VisFlowConnect also has other features such as statistical functions that give the users statistical figures of traffic flow in bytes to and from a host. The users can filter the data by port, protocol, packet size and transfer rate.

Krasser [35] also used parallel coordinates plots in his tools but added time animation and coupled the parallel coordinated plot with scatter plots in 2D and 3D. The result shows the combination provides security analysts with a rapid examination of network traffic and quick detection of network anomalies. The visualisation design consists of two vertical lines. The line on the left is for the source IP address and the line on the right is for the destination port number. The connectivity lines in green show the TCP protocol and lines in blue show the UDP protocol. Glyphs on the left and the right show the age of the packet. The further the glyphs distance from the centre, the older will be the packet and the height of the glyph suggests the packet size (see the picture on the right in Figure 2-2).

Rumint is a system that visualises raw network packets in Tcpdump format into a seven visualisations view [36]. The objective of this system is to analyse the underlying network packets that cause the alarm. The seven visualisations are the scrolling text, parallel coordinates plot, glyph-based animation, thumbnail toolbar, binary rainfall visualisation, byte frequency and detail displays.

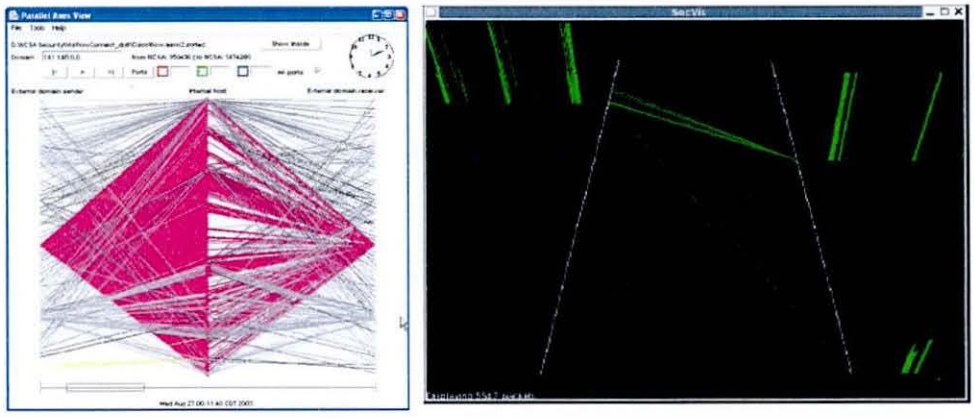


Figure 2-2: On the left, a visualisation from VisFlowConnect shows a virus outbreak pattern, and on the right is the visualisation snapshot from Krasser.

2.4.3 Circle View

CircleView is a new approach for visualizing multidimensional time-referenced data sets proposed by Keim [37]. The work by Livnat [38] used ideas from CircleView in their visualisation (see Figure 2-3). Livnat [38] presented a novel paradigm for visual correlation of Snort alerts. This paradigm provides the situational awareness in complex network environments. The approach is based on the intuitive of *what, when*, and *where*, which they term as w^3. In their system, alerts are grouped and viewed hierarchically with their *what,* and *where* attributes. Then the *when* attribute reveals the temporal distribution of attack trends.

The visualisation display is designed in a circular form, which consists of local network topology in the centre and the alerts are grouped according to the alert types in the surrounding ring. The time is indicated by the radius of the ring. Moving from the inner ring towards the outside ring is like moving back in history. In other words, alerts in the inner ring are later than those in the outside ring. The internal node size is proportionate to the quantity of alerts it receives and the line shows the alert type.

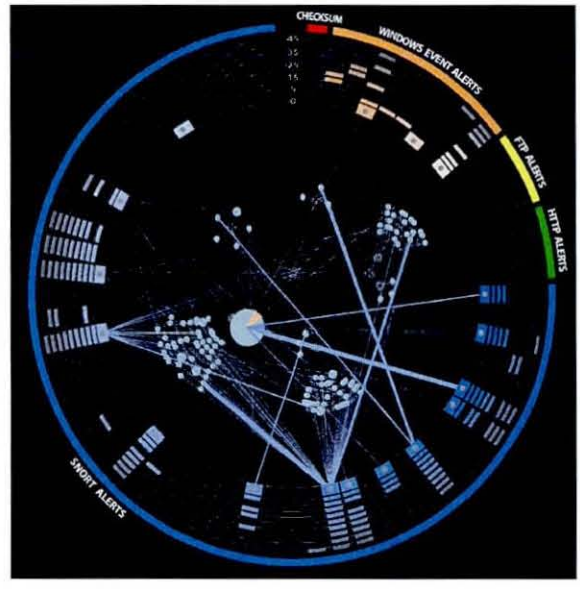


Figure 2-3: Example of VisAlert visualisation using CircleView idea.

2.4.4 Matrix and Graphs

Matrix and graphs are another visualisation technique used by researchers in this area to explore network security data. Koike [4] used complex tables with symbols and colours to present an overview of Snort alerts. The display comprises of three 2D frames, known as the source address frame, alert frame and source-destination matrix frame (see the picture on the left in Figure 2-4). Based on heuristic rules, Koike considers the alerts that are false positives when they appear successively; appearing many times in the entire log, conflicting with the provided services in the network, and alerts concerning other networks. The false alerts detected from the heuristic rules are not filtered but are displayed onscreen to allow the administrators to judge them.

Icons with different shapes are used to classify the attacks and colour-coding is employed to prioritise the alerts. Event time, alert type, attacker source, attack destination, and details of access are considered the essential information to be displayed.

Abdullah [39] in other work, used packet header information to plot histogram graphs and uncover security related events from them (see the

picture on the right in Figure 2-4). The technique involves plotting packet count histograms and stacking them using different colour-codes according to the range of port numbers versus the horizontal time axis. In this technique, any histogram blocks with high height represent anomaly traffic as high histogram blocks suggest high network activity during that period. The dataset used in this research are the Honeynet traffic, because all Honeynet traffic is considered malicious, and this will give a good visualisation illustration.

A Honeynet is a network of computers or servers that is used to attract malicious traffic [40]. They appear to contain information or resources that would be of value to attackers. In practice, they have no production value and therefore should not see any legitimate traffic or activity. However, the traffic seen may also contain benign traffic which was wrongly arrived at the Honeynet due to incorrect configuration etc. As the quantities of benign traffic are small, whatever traffic is captured can then be assumed to be malicious or unauthorized.

Other tools use graphs, tables and pie charts to simplify reporting and querying of alerts. Examples include SnortAlog [41], logparser [42] and analyse console for intrusion detection (ACID) [43].

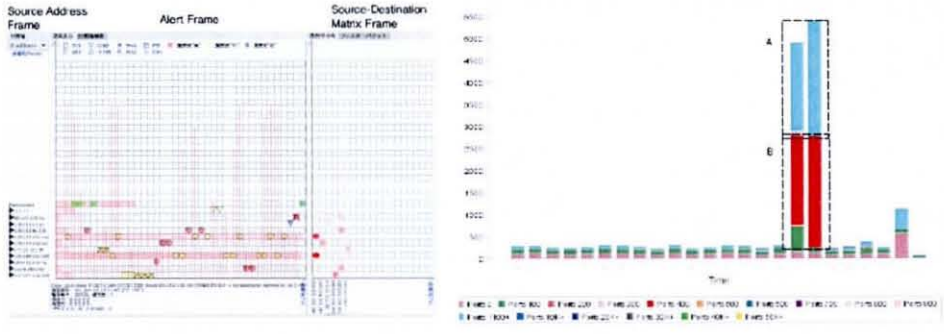


Figure 2-4: On the left, visualisation from SnortView and on the right, example of histogram graph.

2.4.5 Visual Listing

This method consists of visualising information in a 2D display where the

vertical axis is the time and the horizontal axis is the value of the data. Work by J. Colombe [5] used this method with multivariate Bernoulli event representation to convert the text alerts from RealSecure IDS to a visual form. In this method, each comma limited text alarm is converted to a binary form (0 or 1) that marks the presence or absence of a text descriptor in the particular field. The binary string will then form an alarm vector that corresponds to a specific descriptor token. The typicality score of each alert is also calculated. This will give a differentiation of highly typical alerts and anomaly alerts by colour coding.

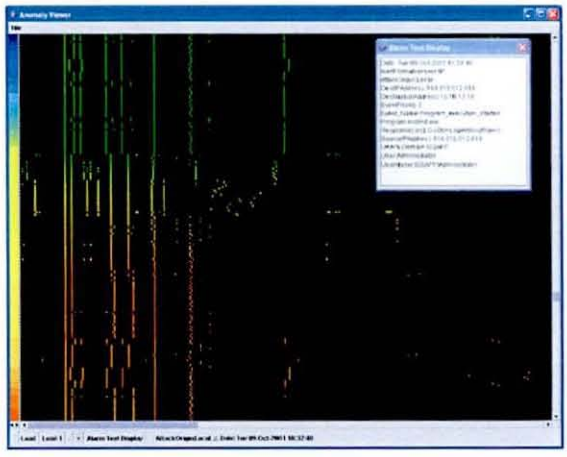
The token vectors are then arranged in chronological order, from the top to the bottom in a visualisation display. Each row will represent an alert and each column shows the present or absence of a specific descriptive token. The illuminating column shows the presence of the token in the alert while the colour suggests the typicality score. An empty background represents the absence of the token. The security analyst gets the details of the alarm in textual form by mouse-click of the specific row.

Work by Stefan Axelsson [44] also used the visual listing idea. The visualisation lists the web server access request logs with a colour-code (see Figure 2-5(a)). The colour-code is achieved by calculating for each alert a token score using a self-learning Bayesian system. The Bayesian system is similar to the approach used for Bayesian e-mail spam detection.

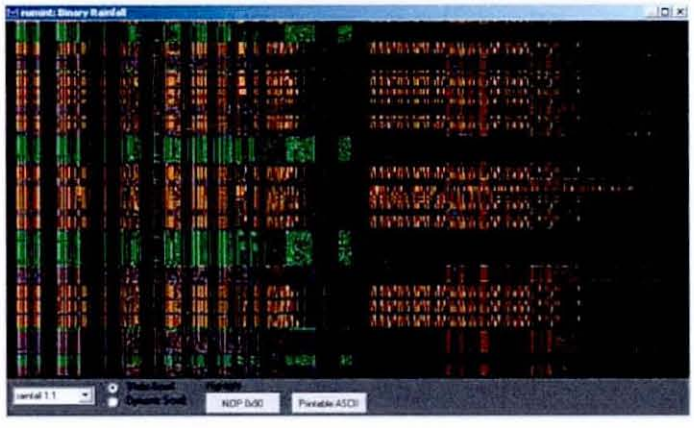
The token score is the probability that a token appears in a malicious message. All alerts that have a token score greater than a threshold will be considered as harmful and intrusive. Based on the token score calculated, colours are mapped from green via yellow to red to show if the message was benign or malicious. The prototype is called the 'Bayesvis' tool.

Rumint [36] also applied this technique in three of its visualisation windows. The scrolling text display visualises network packets, one in each horizontal row according to user encoding preference (ASCII, hexadecimal and decimal) in chronological order. The packet contents are displayed in the binary rainfall visualisation and the byte frequency visualisation as in scrolling text

display (see Figure 2-5(b)).



(a)



(b)

Figure 2-5: (a) Work by Colombe and (b) Example from Rumint. In both pictures, each row represents an alert or a network traffic packet and is in chronological order.

2.4.6 Network layout

This technique involves arranging nodes that represent computer hosts in 3D space and linking them with lines to show connectivity. Nyarko [8] applied this idea and incorporated also ideas from electromagnetics, fluid dynamics and gravitational theory with haptic technologies to provide another visualisation dimension of network intrusion data. The application, network intrusion visualisation application with haptic integration (NIVA), allows the analyst to

examine interactively and to detect attack across time using three-dimensional display. In this model, glyphs are coloured according to their domain and the severity of the attack is represented by the link line colour. An alternative view in NIVA is called a helix technique model (see the picture on the left in Figure 2-6). In this model, glyphs are arranged at the circumference of a helix. The glyph colour shows the domain and the connectivity line colour signals the severity of an attack.

VISUAL [12] also used this technique to visualise the network traffic, based on home centric perspective. The network traffic data are in the Tcpdump format. Each node is represented as a square with internal hosts in the central grid and external hosts outside it (see the picture on the right in Figure 2-6). The line between a pair of hosts shows the connectivity between them. The software includes other information, such as port number, external host activity and timeline filter.

Work by G. Vert [45] presents the network intrusion information using a spicule[4] with different diameter, vector and line to show the security risk of the computer, service and amount of traffic used. To understand the model, let consider bacteria attacks[5] to a computer A as an example. Bacteria attacks are generally localised on a single computer, and are characterised by the use of large amounts of computer resources i.e. high CPU usage and user processes.

When the CPU usage and user processes are high, the CPU usage and user processes vectors move towards the prime axis, y. As CPU utilisation increases, other user processes starve. Therefore, system processes and system files vectors move towards the x-axis. Another visual that associates with the spicule model is a singularity model. This singularity model shows the computer activity relatively to other known computers.

[4] A spicule is a spheroid geometric primitive.

[5] A Bacteria attack is a virus that only replicates itself but did not damage files. However, it can damage the system.

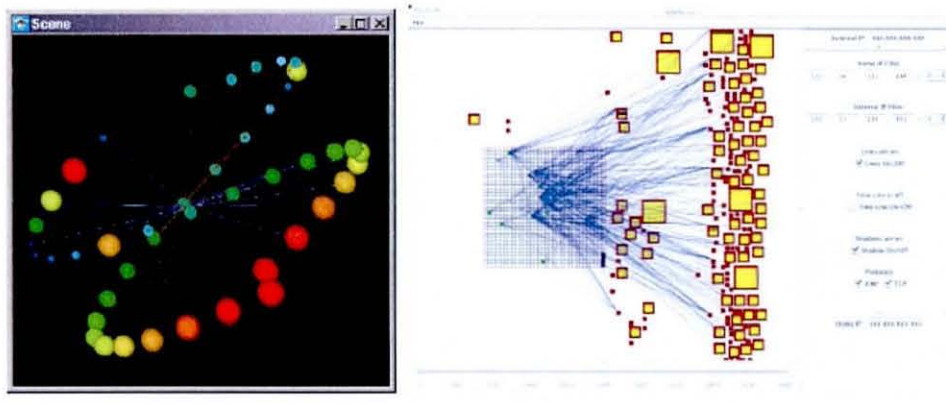


Figure 2-6: On the left, visualisation from NIVA using the helix technique and on the right, a snapshot from VISUAL.

2.5 Summary

Analysis of visualisation tools in the previous section reveals there are some areas that can be improved to make the visualisation more effective. One of the improvements is to link the visualised alert with the raw packet that causes it. Nyarko [8] and Abdullah [7] suggested the use of other tools, such as RealSecure and Black Ice for NIVA, and Rumint for IDSRainstorm, to analyse related packets. The advantage of having this function available in the visualisation is it allows the network security analysts to examine directly the traffic that causes the alert and make a correct judgement of it.

Researchers have adapted various techniques to create an effective visualisation tool. Among the techniques are: the parallel coordinates plot, the 2D and 3D scatter plots, the network layout, the complex matrix and graphs as well as the colour-coding technique. The author considers tools developed by Yin, (VisFlowConnect) [10] and Krasser [35] using the parallel coordinates plot technique. Work by McPherson (PortVis) [32], Lau (The Spinning Cube) [30], Abdullah (IDSRainstorm) [7], Conti (Rumint) [36] and Lakkaraju (NVisionIP) [3] either use the 2D or 3D scatter plot technique. These scatter plot techniques are popular among researchers.

Another popular technique is the network layout in 3D space to present the network security data. This technique has been employed by Nyarko [8] Ball [12] and Vert [45]. This technique sometimes suffers from scalability issues in

comparison to parallel coordinates plot and scatter plot techniques.

Scalability is the main concern in security visualisation because of the huge amounts of data involved and the large number of machines under supervision. Visualisation tools [3, 5, 30, 32] are scalable, as thousands of records can be displayed in one visualisation display. On the other hand, the network layout in 3D space uses objects such as spheres [8, 45] or squares [12] as a computer host, and lines to show communication.

Chapter 3: Designing Visualisation Tools for Network Security Data

This chapter discusses the approach of this visualisation prototype follows. The need analysis of the network security visualisation tool and issues about it are also presented. In addition, the block diagram and the object oriented design of the prototype software are presented. The features of the visualisation prototype, it advantages, and how it fulfils the need analysis are discussed and explained.

3.1 The Need Analysis

The need analysis for this prototype was gathered from various published works in the literature. The findings by Yurcik and others [46] from interviews with the security operators at the National Centre for Supercomputing Applications (NCSA), and two incident response centres, highlighted the need of 'situational awareness'. Their survey found that any visualisation tool should provide the analyst with an overview of the situational awareness of the supervised network. The suggested way to achieve this was to present an overview of the supervised network on a single display.

Paulson [47] also quoting the NSCA researchers stressed that visualisation tools should include forensic support as is impossible for someone to watch the screens all the time. When problems occurred, these tools could easily find out when the events occurred, look at what happened, and drill down for more detail.

In a study, Ball and others [12] determined the needs of the user community by interviewing 22 professional system administrators specialising in computer and network security from two large universities. The outcome from their work suggested that network administrators need to know what was happening in their own network. They were not interested in information about other networks which were not under their responsibilities.

Komlodi [48] identified the visualisation needs and proposed a task

model that would help the security analyst to perform their typical tasks. In this model, the basic analyst tasks consist of monitoring, analysis and response. Monitoring tasks involve checking and identifying the potential true alerts. Analysing tasks involve analysing the potential alerts and diagnosing them. Lastly, the response tasks are the analyst responses to the true attacks. From the findings, Komlodi recommended that visualisation tools should contain features of simple displays, filter options, interaction, multiple view options and reporting tools.

Meanwhile, D'Amico [49] considered the analyst tasks to be either reactive or proactive. According to D'Amico, an ideal visualisation tool should help the analyst to perform both tasks. The reactive mode is where the analyst usually works in real-time. To visualise in real-time, the tool should be able to update data automatically and highlight it according to the alert priority. Conversely, in the proactive mode, the analyst needs data exploration, navigation and animation tools to help identify the anomalous patterns.

In other work, Goodall [50] found similar findings as Komlodi [48] but added the need of analysing not only the packet header information but also the packet payload. Goodall also added that a visualisation tool must be intuitively understood by the user. The Table 3-1 summarises the details of the user need analysis found in the literature.

Table 3-1. The summary of the need analysis for Network Security Visualisation tools.

	Needs
Yurcik and others [46]	• The user needs to get an overview of the monitored system – "*situational awareness*". • The tool should have forensic support.
Ball [12]	• The tools should be able to focus on events happening in the monitored network.
Komlodi [48]	• The tools should address the user needs in performing: o Monitoring tasks o Analysing tasks o Response tasks

D'Amico [49]	• The tools should address the proactive and reactive nature of an analyst's tasks.
Goodall [50]	• The tools should be able to view and analyse the packet payload. • The visual layout should be intuitively understood.

3.2 Some Issues in Network Security Visualisation Tool Design

Apart from the need analysis, researchers should also address other important issues in the network security visualisation. Erbacher [51] states that the main issue in designing network security visualisation tool is scalability. The scalability is an issue because of huge amounts of related data for network security. The data are not only numerous but also of high dimensionality. As an example, port numbers, and the class B IP addresses, can have a range up to 2^{16} (65536).

Another characteristic of network security data is that they are temporally related. This is because some attacks are distributed over time. Therefore, a visualisation tool should address this issue, to allow the tool to expose the attacks. The data should also be studied according to the time sequence of the events. Sometimes, it is the correlation of events over time that distinguishes attacks in concern. A successful network security visualisation tool should be able to address this issue.

Ren [52] mentioned the advantages of using visualisation to uncover network security events were indisputable. However, Ren argued that a good visualisation tool should have the following characteristics:

- Being representational: convey information accurately and effectively.
- Being cognitive: lessen the burden of analysis of huge and complex data, and easy to understand.
- Being analytical: allow the user to investigate and analyse related data.
- Being exploratory: allow walkthrough of the data set and detail on

demand support.

- Being situational aware: allow top level overview of a supervised network with minimum human effort.

3.3 The Design Framework for a Network Security Visualisation Tool

The author proposes a framework for the design of the visualisation tool as in Figure 3-1. The design framework is composed of data storage, data interface, data processing (adaptation and mapping), user interfaces and interaction, and visualisation display. This design framework became the main prototype software architecture. The author noted there were likenesses in the architecture with other visualisation systems, but the author could not find any literature that suggested the most effective one. However, the prototype design framework supports communication between the data and the visualisation engine. It also gives the flexibility to design the GUI and the interactive features.

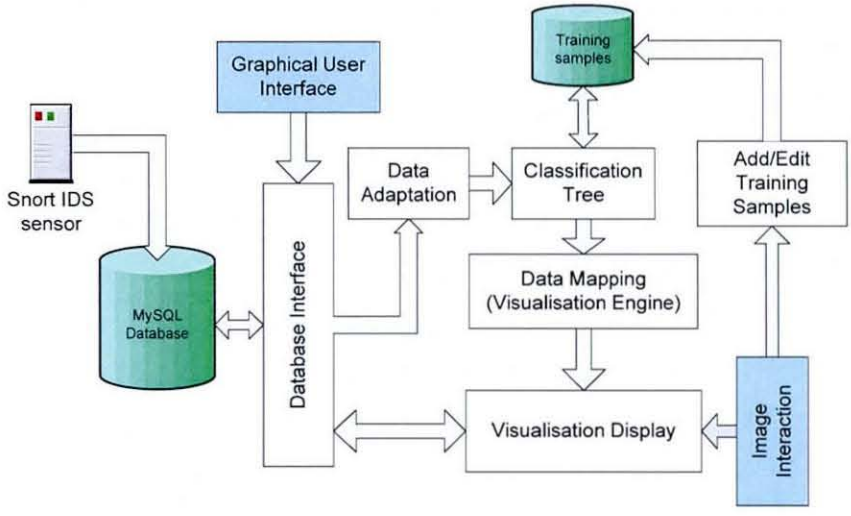


Figure 3-1. Design framework for the visualisation tool.

The IDS environment used in this study was Snort with alert logging to a MySQL database. The Snort engine used the standard signature rules and configuration provided by the Snort community group. The Snort IDS was chosen because of its widely used in the research community and industry, and its ability to log alerts to the database. The Snort IDS can monitor traffic in real-

time and alerts the user when it detects anomalous traffic. Snort can also examine off-line Tcpdump traffic by processing the file in Tcpdump reading mode. However, Snort produces as many alerts as many other IDS.

The database used in this work was the MySQL [53] open source database. MySQL is known for its reliability and reputation as the most used open source database in the world. Using the database storage allows better filtering, querying and fetching of data for the visualisation software.

The visualisation software was developed using the Python [54] programming language. The Python version used is the Python Enthought Edition 2.4.3. Python Enthought edition is a free distribution of Python under the Windows operating system which includes several pre-compiled libraries for scientific data visualisation and manipulation. Python itself is an interpreted, interactive, object-oriented programming language that combines remarkable power with clear syntax. The Python implementation is portable and runs on UNIX, Linux, Windows and Mac OS/X. However, the author used the Windows environment in performing this work. The designed of the graphical user interface (GUI) in the visualisation software was done with the help of wxPython 2.6 module. The wxPython module is an add-on module for python to design the GUI.

The graphical engine used in the software was the Visualisation Toolkits (VTK) [55] version 5.0. The VTK is an open source object oriented software for 3D computer graphics, image processing, and visualisation. VTK is used by thousands of researchers and developers around the world. VTK consists of a C++ class library and interpreted interface layers for Tcl/Tk, Java, and Python.

3.3.1 The Object Oriented Visualisation Software

Python as an advanced scripting programming language has the advantage of object oriented programming. This has enabled the author to develop the prototype software using object oriented software design. The simplified class diagram (see Figure 3-2) shows the relationship of each class in

the software. The class diagram shows there are three main classes in the visualisation prototype. The classes are the GUI controls, the Visualisation classes and the VTK windows classes.

There are three databases in the visualisation software. The databases are the geographical location database (Geo Database), the Snort database and the classifier training sample database. The geographical location database used is from the MaxMind GeoIP database [56]. The GeoIP database is a geographical location database of IP addresses developed and maintained by the commercial company MaxMind. The database is available either with an open source licence or with a commercial licence. The open source version contains less accurate information compare to the commercial one. This project uses the open source licence database and the database is installed locally. The interface between the software and the databases are either through the Snort interface class, GeoLocation Interface class or Update training sample class according to the information requests.

Node, line, label, colour setting, world Map and coordinate are classes to create a 3D image with its specific attributes. The visual interaction of the user with the objects is managed by the Pick Actor class. Meanwhile the machine learning of the software is performed by the Classifier class.

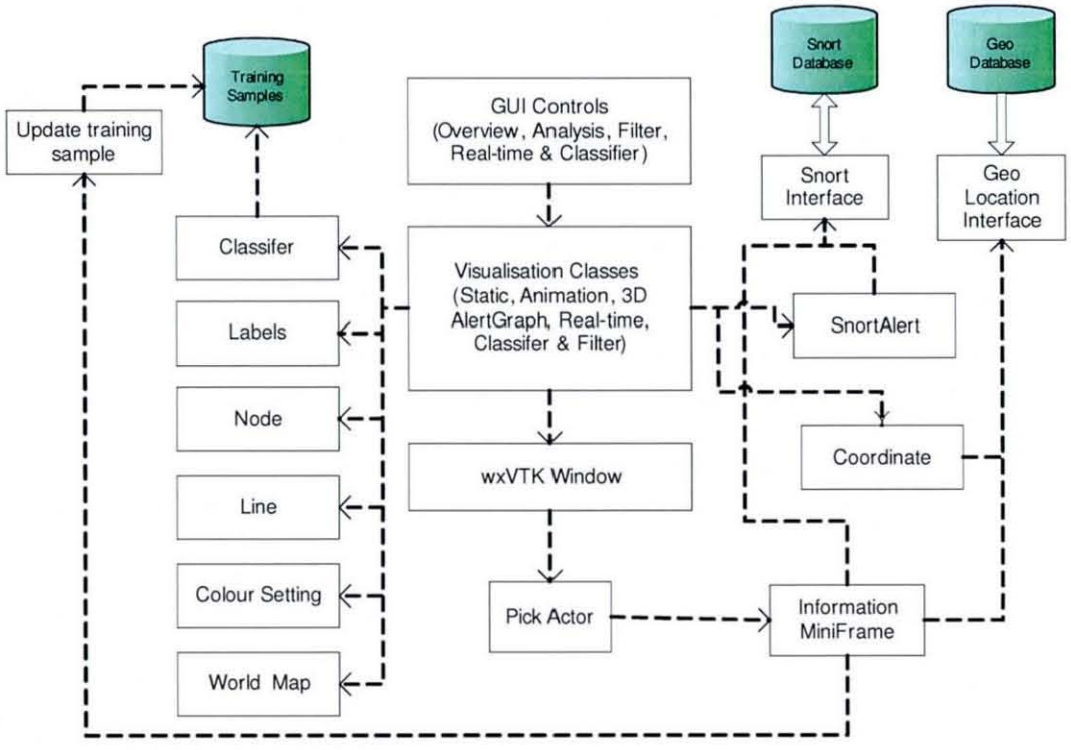


Figure 3-2: The simplified class diagram of the software.

3.3.2 The Data Input

The inputs for the prototype are the data produced by Snort. Snort when detecting suspicious network traffic will produce alerts and log them to the Snort database and logs file. The textual logs file is normally saved in the log directory of the Snort installation. The information from the logs file is minimal compared to the information in the logs database. Figure 3-3 shows the sample of Snort alerts found in the alert logs file.

```
[**] [1:1560:6] WEB-MISC /doc/ access [**]
[Classification: access to a potentially vulnerable web application] [Priority: 2]
04/09-15:19:46.079304 197.218.177.69:16511 -> 172.16.114.50:80
TCP TTL:63 TOS:0x0 ID:5357 IpLen:20 DgmLen:298 DF
***AP*** Seq: 0x97531B46  Ack: 0xDADDDD8F  Win: 0x7D78  TcpLen: 20
[Xref => http://cve.mitre.org/cgi-bin/cvename.cgi?name=1999-0678][Xref =>
http://www.securityfocus.com/bid/318]
```

Figure 3-3. A sample of Snort alert in text form.

The Snort log database provides more information and is easy to access. The relationship diagram of Snort tables used in this visualisation software is

shown in the Figure 3-4. The advantages of using a database apart from accessibility are the portability with other application, ease of querying and of filtering information. The 'defaultclasstype' table is a new table that the author has created in the Snort database. This table refers the Snort signature to its class type. The default class type of each Snort signature is available in the text file in the Snort installation directory. Table 3-2 describes the role of the tables in the Snort database. There are some other tables in the Snort database which are not important here and therefore those tables are not mentioned.

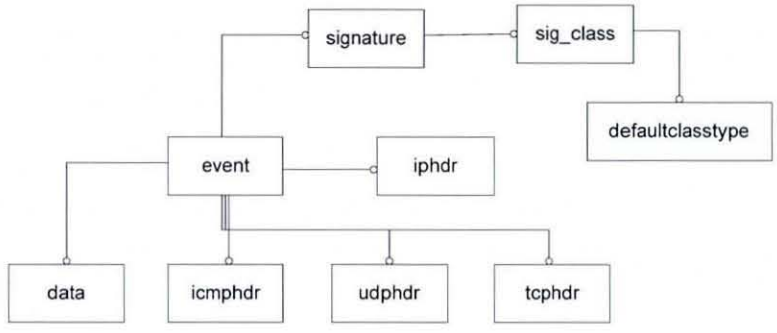


Figure 3-4. The Snort database log structure.

Table 3-2. The description tables in the Snort database.

Table	Description
event	Meta-data about the detected alert
signature	Normalized listing of alert/signature names, priorities, and revision IDs
sig_class	Normalized listing of alert/signature classifications
Data	Contents of packet payload
iphdr	IP protocol fields
tcphdr	TCP protocol fields
udphdr	UDP protocol fields
icmphdr	ICMP protocol fields
defaultclasstype	Classtype reference

The visualisation software extracts the data from various tables in the Snort database and prepares them for the visualisation software. Each alert is represented as a tuple (a finite sequence of information) of 13 fields before sending them to the visualisation engine. The list of the fields is as in the Table

3-3.

Table 3-3. The alert tuple.

Field	Desciption
Sid	The sensor id.
cid	The alert id.
signature	The alert signature id.
timestamp	The date and time.
sig_name	The alert signature name.
sig_priority	The alert signature priority.
ip_src	The source IP address.
ip_dst	The destination IP address.
ip_proto	The IP protocol.
Sport	The source port number.
Dport	The destination port number.
classtype	The class type information
datalen	The datagram length.

3.4 The Visualisation Design

The biggest challenge in visualisation software is to design a visual layout that is intuitive. With the user needs and the issues discussed in 3.1 and 3.2 as a guide, the author designed prototype software that incorporated multiple views options with interactivity, drill down and graphical user interface feature to ease the monitoring and analysis tasks.

The author approach is to combine the visualisation techniques with a machine learning classifier. Therefore, this tool is able to learn from previous examples and classifies alerts into false and true alerts. The false alerts can then be hidden or coloured differently thus allowing the pattern of true alerts to emerge.

Another approach is to use glyphs layout technique which are parallel coordinates plot and scatter plot techniques. The author also designed a novel time series 3D AlertGraph, an extension of 2D histographs [33] technique into three dimensions.

This prototype software has seven 3D views, which are the time series 3D AlertGraph, timeline, plane, world globe, world plane, parallel coordinates plot and scatter plot views. The 3D time series AlertGraph view is used as the general overview of the network security alerts data.

The objective of the multiple views is to give viewers different perspectives of the data because some abnormal patterns are not obvious to identify in one display but might be easy in another.

3.4.1 Time Series 3D AlertGraph

The 3D time series AlertGraph is served as a top level overview of the security alerts status. This view gives the quantity of alerts received from the pair of source IP address and destination port in a time interval. This view also summarises the alerts into different colours to indicate the quantity of alerts from (SRCIP, DPORT) pairs and uses false alert classification to highlight the true alerts. Through interaction tools, the alerts can be highlighted according to the source IP, destination IP or destination port. In this view, a huge number of alerts can be viewed in a single display and a temporal characteristic of the attacks can be discovered.

The time series 3D AlertGraph was plotted by mapping the quantity of alerts received from the source IP and destination port (SRCIP, DPORT) pair on the Z-axis, the destination IP addresses on the Y-axis and the time ordered sequence on the X-axis of the 3D plot. Each mapped point in the 3D plot was attached with a coloured sphere. The sphere was coloured according to the number of the different pairs of (SRCIP, DPORT) and the number of alerts received at that point. Pictures in Figure 3-5 show the schematic diagram and a day view of 3D AlertGraph.

In this view, each mapped point represented the number of alerts received by a destination IP in a given interval from (SRCIP, DPORT) pairs. In an interval, there might be more than one (SRCIP, DPORT) pair that had the

same number of alerts. The author counted the (SRCIP, DPORT) pairs and its value was used as one of the variables to colour the sphere. Another variable used to colour the sphere was the number of alerts at that point which in this case was the Z-axis value. The scalar coefficient to colour the sphere was the sum of number of alerts and number of (SRCIP, DPORT) pairs at a point. Therefore, the coefficient was calculated as:

- ScalarToColour coefficient, C_{PTS}
  - $C_{PTS} = P_{PTS} + Z_{PTS}$
- where
  - P_{PTS}= number of (SRCIP,DPORT) pairs at the point
  - Z_{PTS}=number of alerts for each (SRCIP,DPORT) pair at the point

By considering the quantity of alerts and the number of pairs available at a point, ranges of colours can be achieved. In this method, the quantity of alerts and the number of pairs available at a point will have the same importance in the colour scheme. This variation of colours is helpful to understand the network status and identify the true attacks. When the false alert classifier is applied, the author assigns to the true attacks a scalar coefficient in such a way that the sphere will be coloured in red. This is achieved by assigning the maximum value between the number of pairs and the number of alerts in each pair to the scalar coefficient of the true alert sphere. The pseudo-code to calculate the scalar coefficient in this way is as follows:

- ScalarToColour coefficient, C_{PTS}
  - If PTS contains 'TRUE ALERT':
    - C_{PTS} = Max_Pairs or Max_Z whichever is higher
  - Else:
    - $C_{PTS} = P_{PTS} + Z_{PTS}$
- Where
  - Max_Pairs is the maximum number of pairs in the whole period
  - Max_Z is the maximum number of alerts from all the pairs

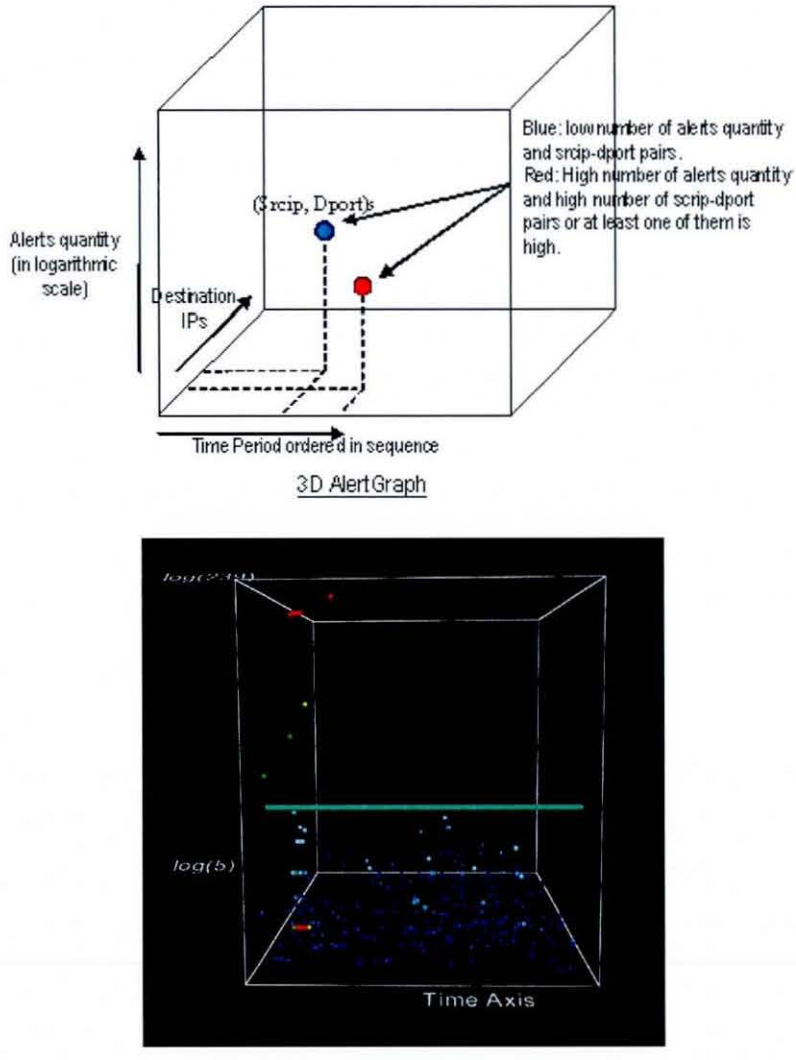


Figure 3-5: The time series 3D AlertGraph – The upper picture shows the schematic drawing. The bottom picture shows a view from 3D AlertGraph.

Mapping the scalar coefficient to colour was achieved by using the VTKLookupTable class in the VTK module. For the VTKLookupTable class to create the range of colours, the class was supplied with the minimum and the maximum range of the scalar coefficient in the monitoring period. The mapping colour table was set to vary from blue to red. The colour transformation was using the logarithmic function in the VTKLookupTable. The blue colour represented the lowest value and the red was the highest.

This colour variation is to highlight to the user, the density of alerts and

the (SRCIP, DPORT) pairs in the data. The red colour will suggest either

- many alerts from a unique (SRCIP, DPORT) pair or
- many alerts from many (SRCIP, DPORT) pairs but with few alerts or
- both having high values or
- the alerts are the true alerts

This 3D AlertGraph could also highlight the alerts received either by the destination IP address, source IP address or destination port. These features can be accessed from the interaction tools of the software.

The interactive tools were added in such a way as to perform Shneiderman's visual information-seeking Mantra, 'Overview first, zoom and filter, details-on-demand' [57] and to understand the data easier. The first interaction was by 'picking' the actor (section 3.4.7). The details of the source IP, destination IP and destination port will be popped-up with the time period of the event, the alerts quantity and the number of (SRCIP, DPORT) pairs.

The second interaction tool was the highlight of a specific source IP or destination IP or destination port by clicking the child item in the SRCIP-DSTIP-DPORT tree panel. By double clicking a destination IP, a transparent green plane will appear with yellow lines from the 3D plot base to all data points in that respective destination IP (see Figure 3-6). The yellow lines make a histogram graph of the number of alerts received according to the time (see Figure 3-7). However, double clicking the source IP or destination port will highlight the spheres having that information. At the same time, yellow lines will be drawn from the 3D plot base to the respective data points. Other settings the users can modify include the start time, the monitoring period, the time interval and the scales of the 3D plot.

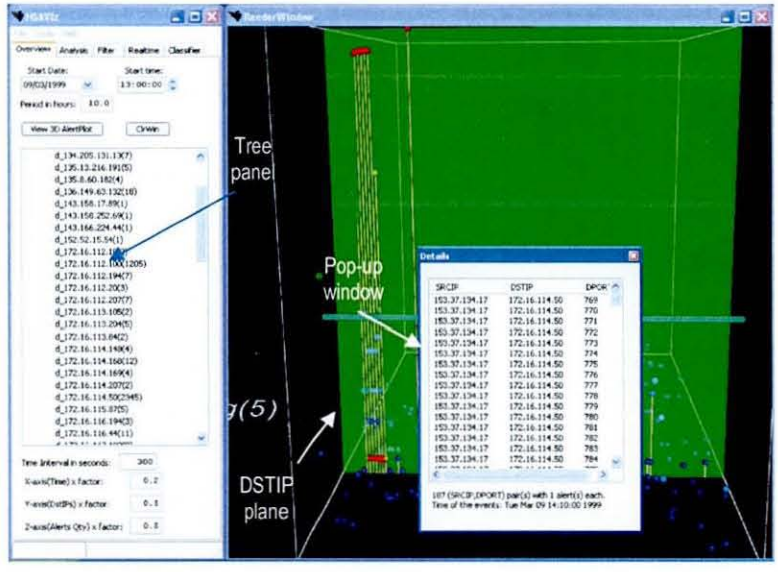


Figure 3-6: The graphical user interface and pop-up window in the 3D AlertGraph design.

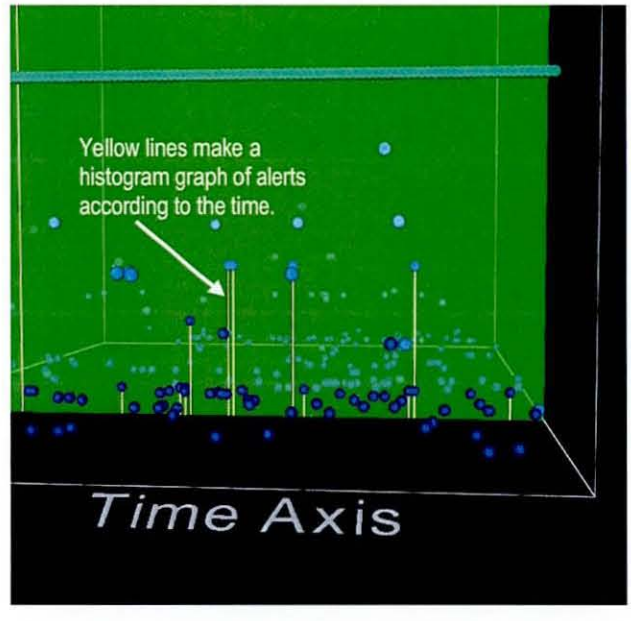


Figure 3-7: A view of a destination IP selection that shows the green transparent layer with the histogram graph of vertical yellow lines.

In summary, the 3D AlertGraph shows the alerts received by the destination IP address from the pair of (SRCIP, DPORT) in a time interval. The colour of the sphere showed the sum of the (SRCIP, DPORT) pairs with the number of alerts at a point. A single red sphere at the top of the 3D plot may suggest a possible DOS attack and continuous horizontal red spheres at the

bottom of the 3D plot may point out a port scanning or a port sweep attack.

Advantages of this view are

1. It summarises the alerts into different colours to indicate the quantity of alerts from (SRCIP, DPORT) pairs.
2. It highlights the true alerts.
3. Through interaction tools, the alerts can be highlighted according to the source IP, destination IP or destination port.
4. A huge numbers of alerts can be viewed in a single display.
5. A temporal characteristic of attacks can be discovered.

3.4.2 Parallel Coordinates Plot View

Figure 3-8 shows the schematic drawing of the parallel coordinates plot. The parallel coordinates plot consists of plotting five attributes of the alert and connecting them with lines. The attributes are the source port, source IP address, alert signature, destination IP address and destination port with the third dimension is the indicative quantity of alerts having that particular attributes. The lines connecting the attributes are coloured according to the transport layer protocol used. Users can change the colour setting in the colour setting window. Previous research has shown that a parallel coordinates plot is good in detecting IP sweep, Port Sweep, NMAP or similar attack. Figure 3-9 shows the visualisation display of parallel plot view.

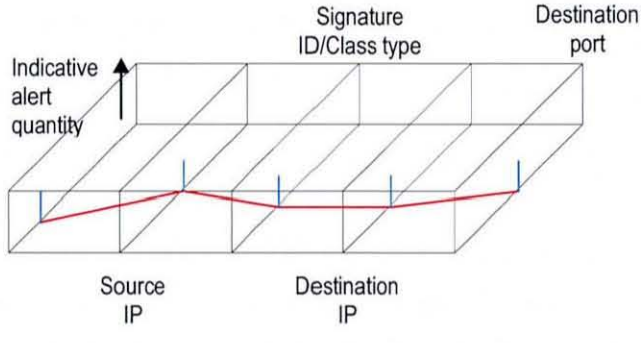


Figure 3-8: The schematic diagram of parallel plot view.

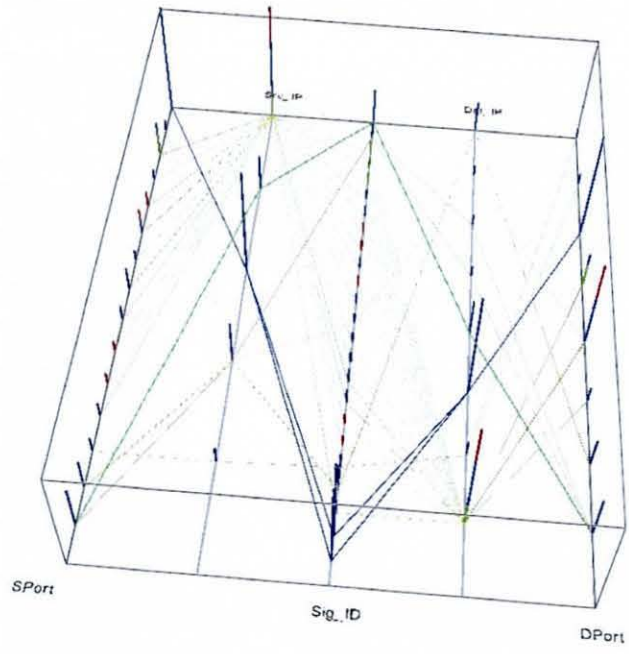


Figure 3-9: The parallel plot view.

3.4.3 Scatter Plot View

In the scatter plot view, six dimensions of alert attributes are displayed at a time. The attributes are the local host, alert signature, event time, alert priority, alerts quantity and attacker's origin. The alert priority is shown by the alert object colour which is red for high, blue for medium and green for low. The alert priority colour is consistence through out the visualisation tool. Alert objects used (as used later in section 3.4.9) are the

- Cube for alerts received by the local network,
- Cone for group alerts received by the local network,
- Sphere for alerts originating from the local network, and
- Cylinder for the alerts originating from the local network.

The Figure 3-10 shows the schematic diagram of the scatter plot. In the scatter plot view, the users have five further views which are:

- the alerts view,
- the alerts indicative quantity view,
- the alerts and their indicative quantity view,

- the alerts and their indicative quantity view with geographical location of the attacker origin, and
- the alerts and their indicative quantity view with the attackers are grouped in their network domain.

Figure 3-11 shows the last two scatter plot options. The first image, noted (a) shows a view of the attacker's origin by connecting alerts with lines to a world map. The second image, noted (b) shows how the attackers are grouped in their network domain. The line colour identifies the internet transport protocol used in the attack. With these views, users can identify the geographical location of the attackers and whether the attackers are from the same network domain or not.

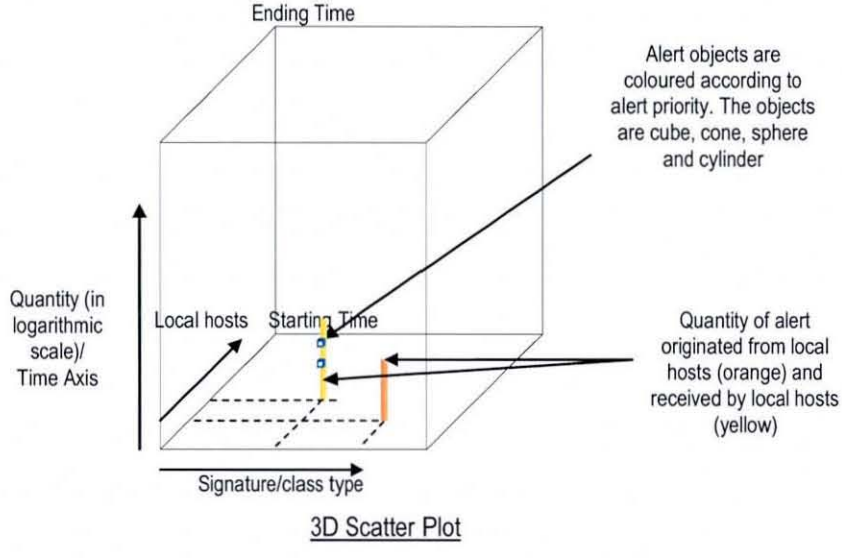
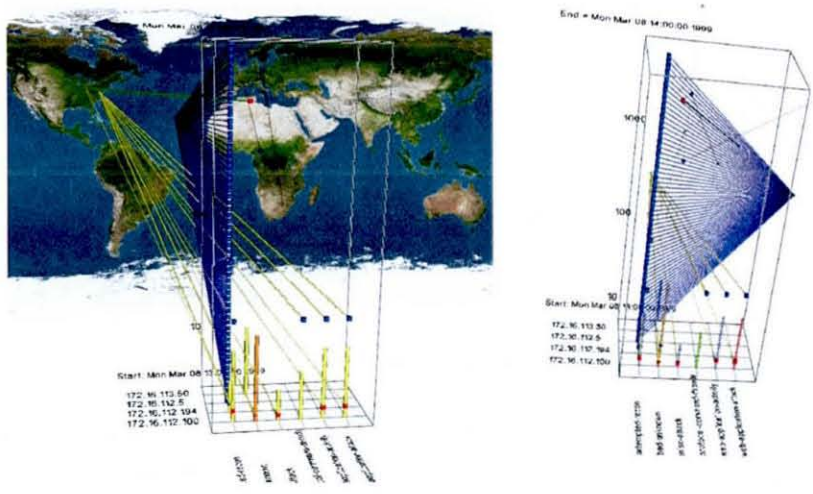


Figure 3-10: The schematic diagram of scatter plot view



(a) (b)

Figure 3-11: (a) Geographical view of an attack to local network. (b) Attackers were grouped in their network domain.

The domain is grouped using the attacker's IP address information. The IP addresses which are assigned to organisations in blocks belong to one of three classes: class A, class B, or class C. By knowing the value of its first octet, the class of an IP address is known. The attackers were grouped by using the default subnet mask of the attacker's network class. The Table 3-4 shows the network classes and their default subnet masks and Figure 3-12 shows the domain arrangement.

Table 3-4: IP network classes.

First Octet	Network Class/Default subnet mask
1 - 126	Class A / 255.0.0.0
128 - 191	Class B / 255.255.0.0
192 - - >	Class C / 255.255.255.0

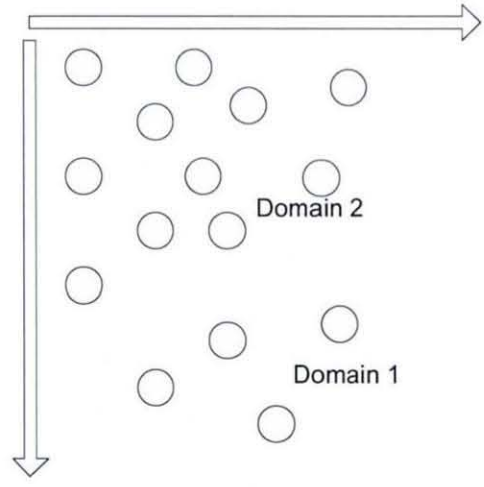


Figure 3-12: The attacker's subnets were arranged randomly. Each sphere represents a subnet.

The indicative quantity of the alert is shown by the height of the coloured bar that is positioned at the intersection of the local IP address and the signature name or class type. The bar height shows the quantity of alerts of the host with its respective signature and the bar colour signals the alert priority (green: low, blue: medium, red: high). Users can change these colours to other colours as they wish. When the bar colour is orange, it shows the local host is the origin of the attacks. The height of the bar is computed by changing the alert quantity using a logarithmic function (bar height = log(alert quantity)).

The scatter plot view becomes the summary display of the alerts for the local network. The advantage of these views is that the alerts are clustered according to alert type, destination IP address, source IP address, geographical location and attacker network domain. The users can use the advanced subnet mask to reduce the hosts in the monitored network. This can be done in the local network settings by changing the bits number in the Classless Inter-Domain Routing[6] (CIDR) notation.

[6] A CIDR network address looks like this '192.30.250.00/18'. The "192.30.250.00' is the network address itself and the '18' says the first 18 bits are the network part of the address and the remaining 14 bits are for specific host addresses in IPv4 system.

3.4.4 Timeline and Plane View

What, when and *where* are trivial questions asked by any analyst doing any investigation task. The *what, when* and *where* ideas were also used by Livnat and others [38] in their work (see section 2.4.3). The author work differs from theirs as they presented the ideas in a Circle View with network topology while the author presents it in a 3D display.

Alternatively, the author answered these questions by visualising the *what, when* and *where* through the design as in the schematic diagram in the Figure 3-15 (3D view with timeline). In this view, an alert is displayed using a line connecting from the source IP address to the destination IP address passing through the time axis which suggests the alert instance. The lines are clustered according to their signature id. Signature ids are placed at the middle of the visualisation that separates the local network from the external network. The signature ids are labelled by the ID number with respective attack priority colour (low: green, medium: blue, high: red). The vertical axis represents the selected monitoring period with start time at the bottom and end time at the top.

The *what* is the alert represented by a line from the source to the destination through a specific signature. The *when* is the intersection of the connection line with the time plane. While the *where* is the source and the destination IP address of the attack.

The *what* attribute will answer questions about the attack signature, the protocol used and the details of the alert. The *when* attribute will provide the time perspective of the alert and the instances of the attack. Lastly, the *where* attribute will provide information about the source and the destination of the attack, information about IP address and their geographical location.

The alert object position on each line suggests whether the local network is the target or victim of an attack. If the object is on the local network side, it means the attack is targeting the local network. On the other hand, if the object is located on the external network side, it means the local network is targeting

an external network.

The local and external hosts are arranged in semicircles. If the semicircle is full, a new circle will be created with an increased radius and height. With this arrangement many hosts can be visualised. The hosts coordinate calculation is shown in the flowchart in the Figure 3-13, while Figure 3-14 shows the simulated host arrangement with 4554 hosts. More hosts can be visualised, but the only constraint will be the system resources.

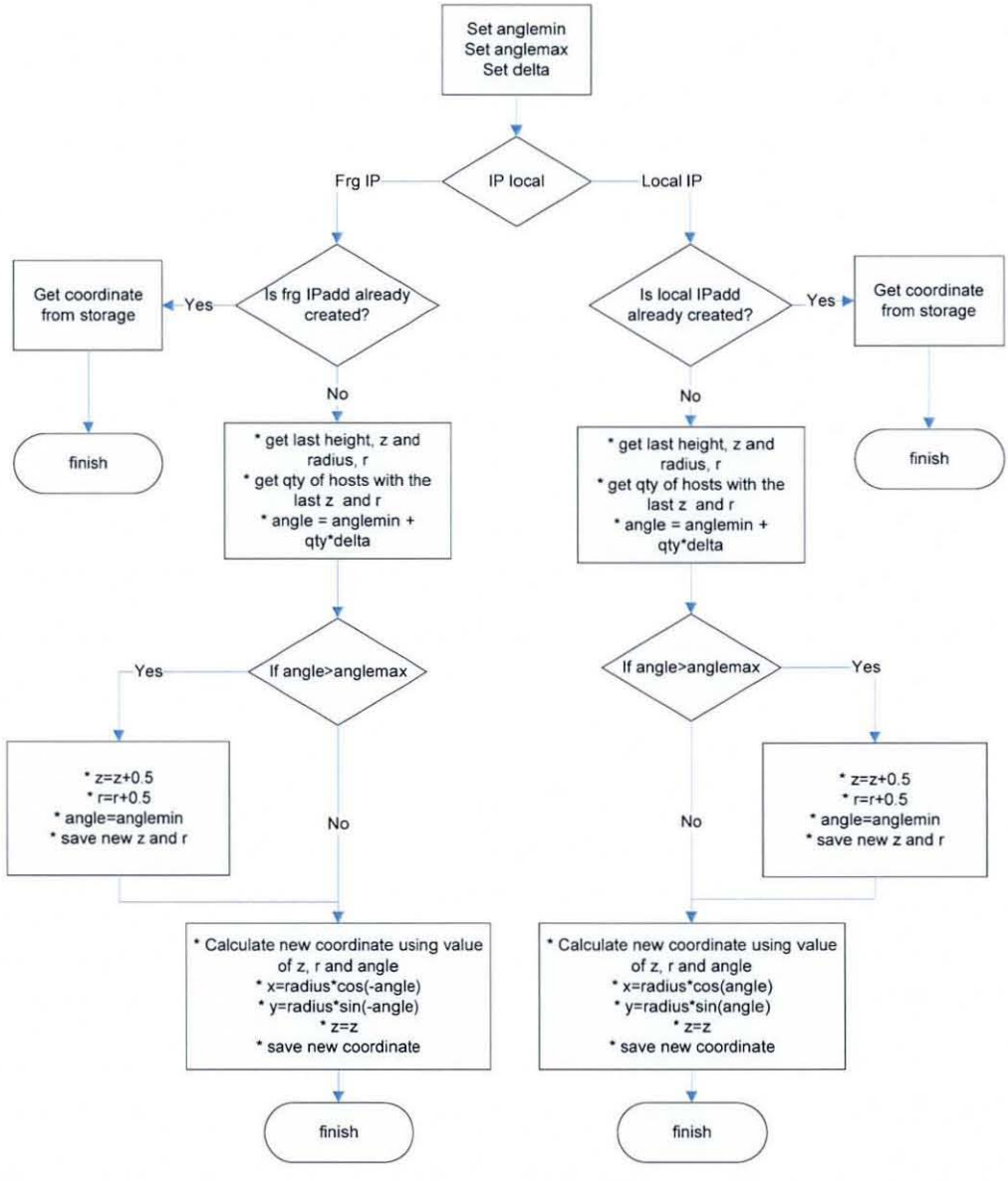


Figure 3-13. The flow chart of the host coordinates algorithm in the timeline view.

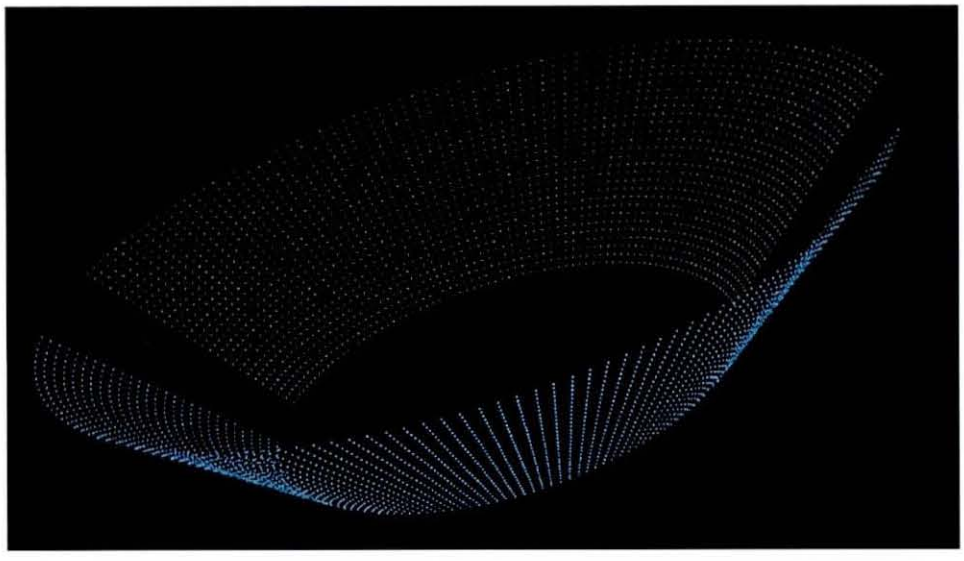


Figure 3-14. A view of 4554 hosts arrangement in the timeline view.

On the other hand, the plane view projects the timeline view without the time axis. This projection will remove the temporal notion of the display and will show fewer lines as attacks with the same signature, source and destination will appear only once. The Figure 3-15 shows both the schematic diagram of 3D view with timeline and without timeline, and pictures in Figure 3-16 show the actual views.

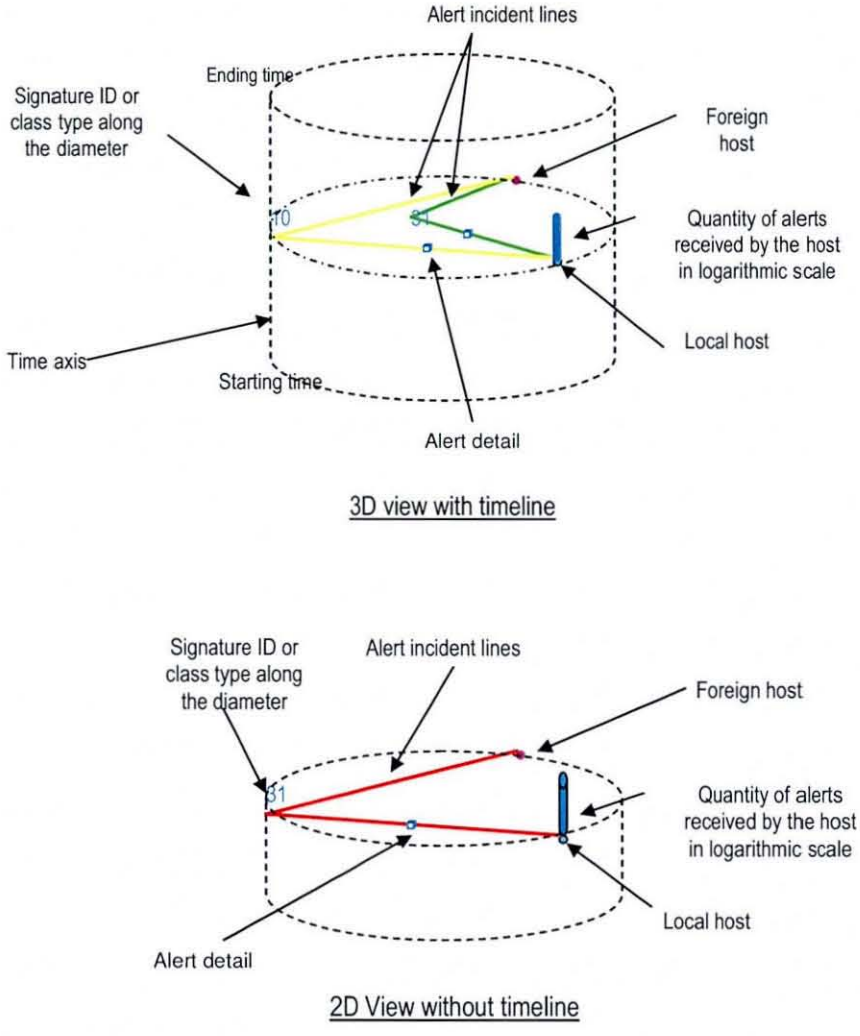
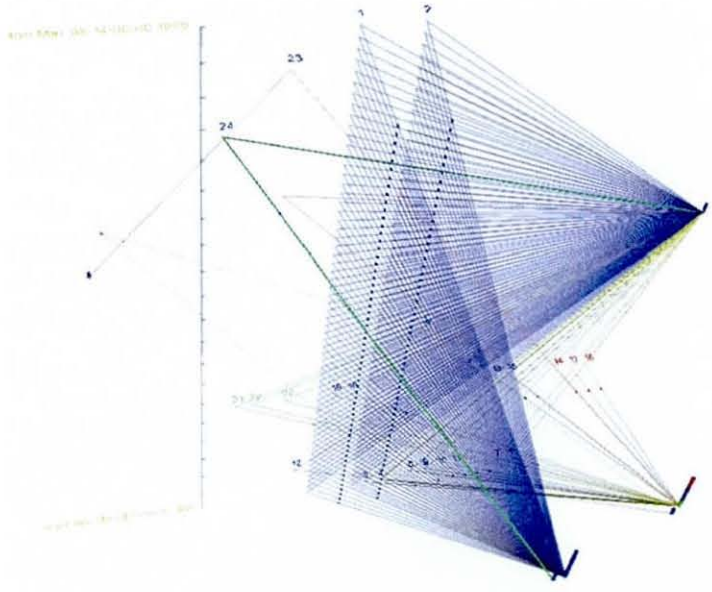
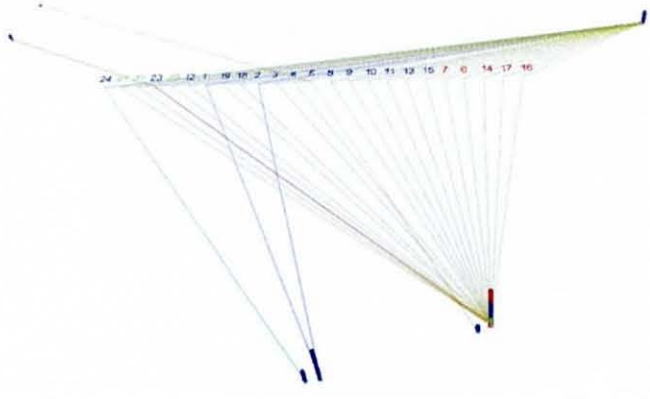


Figure 3-15: The schematic diagram of the timeline and plane views.



(a) Timeline View



(b) Plane View

Figure 3-16: The timeline and plane views.

3.4.5 Geographical Display

Another two views used to show all related IP addresses in their geographical location. The display is either in a world globe or in a world plane view. In these views, the author displays the geographical location of the external IP address that causes the attacks to the local network with their

indicative quantity of alerts. The height of the logarithmic scale bars represents the indicative quantity of the alerts and the colour of the bars shows the alert priority (low: green, medium: blue, high: red). The geographical location of the attackers can also be viewed in scatter plot view (see Figure 3-11) with alert lines connecting the alerts to their respective source.

Figure 3-17: The Geographical View in a world globe.

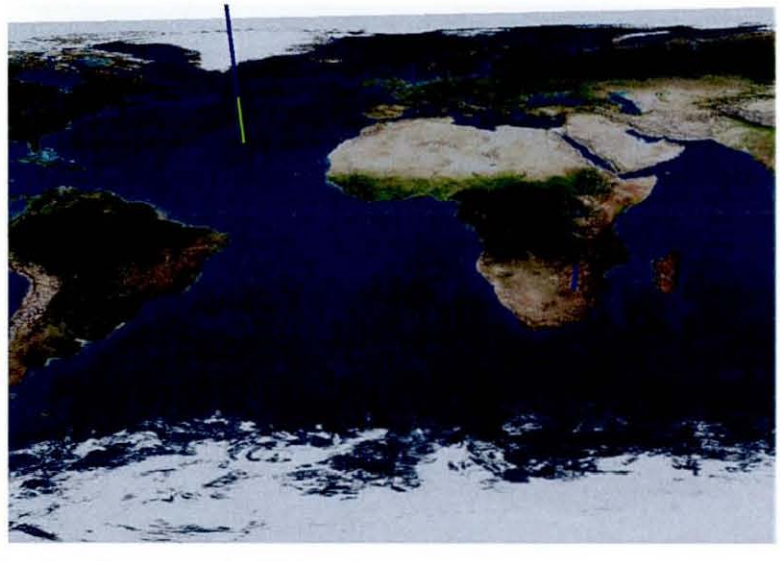


Figure 3-18: The Geographical View in a world plane.

3.4.6 Timeline animation and Real-time monitoring

The timeline animation is designed to give the user the exact sequence of the attacks. The user can replay the attacks from a selected period. The

default replay speed is set to 60 times faster than real-time; which the user can change the speed setting. A video player type control panel is also provided to allow the user to control the animation. The user can also use the slider to slide through the animation. This feature offers the possibility of exploration and navigation through the alerts. The animation can be displayed in the timeline view and in the parallel coordinates view.

The software also has the ability to monitor alerts in real-time. This is performed when Snort sent alerts to the database in real-time. The software will query the database and refresh the display every three seconds. A three-second period or slightly more is taught to be sufficient to inform users of new attack. The user also receives an update of the alerts received by the source and destination IP address pair (Figure 3-19). They also have access to the detailed information on IP address, alert and attack signature by using 'picking' actors as mentioned in section 3.4.7.

The observation window for real-time monitoring is set for a four hour period. At the end of this period, the window will be emptied and reset to clear the display. Figure 3-19 shows the control panel for real-time monitoring. The user can choose either to display in the timeline view or the scatter plot view with the geographical location of the attacker.

To make the monitoring in real-time effortless, users can apply the false alert classifier feature. In this feature, only the predicted true alerts will be displayed. Therefore, users will not be distracted with false alerts. The details of the false alert classifier are explained in the chapter 4.

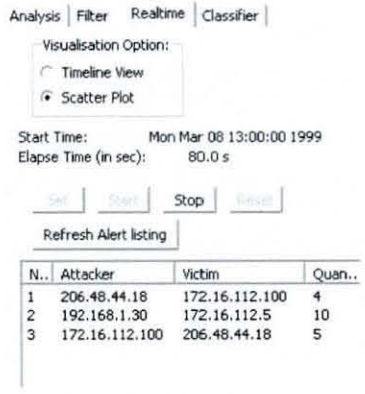


Figure 3-19. The Real-time control panel.

3.4.7 Interactivity, drill down and zoom

One of the issues pointed out in usability studies is the user interactivity with the software. The author suggested various user interactions with the software, which can be categorized into interactions using the 3D images and interactions using the graphical user interface to aid the analysis. These interactions allow 'drill down' and 'detail on demand' supports which follow the Shneiderman's visual information seeking Mantra [57]. The user can interact with the 3D images by using the VTK built-in mouse and keyboard interaction, such as pan, zoom and rotate. Users can get details about IP address, alert and attack signature, by picking the objects that represent them. In the timeline and plane views, the spheres represent IP hosts; the signature ids represent the alert signature and the small cube on each line represents alert details. Picking an actor object can be activated by bringing the mouse pointer on to the object and then pressing the keyboard key, 'p'. A window containing the requested information will appear.

In the parallel coordinates plot view, picking on where the line intersects the attributes axis will display the value of the attribute. In the scatter plot, picking the alert object will display the alert details. Figure 3-20 shows the example of a pop-up window after picking objects in the scatter plot view.

Users can choose the monitoring period accordingly. A smaller monitoring period is preferred to avoid occlusions. This is because, when the alerts become numerous, the lines will congest the display and this will increase

the occlusions. Keeping the height of the time axis constant, and selecting a smaller monitoring period, will scatter the connection lines. The smallest monitoring period that can be chosen is one second.

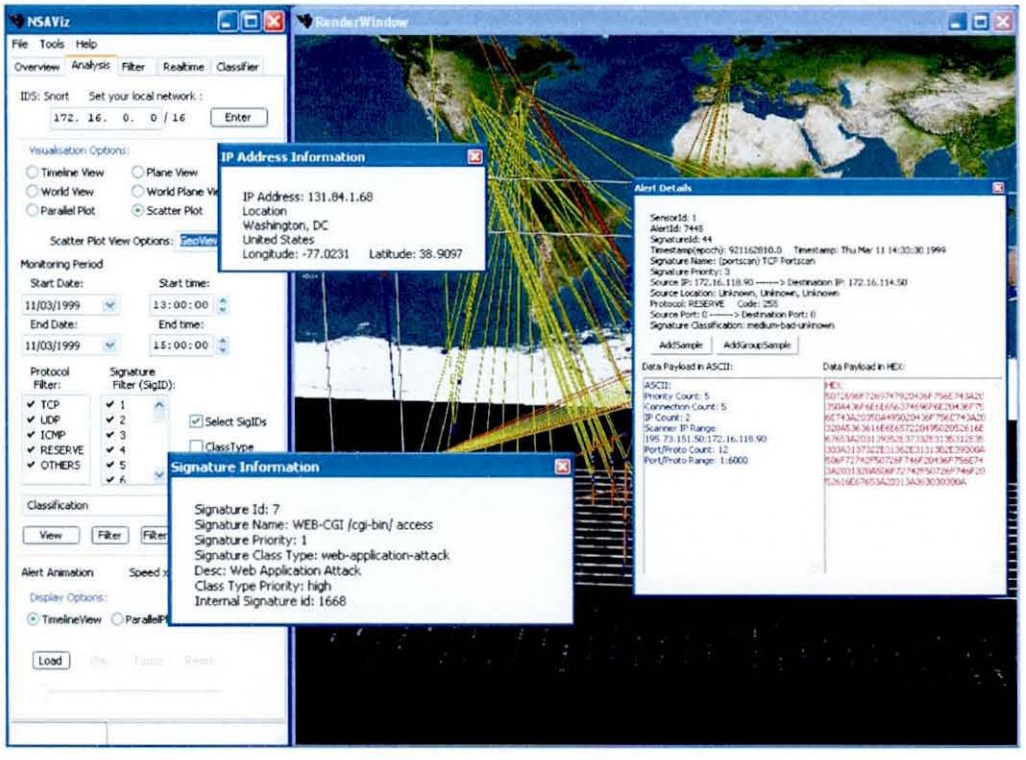


Figure 3-20. Examples of pop-up windows.

Another important interactivity feature is the ability to filter the alerts. The software allows the user to filter alerts according to the internet transport protocol, alert signature, alert class type, alert priority, source IP, destination IP, source port and destination port of the attacks. Users can also view and print the alerts listing and signatures listing in a selected monitoring period. These features can be accessed from the tools menu bar. Other information includes the quantity of alerts according to the source IP address, destination IP address, source port and destination port and users can plot a bar chart of these (Figure 3-21). These features can be accessed from the filter panel and are designed to provide the analysts tools that will help them to perform their tasks better.

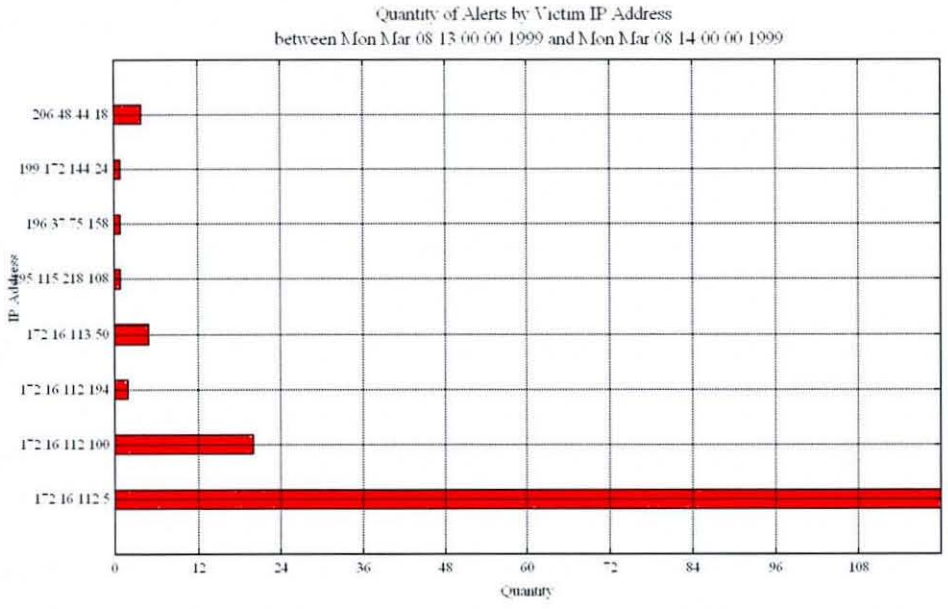


Figure 3-21: A bar chart of alerts received by the victim IP address in a specified period.

3.4.8 The Graphical User Interface

The graphical user interface was designed to help the user to perform monitoring, analysis and response tasks. In designing for an effective graphical user interface, Nielson [58] suggested the design should be simple, clear and address the user tasks. Edward [59], on the other hand, suggested the use of common sense to design a graphical user interface. The common sense or heuristic rules include clear design, use of user language, simple layout and addressing the user tasks.

The user interface was designed by analysing the user tasks and from that the graphical user interface structure was proposed as in the Figure 3-22. The structure divides the tasks using three main designs which are the menu bar, notebook panel and image interaction. The menu bar has two submenus which are the file and tools menus, while the notebook panel has five different panels which are: the overview, analysis, filter, real-time and classifier panels.

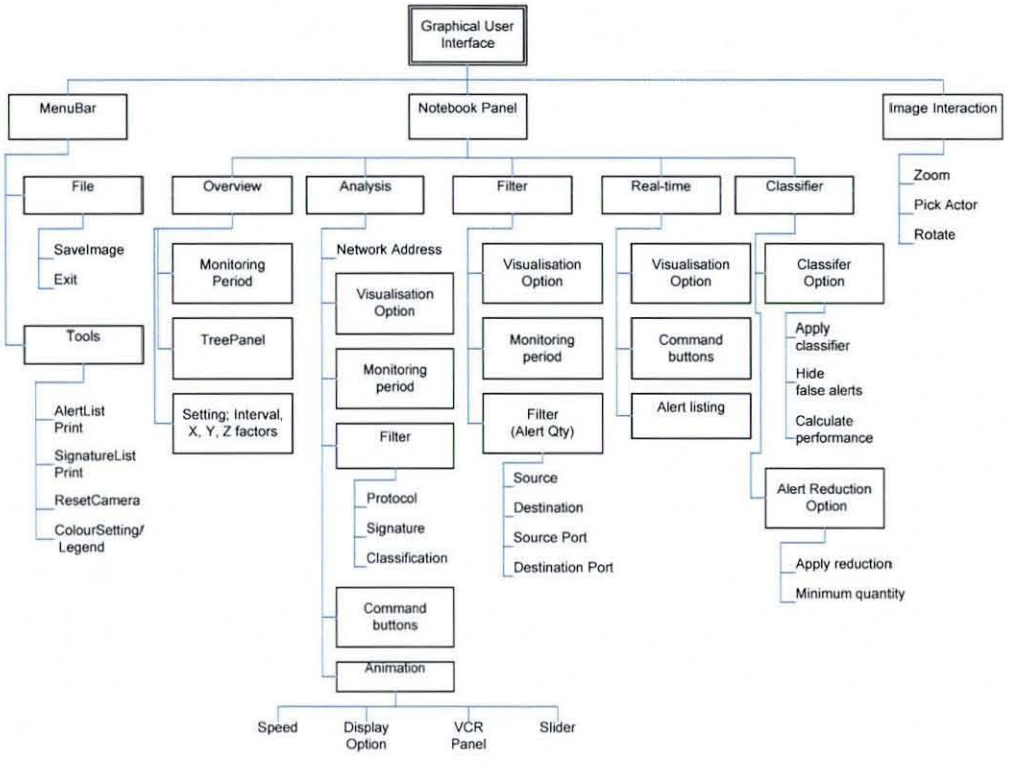


Figure 3-22. The graphical user interface structure.

The overview panel is used for controlling the 3D AlertGraph. The analysis and filter panels are to help the user analysis tasks. The real-time panel is used for monitoring tasks. The response tasks are assisted by the save-image, alert-signature listing and printing features. The colour setting and the classifier menus are for the software settings. The colour setting is used to change the colour setting of the visualisation, such as the background colour and the objects colours. The colour setting also acts as a colour legend. The classifier menu is to allow the user to configure the classifier and the alert reduction settings.

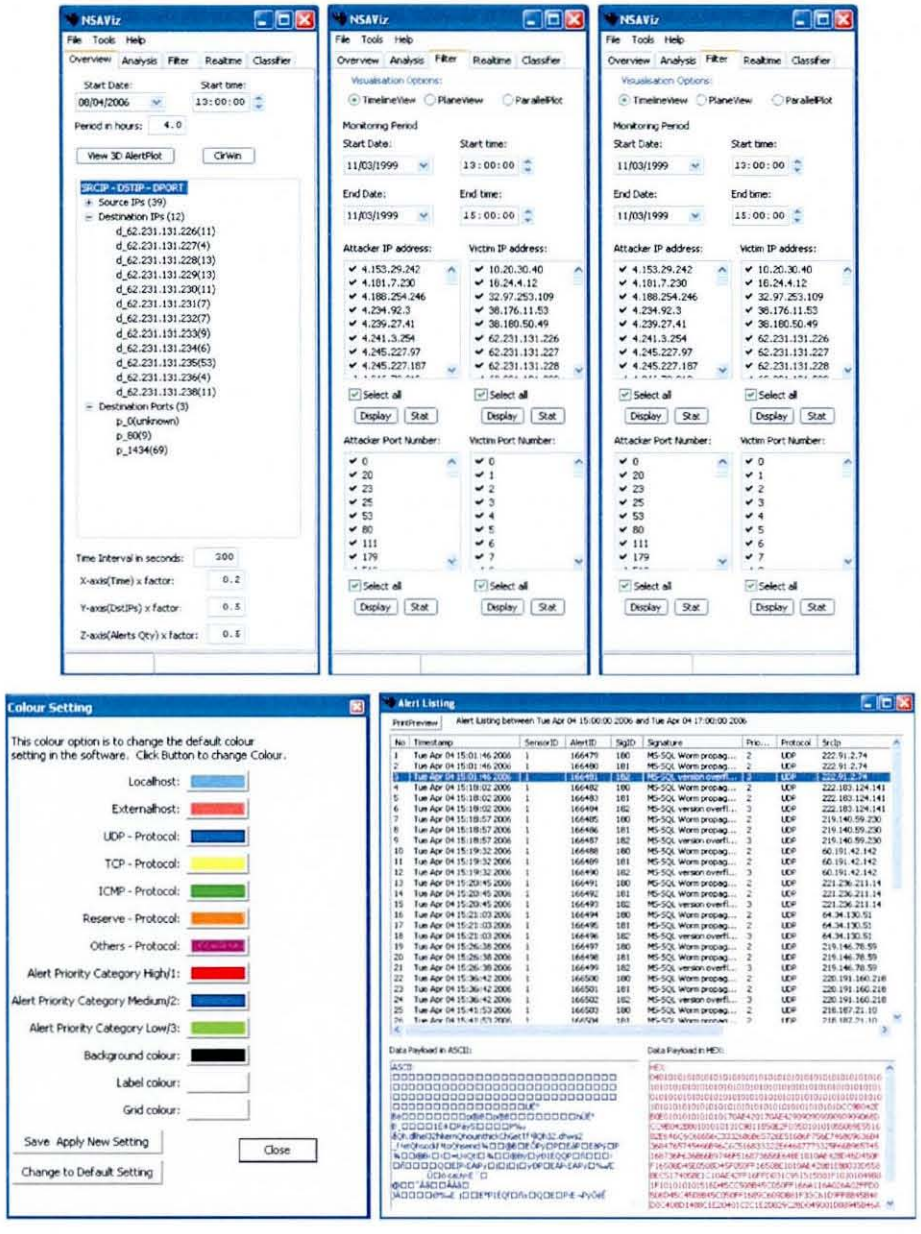


Figure 3-23. Some the GUI control panels, the colour settings and the alert listing.

3.4.9 Alert Reduction and a False Alert Classifier

Displaying thousands of alerts in the graphic display will slow down the associated computer. To solve this problem, an alert reduction algorithm which groups similar alerts is used. This technique has been used by Tedesco [60] to solve the alert flooding attack problem. Alert flooding attack is simulated intrusion attempts that the attacker creates to make the analyst become

deluged with false information.

To reduce the number of alerts, alerts were grouped with the same source IP address, alert signature and destination IP address. Then, different objects were used to represent the grouped alerts. A cube is for an alert targeting the local host, while a sphere is for an alert originating from the local host. In alert reduction mode, the cube will be replaced by a cone and the sphere will be replaced by a cylinder.

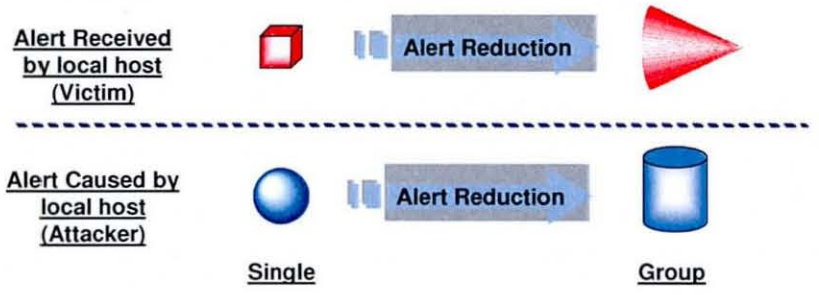


Figure 3-24. Alert objects in scatter plot.

With this alert reduction approach, large numbers of alerts can be displayed at a time. The author has tested the prototype software's ability to display up to 120,000 alerts in one display. According to Abdullah [61], 30,000 to 35,000 networked computers such as at the Georgia Institute of Technology, USA receive an average of 50,000 alerts per day. This prototype is scalable to monitor such scale of networked computers and alerts. Another novel feature that the author have applied in the visualisation is the use of a machine learning classifier to classify alerts into false and true positive alerts. The details of this alert classifier are discussed in chapter 4.

3.5 Summary

In this chapter, the author has explained the main features of the visualisation approach starting from the visual design to alert reduction technique. The author has also explained the user needs analysis and some issues about network security visualisation. The scalability issue was addressed by introducing the 3D AlertGraph, the alert reduction, the false alert classifier

and filter option to overcome the problem of huge amounts of data. The author also carried out special arrangement of the network host as in 3.4.4 to allow many hosts to be displayed in the timeline and plane views.

This visualisation approach attempts to displays the optimum amount of information to users. Many researchers have suggested there is limited information that human beings can process effectively. The limit is still under study but the range within 7 to 9 pieces of information is a typical consideration [62]. The piece of information could be a number, a line, a colour etc. The pieces of information that this visualisation displays are within the effective range that a human being can process at a time. Table 3-5 shows the information displayed in the visualisation prototype. In the Overview, five pieces of information were displayed using colour for alert priority or classification, three information on the X, Y and Z axis (time, destination IP, quantity) and a coloured sphere to represent source IP and destination port. The author also made the monitoring in real-time and in analysis to be easy by implementing the false alert classifier.

Table 3-5. The information displayed in the visualisation prototype.

	Information	Overview	Scatter Plot View	Parallel Plot View	Timeline View
1	The IP transport protocol used in the attack.	-	Yes	Yes	Yes
2	The indicative time of the attack.	Yes	Yes	-	Yes
3	The indicative quantity of alerts.	Yes	Yes	Yes	Yes
4	Geographical location	-	Yes	-	-
5	Alert signature	-	Yes	Yes	Yes
6	Alert priority / Classifier	Yes	Yes	-	Yes
7	Attacker / victim machine	Yes	Yes	Yes	Yes
8	Source Port	-	-	Yes	-
9	Destination Port	Yes	-	Yes	-
10	Attackers domain	-	Yes		

Table 3-6 summarises the need analysis and the features of the visualisation prototype that addressed the user needs. The 3D AlertGraph, scatter plot view and real-time monitoring in the visualisation prototype addressed the monitoring tasks. The drill down, filters, alert animation, false alert classifier, multiple views and geographical clustering features addressed the analysis tasks. For response tasks, listing alerts and saving images are used to help the users in reporting the attacks. In general, this visualisation prototype has addressed all the issues in network security visualisation and the need analysis mentioned earlier.

Table 3-6. The summary of needs analysis and the software features addressing the needs.

Analyst Task	Visualisation needs (see section 3.1)	Visualisation features of the prototype
Monitoring (reactive) - Monitoring all attack alerts - Identifying potentially suspicious alerts	- Simple display - Overview display - Flexibility	- Overview in 3D AlertGraph - Real-time monitoring - Setting of local network
Analysis (proactive) - Analysing alert data - Analysing other related data - Diagnosing attack	- Filtering and interaction - Exploration - Multiple data source and correlation - Multiple view and level of data	- Drill-down - Packet payload view - Filtering - Flexible monitoring period - Multiple view option - Alert Animation - False Alert Classifier - Geographical clustering - Bar charts of alert quantity
Response - Responding to attack - Documenting and reporting attack - Updating IDS	- Save image	- Alert listing and printing - Signature listing printing - Save image in JPEG or BMP - Flexible colour setting

Chapter 4: Integrating a False Alert Classifier with Network Security Data Visualisation

4.1 Introduction

Overwhelming alerts are a well-known problem in a signature-based Intrusion Detection System (IDS), such as Snort. The alerts produced from the IDS may reach thousands or even ten of thousands daily and most of them are false [4, 5]. The task to identify the true alerts is left in the hands of the analyst. As the quantity of the alerts is numerous, manual analysis is almost impossible. This becomes even worse when if there is a lack of expertise to identify the true alerts amongst the thousand of alerts received.

Previous researchers studied this problem and proposed various ways to solve it. Some of the approaches were to develop an analysis tool to explore and examine the alerts such as SnortAlog [41], logparser [42] and ACID (Analyse Console for Intrusion Detection) [43]. The drawbacks of these tools were that the analyst still needs to deal with the alerts individually, and most of the output displays are in text form.

4.2 Related Work

Some researchers have opted for visualisation techniques to tackle this problem. Current research in network security visualisation has grown and many display techniques have been explored. Some of the tools developed were NVisionIP [3], IDSRainstorm [7], SnortView [4], VisFlowConnect [63] and many others. In general, the main focus of visualisation software is to achieve visual patterns of the true alerts and to help the analyst to explore them. Even with the aid of the visualisation software, identifying the attack patterns is still a difficult task. This may be due to occlusion, or due to the numerous alerts crowding the visualisation display.

One way to overcome overcrowding alerts is by reducing the numbers of alerts to ease the analyses. Lee [64] and Viinikka [65] employed statistical methods, such as Granger Causality Analysis and EWMA Control Charts to

remove alerts that formed the normal behaviour of the monitored network. Tedesco [60] used token bucket filter to limit the quantity of alerts in each alert category and monitoring window. Pietraszek [66] and Bloedorn et al. [67] employed machine learning techniques to classify alerts into false and true alerts and then removed the false one. In these methods though, the classified alerts were not visualised and they were still presented to the network security analysts in text form. The network security analysts then had to analyse these alerts manually.

Bloedorn et al. [67] showed that the use of the C4.5 classifier algorithm with domain-knowledge (DK) successfully reduced the false alerts. In their implementation, the DK rules were directly provided by experts and labelled training samples. The use of these rules was to adjust the attributes selection during the tree construction so lower scoring attributes could be chosen if the DK rules suggested such preference.

Alternatively, Pietraszek used the RIPPER algorithm to create associative rules that classified the alerts into false and true alerts [66]. RIPPER is a rule learning algorithm described by Cohen [68]. It is a propositional learner designed for efficient performance on large and noisy datasets and therefore is a noise-tolerant algorithm.

It was thought that, in their implementations, there were difficulties for the analyst to add new training examples to the classifier model and no visualisation aides were used. The author extended the use of the classifier in the visualisation tool by providing the user with the ability to add more training samples to the classifier model using the interactive visual display. So, the classifier would learn new examples of false and true alerts, and then build a new model from them. Lastly, the classifier result was used in the visualisation by colour coding the false and true alerts, or simply hides the false ones. The output in the visual form will make the analysis more effective.

Some researchers have applied machine learning and data mining in intrusion detection. Axelsson [44] used a Bayesian classifier with visualisation to

build a new IDS. Recent work by Sandford [69] introduced a new network intrusion detection system based on a novel signature detection filtering mechanism with statistical summaries and data mining algorithms. This work differed from theirs as the author aim was not to develop a new IDS but a tool that would aid the analysis of the alerts produced by Snort IDS.

The author approach combined the visualisation technique with a machine learning classifier. Therefore, the visualisation software could learn from the previous examples, and classify alerts into false and true alerts. Another advantage of this tool is that it could reduce the alerts in the visualisation display, as the software could filter the false positive alerts.

4.3 Machine learning classifier

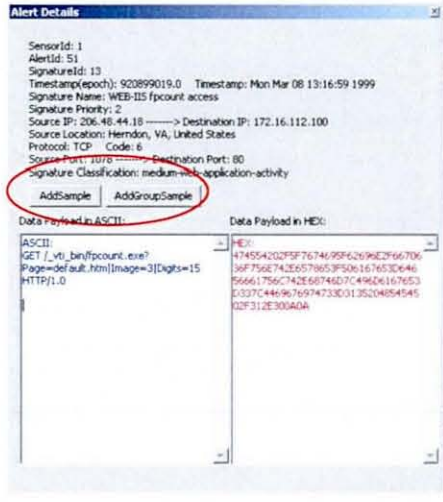
4.3.1 Approach

This approach used the classification tree algorithm as the machine learning classifier. The classifier will learns from the labelled training samples provided by the user and build a classifier model from it. As the objective was not to build a new classification algorithm, a decision tree learner provided by an open source machine learning module, Orange [70] was used.

The classification tree learner in Orange is actually based on the C4.5 classification algorithm [71], and, in default setting, provides the same result as in the C4.5 classification algorithm. However, the Orange classification tree does not use codes from C4.5 algorithms. As in ID3 [72] and C4.5-based algorithms, the Orange classification tree learner also uses information entropy theory to measure the information gain and find the best attribute for the tree division.

Add and edit support were also privided where users could add a new training sample to the classifier or edit the existing training sample. Users could access these features through a window after selecting the alert object in the visualisation display. To select an alert object, users should bring the mouse

pointer on to the alert object and then pressed the keyboard key 'p'. This is also known as 'pick object'. The block diagram of this procedure is in the Figure 4-1.



The circle shows the button to add a new sample or a group of samples.

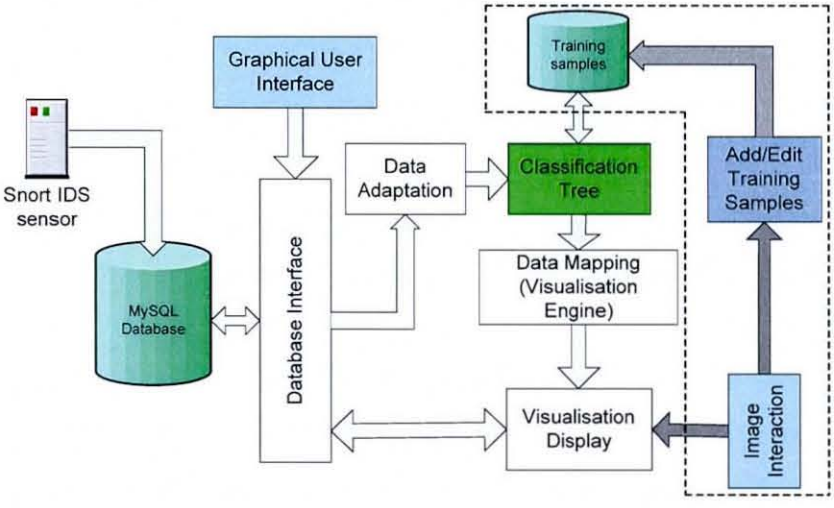


Figure 4-1: The block diagram shows the classification tree and user interaction architecture.

4.3.2 Input Attributes

The information used to teach the classifier was the source IP, source port, destination IP, destination port, alert class type, IP datagram length and IP protocol. Before sending the alerts to the classifier, they were processed into a general form. The source IP address and destination IP address were generalised to local or foreign hosts. For the source and destination ports, they were generalised into standard ports, ephemeral ports, unassigned ports or

unknown ports. Standard ports were the ports with port number less than or equal to 1024. Ephemeral ports were the port numbers between 1024 and 4999. The unassigned port numbers, were the port numbers greater than or equal to 5000. An unknown port number was assigned when there was no information regarding the port number in the IDS alert data. All of the attributes were discrete attributes except for the IP datagram length, which was a continuous attribute. The objective of this generalisation process was to prevent the classifier from memorising a specific example in the training set rather than producing general predictive rules.

Table 4-1: Summary of Classifier Input and generalisation.

Alert attributes	**Generalisation**	**Note (Input Type)**
Source IP address	Local host Foreign host	(Discrete attribute)
Source port	Standard Ephemeral Unassigned Unknown	If port number less than 1024 If port number used is between 1024 and 4999 If port number used is bigger than 5000 If port number used is unknown or unspecified (Discrete attribute)
Destination IP address	Local host Foreign host	(Discrete attribute)
Destination port	Standard Ephemeral Unassigned Unknown	If port number less than 1024 If port number used is between 1024 and 4999 If port number used is bigger than 5000 If port number used is unknown or unspecified (Discrete attribute)
Alert Classtype	Class type as specified by Snort	(Discrete attribute)
IP Datagram length	Actual byte value	(Continuous attribute)
IP protocol	UDP TCP ICMP Reserve Other	(Discrete attribute)

4.3.3 Classification tree

The classification tree algorithm used was the induction tree learner in the Orange module. In the Orange module, the classification tree could be either a decision or a regression tree. As in this case, the decision tree learner was used because the result of the classification would be in binary, which is either true or false. In a decision tree learner, the end leaf will be the attribute associated with the classification result, true or false. The internal nodes correspond to each value of its associated attributes takes. The decision tree is generated by a recursive loop of the learning element, and by splitting the best attributes that splits the training example into their proper class.

A post-pruning (backwards pruning) was also applied to simplify the classification tree. Pruning is one of the techniques employed in machine learning to tackle overfitting. Overfitting is an event in which the algorithm memorises the training data, but fails to predict a new instance well in the future.

4.3.4 Classification Accuracy

The k-fold cross-validation technique [73] was used to measure the classifier performance. In this technique, the data set was divided randomly into k equal subsets. The classifier was then built from the (k-1) data subsets and tested against the remaining subset. This procedure was repeated k times with different training and test sets. The average performance measures from all tests were then calculated. The k value of 10 was chosen, as recommended in the literature. The standard machine learning performance scores calculated were the classification accuracy (CA), the Brier Score (BS) and the area under the receiver operating characteristic (ROC) curve (AUC).

The performance scores terms and formulas are explained briefly below:

- Classification Accuracy, CA: This measure calculates the ratio of the correct prediction by the classifier against the total number in the test.

- $ca = \frac{no.\_of\_correct\_prediction}{total\_number\_in\_the\_test}$

- Brier Score, BS: A Brier score measures the accuracy of a set of probability assessments. It is the average deviation between predicted probabilities for a set of events and their outcomes, so a lower score represents higher accuracy. In Orange, BS is calculated as
  - $average\sum(t(x)-p(x))^2$ where:
    - x is a class,
    - $t(x)$ is 1 for the correct class and 0 for the others, and
    - $p(x)$ is the probability that the classifier assigned to the class x.
- Area under the ROC curve: This statistical measure is popular in medicine. It is a discrimination measure that ranges from 0.5 to 1.0. A rough guide for classifying the accuracy of a diagnostic test is the traditional academic point system:
  - 0.90-1 = excellent (A)
  - 0.80-0.90 = good (B)
  - 0.70-0.80 = fair (C)
  - 0.60-0.70 = poor (D)
  - 0.50-0.60 = fail (F)

The classifier performance was measured using a training sample that consisted of 561 alerts. These alerts had been added manually to the classifier through Add/Edit training sample feature. As the dataset was a skewed class distribution where there were high numbers of false alerts and low numbers of true alerts, the training set was sampled with a balance number of true and false alerts. The training sample consisted of 273 false alerts and 288 true alerts. The performance scores and the confusion matrix were shown in Table 4-2 and Table 4-3.

Table 4-2: Classifier Performance Scores

	CA	BS	AUC
Scores	*0.9857*	*0.0265*	*0.9892*

Table 4-3: Classifier Confusion Matrix

	Negative (false) Predicted	Positive (true) Predicted	Accuracy	Precision	Recall
Negative (false) in the training set	*a* 267 (TN=0.9780)	*b* 6 (FP=0.0220)	*0.9804*	*0.9792*	*0.9826*
Positive (true) in the training set	*c* 5 (FN=0.0174)	*d* 283 (TP=0.9826)			

A confusion matrix shows the predicted and the real classification count. Each row in the confusion matrix represents the classification category. The following terms are defined for a two-by two confusion matrix and the formulas used to calculate them are:

- Accuracy: $\frac{(a+d)}{(a+b+c+d)}$.
- True positive rate, TP (Recalled, Sensitivity): $\frac{d}{(c+d)}$.
- True negative rate, TN (Specificity): $\frac{a}{(a+b)}$.
- Precision: $\frac{d}{(b+d)}$.
- False positive rate, FP: $\frac{b}{(a+b)}$.
- False negative rate, FN: $\frac{c}{(c+d)}$.

The result showed the performance scores of the classifier were excellent. The area under the curve (AUC) ROC and classification accuracy (CA) were above 0.9857. For the Brier score (BS), the score was 0.0265 which

showed high accuracy. In contrast, Table 4-3 showed the confusion matrix score and the achieved true positive (TP), true negative (TN), false positive (FP) and false negative (FN) rates. The result showed the classifier was not biased towards any class category in the training sample. The Table 4-8 shows the list of the training sample.

4.3.5 Testing

Finally, the classifier was tested with three weeks of alerts data from the DARPA 1999 dataset. The first week and the third week of the data did not contain any intrusions. The second week, however, contained some intrusions. During these 3 weeks, Snort produced 142871 alerts with 125955 alerts in the first week, 14230 alerts in the second week and 2686 alerts in the third week.

By testing the classifier with the data from first and third weeks, the result showed the classifier effectively classified almost all alerts as false alerts. The summary of the results is given in Table 4-4. The results were excellent. The classifier wrongly classified nine alerts from 2686 alerts of the third week, and none for the first week. These results showed the classifier was excellent in the baseline condition where there were no intrusions. Actually, there should be no true attacks at all. However, analysing the nine wrongly-classified alerts showed that the destination hosts of the attacks were not from the local host. Therefore, the network administrator could easily classify them as false alerts.

Table 4-4: The classification results of the first and the third week.

	Total Alerts	Classified as false	Classified as true	% of detected false alert
Week1	125955	125955	0	100%
Week3	2686	2677	9	99.66%

To measure the classifier performance in the attack environment, the classifier was tested against the second week dataset. The classifier detections were compared with the attacks in the intrusions list (Table 5-1). The classifier performance was assessed by counting the number of its true detections. From

the intrusions list (Table 5-1), there were initially 43 attacks. However, as Snort missed some of the attacks, the available attacks left were 18. It was noted that five of the attacks were in the training sample. The author assumed it was appropriate to test the classifier with the dataset, as the parameters used to build the classifier model were in general form (section 4.3.2). There was no detailed information of the attacks available to the classifier such as IP address and port number.

From the testing, the classifier detected 17 out of 18 attacks in the intrusions list. The author believes the result was excellent. Furthermore, the one wrongly-classified attack could still be identified with the help of visualisation. The author will show these in the next section.

Table 4-5: Classifier detection table - The green lines were the attacks in the training sample and the orange line was the wrongly classified attack.

ID	Date	StartTime	Destination IP	Name	Classifier Detection
1	08/03/1999	13:01:01	172.16.112.100	Ntinfoscan	TRUE
2	08/03/1999	13:50:15	172.16.113.50	pod	TRUE
3	08/03/1999	14:39:16	172.16.114.50	back	TRUE
4	09/03/1999	13:44:17	172.16.114.50	portsweep	TRUE
5	09/03/1999	15:06:43	172.16.114.50	back	FALSE
6	09/03/1999	21:11:15	172.16.114.50	phf	TRUE
7	10//3/1999	17:02:13	172.16.114.50	satan	TRUE
8	11/03/1999	13:04:17	172.16.112.100	crashiis	TRUE
9	11/03/1999	14:33:17	172.16.114.50	satan	TRUE
10	11/03/1999	15:50:11	172.16.114.50	portsweep	TRUE
11	11/03/1999	16:04:16	172.16.114.207	neptune	TRUE
12	12/03/1999	00:16:18	172.16.112.50	ftp-write	TRUE
13	12/03/1999	13:07:17	172.16.114.50	phf	TRUE
14	12/03/1999	14:18:15	172.16.113.84	pod	TRUE
15	12/03/1999	16:20:15	172.16.114.50	neptune	TRUE
16	12/03/1999	17:40:12	172.16.112.100	crashiis	TRUE
17	12/03/1999	22:13:10	172.16.112.50	portsweep	TRUE
18	12/03/1999	22:43:18	172.16.112.50	ftp-write	TRUE

The author also tested the classifier with alerts produced from Honeynet traffic between 1st of April 2006 and 30th of April 2006, and the results are given in Table 4-6. The author noted that there were five alert samples from the Honeynet traffic in the training sample. The alerts were highlighted in yellow in Table 4-8 at page 85.

The training sample signature IDs was also compared with the signature

IDs from the Honeynet alerts (see Table 4-7). This was to ensure that alert signatures from the Honeynet alerts were minimum in the training sample signatures ID. The author found that only seven out of 34 signature IDs from the Honeynet alerts were in the training sample signature IDs. This suggests that the alerts were new and unknown to the classifier. From the results, the classifier detected true alerts in this environment with a 99.1% detection rate.

Sample alerts from Honeynet traffic can also be used to train the classifier. However, the traffic which is free from intrusions, is needed. This will give the samples of false alerts.

Table 4-6: Classification results for the Honeynet traffic.

	Total Alerts	Classified as false	Classified as true	% of detected false alert
April 2006	16521	149	16372	0.9%

Table 4-7: The quantity of alerts according to their alert signatures for April, 2006.

no.	SigID	SignatureName	Priority	Class	Quantity
1	1	SNMP public access udp	2	medium:attempted-recon	36
2	2	SNMP request udp	2	medium:attempted-recon	36
3	15	WEB-FRONTPAGE /_vti_bin/ access	2	medium:web-application-activity	8
4	23	(portscan) UDP Portsweep	3	medium:bad-unknown	255
5	42	ICMP PING NMAP	2	medium:attempted-recon	111
6	44	(portscan) TCP Portscan	3	medium:bad-unknown	312
7	45	(portscan) Open Port	3	medium:bad-unknown	2024
8	52	WEB-MISC cat%20 access	2	medium:attempted-recon	4
9	61	(portscan) UDP Portscan	3	medium:bad-unknown	143
10	73	(portscan) TCP Portsweep	3	medium:bad-unknown	1021
11	180	MS-SQL Worm propagation attempt	2	medium:misc-attack	3832
12	181	MS-SQL Worm propagation attempt OUTBOUND	2	medium:misc-attack	3832
13	182	MS-SQL version overflow attempt	3	low:misc-activity	3832
14	183	BAD-TRAFFIC udp port 0 traffic	3	low:misc-activity	137
15	184	ICMP Destination Unreachable Communication Administratively Prohibited	3	low:misc-activity	20
16	214	(http_inspect) BARE BYTE UNICODE ENCODING	3	medium:bad-unknown	31
17	216	(http_inspect) OVERSIZE REQUEST-URI DIRECTORY	3	medium:bad-unknown	71
18	217	ICMP redirect host	2	medium:bad-unknown	1
19	218	WEB-PHP xmlrpc.php post attempt	1	high:web-application-attack	115
20	219	WEB-MISC Cisco IOS HTTP configuration attempt	1	high:web-application-attack	14
21	220	WEB-PHP remote include path	1	high:web-application-attack	6
22	221	NETBIOS SMB-DS Session Setup NTMLSSP unicode asn1 overflow attempt	3	low:protocol-command-decode	6
23	222	WEB-MISC Chunked-Encoding transfer attempt	1	high:web-application-attack	8
24	223	WEB-FRONTPAGE rad fp30reg.dll access	2	medium:web-application-activity	8
25	224	ICMP PING CyberKit 2.2 Windows	3	low:misc-activity	80
26	225	WEB-MISC cacti graph_image.php access	2	medium:web-application-activity	3
27	226	ICMP Destination Unreachable Communication with Destination Network is Administratively Prohibited	3	low:misc-activity	1
28	227	SCAN SolarWinds IP scan attempt	3	low:network-scan	40
29	228	WEB-CGI awstats access	2	medium:web-application-activity	27
30	229	ICMP Destination Unreachable Communication with Destination Host is Administratively Prohibited	3	low:misc-activity	4
31	230	WEB-CGI awstats.pl configdir command execution attempt	1	high:attempted-user	4
32	231	BAD-TRAFFIC tcp port 0 traffic	3	low:misc-activity	23
33	232	WEB-MISC WebDAV search access	2	medium:web-application-activity	1
34	233	ICMP superscan echo	2	medium:attempted-recon	13

4.4 Integrating the Classifier Output with Visualisation

The outcomes from the classifier were applied to the visualisation software. The users were given the choice of either to apply the classifier or not. In the overview display (3D AlertGraph), when the classifier is applied, the true alerts will be coloured in red to highlight the area of interest. In other views, the true alerts will be coloured in red, the false alerts will be coloured in blue and can also be hidden.

Figure 4-2 shows a visual image of the classifier outcomes in the scatter plot view. Alerts in blue showed the false alerts. The image in Figure 4-3 shows there were no alerts which suggested there were no true alerts. The image was

visualised with filtering of the false alerts. The images are produced from the monitoring period Wednesday 17/03/1999 from 13:00 to 15:00.

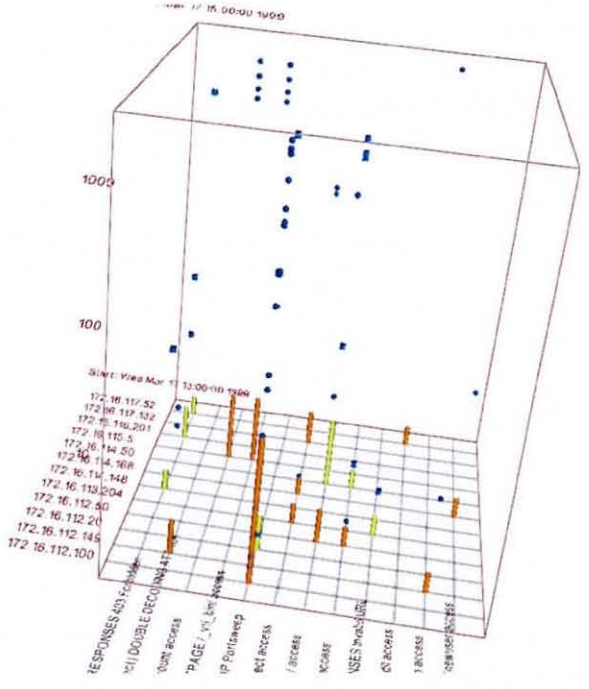


Figure 4-2: Visualisation with classifier and all alerts were displayed.

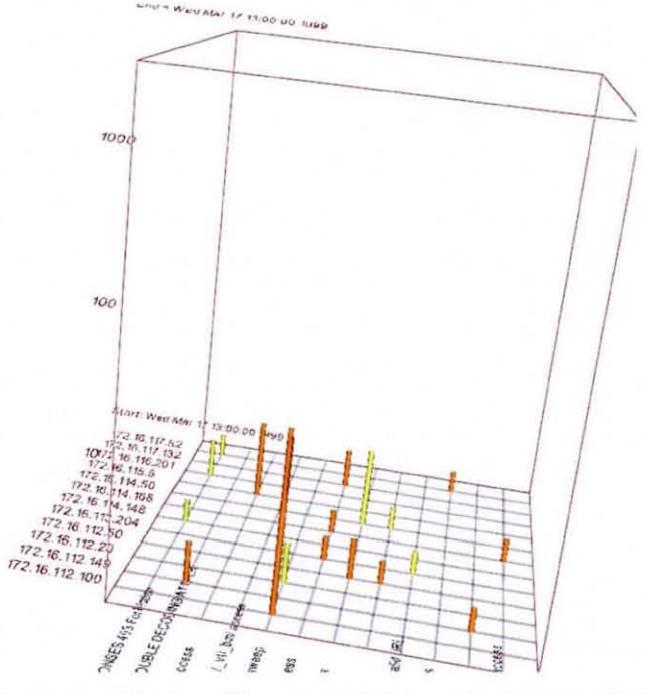


Figure 4-3: Visualisation with classifier and all false alerts were hidden.

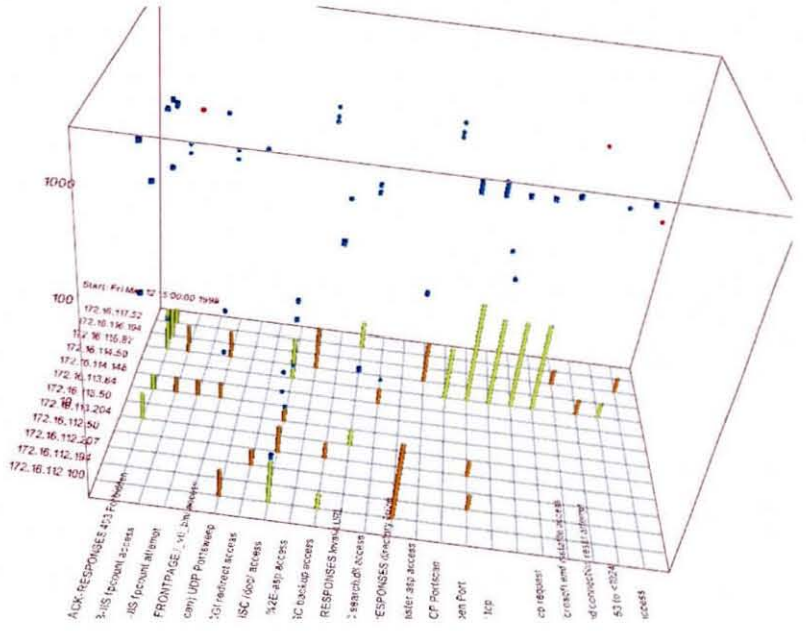


Figure 4-4: Visualisation without applying the classifier.

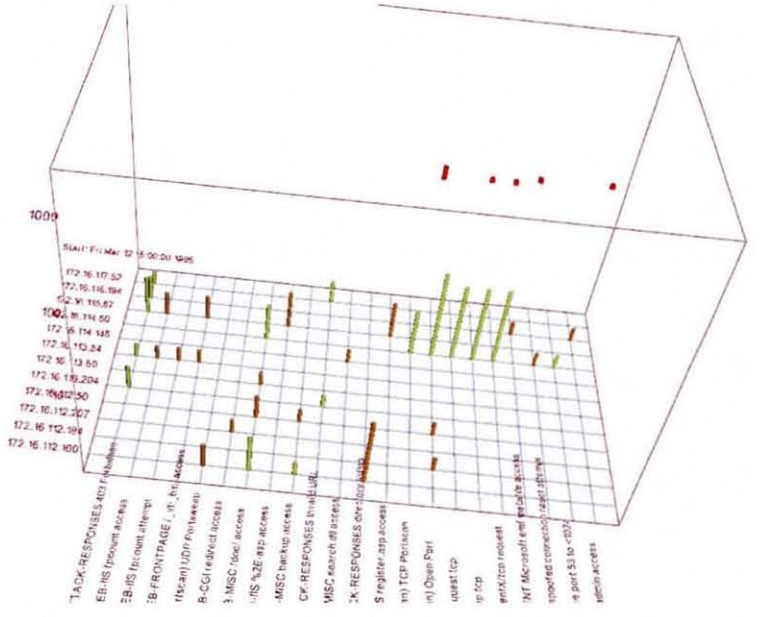


Figure 4-5: Visualisation with classifier and only the classified true alerts were displayed.

The images in Figure 4-4 and Figure 4-5 were from the monitoring period Friday 12/03/1999 between 15:00 and 17:00. Figure 4-4 shows the visualisation without applying the false alert classifier. By looking at the image, it was difficult for the user to identify the true alerts in the image. However, when the classifier was applied and the false alerts were filtered, the attacks to the host 172.16.114.50 became observable. The true attacks were coloured red in

Figure 4-5.

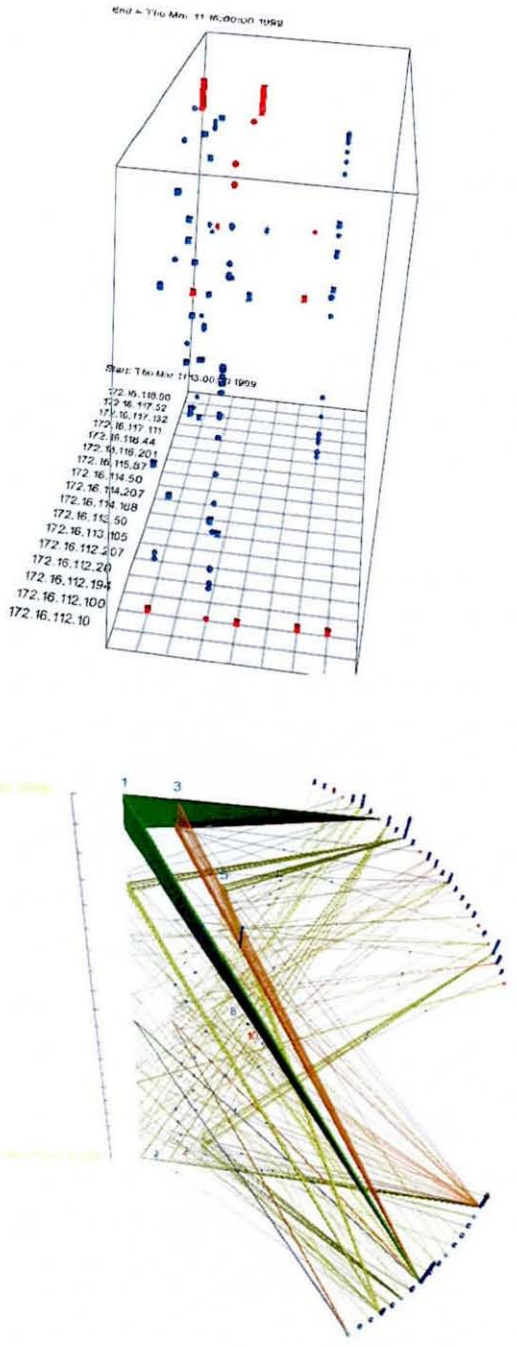


Figure 4-6: These figures show the observation in the scatter plot (top) and the timeline view (bottom) without hiding the false alerts in the same monitoring period.

The images in Figure 4-6 and Figure 4-7 show another advantage of using the false alert classifier with visualisation. The scatter plot pictures in Figure 4-6 were the image with this classifier, but the false alerts were unfiltered. As the true alerts were coloured red and the false alerts were coloured blue, one could still identify the true alerts. However, the image in the bottom of Figure 4-6, shows the alerts in the timeline view and it was not easy to identify the true alerts in this image. When the false alerts were filtered after the false alert classifier had been applied, the pattern of the attacks emerged (image of the bottom of Figure 4-7). The monitoring period of the images was Thursday 11/03/1999 between 13:00 and 15:00.

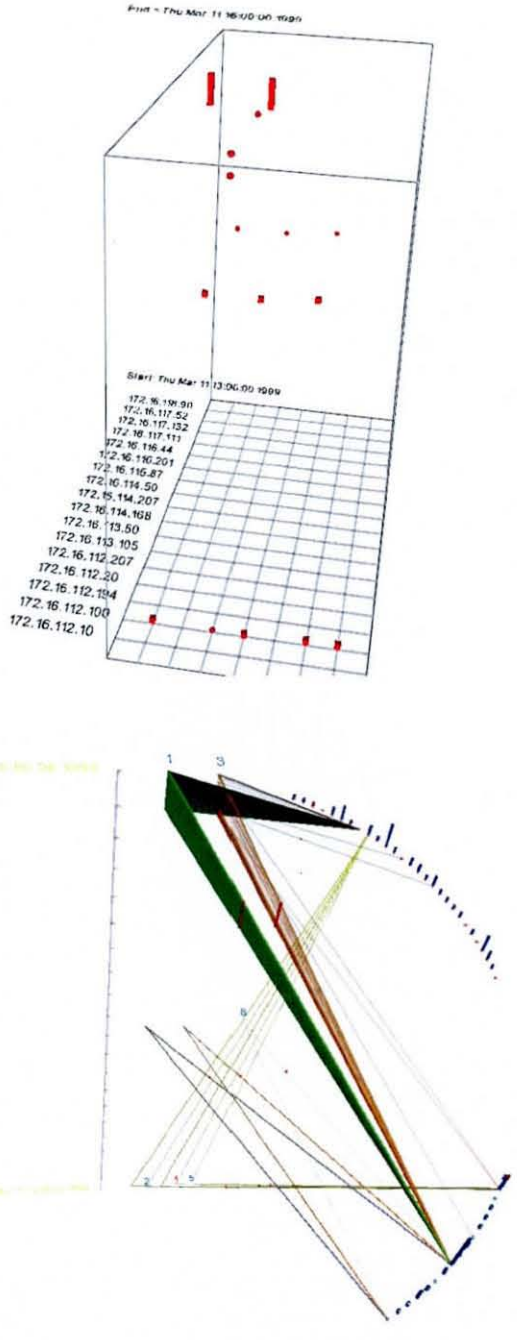


Figure 4-7: These were the images after applying false alert classifier and hiding the false alerts.

The pictures in Figure 4-8 show the different views before and after applying the classifier in the scatter plot view with geographical location.

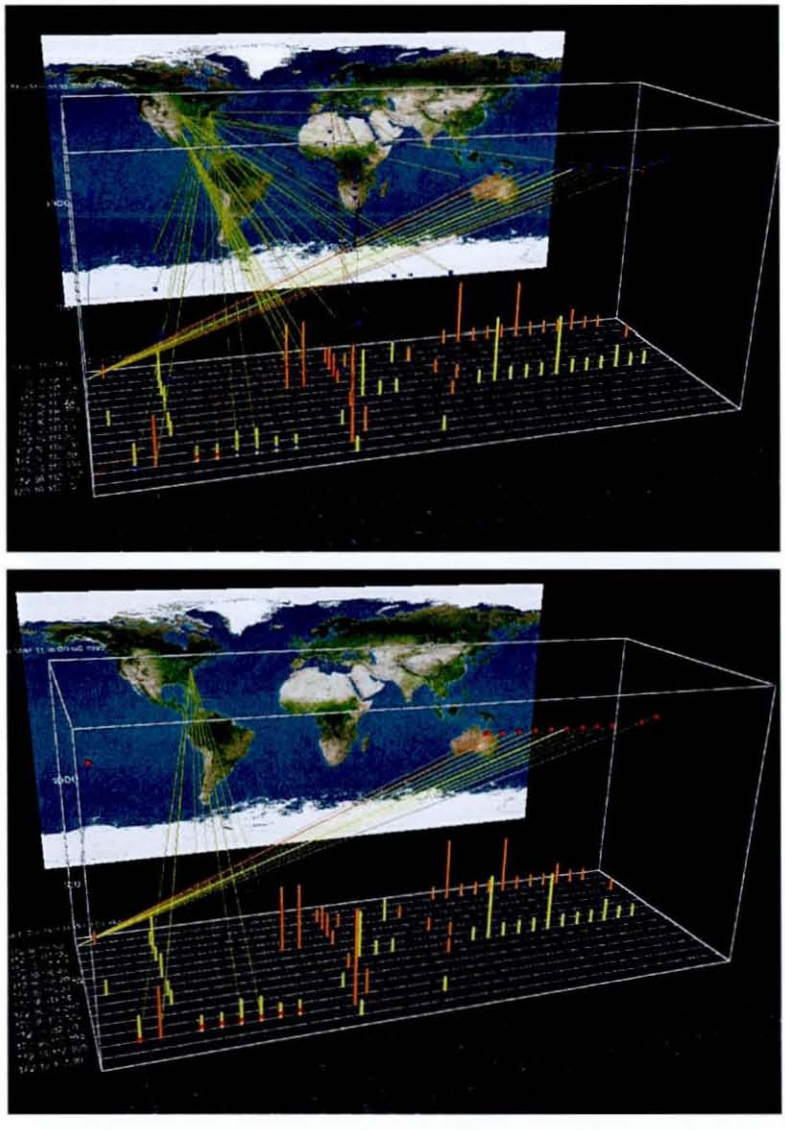


Figure 4-8: Images from the scatter plot with geographical location before (top) and after (bottom) applying the classifier.

In the situation where the classifier wrongly classified the attack as mentioned in the previous section, the visualisation could help to point out this event. An attack that was wrongly classified was attack number 5 as in the Table 4-5 at page 74. In the 3D AlertGraph, the attack number 5 was highlighted in red as it had a high number of alerts in the time interval.

Figure 4-9: Showing the occasions where the classifier wrongly classified the attack.

4.4.1 Real-time Monitoring

The false alert classifier can also be applied with real-time monitoring. If the false alert classifier option was chosen, the alert received would go through the classifier to predict the category of the alert before being visualised. The software would display the true alerts and filter the false ones according to the user settings. With this function users could directly identify the true alerts as all alerts displayed were the true alerts. As there were some internal procedures involved to visualise the alerts, the users should expect a delay of between 3 to 20 seconds before getting new alerts to be displayed. This delays also depending upon hardware performance of the system.

4.5 Conclusions

The author has shown that a decision tree classifier could help the analyst to identify true alerts and allow a pattern of true alerts to emerge in the visualisation. As the classifier did not remove the false alerts, the analyst could analyse them, if it was felt necessary. As a matter of fact, they could provide the classifier with new samples and updated the classifier model.

In summary, this work incorporated into the visualisation method a false alert classifier that helped users to identify the real alerts. The classifier used a decision tree learner and built a classifier model from seven alert attributes that had been generalised. The outcome was visualised with a different alert colour and the false alerts could be hidden. A utility has been created to help users to add new training samples into the classifier model. The classifier has been tested with the DARPA1999 dataset and a month of data from Honeynet traffic. The results showed the classifier performed well and helped the analysts to identify the real alerts.

Table 4-8: The training sample lists. The yellow area shows alerts from the Honeynet data.

No	Timestamp	DstIP	SrcIP	SigID	Classifier	Quantity
1	Mon Mar 08 13:00:27 1999 (920898027.0)	172.16.112.5	192.168.1.30	1	false	120
2	Mon Mar 08 13:00:27 1999 (920898027.0)	172.16.112.5	192.168.1.30	2	false	120
3	Mon Mar 08 13:00:59 1999 (920898059.0)	172.16.112.100	206.48.44.18	3	true	1
4	Mon Mar 08 13:01:38 1999 (920898098.0)	172.16.112.194	137.245.85.134	5	false	2
5	Mon Mar 08 13:01:59 1999 (920898119.0)	172.16.112.100	206.48.44.18	9	true	1
6	Mon Mar 08 13:01:59 1999 (920898119.0)	172.16.112.100	206.48.44.18	6	true	1
7	Mon Mar 08 13:01:59 1999 (920898119.0)	172.16.112.100	206.48.44.18	10	true	1
8	Mon Mar 08 13:01:59 1999 (920898119.0)	172.16.112.100	206.48.44.18	7	true	1
9	Mon Mar 08 13:01:59 1999 (920898119.0)	172.16.112.100	206.48.44.18	11	true	1
10	Mon Mar 08 13:01:59 1999 (920898119.0)	172.16.112.100	206.48.44.18	8	true	1
11	Mon Mar 08 13:16:59 1999 (920899019.0)	172.16.112.100	206.48.44.18	17	false	1
12	Mon Mar 08 13:16:59 1999 (920899019.0)	172.16.112.100	206.48.44.18	14	false	1
13	Mon Mar 08 13:16:59 1999 (920899019.0)	172.16.112.100	206.48.44.18	18	false	1
14	Mon Mar 08 13:16:59 1999 (920899019.0)	172.16.112.100	206.48.44.18	15	false	1
15	Mon Mar 08 13:16:59 1999 (920899019.0)	172.16.112.100	206.48.44.18	19	false	1
16	Mon Mar 08 13:16:59 1999 (920899019.0)	172.16.112.100	206.48.44.18	16	false	1
17	Mon Mar 08 13:16:59 1999 (920899019.0)	172.16.112.100	206.48.44.18	13	false	1
18	Mon Mar 08 13:17:00 1999 (920899020.0)	172.16.112.100	206.48.44.18	21	false	1
19	Mon Mar 08 13:17:00 1999 (920899020.0)	172.16.112.100	206.48.44.18	22	false	1
20	Mon Mar 08 13:17:00 1999 (920899020.0)	172.16.112.100	206.48.44.18	20	false	1
21	Mon Mar 08 13:50:12 1999 (920901012.0)	172.16.113.50	206.229.221.82	24	true	5
22	Mon Mar 08 14:05:00 1999 (920901900.0)	172.16.112.100	135.8.60.182	26	false	1
23	Mon Mar 08 14:07:58 1999 (920902078.0)	172.16.113.105	206.253.217.8	5	false	1
24	Mon Mar 08 14:14:38 1999 (920902478.0)	172.16.113.105	131.84.1.68	5	false	2
25	Mon Mar 08 14:26:54 1999 (920903214.0)	172.16.114.50	197.182.91.233	29	false	1
26	Mon Mar 08 14:27:19 1999 (920903239.0)	172.16.117.103	207.200.75.201	30	false	2
27	Mon Mar 08 14:29:28 1999 (920903368.0)	172.16.112.207	207.200.75.201	30	false	1
28	Mon Mar 08 14:39:12 1999 (920903952.0)	172.16.114.50	199.174.194.16	32	true	240
29	Mon Mar 08 14:48:24 1999 (920904504.0)	172.16.112.100	172.16.112.50	26	false	1
30	Mon Mar 08 14:59:19 1999 (920905159.0)	172.16.113.84	137.245.85.134	5	false	1
31	Mon Mar 08 15:02:49 1999 (920905369.0)	172.16.113.84	172.16.112.100	34	false	7
32	Tue Mar 09 13:44:21 1999 (920987061.0)	172.16.114.50	153.37.134.17	44	true	16
33	Fri Mar 12 13:07:15 1999 (921244035.0)	172.16.114.50	209.117.157.183	51	true	1
34	Fri Mar 12 22:05:12 1999 (921276312.0)	172.16.113.105	207.200.75.201	30	false	1
35	Fri Mar 12 22:08:28 1999 (921276508.0)	172.16.113.105	137.245.85.134	5	false	2
36	Fri Mar 12 22:18:09 1999 (921277089.0)	172.16.114.50	172.16.112.50	26	false	1
37	Sat Apr 01 14:49:51 2006 (1143899391.0)	62.231.131.227	222.182.103.122	180	true	1
38	Sat Apr 01 14:49:51 2006 (1143899391.0)	62.231.131.227	222.182.103.122	181	true	1
39	Sat Apr 01 14:49:51 2006 (1143899391.0)	62.231.131.227	222.182.103.122	182	true	1
40	Sun Apr 02 14:29:33 2006 (1143984573.0)	62.231.131.226	203.116.29.175	218	true	9
41	Sun Apr 02 14:29:33 2006 (1143984573.0)	62.231.131.229	203.116.29.175	218	true	7

Chapter 5: The Simulation Analysis

This chapter explains the simulation analyses to test the visualisation prototype software. The method and the dataset used are described. The examples of known attacks and simulated intrusion attempt are also explained.

5.1 The simulation framework

To test the usefulness of the prototype software, the author evaluated it in two different scenarios. In the first scenario, the prototype was tested using off-line traffic from the DARPA 1999 dataset and Honeynet traffic data from the High Speed Networks laboratory. The second scenario used real-time traffic in an experimental setting with an attacker machine injecting intrusion traffic into the network.

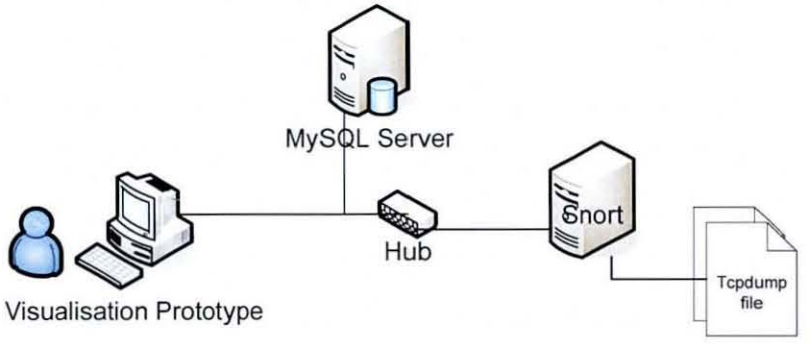


Figure 5-1: The off-line software evaluation – scenario 1.

In the first scenario, the dataset [74] used was the dataset created by The Information Systems Technology Group (IST) at the Lincoln Laboratory, Massachusetts Institute of Technology (MIT). The project was carried out under Defence Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory sponsorship. The dataset was collected and distributed for evaluation of computer network intrusion detection systems. The project had distributed two evaluation datasets known as DARPA1998 and DARPA1999. In this study, the author used only the DARPA1999 dataset.

The dataset contained five weeks of Tcpdump traffic. Traffic from the first week and the third week did not contain any intrusions while the second week contained intrusions. The intrusion events in the second week were labelled so

the researchers knew what the intrusions were and when the events occurred. In analysing the dataset using Snort IDS, the Snort IDS had missed some of the intrusions. Thus any alert on those attacks was not received. Table 5-1 shows the intrusions list with the attacks detected by Snort are highlighted in yellow. The author noticed that the events in the dataset appeared five hours late. This is because the timestamp was converted to a local Coordinated Universal Time (UTC) time setting which was the London time used by Snort.

Table 5-1: Intrusions list for the second week of DARPA 1999 dataset.

ID	Date	StartTime	Destination	IP Address	Name	Snort Detection
1	08/03/1999	13:01:01	hume.eyrie.af.mil	172.16.112.100	Ntinfoscan	Yes
2	08/03/1999	13:50:15	zeno.eyrie.af.mil	172.16.113.50	pod	Yes
3	08/03/1999	14:39:16	marx.eyrie.af.mil	172.16.114.50	back	Yes
4	08/03/1999	17:09:18	pascal.eyrie.af.mil	172.16.112.50	httptunnel	Missed
5	08/03/1999	20:57:15	pascal.eyrie.af.mil	172.16.112.50	land	Missed
6	08/03/1999	22:27:13	marx.eyrie.af.mil	172.16.114.50	secret	Missed
7	09/03/1999	00:09:17	pascal.eyrie.af.mil	172.16.112.50	ps attack	Missed
8	09/03/1999	13:44:17	marx.eyrie.af.mil	172.16.114.50	portsweep	Yes
9	09/03/1999	14:43:51	pascal.eyrie.af.mil	172.16.112.50	eject	Missed
10	09/03/1999	15:06:43	marx.eyrie.af.mil	172.16.114.50	back	Yes
11	09/03/1999	15:54:19	zeno.eyrie.af.mil	172.16.113.50	loadmodule	Missed
12	09/03/1999	16:49:13	pascal.eyrie.af.mil	172.16.112.50	secret	Missed
13	09/03/1999	19:25:16	pascal.eyrie.af.mil	172.16.112.50	mailbomb	Missed
14	09/03/1999	18:05:10	172.016.112.001-114.254		ipsweep	Missed
15	09/03/1999	21:11:15	marx.eyrie.af.mil	172.16.114.50	phf	Yes
16	09/03/1999	23:06:17	pascal.eyrie.af.mil	172.16.112.50	httptunnel	Missed
17	10//3/1999	17:02:13	marx.eyrie.af.mil	172.16.114.50	satan	Yes
18	10//3/1999	18:44:18	pascal.eyrie.af.mil	172.16.112.50	mailbomb	Missed
19	10//3/1999	20:25:18	marx.eyrie.af.mil	172.16.114.50	perl(failed)	Missed
20	11//3/1999	01:17:10	172.016.112.001-114.254		ipsweep	Missed
21	11//3/1999	04:23:00	pascal.eyrie.af.mil	172.16.112.50	eject(console)	Missed
22	11//3/1999	04:56:14	hume.eyrie.af.mil	172.16.112.100	crashiis	Missed
23	11/03/1999	13:04:17	hume.eyrie.af.mil	172.16.112.100	crashiis	Yes
24	11/03/1999	14:33:17	marx.eyrie.af.mil	172.16.114.50	satan	Yes
25	11/03/1999	15:50:11	marx.eyrie.af.mil	172.16.114.50	portsweep	Yes
26	11/03/1999	16:04:16	pigeon.eyrie.af.mil	172.16.114.207	neptune	Yes
27	11/03/1999	17:57:13	marx.eyrie.af.mil	172.16.114.50	secret	Missed
28	11/03/1999	19:25:17	marx.eyrie.af.mil	172.16.114.50	perl	Missed
29	11/03/1999	20:47:15	pascal.eyrie.af.mil	172.16.112.50	land	Missed
30	11/03/1999	21:36:10	172.016.112.001-114.254		ipsweep	Missed
31	12/03/1999	00:16:18	pascal.eyrie.af.mil	172.16.112.50	ftp-write	Yes
32	12/03/1999	13:07:17	marx.eyrie.af.mil	172.16.114.50	phf	Yes
33	12/03/1999	13:10:40	marx.eyrie.af.mil	172.16.114.50	perl(console)	Missed
34	12/03/1999	13:16:46	pascal.eyrie.af.mil	172.16.112.50	ps(console)	Missed
35	12/03/1999	14:18:15	duck.eyrie.af.mil	172.16.113.84	pod	Yes
36	12/03/1999	16:20:15	marx.eyrie.af.mil	172.16.114.50	neptune	Yes
37	12/03/1999	17:40:12	hume.eyrie.af.mil	172.16.112.100	crashiis	Yes
38	12/03/1999	18:12:17	zeno.eyrie.af.mil	172.16.113.50	loadmodule	Missed
39	12/03/1999	19:06:17	marx.eyrie.af.mil	172.16.114.50	perl(failed)	Missed
40	12/03/1999	19:24:18	pascal.eyrie.af.mil	172.16.112.50	ps(console)	Missed
41	12/03/1999	20:24:16	pascal.eyrie.af.mil	172.16.112.50	eject	Missed
42	12/03/1999	22:13:10	pascal.eyrie.af.mil	172.16.112.50	portsweep	Yes
43	12/03/1999	22:43:18	pascal.eyrie.af.mil	172.16.112.50	ftp-write	Yes

Another set of data for this study was taken from the Honeynet traffic data generated by the High Speed Network research lab in the Electronic and

Engineering department, Loughborough University. Two months of data from 01/03/2006 to 30/04/2006 were visualised and studied. The off-line Tcpdump datasets were examined using Snort IDS with alerts logging to the MySQL database. The signatures and rules used in the Snort IDS were obtained from the Snort community group (www.snort.org).

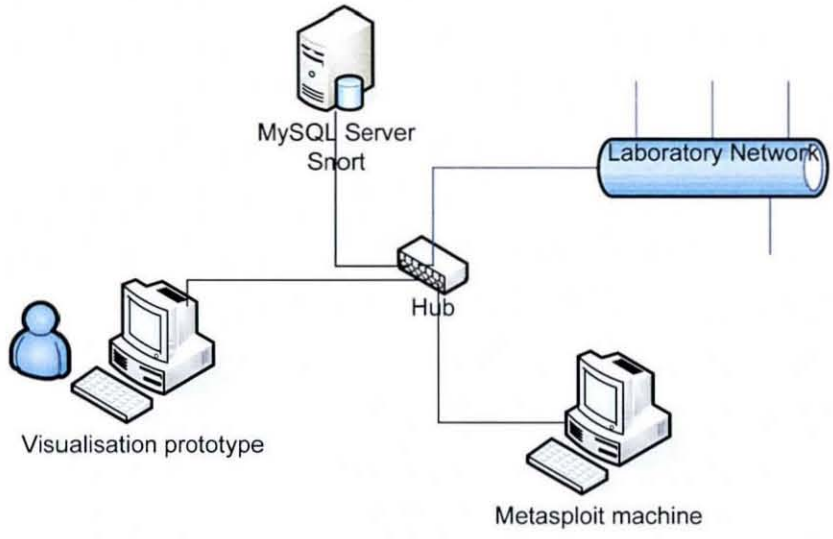


Figure 5-2: The real-time software evaluation – scenario 2.

In the second scenario, the setting was as shown in Figure 5-2. In this setting, an attacker machine was added into the network. The attacker machine was a machine with Metasploit Framework 3.0 software installed. Metasploit Framework software (www.metasploit.com) is an open source development platform for creating security tools and exploits. The software is used by network security professionals to perform penetration tests; system administrators to verify patch installations; product vendors to perform regression testing; and security researchers. In this scenario, the Snort IDS was set with alerts logging to the MySQL database and the real-time monitoring was performed in the visualisation prototype. Then, some exploits were run and the visualisation display was observed.

5.2 Samples analyses

To analyse the dataset using this visualisation prototype, some basic

approaches to identify the anomalous alerts were followed. Firstly, the alerts were displayed in the 3D AlertGraph view and then in the scatter plot view. In the 3D AlertGraph view, the focus was on the area where there were many alerts in the time interval and the area with colour red or other colours except blue.

In the scatter plot view, the focus was on irregularity alert flows, because the irregularities were normally caused by abnormal system behaviour [65]. The irregularities could be identified in the visual display either by a host:

- having very few alerts
- having too many alerts with different signatures
- was the source of attacks.

When suspecting a suspicious alert, the alert details were examined and the alert priority and the packet payload were used to understand the alert cause. Then, try to view it in the other displays and use the filter options to confirm and to corroborate the abnormal behaviour.

5.2.1 Ntinfoscan and Pod DOS

An observation on Monday 8 March 1999 between 13:00 to 14:00 GMT from the DARPA1999 dataset showed there were some suspicious events that needed further attention. Figure 5-3 shows the images from two different views.

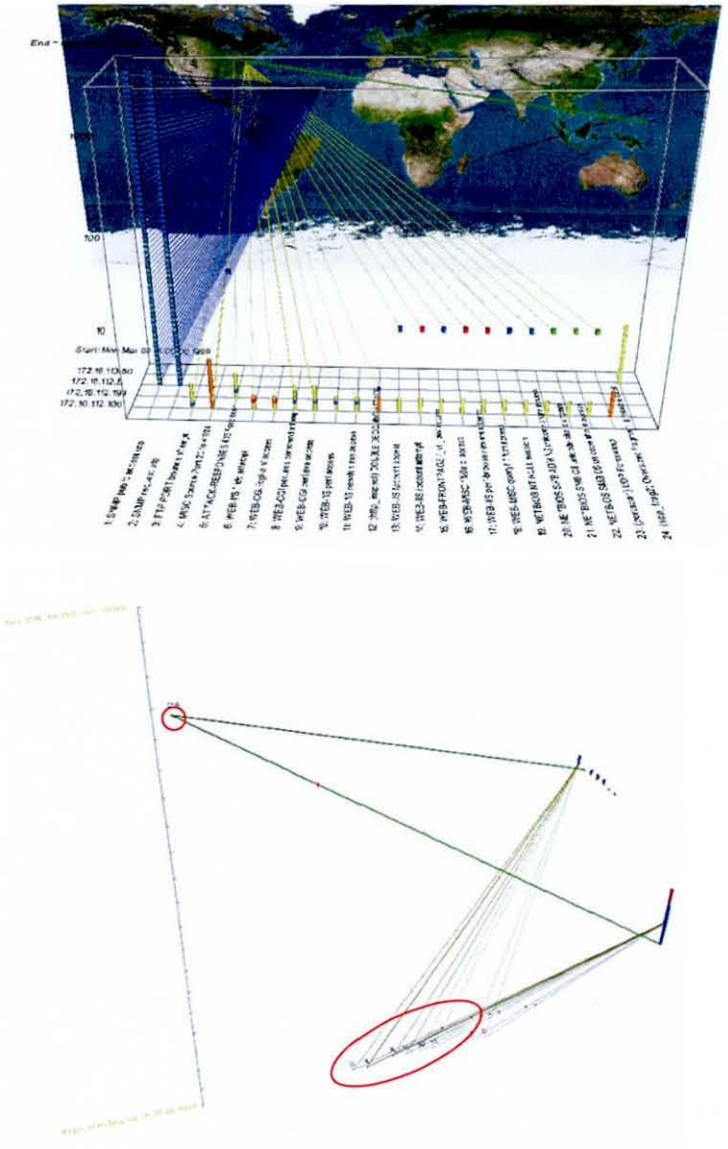


Figure 5-3: The top image shows various attacks to local hosts in the geographical view. The lower image shows the attacks after false alert filtering in the timeline view.

The top image in Figure 5-3 showed there were various attacks from the same source targeting the same destination at the start of the observation and a few minutes later. These were shown as yellow lines which referred to the TCP protocol. Other suspicious events were the continuous attacks using the UDP protocol (blue lines). Then, there were attacks with an unknown protocol which were coloured in green with a high alert quantity bar.

When analysing the first group of alerts, the attacks could be assumed from a unique source targeting a unique host, using the TCP protocol and occurring roughly at the same time but with a different alert signature. Visualising in the signature class type revealed the signature class types were "attempt-recon", "misc-attack", "web-application-activity" and "web-application-attack". Analysis from the packet payload of the alerts showed that these were perl command scripts launched by the external host at the targeted host. Therefore, there were grounds to suspect the alerts were true alerts. From the dataset intrusions list (Table 5-1), the multiple attacks from the same source and destination were "Ntinfoscan" attack. The "Ntinfoscan" were used by the attacker to the targeted machine to gather information such as services, login and file system details.

From the analysis of the packet payload of the attacks a few minutes later, the author found that the attacks were caused by the 'fpcount.exe' script. 'Fpcount.exe' is the Microsoft FrontPage common gateway interface (CGI) script to count visitors accessing a web page. The continuous attacks using the UDP protocol (blue lines) were the 'SNMP Public Access UDP' and 'SNMP request UDP'. These alerts are classified by the classifier as false alerts.

Meanwhile, the attack using an unknown protocol and having many alerts may also indicate suspicious events. Details of the alert showed the attack was using an unknown computer port and the packet payload contained a series of '0' hexadecimal values. As the computer port was unknown, and the packet payload was a series of '0' hexadecimal values, there were good grounds to be suspicious. From the alert signature, the attacks were a DOS attack.

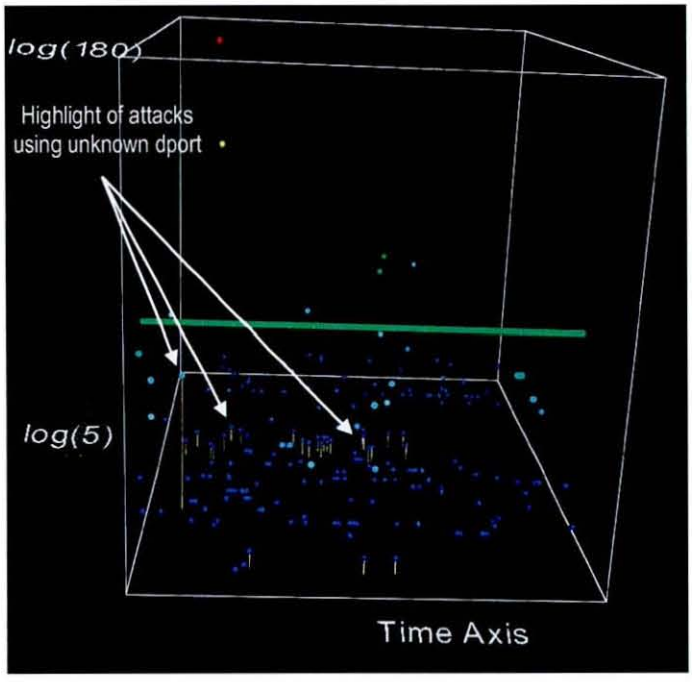


Figure 5-4: The yellow bars highlighted attacks using unknown destination port in 3D AlertGraph.

From the intrusions list (Table 5-1) of that day, it appeared that "Ntinfoscan" and Pod DOS attacks did happen, and the visualisation was able to direct attention to these attacks. Moreover, the visualisation prototype allows the users to understand better the information and gives access to detailed information of the attacks. The detailed information of the alerts and traffic are essential to determine the true and false alerts.

5.2.2 Portsweep

Figure 5-5 shows another example of analysis. The display in the 3D AlertGraph for 24 hours period, starting from 13:00 Monday 9 March 1999, highlighted a continuous horizontal red spheres and a high quantity of alerts at the beginning hours of the monitoring. The details from the pick object revealed that, at the red spheres area, there was a unique host attacking a unique local host using multiple destination ports. This was an indication of a portsweep attack. Then, in the first high quantity alerts (Figure 5-5), the attacks were from the same previous host attacking the same local host, but using a unique destination port number 7. The second high quantity alerts, about one and a

half-hours later, were attacks to the same local host using http port 80.

The observation in the parallel coordinates plot view, on Monday 9 March 1999 between 13:00 to 14:00 GMT from the dataset, showed there were lines covering the entire destination port axis (see Figure 5-5). The attack list showed the particular machine was under a portsweep attack. This confirmed the observation in the previous 3D AlertGraph view.

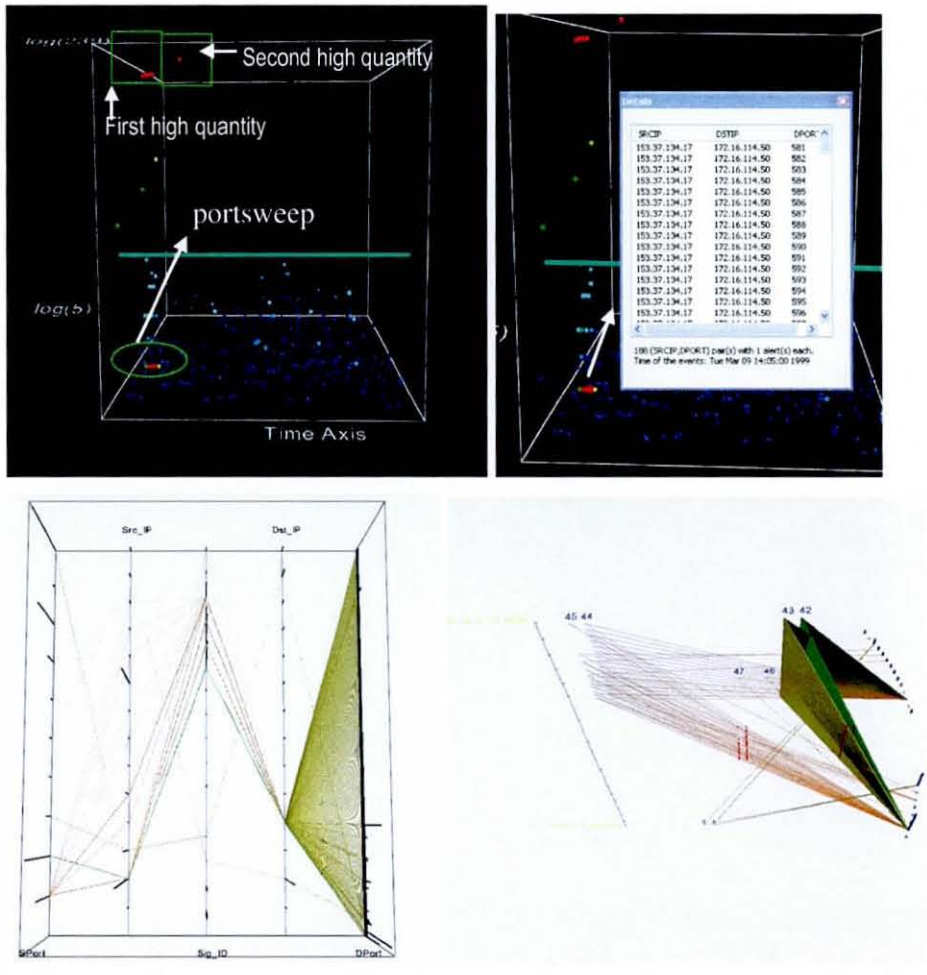


Figure 5-5: Portsweep Attack – The upper left picture is the 3D AlertGraph that shows the area of portsweeps and high quantity of alerts. The upper right picture shows the display of the pop-up window and green transparent layer highlighting the IP address 172.16.114.50. The bottom left picture is from the parallel coordinates plot that confirms the portsweep attack. The bottom right is the data in the timeline view.

The image in the timeline view revealed more information. The Snort IDS detected four attacks from attacker host, 153.37.134.17 to victim host 172.16.114.50. The alert signatures were 'ICMP ping nmap (id=42)', 'scan fin (id=43)', 'TCP portscan (id=44)' and 'portscan open port (id=45)'.

5.2.3 Slammer worm

The slammer worm is a self-propagating malicious code that exploits the vulnerability in the Resolution Service of Microsoft SQL Server 2000 and Microsoft Desktop Engine (MSDE) 2000. It attacks port 1434 using the UDP protocol. An observation of a day (07/04/2006) from the Honeynet traffic showed there were attacks by the slammer worm to the Honeynet.

Observation in the 3D AlertGraph on that day showed there were a constant number of alerts in each interval for the whole period attacking the same destination port 1434 to the multiple hosts. The two upper pictures in Figure 5-6 show the view in the 3D AlertGraph. The bottom picture in Figure 5-6 is the picture from the scatter plot view. From that picture, there were three attacks with alert signature ID of 180, 181, 182 and all the attacks used the UDP protocol. The total alerts for that period was 1354. Details of the three signature IDs showed the attacks were related to the MS-SQL worm which was a slammer worm. Other attacks were with signatures 42,184 and 216. Further investigation showed the signature alert 42 was ICMP NMAP ping. The signature 216 was an oversize uniform resource indicator (URI) request to http port 80. Signature 184 was an ICMP destination unreachable. The advantage of using the scatter plot view was that the attacks were clustered according to the destination host and the attack signature. This means the attacks were grouped and easy to analyse.

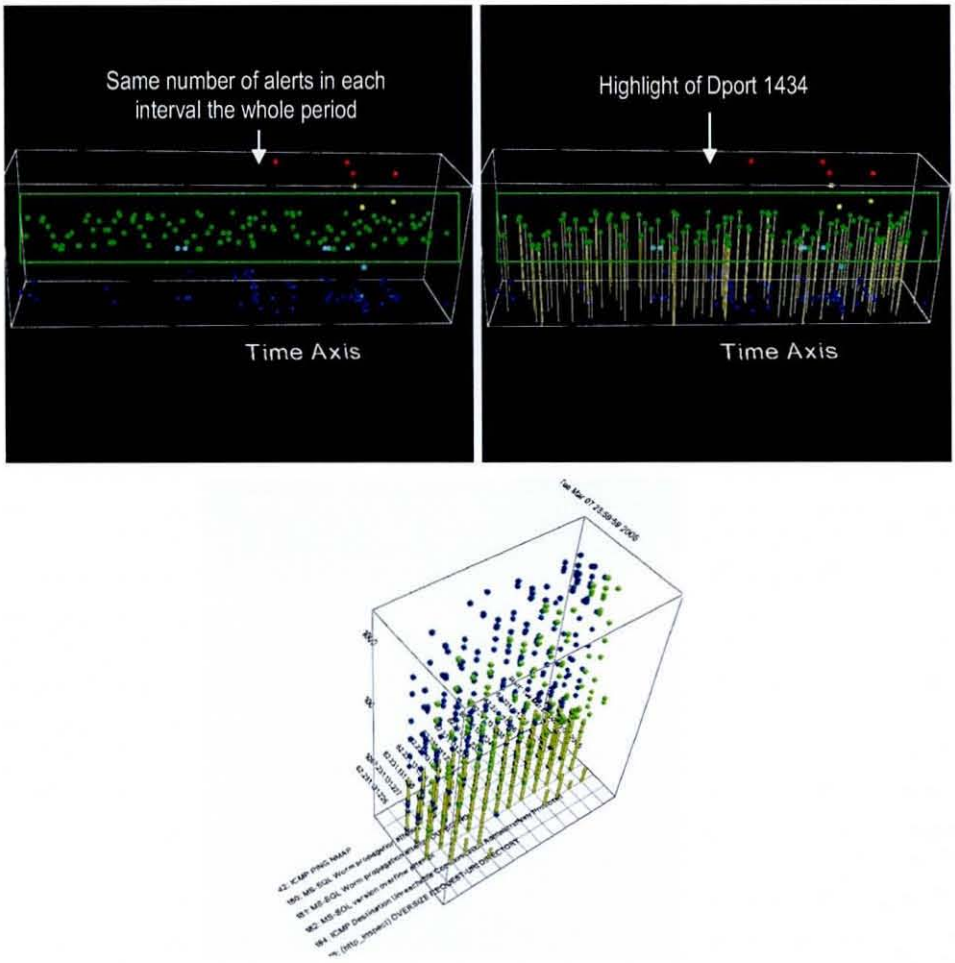


Figure 5-6: A day image of Honeynet alert data in 3D AlertGraph (upper) and scatter plot view (bottom). Signature 180, 181 and 182 show the Slammer worm attacking the local hosts.

5.2.4 Real-time Monitoring

To test the real-time monitoring of the prototype software, the scenario as in Figure 5-2 was set. Snort IDS was run on one machine in promiscuous mode with the alert logging to the MySQL database server. On the other machine, the visualisation software was set in the real-time monitoring mode with the false alert classifier turned on. By using metasploit framework 3.0 software, some attacks were launched to a computer in the network. The objective of this experiment was to observe the visualisation prototype effectiveness in the real-time monitoring.

In the real-time monitoring, the visualisation display is either in timeline view or scatter plot view. In both visualisation displays, the users have choices either to switch on the false alert classifier and filter the false alerts, or to display all alerts with the Snort alert priority using colour-codes. In the test, a machine in the network was first attacked by NMAP and then by a TCP Overflow attack. The visualisation prototype displayed both attacks within a second after the attacks were launched. It was concluded the visualisation prototype was able to display the attacks in the network within seconds of the attacks, if Snort IDS detected the attacks.

5.3 Conclusions

The off-line traffic analysis and the real-time monitoring scenarios showed the prototype visualisation software was able to direct analyst attention to the potential intrusion attempts. The visualisation prototype also allowed the analyst to view archive data and display the alerts in different views. Combining the false alert filter and the visualisation also helps the analyst to identify the true alerts. The visualisation prototype also allowed the user to access detail information about the alerts almost immediately including the packet payload information. The observation in 3D AlertGraph also revealed that there might be

- True attacks when there were high numbers of alerts in a specific interval.
- A portsweep attack when there was continuous horizontal red or other colours except blue in a particular host.
- A worm propagation when there were a constant number of alerts in many hosts throughout the monitoring period using the same destination port number.

It was concluded that the prototype could usefully direct the user's attention, and would also give better understanding of the attacks.

Chapter 6: User Evaluation of the Network Security Alerts Visualisation Tool

6.1 Introduction

Network security attack visualisation is a fairly new area of research and there are not many reports on user evaluation. Even in the subject of visualisation in general, few papers on visualisation applications or techniques reported their findings on user evaluation [75]. Furthermore from 12 studies that had user evaluation, Ellis [75] reported that two were flawed, five were problematic, one was an informal study, two were foregone conclusions and only two could be considered successful. These show that conducting user evaluation is not an easy process.

Despite that, the author believes evaluating network security visualisation tools is important, especially with human subjects. It is important because visualisation itself is about the representations of abstract data on the computer screen which the user has to interpret, and to make sense of the visual coded images. Moreover, the objective of most visualisation tools is to help users solve specific tasks, or have a bigger picture of the abstract information. Therefore, evaluating visualisation tools with their intended users plays an important part in measuring the tools effectiveness.

However, conducting user testing is somewhat challenging. First, to identify the real-world users or close to the real-world users of the intended visualisation tools were needed, rather than heavily rely on student participants. To find real-world users is difficult, and if it is found, he or she is normally a busy person.

The user evaluation is also time-consuming. The users have to familiarise themselves with the tools before conducting the test. The time needed for the training also varies from one participant to another, as each human has different ability to absorb and learn new information. As the prototype is only available on a single machine, the test has to be conducted in a one-to-one session, rather than in a group. Thus, it needs a lot of time.

Another challenge faced is to test the prototype in the real environment.

As the prototype does sniffs of the network traffic, it may violate the IT networking policy of some organisations. Most organisations are reluctant to allow such tests. Despites these challenges, an acceptable evaluation of the visualisation tool must be found.

6.2 Current Evaluation Practices in Information Visualisation

Plaisant [76] summarised reports from a survey of literature of fifty user studies on information visualisation systems. From the study, the authors found four categories of user evaluation:

- Controlled experiments comparing design elements. The studies in this category might compare specific widgets, or GUIs, or compare mapping information to graphical displays.
- Usability evaluation of a tool. These studies might provide feedback on the problems users faced with a tool and show how designers went on to refine their design.
- Controlled experiments comparing two or more tools. This is a common category of user evaluation study. For example, the comparison studies of three tree visualisation tools: SpaceTree, Hyperbolic and Window Explorer by [77]. Those studies usually try to compare the novel technique with the current technique.
- Case studies of tools in a realistic setting. This is the least common category. The advantage of case studies is that they report on users in their natural environment doing real tasks, demonstrating feasibility and in-context usefulness. The disadvantage is that they are time-consuming to conduct, and the results may not be replicated and generalised.

In usability evaluation, one popular technique is heuristic evaluation as proposed by Nielsen [58]. Heuristic evaluation is a usability evaluation method for computer software that helps to identify usability problems in the user interface design. The heuristic method consists of checking the software or tool against the standard usability principals, such as realistic user interface design, system status, user control and freedom, consistency, flexibility and ease of

use. It is low cost in terms of time, since it can be completed quite quickly. According to Neilson [78], only 3-5 evaluators are needed to identify 75-80% of all usability problems.

For network security visualisation tools, the works by Conti [79], Abdullah [61] and Komlodi [80] reported their user evaluation. Komlodi and others [80] used heuristic evaluation and focus group usability testing of their prototype named as IDtk (intrusion detection toolkit). From the study, they identified some usability problems in their visualisation tool. The summary of the problems are:

- The problem of using three mouse buttons to achieve rotational movement and zoom the 3D display to understand the data.
- Users getting lost in the 3D display by over-zooming or over-panning, a typical problem of 3D spaces. In these cases, there is no visual home position for the user to get back to the initial position.
- Users complaining that, after they exerted much effort to achieve a satisfactory view, then they are not able to save views in a history, or to mark certain views as default views to which they could return.

In this tool, the first usability issue found by Komlodi was addressed by applying the visualisation toolkit (VTK) zoom, rotate and pan default features with speed control. The user can choose to zoom or to rotate at a slow speed by bring the mouse control near to the centre of the image before zooming or rotating. The nearer the mouse cursor to the centre, the slower will be the motion. In VTK, the left button is for rotating, the right button is for zooming and the middle button is for panning. The zoom-in and zoom-out features depend on the mouse cursor position on the screen. Zoom-in is when the mouse cursor is on the upper area of the screen and zoom-out is when the mouse cursor is on the lower area of the screen.

The second and third issues were addressed by adding a default visual home position and save image feature. With the default home position, the user can go back to initial position if they lost image orientation. These features can be assessed from the menu bar.

The work by Abdullah [61] involved a case study of the prototype, IDSRainstorm in a real world setting with real world users at Georgia Institute of Technology, USA. The network administrators there used this tool in their work and gave feedback to the developer. The feedback was gathered in the form of questionnaires and user ratings. There are between 30,000-35,000 networked computers operational at any given time at Georgia Institute of Technology. Some of the questions asked were:

- About the user preference between using text logs and the visual tool.
- About any missing information from the overall view that could have pointed the important events.
- About the least useful or distracting features and the best features in the tool.

The users were also asked to rate the tool according the following areas:

- The information representation.
- The zoom and pan features.
- The mouse button features.
- The comparison of using the tool with manual investigation
- Was there enough information in the overview to get a general idea of network security alerts, identify alerts of interest and drill down for more detail.
- Is the tool easy to use and to learn.

On the other hand, Conti [79] evaluated their prototype effectiveness by addressing the key weaknesses of Ethereal[7] to detect malicious traffic. The prototype was tested by a group of students who took a network security course. They were asked to perform certain tasks, and their responses were recorded to identify the ability to detect malicious activity in the playback traffic, the precision of the detection, and the time they took to perform the detection tasks.

[7] Ethereal is an open source network protocol analyser.

6.3 Evaluation Method

Spence [81] describes information visualisation as a cognitive activity where users engaged with the potential gaining of an insight and understanding of the represented data. It involves perception and interpretation in the viewer or user's mind. The graphic in Figure 6-1 illustrates the information visualisation process. Therefore, evaluating information visualisation tools should not only address the human computer interaction (HCI) issues, such as the GUI and interaction, but must also include the understanding of the visual language. Assessment of the final objective of the visualisation tool, which is to help users to perform their daily tasks, should also be included.

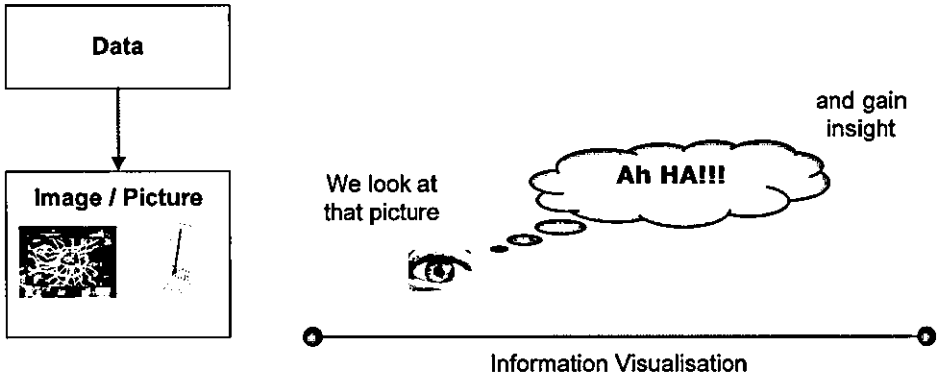


Figure 6-1: Information Visualisation process. The graphically encoded data are viewed to form a mental model of that data. The picture was redrawn from [81].

To evaluate this visualisation prototype tool, the tool was tested with researchers at the High Speed Networks laboratory and network administrators at Loughborough University. All of them were familiar with intrusion detection, and have a good background in network security in general. Their participation in this study was voluntary, and they have never used this tool before.

Before the evaluation session, the author showed them a quick demonstration of the prototype software, and they had some time to familiarise themselves with it. To test whether the users are gaining insight from the visual tool or not, they were asked to identify some true attacks to the network using

off-line data. The data used in this study was the DARPA 1999 dataset. The author asked them to interact with the visual image and to get the attack details, such as the attacker location, the IP address, and the attack types. The author also asked them to switch to different displays to get more insight and different perspectives of the data. The displays were: the overview, scatter plot view, parallel plot view, timeline view, plane view and world views. For each display, the author asked them to rate whether the visual display and design were intuitive and made sense or not.

To check the usability of the tool in general, the participants were asked to give their rating according to the following questions using the Likert[8] Scale 1 to 5 with 1, for the most disagree, and 5, for the most agree:

- Are the graphical user Interfaces (GUI) user-friendly?
- Does the image interaction (pick actor) help users to get more information?
- Does the false alert classifier help users to identify the true alerts?
- Is the alert filter feature helpful?
- Is it necessary to visualise the alerts in real-time?
- Are the reporting tools such as saving the image in JPEG or BMP and alert listing helpful in preparing a network security report?
- How do you rate this prototype software compared with other tools that you have used before?
- Does this prototype software help you in performing network security tasks?

At the end, the participants were asked to give their opinion or any suggestion to further improve this tool. In this evaluation, the author did not take any time measurement for the user to complete the task. This is because the author was not able to find any complete analyst software to compare the time with the visualisation tools.

[8] Likert Scale is a type of survey question where respondents are asked to rate the level at which they agree or disagree with a given statement.

During the user testing, the author was around to help the participant carries out the tasks when necessary. The author also discussed with them and noted down the areas where they felt there were usability problems.

6.4 The Usability Report

Following the Nielsen [78] findings, the usability test has been conducted with five participants. Even though the number of participants was small, the author managed to identify bugs[9], usability problems and suggestions for further improvement.

The bugs found were:

- Image interaction was not functioning after the user pressed the 'Stat' button in the filter panel. This happened because the module for plotting the bar chart was conflicting with wxVTK window. The wxVTK was the class responsible for displaying the visual. The author has fixed the problem by properly displaying the bar chart using wxFrame.
- The pop-up window after the 'pick' actor was not functioning for some alerts. This was caused by the class responsible for decoding the data payload into ASCII. Sometimes, the data payload was unable to be decoded, and therefore raised a run-time error. The author fixed the problems by making a control whether the data payload can be decoded or not before displaying the data payload in the text box.
- The previous image displayed in the overview was not cleared up when the user started to visualise using the analysis panel or the filter panel. This is a minor problem and has been fixed.

The usability problems found were:

- Some users felt difficulty in controlling the 3D image movement using the VTK mouse control built-in features. This is due to lack

[9] Bug is a fault in a computer program that prevents it from behaving as intended.

of training. However, after several trials, they managed to control it correctly. One participant suggested a navigation control as in the 'Google map' application could be used. The author suggests this feature in the future work.

- The participants felt the overall GUI was fine except for the GUI in the analysis panel. They suggested a new organisation of the GUI to make it less crowded.
- Another suggestion from the participants was to improve the interaction by displaying the mouse-over information when the user brings the mouse cursor onto an object. The basic information such as IP address, signature ID and time should be made available. The author agreed with this suggestion and the author will try to implement it in future work.

In the user evaluation, apart from finding the usability problems and bugs, the participants were also asked to analyse a day of alert data which contained 1539 alerts. From the data, the participants were asked to list three local hosts that were under attack and to identify the one that was severely attacked using the time series 3D AlertGraph. All the participants successfully identified the local hosts, and the one that was severely attacked almost immediately. They agreed that the time series 3D AlertGraph was easy to understand, and easy to identify the potential true attacks in the network. The average rating given was 4.67 from the Likert scale, of 1 for very difficult, and 5 for very easy. The standard deviation was 0.58.

On the other views, the author also asked them to give their opinion whether the views were intuitive and made sense or not. The timeline and the plane views looked difficult to understand, with averages of 3.67 and 4.0 respectively. The standard deviation was 1.53 for timeline view, and 1.73 for plane view. In other views, the average rating was 4.67, with a standard deviation of 0.58. These showed the visuals were intuitive and comprehensible.

Lastly, the author asked the participants to evaluate the software on the following themes:

- Graphical User Interface design
- Image interaction
- False alert classifier feature
- Filters features
- Real-time feature
- Reporting tools
- Ability to perform network security tasks
- General comparison with other or similar tools that they have seen.

The participants were asked to give the rating, using a Likert Scale 1 to 5, with 1 for the most dissatisfied, and 5 for the most satisfied.

For the themes number C1 to C8 in Table 6-1, each average score was above 4.0, with a standard deviation less than or equal to 0.58. Table 6-1 showed the average scores of the Likert Scale in each category. These were positive results, and showed that this prototype had been successfully designed and addressed the needs of the user.

Table 6-1: Analysis of user evaluation.

no		**Average**	**Std dev**
A1	Overview	4.67	0.58
B1	Scatter plot	4.67	0.58
B2	Parallel plot	4.67	0.58
B3	Timeline view	3.67	1.53
B4	Plane view	4.00	1.73
B5	World globe	4.67	0.58
B6	World plane	4.67	0.58
C1	GUI - user friendly	4.33	0.58
C2	Interaction	4.67	0.58
C3	Classifier features	4.67	0.58
C4	Filter features	4.67	0.58
C5	Real-time	4.00	0.00
C6	Reporting features	4.67	0.58
C7	Comparison with similar tool	na	na
C8	Perform Security tasks	4.67	0.58

6.5 Conclusion

Conducting user evaluation is not the end process in software development. It should be a continuous process between the developers and the users. With these, the prototype software can be refined and meet the expectation of the industry. The author has fixed all the bugs found during the usability test, and the author has also implemented some of the suggestions by the users. However, the author reserved some suggestions for future work, due to the constraint of time.

Chapter 7: Conclusions and Future Work

7.1 Conclusions

The difficulty in exploring and analysing large quantities of network security alerts in text form has inspired many researchers to use visual methods as an alternative. The author has discussed some of their methods and applications in chapter 2. This application, NSAViz used visual methods to explore and give an insight from large quantities of network security alerts data. This application differed from other applications, and used three new components and techniques which are:-

1. A novel time series 3D AlertGraph to overview the network security status,
2. Multiple 3D visualisations in the scatter plot, parallel plot and timeline view for detail analysis, and,
3. Integrating a false alerts classifier to help the user to identify the true attacks to the network.

This application was designed with graphical user interfaces and user interaction features to help the analyses. The author has also addressed the main issues in security visualisation, and answered the research questions which are:

1. To visualise network security information in a way that gives understanding of the data. From the usability study, it was shown the visual design was intuitive and easy to understand.
2. To inform effectively the security status of the supervised network. The author has designed a novel 3D AlertGraph that gives a temporal outline of the attacks to the network. From the visual image, the user can understand and identify the potential true attacks to the network.
3. To support large volumes of network security data and internet nodes. The 3D AlertGraph was designed to receive huge data, but for other views, filtering, alert grouping and smaller monitoring periods were allowed to reduce the quantity of information to be displayed.

4. To recognise the patterns of true attacks. By using a colour scheme and a false alert classifier, the possible true attacks to the network were able to be highlighted.
5. To allow interactivity with the visualisation tool. The interaction using the graphical user interface and the visual objects allowed detailed information at a point of interest such as the source IP, destination IP and the details of the alert to be seen.
6. To provide reporting tools in the visualisation application. The author has provided printing and saving features, such as the ability to print the alerts listing and alert signatures, and to save the images in JPEG and BMP formats.

7.2 Research Contribution

This thesis has described solutions for the problems outlined in the statement of the problem (see section 1.4) using visualisation methods. The author used the visualisation to represent network security attack data for effective exploration, monitoring and analysis. This thesis has provided the following main contributions in the area of security visualisation:

- Designing novel 3D visuals of network security alerts for better viewing and understanding of network security alerts. The design of a 3D AlertGraph that gives a temporal outline of the number of alerts according to destination IP, source IP and destination port. This 3D AlertGraph is an extension of a 2D histograph plot into 3D. The 3D scatter plot was designed for detail alert analysis. Other views include the 3D timeline view and parallel plot view.
- Using a classification tree based on the C4.5 classification algorithm with visualisation to help users to identify the true alerts. This classification algorithm was integrated in each view, to provide a better visual pattern or highlight of true attacks.
- Using animation, real-time monitoring, interaction, user-friendly graphical user interface, drill-down and filtering with the visualisation, to get a better insight of the network security alerts

data.

7.3 Future Work

From the usability study that was conducted, positive feedback was received from the users. The author plans in the future to incorporate the suggested improvements. The first suggestion was to allow pop-up information, such as IP address and alert signatures, to appear above the object when a mouse cursor passes over the visual object. The second suggested improvement was related to zoom-in, zoom-out and rotation of the 3D images using the mouse. The suggestion was to apply a navigation panel such as that in 'Google map'. At this moment, such controls use the three key mouse buttons, and it seemed difficult for the new user to control it.

The author also plans to allow users to save some points of interest from the 3D AlertGraph, and then to use the saved information to be displayed in other views. Lastly, a greater scope of research can be conducted by upgrading this tool to be an Intrusion Detection System based on the visualisation method. The author mentioned previously that Snort missed some of the attacks from the DARPA 1999 dataset. With visualisation and integration of artificial intelligent techniques, it might be possible to resolve the problem of missed detection like in Snort. Currently, there is on going research on this subject, and there is still room for improvement.

Bibliography

[1] C. Brenton and C. Hunt, Mastering Network Security: Sybex, 2002.

[2] S. K. Card, J. D. Mackinlay, and B. Sheiderman, Readings in Information Visualisation: Using Vision to Think. San Francisco, CA: Morgan Kaufmann, 1999.

[3] T. Escamilla, Intrusion Detection: Network Security Beyond the Firewall. New York: John Wiley, 1998.

[4] P. R. Keller and M. M. Keller, Visual Cues: Practical Data Visualization: IEEE Computer Society Press, 1993.

[5] E. Maiwald, Fundamentals of Network Security. U.S.A: Mc Graw-Hill Technology Education, 2004.

[6] M. A. Maloof, Machine Learning and Data Mining for Computer Security: Springer, 2006.

[7] S. Russel and P. Norvig, Artificial Intelligent: A modern Approach. USA: Prentice Hall, 1995.

[8] W. Schroeder, K. Martin, and B. Lorensen, The Visualization Toolkit An Object-Oriented Approach To 3D Graphics. USA: Kitware, Inc., 2004.

[9] I. Sommerville, Software Engineering. USA: Addison-Wesley, 1995.

[10] C. Ware, Information Visualization: Perception for Design: Morgan Kaufman., 2004.

[11] I. H. Witten and E. Frank, Data Mining: Morgan Kaufmann Publisher, 2000.

[12] J. Zelle, Python Programming: An Introduction to Computer Science. USA: Franklin, Beedle & Associates, 2004.

References

[1] CompTIA, "Internet Security Threats Increasing in Maliciousness and Criminal Intent, CompTIA IT Study Reveals" Business Wire 2005 Accessed Date: 30/08/2008
Available: http://findarticles.com/p/articles/mi_m0EIN/is_2005_June_14/ai_n13812162

[2] BBC, "Net security threats growing fast" BBC News - Technology 2004 Accessed Date: 30/09/2004
Available: http://news.bbc.co.uk/1/hi/technology/3666978.stm

[3] K. Lakkaraju, W. Yurcik, and A. J. Lee, "NVisionIP: netflow visualizations of system state for security situational awareness," *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security,* vol. Washington DC, USA, pp. 65-72, 2004.

[4] H. Koike and K. Ohno, "SnortView: visualization system of snort logs," *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security,* vol. Washington DC, USA, pp. 143-147, 2004.

[5] J. B. Colombe and G. Stephens, "Statistical profiling and visualization for detection of malicious insider attacks on computer networks," *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security,* vol. Washington DC, USA, pp. 138-142, 2004.

[6] S. K. Card, J. D. Mackinlay, and B. Sheiderman, *Readings in Information Visualisation: Using Vision to Think.* San Francisco, CA: Morgan Kaufmann, p. 7, 1999.

[7] K. Abdullah, C. P. Lee, G. Conti, J. A. Copeland, and J. Stasko, "IDS rainStorm: visualizing IDS alarms," *VizSEC 05 IEEE Workshop on Visualization for Computer Security* vol. Minneapolis, Minnesota, USA, pp. 1-10, 2005.

[8] K. Nyarko, T. Capers, C. Scott, and K. Ladeji-Osias, "Network intrusion visualization with NIVA, an intrusion detection visual analyzer with haptic integration," *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems* pp. 277-284, 2002.

[9] R. F. Erbacher and D. Frincke, "Visualization in Detection of Intrusions and Misuse in Large Scale Networks," *IV '00: Proceedings of the International Conference on Information Visualisation,* pp. 294-299, 2000.

[10] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju, "VisFlowConnect: netflow visualizations of link relationships for security situational awareness," *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security,* vol. Washington DC, USA, pp. 26-34, 2004.

[11] W. C. Graham, "Buckles: A Configurable Information Visualization System." 2007 Accessed Date: 21/01/2008
Available: http://www.freshvista.com/WhitePaper.pdf

[12] R. Ball, G. A. Fink, and C. North, "Home-centric visualization of network traffic for security administration," *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security,* vol. Washington DC, USA, pp. 55-64, 2004.

[13] C. Ware, *Information Visualization: Perception for Design*: Morgan Kaufman., pp. 1-23,116-117, 2004.

[14] R. F. Erbacher and D. Frincke, "Visual Behavior Characterization for Intrusion and Misuse Detection," *VIII Proceedings of the SPIE '2001 Conference on Visual Data Exploration and Analysis,* pp. 210-218, 2001.

[15] Snort, "Snort - the de facto standard for intrusion detection/prevention" Accessed Date: 20/11/2007
Available: http://www.snort.org/

[16] S. Panjwani, S. Tan, K. M. Jarrin, and M. Cukier, "An experimental evaluation to determine if port scans are precursors to an attack," *International Conference on Dependable Systems and Networks, 2005. DSN 2005. Proceedings,* pp. 602-611, 2005.

[17] S. Ansari, S. G. Rajeev, and H. S. Chandrashekar, "Packet sniffing: a brief introduction," *Potentials, IEEE,* vol. 21, pp. 17-19, 2002.

[18] A. Piskozub, "Denial of service and distributed denial of service attacks," *Proceedings of the International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science,* pp. 303-304, 2002.

[19] B. Harris and R. Hunt, "TCP/IP security threats and attack methods," *Computer Communications,* vol. 22, pp. 885-897, 1999.

[20] C. Brenton and C. Hunt, *Mastering Network Security*: Sybex, pp. 435-435, 2002.

[21] T. Javvin, "RIP Routing Attacks" 2008 Accessed Date: 20/01/2008
Available: http://www.javvin.com/networksecurity/RIPRoutingAttacks.html

[22] E. Maiwald, *Fundamentals of Network Security.* U.S.A: Mc Graw-Hill Technology Education, pp. 67-70, 2004.

[23] T. Bradley, "Introduction to Intrusion Detection Systems (IDS)" Accessed Date: 30/08/2008
Available: http://netsecurity.about.com/cs/hackertools/a/aa030504.htm

[24] J. B. Kennedy, K. J. Mitchell, and P. J. Barclay, "A framework for information visualisation," *SIGMOD Rec.,* vol. 25, pp. 30-34, 1996.

[25] P. R. Keller and M. M. Keller, *Visual Cues: Practical Data Visualization*: IEEE Computer Society Press, p. 6, 20-21, 1993.

[26] B. Wunsche, "A survey, classification and analysis of perceptual concepts and their application for the effective visualisation of complex information," *APVis '04: Proceedings of the 2004 Australasian symposium on Information Visualisation,* vol. Christchurch, New Zealand, pp. 17-24, 2004.

[27] Wireshark-Ethereal, Accessed Date: 30/08/2008
Available: http://www.wireshark.org/

[28] CiscoSysytem, "Netflow" Accessed Date: 30/08/2008
Available:
http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html

[29] Lancope, "StealthWatch" Accessed Date: 30/08/2008
Available: http://www.lancope.com/products/stealthwatch-management-console/

[30] S. Lau, "The Spinning Cube of Potential Doom," *Communications of the ACM,* vol. 47, pp. 25-26, 2004.

[31] Bro, "Bro Intrusion Detection System" 2005 Accessed Date: 25 Oct. 2005
Available: http://bro-ids.org/manuals.html

[32] J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen, "PortVis: a tool for port-based detection of security events," *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security,* vol. Washington DC, USA, pp. 73-81, 2004.

[33] P. Ren, G. Yan, L. Zhichun, C. Yan, and W. Benjamin, "IDGraphs: Intrusion Detection and Analysis Using Histographs," *Proceedings of the*

IEEE Workshops on Visualization for Computer Security, pp. 39-46, 2005.

[34] F. J. Dean and T. S. John, "The Information Mural: A Technique for Displaying and Navigating Large Information Spaces," vol. 4, pp. 257-271, 1998.

[35] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen, "Real-time and forensic network data analysis using animated and coordinated visualization," pp. 42-49, 2005.

[36] G. Conti, K. Abdullah, J. Grizzard, J. Stasko, J. A. Copeland, M. Ahamad, H. L. Owen, and C. Lee, "Countering security information overload through alert and packet visualization," *Computer Graphics and Applications, IEEE,* vol. 26, pp. 60-70, 2006.

[37] D. A. Keim, J. Schneidewind, and M. Sips, "CircleView: a new approach for visualizing time-related multidimensional data sets," *AVI '04: Proceedings of the working conference on Advanced visual interfaces,* vol. Gallipoli, Italy, pp. 179-182, 2004.

[38] Y. Livnat, J. Agutter, S. Moon, R. F. Erbacher, and S. Foresti, "A Visualisation Paradigm for Network Intrusion Detection," *Proceeding of the 2002 IEEE Workshop on Information Assurance and Security,* vol. United States Military Academy, West Point, NY, pp. 92-99, 2002.

[39] K. Abdullah, C. Lee, G. Conti, and J. A. Copeland, "Visualizing network data for intrusion detection," *Information Assurance Workshop, 2005. IAW '05. Proceedings from the Sixth Annual IEEE SMC,* pp. 100-108, 2005.

[40] M. S. De-Silva, D. J. Parish, P. Sandford, and J. M. Sandford, "Automated Detection of Emerging Network Security Threats," *ICN '07 Sixth International Conference on Networking,* pp. 98-105, 2007.

[41] J. Chartier, "SnortALog : Snort Analyser Logs" 2005 Accessed Date: August 2005

Available: http://jeremy.chartier.free.fr/snortalog/index.html#sample

[42] Microsoft, "Log Parser" Accessed Date: 20/11/2007
Available: http://www.logparser.com/

[43] R. Danyliw, "The Analysis Console for Intrusion Databases (ACID)" Accessed Date: 20/11/2007
Available: http://www.cert.org/kb/acid/

[44] S. Axelsson, "Combining a bayesian classifier with visualisation: understanding the IDS," *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security,* vol. Washington DC, USA, pp. 99-108, 2004.

[45] G. Vert, D. A. Frincke, and J. C. McConnell, "A Visual Mathematical Model for Intrusion Detection," *Proc. 21st {NIST}-{NCSC} National Information Systems Security Conference,* p. 337, 1998

[46] W. Yurcik, J. Barlow, K. Lakkaraju, and M. Haberman, "Two Visual Computer Network Security Monitoring Tools Incorporating Operator Interface Requirements" Proc. of the Workshop on Human-Computer Interaction and Security Systems (HCISEC 2003) 2003 Accessed Date: 30/08/2008
Available: http://www.andrewpatrick.ca/CHI2003/HCISEC/hcisec-workshop-yurcik-2.pdf

[47] L. D. Paulson, "Researchers develop network-security visualization tools," *Computer,* vol. 37, pp. 17-18, 2004.

[48] A. Komlodi, J. R. Goodall, and W. G. Lutters, "An Information Visualization Framework for Intrusion Detection," *CHI '04: CHI '04 extended abstracts on Human factors in computing systems,* vol. Vienna, Austria, p. 1743, 2004.

[49] A. Dapos Amico and M. Kocka, "Information assurance visualizations for

specific stages of situational awareness and intended uses: lessons learned," *Visualization for Computer Security, 2005. (VizSEC 05). IEEE Workshop on,* vol. Minneapolis, Minnesota, USA, pp. 107-112, 2005.

[50] J. R. Goodall, "User Requirements and Design of a Visualization for Intrusion Detection Analysis," *Information Assurance and Security (IAW),* pp. 394-401, 2005.

[51] R. F. Erbacher, K. Christensen, and A. Sundberg, "Designing Visualization Capabilities for IDS Challenges," *Proceedings of the IEEE Workshops on Visualization for Computer Security,* pp. 121-127, 2005.

[52] P. Ren, "Ensuring the continuing success of vizsec," *Proceedings of the 3rd international workshop on Visualization for computer security,* pp. 67-70, 2006.

[53] MySQL, Accessed Date: 30/08/2008
Available: http://www.mysql.com/

[54] Python, Accessed Date: 30/08/2008
Available: http://www.python.org/

[55] VTK, Accessed Date: 30/08/2008
Available: http://www.vtk.org/

[56] L. L. C. MaxMind, "MaxMind, GeoIP" 2006 Accessed Date: 26 May 2006
Available: http://www.maxmind.com/

[57] B. Shneiderman, "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations.," *In Proceedings of the IEEE Symposium on Visual Languages,* pp. 336-343, 1996.

[58] J. Nielsen and R. Molich, "Heuristic evaluation of user interfaces," *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people,* pp. 249-256, 1990.

[59] J. S. Edward and C. W. Douglas, "Effective user interfaces: some common sense guidelines," *Proceedings of the 5th annual international conference on Systems documentation,* pp. 87-94, 1986.

[60] G. Tedesco and U. Aickelin, "Data Reduction in Intrusion Alert Correlation," *WSEAS Transactions on Computers,* pp. 186-193, 2006.

[61] K. Abdullah, "Scaling and Visualizing Network Data to Facilitate in Intrusion Detection Tasks," *School of Electrical and Computer Engineering,* vol. Phd, p. 59, 7 Apr. 2006.

[62] K. C. Larson, M., "Web page design: implications of memory, structure and scent for information retrieval.," *Proc. ACM Conf. on (CHI) Human Factors in Computing Systems,* pp. 25-32, 18-23 April 1998.

[63] X. Yin, W. Yurcik, and A. Slagell, "The Design of VisFlowConnect-IP: A Link Analysis System for IP Security Situational Awareness," *IWIA '05: Proceedings of the Third IEEE International Workshop on Information Assurance,* pp. 141-153, 2005.

[64] W. Lee and X. Qin, "Statistical Causality Analysis of INFOSEC Alert Data," *Lecture Notes in Computer Science - RAID2003,* vol. Volume 2820/2003, pp. 73-93, 2003.

[65] J. Viinikka, H. Debar, L. Me, and R. Seguier, "Time series modeling for IDS alert management," *ASIACCS '06: Proceedings of the 2006 ACM Symposium on Information, computer and communications security,* vol. Taipei, Taiwan, pp. 102-113, 2006.

[66] T. Pietraszek, "Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection," *Lecture Notes in Computer Science - RAID2004,* vol. Volume 3224/2004, pp. 102-124, 2004.

[67] E. E. Bloedorn, L. M. Talbol, and D. D. DeBarr, "Data Mining Applied to Intrusion Detection: MITRE Experiences," *Machine Learning and Data*

Mining for Computer Security, pp. 65-87, 2006.

[68] W. W. Cohen, "Fast Effective Rule Induction," *Proc. of the 12th International Conference on Machine Learning,* pp. 115-123, 1995.

[69] P. Sandford and D. J. Parish, "Identifying Abuse in ISP Networks" Oxford Internet Conference: Safety & Security in a Networked World 2005 Accessed Date: 30/08/2008
Available:
http://www.oii.ox.ac.uk/microsites/cybersafety/extensions/pdfs/papers/peter_sandford.pdf

[70] J. Demsar, B. Zupan, and G. Leban, "Orange: From Experimental Machine Learning to Interactive Data Mining" White Paper (www.ailab.si/orange), Faculty of Computer and Information Science, University of Ljubljana 2004 Accessed Date: 30/08/2008
Available: http://www.ailab.si/orange

[71] J. R. Quinlan, *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[72] H. Ding and X.-K. Wang, "Research on algorithm of decision tree induction," *Proceedings of International Conference on Machine Learning and Cybernetics,* pp. 1062-1065 4-5 Nov. 2002.

[73] R. Kohavi, "A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection.," *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI),* pp. 1137–1143, 1995.

[74] L. L. MIT, "DARPA Intrusion detection evaluation dataset" 1999 Accessed Date: 20/11/2007
Available: http://www.ll.mit.edu/IST/ideval/data/data_index.html

[75] G. Ellis and A. Dix, "An explorative analysis of user evaluation studies in

information visualisation," *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization,* pp. 1-7, 2006.

[76] C. Plaisant, "The challenge of information visualization evaluation," *Proceedings of the working conference on Advanced visual interfaces,* pp. 109-116, 2004.

[77] C. Plaisant, J. Grosjean, and B. B. Bederson, "SpaceTree: Supporting Exploration in Large Node Link Tree, Design Evolution and Empirical Evaluation," *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02),* pp. 57-64, 2002.

[78] J. Nielsen, "Heuristic Evaluation," *Usability Inspection Methods,* pp. 25-64, 1994.

[79] G. Conti, "Countering Network Level Denial of Information Attacks using Information Visualisation," *College of Computing,* vol. Phd, p. 183, 27 March 2006.

[80] A. Komlodi, P. Rheingans, A. Utkarsha, J. R. Goodall, and J. Amit, "A user-centered look at glyph-based security visualization," *VizSEC 05 IEEE Workshop on Visualization for Computer Security,* vol. Minneapolis, Minnesota, USA, pp. 21-28, 2005.

[81] R. Spence, *Information Visualization: Design for Interaction* Harlow: Pearson/Prentice Hall, p. 5, 2007.