

Multi-dimensional Visualization for the Analysis of Internet Traffic and the Identification of Intrusive Activity

Submitted in partial fulfilment
of the requirements of the degree
Bachelor of Science (Honours),
in the Department of
Computer Science,
Rhodes University

Jean-Pierre van Riel
10 November 2005

Supervised by:
Barry Irwin

Abstract

The research presented by this paper is an exploration of network security visualizations. It is argued that visual representations of data allow us to pick up anomalous and suspicious traffic patterns. They also afford an intuitive holistic understanding of data; an understanding of data that would otherwise be lost (or hard to come by) if data were reviewed in a textual format.

As an integral part of this research, graphical methods for representing network events are evaluated, and the ability to identify intrusive traffic patterns is a primary criterion for this evaluation. A literature survey of several network visualizations aims to provide a perspective on the current ‘state of the art’ in the field. From this survey, scalability is identified as a concern. Points are argued for as a simple and scalable visual metaphor to represent events (instead of the conventional line metaphors employed in many visualizations).

Practical research of this involves a 3-D animated and interactive scatter-plot visualization, dubbed InetVis (Internet Visualization), which is developed to confirm and extend the original concept based on Stephen Lau’s ‘Spinning Cube of Potential Doom’. Isolated network scanning activity confirms the concept by providing anticipated visual signatures. The use and value of several extensions is demonstrated in the analysis of Internet traffic. Specifically, traffic captured from a class C darknet (network telescope) over August, September and October 2005 is reviewed. In this review, extensive scanning activity is visually observed, and intriguing anomalous traffic patterns are discovered and documented.

In conclusion, the 3-D scatter-plot plotting scheme is proven to convey intrusive activity in a salient, intelligible, and scalable manner. Finally, the merits of several future extensions are described.

Contents

Introduction	5
1.1 Central Thesis Outline	6
1.2 Research Methodology	8
Background	10
2.1 The Internet as a Security Threat	10
2.1.1 Growth in Size, Connectivity and Utilization	11
2.1.2 Escalating Creation of Malicious Code	11
2.1.3 A Growth in Security Vulnerabilities	13
2.1.4 Lack of Accountability	14
2.1.5 Blacklisting as a Measure of Accountability	14
2.1.6 Lack of Security Awareness.....	15
2.2 Characteristics of Intrusive Activity.....	15
2.2.1 Automated Code versus Human Intrusion Attempts.....	16
2.2.1.1 Network Security Tools	16
2.2.2 Classification of Malicious Code and Activity	16
2.2.2.1 Worms, Viruses, Trojan-horses, Spyware, and Backdoors.....	17
2.2.2.2 Bot Networks and Denial of Service	17
2.2.3 Classification of Intrusive Network Probing Techniques.....	18
2.2.3.1 Network Scans.....	18
2.2.3.2 Port Scans.....	18
2.2.3.3 Stealth Scans, and Distributed, Coordinated Scans	18
2.3 Conventional Methods for Network Monitoring	19
2.3.1.1 Packet Capture.....	19
2.3.1.2 Manual Review with Textual Tools.....	20
2.3.1.3 Network Intrusion Detection Systems	20
2.3.1.4 Honeynets and Darknets (Network Telescopes)	21
2.4 The Advantages of Visual Methods	22
2.4.1 Parallel and Pre-attentive Visual Cognition.....	22
2.4.2 Multi-Dimensional Visualization	23
2.5 Chapter Summary and Conclusion.....	23
Related Work.....	25
3.1 Selection Criteria for Reviewed Network Visualizations.....	25
3.2 Network Security Visualizations.....	26
3.2.1 ‘The Spinning Cube of Potential Doom’	26
3.2.1.1 Concept and Features	27
3.2.1.2 Observations of Intrusive Activity.....	29
3.2.1.3 Scalability and Performance.....	29
3.2.2 VISUAL – Ball and Fink et al, “Home-centric visualization of network traffic for security administration”	30
3.2.3 Scanmap3D by Daniel Clark.....	34
3.2.4 The ‘Space Shield’ – Fisk et al, “Immersive Network Monitoring” 39	
3.2.5 VisFlowConnect – Yin et al, “NetFlow Visualizations of Link Relationships for Security Situational Awareness”	43

3.2.6	OASC Visualizaions – Teoh at al, “Detecting Flaws and Intruders with Visual Data Analysis”	47
3.3	General Review and Comparative Analysis	50
3.3.1	Points versus Lines as a Visual Metaphor for Representing Connections	50
3.3.2	Use of transparency and colour	52
3.3.3	Animation and Real-Time playback.....	52
3.3.4	Filtering.....	53
3.3.5	Drill downs and details on demand	53
3.4	Chapter Summary and Conclusion.....	54
Design and Implementation		56
4.1	Conceptual Design and Feature Specification	56
4.1.1	The Perspective of an Internal Domain versus the Internet.....	57
4.1.2	Points as a Visual Metaphor for Network Events	58
4.1.3	3-D Plotting Scheme.....	58
4.1.4	Reference Frame.....	61
4.1.5	Colour Schemes.....	61
4.1.6	Variable Point Size	62
4.1.7	Animation and Time Scaling.....	62
4.1.8	Time Window.....	63
4.1.9	Filtering Techniques	63
4.1.10	Navigation and Exploration	64
4.2	System Design and Implementation.....	64
4.2.1	Component Model	65
4.2.2	Development Platform and Software Libraries.....	67
4.2.3	User Interface	68
4.2.3.1	Control Panel.....	68
4.2.3.2	Visualization Pane and Navigation.....	68
4.2.4	Data Processing, Extraction and Representation.....	69
4.2.4.1	Data Extraction – The LibPCap Library	69
4.2.4.2	Packet Representation	70
4.2.4.3	Raw Packet Data Parsing	70
4.2.4.4	Packet Event Buffering	71
4.2.4.5	Event Buffering and Animation	72
4.2.5	Rendering and Timing	72
4.2.5.1	Animation Timing.....	72
4.2.5.2	Graphical Rendering	72
4.3	Chapter Summary and Conclusion.....	73
Results and Analysis		75
5.1	Visual Signatures of Network Scanning.....	75
5.1.1	A Method for Capturing Network Scans in Isolation.....	75
5.1.2	Horizontal Scan Signature – Network Scanning.....	76
5.1.3	Vertical Scan Signature – Port Scanning	78
5.1.4	Block Scan Signature.....	79
5.1.5	Grid Scan Signature with Decoys.....	80
5.2	Darknet Traffic Analysis	82
5.2.1	August 2005	82
5.2.1.1	Bands of activity	82
5.2.1.2	Anomalous Diagonals.....	83
5.2.1.3	The Step Scan	85

5.2.1.4	The ‘Creepy Crawly’ Scan.....	85
5.2.2	September 2005.....	86
5.2.2	September 2005.....	87
5.2.2.1	Diffuse diagonals.....	87
5.3	Performance and Scalability.....	88
5.4	Summary and Conclusion.....	88
	Conclusion.....	90
	References.....	91

Chapter 1

Introduction

Intrusive and malicious activity transpires on computer networks and globally propagates through the Internet. Such activity constitutes a network security threat. It arises as the result of malicious software (i.e. worms) or human attackers (informally referred to as ‘hackers’) probing for vulnerable systems with security flaws. Understanding the vast and varied extents of this intrusive activity is a difficult challenge for security professionals, let alone the ordinary Internet user.

Traditional methods for network monitoring entail the review of network events logged in a textual format. Logs may report rudimentary packet capture data, connection attempt information or more processed information in the form of security alerts (as logged by a network intrusion detection system – abbreviated as NIDS). To understand these logs usually requires a high level of technical understanding and tedious line-by-line inspection. The reviewer may have to correlate and keep in mind numerous network events in order to grasp links and patterns, a practise that becomes intractable for large volumes of data when presented in a textual format. Furthermore, a wealth of literature [Fisk. 2003, Putton 2001, Teoh 2004, Yin 2000] suggests that both intrusion detection systems and algorithmic data mining methods in general have shortcomings (as will be elaborated upon in Chapter2, Section 2.3.1.3). Therefore, an objective is to improve upon current methods by finding efficient and effective ways to monitor and analyse network traffic.

As a prospective solution, visualizations can produce graphical representations of network events that convey anomalous traffic patterns in aid of identifying intrusive probing activity (i.e. network scans). In particular, animated multi-dimensional (n-D) visualizations can exhibit and allow for the correlation of more information than traditional static 2-D graphs; and therefore, utilize the capability of computer graphics to a fuller extent.

This work presents an exploration of graphical methods employed in visualizations of network traffic. The primary objective is to evaluate their application in intrusion detection. In particular, Stephen Lau’s visualization, ‘The Spinning Cube of Potential Doom’, forms the primary initiative behind this

research [Lau 2003, Lau 2004]. Stephen Lau's work, amongst contributions by others [Ball 2004, Fink 2004, Fisk 2003, Scanmap3D, Teoh 2004, Yin 2004], constitutes visual network security research – an emergent field within the broader class of information visualization and information security. Recently, dedicated conferences, such as VizSEC¹, have provided a forum to exchange ideas about network security visualizations.

The ambition of the research presented here is to elucidate the promise of visual network security analysis, argue for its merits, address its challenges, and demonstrate its effective use. Continuing this introduction, section 1.1 outlines the central thesis that this work aims to establish. Section 1.2 follows, providing an overview of the investigative steps taken, whilst outlining the structure and purpose of the remaining chapters.

1.1 Central Thesis Outline

The thesis put forward is as follows:

1. Two similar, but distinct, methods for detecting intrusive network activity involve:
 - a. Positively identifying suspicious traffic patterns known to arise as the result of intrusive activity (e.g. signature traffic patterns caused by worm activity).
 - b. Discovering anomalous traffic patterns that may result from unknown novel intrusive techniques (presumably devised to be undetectable by conventional intrusion detection systems).
2. Multidimensional visual analysis of network events can offer marked advantages over traditional methods, for the following reasons:
 - a. They allow the viewer to correlate multiple facets of traffic capture data at a single glance, and enhance the ability to display complex patterns.
 - b. The parallel and pre-attentive modality of visual cognition is superior to the serial manner in which textual data is interpreted, thereby:
 - i. Granting the viewer with the ability to grasp a holistic view of the data.
 - ii. Allowing the viewer to perceive unexpected patterns that algorithmic signature based methods may fail to anticipate and detect.

¹ A large portion of recent research can be found in the VizSec/DMSec-2004 proceedings, published in '2004 ACM workshop on visualization and Data Mining for Computer Security'

3. Provided the graphical representations employed are salient, intelligible and scalable, visual analysis of network events is a tenable solution allowing, in practise, the identification of intrusive activity via discovery of anomalous traffic patterns.
 - a. Where the term salient refers to the ability to convey selected facets of the information in a manner that the viewer would readily notice (i.e. anomalous patterns should be easy to see).
 - b. The term intelligible means that the viewer can easily interpret how the graphical representation relates back to attributes of the data from which it is constructed (i.e. the viewer easily understands what is being visualized).
 - c. The term scalable implies that visualization is capable of viewing both small and large amounts of data, whilst maintaining acceptable levels of performance and remaining interpretable (i.e. does not become unresponsive or obfuscated by larger data sets).
4. A 3-D scatter-plot visualization, developed to evaluate and extend the concepts behind Stephen Lau's Spinning Cube of Potential Doom, supports the methods in 1, has the advantages of 2, and satisfies 3 because:
 - a. It employs a simple, intelligible plotting scheme according to three spatial coordinates that allow the viewer to interpret the source address, the destination address and the ports used for network communication. It is intended for viewing traffic traversing between an internal subnet and the external Internet.
 - b. Offers further dimensionality by the use of animation which relates the forth dimension of time, and the use of colour to relate various other attributes of the traffic viewed, thereby correlating more information.
 - c. Has the ability to convey substantial amounts of data at once (scalability) – 800,000 network events at a given time – and can relay events in rapid review.
 - d. Provides salient depictions of network probing activity such as:
 - i. Network scanning of an address range in search of hosts, also referred to as 'horizontal' scanning.
 - ii. Port scanning a host for open ports, also referred to as 'vertical' scans.
 - iii. Anomalous patterns that are not conventional methods of scanning, but may be the result of slow 'stealth' scans, coordinated scans from multiple attacking hosts, or other peculiar traffic.

1.2 Research Methodology

In brief summary, the method of investigation entailed:

1. A literature review to assess visualization concepts and features
2. Subsequent to the review, the design and delivery of an animated 3-D scatter-plot visualization serves as a modified reimplementaion of the ‘Spinning Cube of Potential Doom’, and is named InetVis (Internet Visualization).
3. Develop InetVis with the intent to:
 - a. Extend and improve upon Lau’s work by adding valuable features.
 - b. Confirm the concept behind Lau’s work, and establish its viability by documenting visual signature patterns for known intrusive techniques.
 - c. Note which types of intrusion the visualization can reliably detect, and conversely, which types it falsely identifies, or fails to identify (and under what circumstances it fails).
 - d. Establish how much traffic data can be analysed at a given time.
 - e. Apply the visual analysis to darknet traffic capture.
4. Ultimately, as is intended, steps 1, 2, and 3 result in the practical demonstration of identifying and analysing intrusive activity with InetVis as an effective and scalable visualization of network events.

Following this introduction, the next chapter addresses the problem domain and background motivating this research. This includes considering the state of the Internet as a security threat, reviewing characteristic forms of intrusive activity, and classifying various types of malicious software. The advantages of visual cognition are compared to the limitations of conventional textual and automated approaches, providing support for visual network security research. (Should the reader consider himself, or herself, well versed about subjects in network security, such as malicious software, scanning techniques, intrusion detection systems, then he or she may omit Chapter 2, and read on with Chapter 3.

With particular focus on the merits of Lau’s work, Chapter 3 assesses and compares several network visualizations. The application of intrusion detection is the main criteria for evaluating visualizations and graphical methods that depict network events. The foremost concern identified is scalability, and the review surmises that the concept of the ‘Spinning Cube of Potential Doom’ has promising scalability. As part of this review process, the work of others eludes to several features that are useful extensions for the concept behind Lau’s work.

Chapter 4 describes the design of InetVis (Internet Visualization), intended for viewing the Internet as a threat to a subnet (an internal organization network range to be defended). This chapter elaborates upon the visualization's concept and features, with due reference to related work. Chapter4 also discusses some pertinent implementation issues, with special regard to performance and scalability.

Following Chapter5, offering a proof of concept in the form of documenting visual signatures by recording images of intrusive activity in isolation. The rest of the chapter is devoted to darknet traffic analysis, where suspicious activity is identified and further investigated; darknets are also known as network telescopes. In closing, Chapter 6 surmises and concludes the findings of this research project.

Chapter 2

Background

This chapter begins by outlining the broader problem domain of network security with sections that offer an overview of the Internet as a security threat and characterise intrusive network phenomena. Conventional textual network monitoring methods and network intrusion detection systems (NIDS) are briefly discussed. Following from this, motivation for visual network analysis is provided by contrasting the advantages of visual methods with the shortcomings of traditional methods (textual and algorithmic). The purpose of detecting and analysing intrusive network traffic patterns is the main criteria for this comparison.

2.1 The Internet as a Security Threat

There are several factors to consider when viewing the Internet as a security threat:

- The territory to monitor and defend against is vast considering the Internet's massive size, global connectivity, public accessibility, and growing usage.
- There is a lack of accountability for the unsolicited use of the Internet, although blacklists can provide some measure of worst offenders.
- Trends in malicious and intrusive activity are on the incline, with increasing rates of attack that systems must face.
- The growing number of uncovered security vulnerabilities further inflames the problem.
- Firewalls are a common perimeter defence strategy for networks, but are an inadequate measure in lieu of mobile communication technologies that traverse in and out of 'secure' network zones.
- User ignorance and negligence facilitates the propagation of malicious code.

As such, this section aims to bring across a brief overview of Internet security threat that organizations face. In passing, it also covers some common practices and misconceptions that result in inadequate measures of defence.

2.1.1 Growth in Size, Connectivity and Utilization

As a means apart from physical access to local networks, the Internet constitutes a remote security threat. To begin to understand the challenges at hand in monitoring an Internet connection, one must realise the vast extents of the Internet. With figures as of 2002, the Internet had over 500 million users, over 8 million unique website domains², and an excess of 500,000 terabytes of data [O'Neill 2003]. Latest estimates suggest that there are almost 1 billion Internet users (14.9% of the population)³. IPv4 (Internet Protocol version 4) caters for over 4 billion addresses (2^{32}). IPv6 will result in a global address space of over 2^{128} (3.4×10^{38} addresses). The figures presented above give an indication of the expansive territory that network security efforts need to monitor.

2.1.2 Escalating Creation of Malicious Code

Yengneswaran et al conducted a widespread study of intrusive activity in 2002, and presented in their findings in an article titled 'Internet Intrusions: Global Characteristics and Prevalence' [Yengneswaran 2003]. Over 1600 firewall administrators distributed throughout the world contributed to a collection of log data covering May 2002 to June 2002. From this data set, they observed highly

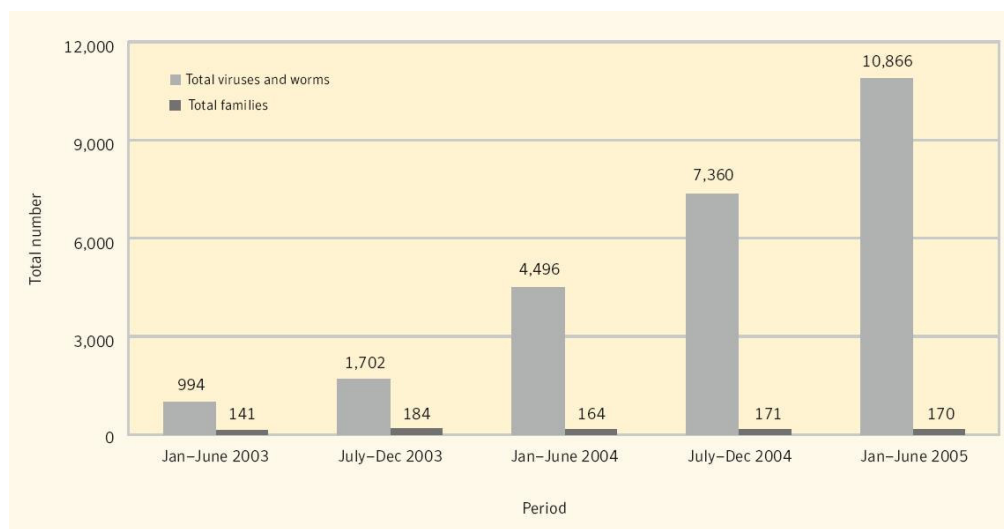


Figure 2-1: New Win32 viruses and worms by six-month period over 2003-2005 [Symantec 2005-Sep: 10]

² The number of unique website domains constitutes IP addresses responding to HTTP request where each IP can maintain many virtual domain names.

³ Statistics as of September 30, 2005 <<http://www.Internetworldstats.com/stats.htm>> (2005-11-03)

varying intrusive activity levels ranging from 1 million to 3 million network scans per day. They projected this dataset to the larger Internet, and inferred that a peak of 25 billion scans occurred on some days. They also noted an increasing trend in intrusive activity, as the projection of the average daily number of scans increased by 25% from 6.5 billion to 8.2 billion during their four-month observation period.

According to Symantec⁴, on average, median “organizations received 13.6 attacks per day”, between July 2004 and December 2004 [Symantec 2005-Mar: 11]. This increased from an average of “11 attacks per day” during the January-June 2004 period [Symantec 2004-Sep: p. 8]. Regrettably, the Symantec reports do not detail the method by which they attain these figures (to a satisfactory level of precision⁵), and it is possible that these figures fail to account for false positives. Added to this concern, the Symantec Internet Security threat Report suggests rapid growth in the development of new Win32 viruses, worms, and malicious scripts.

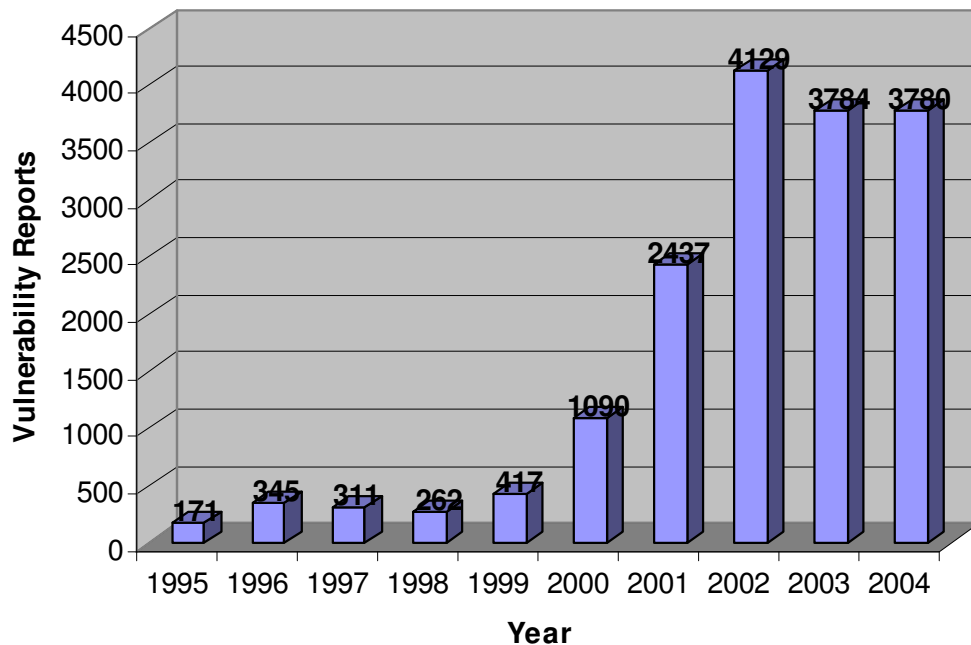


Figure 2-2: Vulnerability reports per year from 1995 to 2005 [CERT]

⁴ Symantec is a well established computer security company. They gather data via feedback from a range of widely deployed products and services, with “over 20,000 sensors monitoring network activity in 180 countries [and ...] gathers malicious code data from over 120 million client, server and gateway systems” [Symantec 2004-Sep: 6: 2]. Given that Symantec sells security products, there is reason to consider the accuracy of their claims, as they would profit from weary consumers.

⁵ In the latest report, Symantec readjusted their metric to include unauthorised access attempts blocked by firewalls or intrusion detection systems which further inflate these figures, rather than “being centred only on the most critical attacks” [Symantec 2005-Sep: 25]. They do not precisely define what they consider a ‘critical attack’ and ‘unauthorised access attempt’, nor do they mention exactly how the figures were derived.

The bar graph in Ball et al recon that in general VISUAL could display approximately 10,000 external hosts with their mapping approach [Ball 2004: 58] Figure 3-5 indicates an incline in the advent of malicious win32 code. For the Jan-June 2005 period, Symantec observed 170 distinct forms of malicious code (families) along with a total of 10,866 new virus and worms inclusive of variants (which are modified versions of prior forms).

2.1.3 A Growth in Security Vulnerabilities

Symantec warns that, “*vulnerabilities continue to increase in number and severity*” [Symantec 2004-Sep: 6]. Between 1995 and the first quarter of 2005, the CERT® Coordination Centre (CERT/CC)⁶ has received 17,946 vulnerability reports, with over 3,500 vulnerabilities reported each year for the last three years, namely 2002, 2003, and 2004 [CERT]. The bar graph in Figure 2-2 reflects the rising number of vulnerabilities discovered in recent years. One possible inference from the data that is that vulnerabilities are becoming more prevalent, but other factors such as improved security research may contribute to inflate the figures above. Furthermore, these statistics relate *reported* vulnerabilities, and are not a complete account of how many security flaws exist, nor an exact indication of how many exploits are available. Security flaws could remain unreported for an indeterminate amount of time, and it stands to reason that unpublished exploits for them some of them could exist indefinitely.

Without improved software engineering practices that heed security concerns, an increasing number of security flaws will ensue, and continue to add to the burden of maintaining secure systems. In an attempt to treat the symptoms of poor software development practises, a lack of effective vulnerability patching practises will fail to contend with an increasing number of security flaws, and ultimately, systems will be more susceptible to attacks.

⁶ Date source: <http://www.cert.org/stats/cert_stats.html> (accessed 26/05/2005). CERT is a well-established organization, with its original roots dating back to 1988, where the Defence Advanced Research Projects Agency (DARPA) formed a computer security, incident response team. SANS is another useful site for information regarding Internet threats and vulnerabilities. <<http://www.sans.org/>> (01/06/2005).

2.1.4 Lack of Accountability

The Internet is publicly accessible worldwide network that falls under the jurisdiction of different states as autonomous entities collectively administer its operation. Consequently, there is a lack of consolidated control over its users and data; and furthermore, there is no central authority to hold users accountable for its abuse. Added to this, “*sharing of information on intrusion activity is complicated by privacy issues*” [Yengneswaran 2003: 138]. A sensible postulate is that many organizations wish to keep their network information private, and many intrusion incidences remain undisclosed in lieu of damaging an organizations reputation. Therefore, consolidating sources of malicious activity can be a tenuous issue.

2.1.5 Blacklisting as a Measure of Accountability

Some organizations attempt to maintain blacklists of known offenders⁷. However, subscription to blacklisting policies is on a voluntary basis. For this reason, blacklists fail to enumerate all sources of malicious activity because blacklisting policies are not compulsory (nor universally adopted). Nonetheless, blacklists are a useful method for identifying and blocking ‘worst offenders’.

In practise, blacklisting seldom amounts to the full justice of convicting a perpetrator. A common reaction is to deny access or filter traffic to and from entities identified as hostile. Ironically, organizations involved in supporting blacklists can potentially face legal action for wrongful blacklisting. For example, a user may not be aware of self-propagating malicious code that has compromised a system they own, code that further exploits their system to perpetuate malicious activity. In this case, the user becomes the unwittingly subject to blacklisting, but this due to ignorance or negligence rather than intentional offence. Arguably, blacklists should discern between intentional and unintentional sources of intrusive activity. Of course, to make this distinction in practise may not be straightforward.

⁷For example, DShield.org allows voluntary contributors to submit firewall logs to form a distributed intrusion detection system. They maintain a list of top 10 offending IP addresses and a block list of IP ranges “*that have exhibited suspicious activity*”. <<http://www.dshield.org/>> (2005-11-03).

2.1.6 Lack of Security Awareness

Whilst security professionals may be well informed, many non-professionals (users and perhaps even some system administrators) are unaware of how prevalent Internet attacks are. A factor that contributes to the spread of worm infections and the like – forms of self-propagating malicious activity – is the lack of ‘security education’ (ignorance and negligence) among non-professionals. Despite the availability of patches (security fixes) that fix vulnerabilities exploited by worms, some users fail to apply updates and secure their systems. Evidence for this comes from observing that some intrusive scanning activity, characteristic of a worm or known exploit, lingers well after a patch is available [Yengneswaran: 143].

There are at least three pertinent points in this regard which explains why users are able to persist in ignorance:

5. Often malicious code runs in background processes that remain hidden, showing no overt signs of being present on a system.
6. Detecting and understanding intrusive forms of network activity requires extensive technical knowledge of the Internet’s infrastructure, its network protocols, and the complex networking architecture of computers. Hence, an ordinary user is ill equipped to understand the workings of network traffic, let alone its intrusive forms.
7. Added to this, firewalls are a ‘shielding’ strategy for organizational networks to protect users, but have also allowed them to persist in ignorance of how hostile an environment the ‘open’ Internet is; a problem if one considers the rapid advent and use of mobile communication technologies that can break the integrity of the perimeter defence strategies⁸.
8. Patch management practises are insufficient with technical difficulties in defending systems against an increasing number of vulnerabilities.

2.2 Characteristics of Intrusive Activity

Intrusive activity threatens the confidentiality and integrity of information as, presumably, it transpires with the purpose of stealing information, committing fraud, or causing disruption. Some types of intrusive activity may be considered benign (harmless pranks), but all forms are unsolicited per say, and result in the

⁸ This is a point made by Lau [Lau 2003].

misuse of computer resources - in particular, computer networks and the Internet. To begin with, managing this threat requires identifying surreptitious network activity - identification in terms of detecting its occurrence, and characterising the form of intrusive activity. If possible, further measures are identifying the source of it, and ultimately inferring the motives of its perpetrators.

2.2.1 Automated Code versus Human Intrusion Attempts

Commonly large volumes intrusive activity originates from malicious software ('malware'). Worms for example have the capability of global self-propagation through the Internet, and can result in outbreaks of 'infection'. Less commonly, human attackers ('hackers') are responsible for manually crafted intrusion attempts employing network-probing techniques to seek out vulnerable systems with security flaws. In some cases, legitimate use of security tools to perform 'penetration testing' for example, can give rise to apparent intrusive activity. Ethical 'Penetration testing' entails scanning a network in order to identify vulnerabilities with the intent of recognising weaknesses and improving defence. The use of various network tools can be ethical, or conversely, they may serve less honourable purposes.

2.2.1.1 Network Security Tools

Network probing tools (such as Nessus and NMap) provide a means to scout networks for vulnerable systems [Nessus, NMap]. Network exploit tools (such as Metasploit) test for the ability to perform manual intrusions [Metasploit]. Other tools (such as Ethereal or TCPDump) are suited to capturing and reviewing traffic off a network interface set in promiscuous mode [Ethereal, TCPDump]. In normal operation, if an interface detects traffic not addressed to it, it discards the traffic. In promiscuous mode, the interface allows the 'sniffing' of traffic that is intended for other recipients – this can be a breach of confidentiality and intercept laws. Some network 'hacking' tools incorporate a hybrid of these features.

2.2.2 Classification of Malicious Code and Activity

The intent for malicious software can be varied, but what categorises software as malicious is that its code performs unsolicited tasks whilst running on an 'infected' system. The term 'malware' refers to malicious software in general, and is

inclusive of other terms such as ‘worms’, ‘viruses’, ‘spyware’ and ‘adware’. These terms distinguish different forms of automated malicious activity. The attack vector (method of intrusion and distribution) as well as the intended purpose, serve as criteria to classify its various types of malicious software.

2.2.2.1 Worms, Viruses, Trojan-horses, Spyware, and Backdoors

Worms are essentially self-propagating code that use networks for distribution. They use infected hosts to launch attacks against other hosts and inject their own code into executable memory (commonly via an exploit such as an unchecked buffer – buffer overrun). Viruses refer to the more traditional forms of malicious software that infects (and corrupts) other pieces of software, such as executable application files or email attachments. Commonly, they spread via users copying and distributing infected files. A Trojan hoarse is harmful or unwanted code distributed and hidden within what appears to be legitimate software. Unlike viruses or worms, Trojan horses rely on deceiving users to install them rather than replicating themselves.

Whilst worms, viruses and Trojan horses are characterised by their attack vector (method of attack and propagation), spyware, and adware are characterised by their purpose. The intent of spyware is to collect information from a system and to report it back to an authorised system without consent. Spyware typically targets and breaks the confidentiality of sensitive information such as credit card numbers and passwords. Similar yet distinct from spyware, backdoors are intentional parts of programs the purpose of breaking access control to a system and enabling unauthorized remote access. A backdoor allows the compromised system to be manipulated or controlled remotely. In some cases, worms and Trojans introduce a backdoor to a system as their ‘payload’.

2.2.2.2 Bot Networks and Denial of Service

As a form of backdoor software, network bots are malicious programs that reside on infect host machines with the intent of building up a coordinated collection of machines to control and manipulate remotely. The controller of a bot network can remotely launch distributed and coordinated denial of service attacks, or intrusion attacks. A denial of service attack is characterised by the intent to disrupt network services. An example of a denial of service attack is to overwhelm a website server with an unmanageable number of connections, thereby disrupting its availability.

2.2.3 Classification of Intrusive Network Probing Techniques

To probe into a network and search for vulnerable systems, requires connecting to hosts and receiving responses. The process of performing this reconnaissance is termed network scanning. Various methods of scanning can be classified according to the manner in which they attempt to connect to systems. In terms of IP and TCP/UDP protocol specifications, network source addresses, destination addresses, source ports, and destination ports can be used to characterise scanning activity.

2.2.3.1 Network Scans

Scanning consecutive IP address in a network range constitutes a network scan. The common use is to distinguish addresses for responsive hosts from unassigned address space, thereby mapping the address space – sometimes referred to as host discovery. Protocols such as TCP, UDP and ICMP can be used to perform such scans. For example, a ICMP ping is intended to test if a host is reachable. To map out network address space, an ICMP ping scan would attempt to ping every IP address in the range, and await a response.

2.2.3.2 Port Scans

When a single host is targeted, TCP or UDP can be used to attempt to establish a connection to it. In order to establish the connection, an open destination port must be found. A port scan is constituted by attempting to connect to a sequence of ports to distinguish closed ports from open ports. Port scanning can be thorough, attempting consecutive ports in a port range. Alternatively, a port scan could just attempt a list of ports assigned to well known services; in this case, for each port scanned, the probability that is open is higher.

2.2.3.3 Stealth Scans, and Distributed, Coordinated Scans

A Network scan or port scan is obvious to detect if it is conducted in a short time by using the same host address. To evade detection by an NIDS, scans can either be conducted very slowly, or performed in unconventional ways. Scanning in slow random sequences may evade intrusion detection. Another example would be scanning across address space using different source ports; even if a port is closed,

the targeted host sends a response indicating it is closed, which is enough to infer its presence. Using multiple coordinated hosts can also complicate detecting scans, and provides an incentive for establishing bot networks to conduct probing activity.

2.3 Conventional Methods for Network Monitoring

For computer security professionals and network administrators, discovering and identifying intrusive activity is far from trivial:

- Firstly, intrusive activity is intended to go unnoticed.
- Secondly, it is obfuscated by the complexity of network protocols.
- Thirdly, it becomes further obscured when hidden in large volumes of traffic.

Intrusion detection requires monitoring network traffic for 'fingerprints' such as anomalous traffic patterns or characteristic packets that are indicative of viral activity, worms, network scanning tools, and the like. To pick up anomalous traffic patterns requires correlating information from numerous network packets. Typically, this information is fragmented across numerous packets and intertwined amongst innocuous traffic. Added to this complication, sophisticated probing methods can cause packets to occur in random sequences, with erratic and unordered timing.

2.3.1.1 Packet Capture

In order to monitor a network, at the lowest level, a 'sensor' system captures and logs packets received at a network interface. Network interfaces (i.e. an Ethernet card) can be set to promiscuous mode in order pick up all traffic signals arriving at the interface, even if not addressed to the host. A strategic point of the network such as Internet firewall forms a 'choke-point' where all in-bound and out-bound Internet traffic must traverse, and is suitable place for monitoring the boundary of a network.

A sensor only needs to log the first few bytes of a packet data in order to capture header information. The presumption is that intrusive activity is identifiable by the header information, or the first few bytes of the payload. Capturing packets in their entirety amounts to fully duplicating network traffic and would require far more storage and. Log entries also include other important information like packet size and time of arrival. Packet filtering is a method by

which only certain forms of traffic is captured and logged, thereby reducing the amount of (presumably disinteresting) traffic that is logged.

TCPDump is an open source command prompt network logging utility that utilizes the LibPCap packet capture library to capture and log packets, with support for Berkley Packet Filtering (BPF) [TCPDump, LibPCap, McCanne 1992]. The Ethereal protocol analyser and Snort NIDS also utilise LibPCap to extract packet header information [Ethereal, Snort]. Apart from monitoring tools run on standard desktop computers, network devices such as switches and routers may also offer logging facilities. One issue is that these logs may be in proprietary formats and vary from vendor to vendor, complicating the consolidation of capture data.

2.3.1.2 Manual Review with Textual Tools

Manual network monitoring and investigation of security incidences predominantly entails textual review of packet capture logs. According to interviews conducted by Ball et al, majority of administrators resort to text-based tools, despite some of the marked advantages of using visual methods; they further site a lack of scalable, concrete, visualizations as a reason [Ball 2004: 56]. To understand the output of textual tools requires technical knowledge and tedious inspection of network event logs and alerts. To attain a holistic overview of traffic patterns is also tenuous because it requires that the reviewer mentally correlate several log entries (which may occur in disparate ordering).

Another difficulty is that textual review of network events becomes unmanageable for high volumes of data. *“In a moderate sized class B network, log files and packet traces may easily approach terabytes of information each day”* [Ball 2004: 56]. For this reason, tools such as tcpdump and Ethereal support filtering techniques to reduce the data. However, excessive filtering can omit facets of a full data set that could possibly give rise to anomalous traffic patterns the reviewer is not anticipating.

2.3.1.3 Network Intrusion Detection Systems

To deal with large volumes of traffic data, network intrusion detection systems (NIDS) automate the task of monitoring networks by inspecting traffic for signs of intrusion. A strategy to monitor the level of threat experienced by an organization is to place an NIDS in front of a firewall, allowing it to sample raw unfiltered incoming Internet traffic (from the ‘open’ Internet). Alternatively, placing an

NIDS behind a firewall can provide a second measure to check the effectiveness of the firewall for breaches; and since firewalls can filter both incoming and outgoing traffic, this placement can also reflect upon outward bound Internet traffic [Rehman 2003: 8].

A conventional NIDS (such as Snort) makes use of intrusion signatures to look for traces of traffic patterns known to arise as the result of intrusive activity. In response to detecting suspicious activity, typically a NIDS logs and alert for review by an administrator. Some systems can take reactive measures (i.e. blocking the attempt by discarding the traffic) and are termed intrusion prevention systems, and are similar to firewalls, with a subtle difference. Both attempt to block malicious activity, but intrusion prevention systems take a black-list approach revoking traffic identified as a threat, whereas firewalls tend to take a white-list approach by defining which types of communication is permissible (i.e. port filtering). Effectively NIDS filter out innocuous traffic information and reduce the textual data. However, *“most successful, algorithmic pattern matching systems are generally limited to recognizing patterns whose general form has been anticipated by the developers of the algorithms”* [Fisk 2003: 1]. . For this reason, NIDS may miss more subtle and novel intrusion activity, and provide a false sense of security.

Another issue with NIDS is the occurrence of false alarms, also termed false positives. False positives occur when an IDS identifies innocuous traffic as intrusive. For example, a misconfigured network device could trigger an alert. According to Putton et al, the problem of IDS false alarms is well documented and *“packets can be crafted to match attack signatures such that alarms on a target IDS can be conditioned or disabled and then exploited”* [Putton 2001]. Network visualizations are not exempt from the problem of false positives either. Therefore, in both cases (IDS and visualizations), the capability to drill down into the packet capture log data can prove useful in resolving false alarms.

2.3.1.4 Honeynets and Darknets (Network Telescopes)

Production networks carry large volumes of traffic which can obscure intrusive activity amongst legitimate traffic. To avoid this complication, network telescopes are essentially empty unused Internet address space. These networks are also sometimes termed darknets, since there should be no traffic observed in the range, as the network offers no services, nor is it intended to use any Internet services. So a high signal to noise ratio is presumed since any traffic entering the darknet is rouge traffic, and warrants suspicion.

A honeynet is a collection of honeypot hosts that form a part of the network. A honeypot is a system that is internally exposed and made to appear attractive targets for attack. The idea is that honeypots lure attention away from production systems and collect intrusion data as it serve as bait. The difference between a honeynet and a darknet (network telescope) is that a honeynet is part of a network and performs active measurement. A darknet is an entirely separate unused range that performs passive measurement. For example, a honeypot machine can respond to connection attempts, but a darknet sensor should only listen and not respond.

2.4 The Advantages of Visual Methods

Whilst majority of IDS are well suited to detecting known intrusion techniques, they are not adept at picking up novel forms of attack (as mentioned before in 2.3.1.3, and stated in several papers promoting network visualizations). In particular, Teoh et al. argue that, “*by incorporating human perception into the data mining process, researchers can detect patterns in data missed by traditional automatic data mining methods*” [Teoh 2004: 27]. Similarly, Yin et al claim, “*the human mind is capable of very fast visual processing outweighing the data mining capabilities of machines*” [Yin 2000: 26].

2.4.1 Parallel and Pre-attentive Visual Cognition

Biologists note that primates have a highly evolved visual system, especially humans, who rely most predominantly on their sense of sight over and above the other senses. The pre-attentive and parallel nature of visual cognition graces a human viewer with an excellent aptitude for recognising and interpreting patterns represented in a graphical format. The use of graphs in many disciplines that correlate and study relationships in data (e.g. scientific disciplines) support this point.

Literature in the field suggests that visual methods offer marked advantages over textually based tools. As Ball et al explain, “*text data is absorbed sequentially via*

the auditory cognitive modality⁹, as is speech” [Ball 2004: 56], whereas visualization tools “*take advantage of the parallel and preattentive nature of the visual-spatial cognitive modality.*” [Ball 2004: 55]. When looking at an image, the human mind perceives the image in its entirety. Textual formats by contrast are perceived in sequential steps as each word and line is read, and requires attentive linking between entries in order to gain insight into relationships and patterns obscured inside the textual form – this is explains why manually reading a security log files is inefficient and feeble. Transforming log data into a suitable visual metaphor can provide an image that is perceptible in a holistic manner.

2.4.2 Multi-Dimensional Visualization

For complex data such as network traffic, multi-dimensional visual techniques can provide an advantage over and above standard 2-D graphical methods. Although commonly projected onto a 2-D display device, using three-dimensional perspective of diminishing lines allows a viewer to perspective an added level of depth and locality. This extra dimension can enhance their grasp of multifaceted relationships within the data. Visualizations can further extend the dimensionality with the use of animation to relate a fourth dimension of time – a benefit of not being restricted to static graphs on paper. The Dimensionality of a visualization can be extended further with the use of colour (including transparency), shape, size, and orientation.

2.5 Chapter Summary and Conclusion

As background to this research, Internet hostility is highly prevalent considering the figures offered in Section 2.1.2. As discussed in Section 2.1.3, the problem is compounded by a rising number of vulnerabilities, along with several other factors (as noted throughout section 2.1). The illusive characteristics of intrusive activity, and its various forms, is brought into bearing by Section 2.2, and suggests that combating this problem is not a trivial matter. The classification of network probing techniques will serve the discussion in the next few chapters, as it outlines what security visualizations should attempt to convey (Sections 2.2.2 and 2.2.3).

⁹ Ball et al. reference this point to: **Wickens C., Sandry, D., and Vidulich, M.** *Compatibility and resource competition between modalities of input, central processing, and output.* Human Factors, 25(2):227-248, 1983.

In the discussion of conventional network monitoring (Section 2.3), the shortcomings of manual textual review and NIDS are briefly highlighted, and in future chapters, packet capture, and darknets will come to bearing. Section 2.4 follows on to contrast the promise of using visual analysis with the disadvantages of textual review and NIDS. In conclusion, this chapter establishes the basis of the problem, outlines the current inadequacies of contemporary methods, and if persuasive, alludes to the role network security visualizations can play in aiding network security research.

Chapter 3

Related Work

This chapter offers a critical review of a select number of network security visualizations, and aims to reflect the state of the art in the field. Amongst the reviews, Stephen Lau's 'Spinning Cube of Potential Doom' is the principle work. As this chapter unfolds, it draws out the merits of Lau's visualization in comparison to other visualizations. In tandem, it also gleans features from the work of others that prospectively would enhance and extend Lau's work.

A list of reviewed visualizations is as follows:

- The Spinning Cube of Potential Doom, by Stephen Lau [Lau 2003, 2004].
- VISUAL, by Ball et al [Ball 2004, Fink 2004].
- Scanmap3D, by Daniel Clark [Scanmap3D].
- The 'Space Shield', by Fisk et al. [Fisk 2003].
- VisFlowConnect, by Yin et al [Yin 2004].
- The OASC Visualization by Teoh et al [Teoh 2004].

3.1 Selection Criteria for Reviewed Network Visualizations

The majority of visualizations reviewed provide a perspective of an internal (home) network domain versus the external Internet, with the intent of viewing traffic traversing a network boundary. This selection choice is justified in lieu of the primary investigative focus – visually identifying network intrusion attempts (i.e. probing scanning activity). Network visualizations with more diverse purposes – for example bandwidth usage and performance analysis – are not subject to review. Although there is a commonality in the intended application of these visualizations, the selection covers a wide variety of plotting schemes and features that provide insight varied approaches.

The assessment of respective visualizations considers several criteria:

- **Visual metaphors** – a visualization should provide clear, salient, and suggestive representations for network events.
- **Plotting schemes** – intuitive, readily interpreted plotting schemes enhance the viewer’s ability to form a ‘concrete’ understanding of the data visualized. Conversely, abstract and complex schemes may make it more difficult to relate the visual elements back to the data attributes from which they are constructed.
- **Salience** – In terms of ‘visual network intrusion detection’, a visualization should *saliently* convey intrusive activity, and make such activity appear distinct from innocuous traffic. The term salient implies automatically perceptible, as phenomena of interest will stand out by presenting themselves to the viewer’s conscience with strong force.
- **Versatility** – a visualization intended for compressive analysis should be capable of exhibiting a wide variety of different scanning methods.
- **Dimensionality** – in relation to versatility, higher dimensions can be used to correlate more data, thereby providing a greater capacity for exhibiting complex relationships between multiple traffic attributes. In this regard, the use of 3-D space, colour, and animation, can help capitalize on the capabilities of computer graphics.
- **Scalability** – A scalable visualization is able to represent both large and small amounts of data, preferably without losing its ability to convey anomalous traffic patterns indicative of intrusive activity.

In passing, the assessment also draws out other important concepts and facets about network visualization. The first review looks at the Spinning Cube of Potential Doom, with the intention of contrasting and comparing to it the works that follow.

3.2 Network Security Visualizations

3.2.1 ‘The Spinning Cube of Potential Doom’

Stephen Lau developed a coloured 3-D scatter-plot visualization of Internet traffic within a cubic reference frame, as seen in Figure 3-1. He demonstrated the at the SC03 conference for high performance computing and networks¹⁰. SCinet

¹⁰ <<http://www.sc-conference.org/sc2003/>> (25/06/2005).

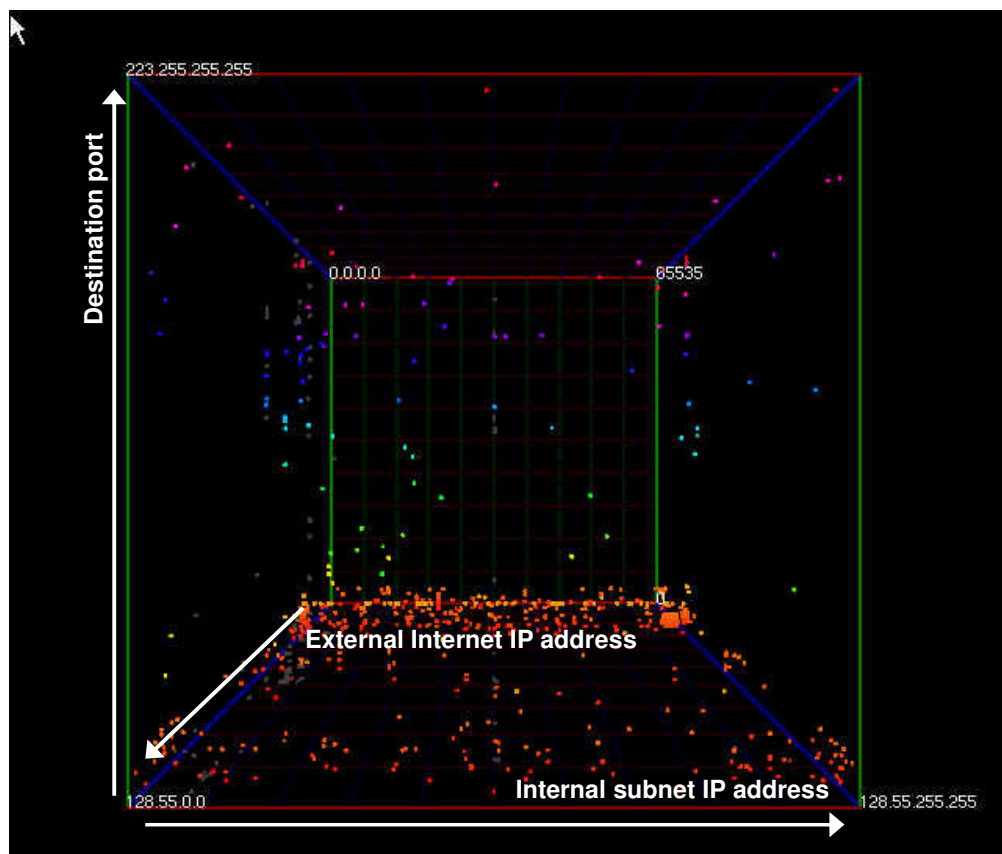


Figure 3-1: The Spinning Cube of Potential Doom – plotting scheme [Lau 2004]

provided the conference provided with extreme bandwidth (10-gigabit Ethernet) and no firewall for un-impinged Internet access¹¹. This allowed an ‘open’ perspective of intrusive activity present in unfiltered Internet traffic.

3.2.1.1 Concept and Features

Annotated axes in Figure 3-1 illustrate the plotting scheme. The red axis (horizontal in picture) references the organizations is referenced along the as it plots the internal domains IP address range. The entire Internet is plotted according to the blue axis (looking inward into depth of picture), and the port numbers are along the green axis (vertical in picture). The user can interact with the display by zooming in and out (scaling), and swivelling (rotating) the display.

As a visual metaphor, points represent TCP connections and connection attempts. The visualization utilizes the Bro-IDS¹² to extract the TCP/IP handshake

¹¹ <http://www.sc-conference.org/sc2003/infra_scinet.html> (25/06/2005).

¹² <<http://www.bro-ids.org/>> (25/06/2005).

information and flag connections as successful or incomplete. As the animated display plays back recorded network data, it draws successful connections in grey and incomplete connection attempts in colour. The use of grey diminishes the visual impact of established connections, presumably innocuous, whilst the coloured points attract attention to the unsuccessful connection attempts. Lau intended the rainbow mapping of colour by port number to assist with locating points in the 3-D view. This reinforces the spatial relation of the destination port number, and so in effect, it does not extend the dimensionality.

Lau assumes that systems will keep majority of their ports closed. Consequently, scanning attempts will leave a coloured pattern as a trail as they fail to make connections. He considers the majority of coloured dots as malicious activity, further speculating that though there may be that false positives, they are negligible. False positives (coloured dots) could be the result of inoperative services or misconfigured networks, but as such are unlikely to account for the noteworthy patterns observed.

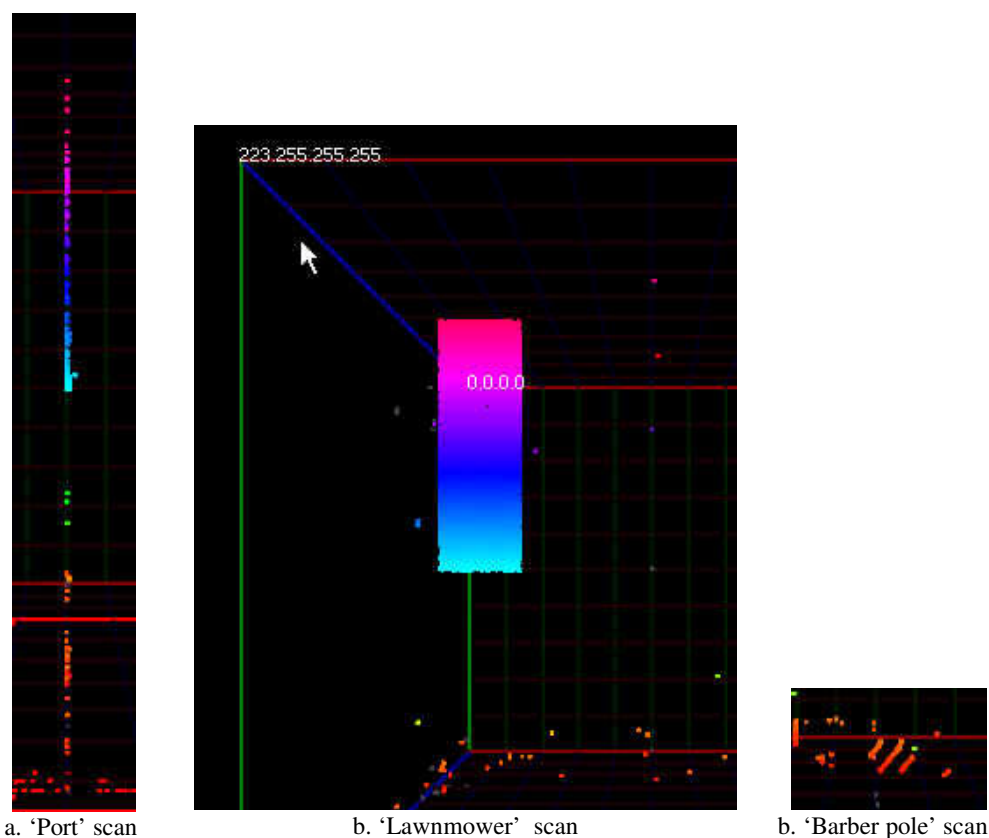


Figure 3-2: Network scanning activity [Lau 2004]

3.2.1.2 Observations of Intrusive Activity

The Spinning Cube of Potential Doom exhibited a range of suspicious network activity that was presumably mapping the address space for target hosts and probing for vulnerabilities. In Figure 3-2a, a port scan of a single host shows up as a vertical line. A second type of scan is dubbed by Lau as dubbed a ‘lawnmower’ because thoroughly covers a range of address over a range of ports. It blatantly shows up on the display as a coloured rectangle as seen in Figure 3-2b. Figure 3-2c reveals a third type of scan dubbed a ‘barber pole’. It forms a series of diagonals as it increments through a series of port numbers whilst scanning across the address space, and tracks back down to a lower port one it reaches an upper bound.

As a visual metaphor for connection attempts, these simple points provide that very appropriate and distinct visual patterns for making rapid port scanning events evident. To notice slower and random scanning techniques would not be as simple. This would require some way of representing the history of connection attempts. Points could be drawn for an extended time afterward, but this may result in obscuring the display when displaying a large number of connections.

3.2.1.3 Scalability and Performance

Exactly how many connections “The Spinning Cube of Potential Doom” could be represent without clutter was not explicitly mentioned, although it presumably handled a high volume of traffic at the SC03 conference. A preliminary overload test was conducted for my clone visualization of “The Spinning Cube of Potential Doom”. With a noise of 100,000 random points (representative of 100,000 connection attempts), and an internal network of 1000 hosts, pseudo port scans were generated and still noticeable. As a rough approximation, this clutter obfuscation test shows that intrusive activity could still be discernable with 100,000 connections for 1000 local (internal) hosts¹³. This simple trial will be brought to bear in assessing the scalability claims made by Ball et al, as presented in the next section.

¹³ Note that the points are smaller than those of the original visualization, and that the density of noise is evenly distributed; one could expect that the lower known port range (0-1023) would have a higher incidence of traffic.

3.2.2 VISUAL – Ball and Fink et al, “Home-centric visualization of network traffic for security administration”

VISUAL stands for Visual Information Security Utility for Administration Live, and this visualization is detailed in the paper entitled “Home-Centric Visualization of Network Traffic for Security Administration” [Ball 2004]. It is a prototype visualization and predecessor to their Network Eye visualization suite [Fink 2004]. Ball et al designed the VISUAL with particular attention to the network monitoring requirements of administrators. As mentioned before a major concern was scalability. In their own words,

VISUAL’s purpose is to provide more concrete visualizations that will require much less training to interpret. In addition, VISUAL is also more scalable, showing up to approximately 10,000 external hosts and 2500 internal hosts. Visual is a security-purpose, visual, concrete presentation of direct data from a home-centric perspective. [Ball 2004: 57-58]



Figure 3-3: VISAUL displaying 80 hours of network data with 915 communication between a home network of 1020 hosts and 183 external hosts [Ball 2004: 58]

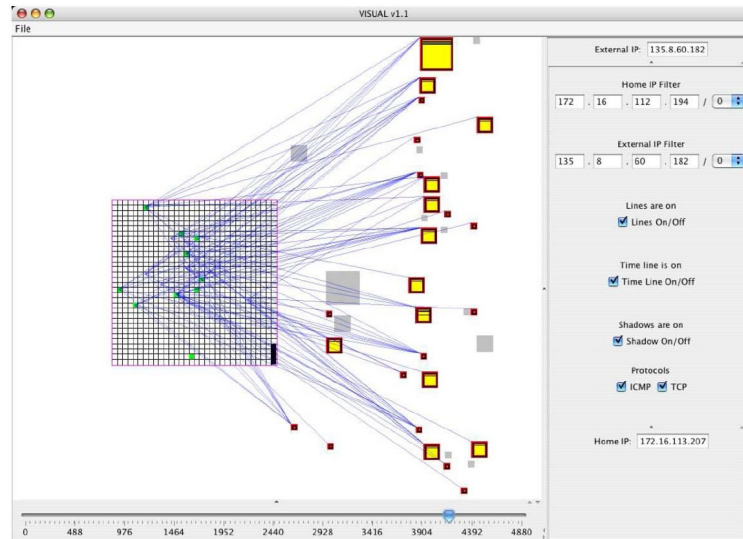


Figure 3-4: Time line example using shadows. The light-grey squares represent inactive hosts that have been active in the last 200 seconds. The yellow squares represent active hosts during the selected second.
[Ball 2004: 61]

The term ‘concrete’ is used in opposition to abstract, and presumably means that the visualization represents real network concepts and characteristics. Of the visualizations reviewed in this survey, the visualization by Teoh et al. (0) is the one example of an abstract form of representation, as opposed to the rest which can be considered concrete. By the term ‘direct data’, Ball et al. mean that the source data is interpreted directly from log files and does not do any pre-processing for the visualization. Packet traces from TCP-dump or Ethereal are mentioned as sources of data for VISUAL.

VISAUL uses a three dimensional head on perspective providing a depth perspective (Figure 3-3). The internal network host markers are small squares in a flat grid which is placed in the background. External host markers are squares of varied size placed in the foreground. Blue connection lines are drawn from green internal host markers to transparent yellow external hosts markers (noting that the transparency helps alleviate visual obstruction problems). The varied sizes of external host markers reflect the amount of traffic exchanged between each external host and the home network. Inside these square markers, coloured lines are drawn to represent the destination ports used in communications. Given that the external host square we big enough, are one were able to zoom in, port scans might show up as colour arrangements of lines.

In mapping both internal and external host markers, the visualization attempts to keep the same relative arrangements. This helps maintain the viewer's ability to recognise repetitive patterns in the traffic. The grid of internal hosts is automatically placed in an empty area not obstructed by external host markers, and may be moved or resized to avoid overlap. A mapping function assigns a unique and static virtual position to all of the external IP addresses (approximately 4 billion possible addresses for IPv4). The first two octets are used to map the x coordinate, and the last two for the y coordinate. As a result, external IP addresses originating from the same class B address space end up clustered in vertical columns. This would assist viewers to see sub-network patterns between external domains and the home network

In the event that two external hosts map too near to each other, an anti-overlap algorithm shifts competing markers downward (in the y direction). If the bottom is reached, y is reset to 0 (the top) and the marker placed to the right. If the display begins to run out of space, all the host markers are scaled down. Where the display animated with a high number of clustered external hosts, some interesting knock on shift effects could occur. As the amount of traffic exchanged grows, so does the size of the external host marker, which may cause shifting of nearby makers. New connections may also result in the shift of several markers.

Ball et al recon that in general VISUAL could display approximately 10,000 external hosts with their mapping approach [Ball 2004: 58] Figure 3-5 shows VISAUL displaying 80 hours of network data with 915 communications between a home network of 1020 hosts and 183 external hosts. Looking at this, a discerning reader might come to question their claim. In Figure 3-3 one can already notice some confusion between overlapping lines, and this is with only 183 external hosts. 10,000 external hosts would entail at least 10,000 connections, and representative connection lines, which (presumably) would render the visualization unintelligible (unless substantial filtering was employed). Recalling that marker size was used to convey the amount of traffic exchanged, drawing 10,000 external host markers would diminish this metaphors effectiveness¹⁴.

¹⁴ A screen resolution of 1024x768 affords 786432 pixels. Assuming a full screen mode, for 10,000 markers, this would leave at most 78 pixels per marker, not considering the space needed between to show separation between markers. Therefore, at best, there is enough for an average square region of 8x8 per marker.

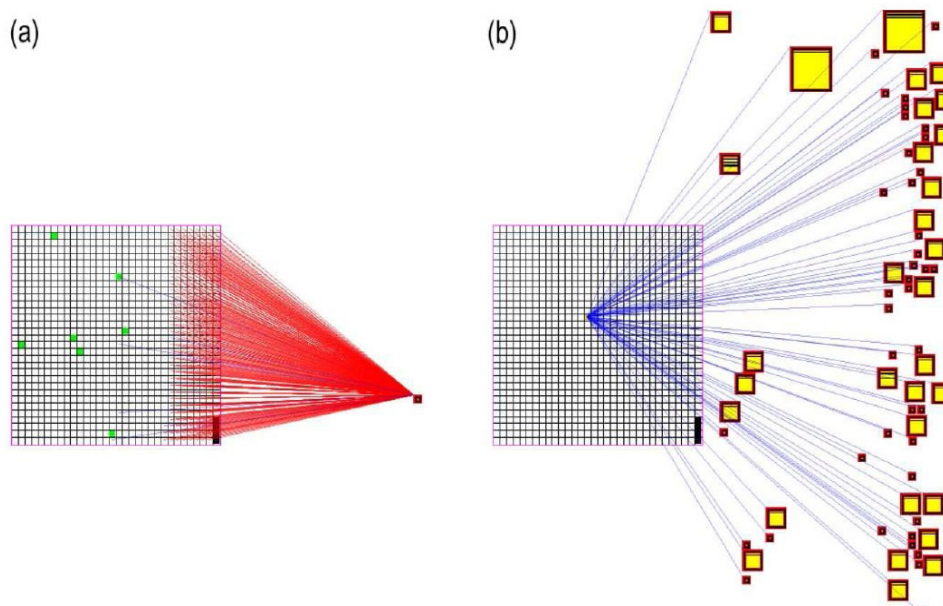


Figure 3-6: (a) Example of fan-in. An external host performing a ping sweep on a subnet in the home network. (b) Example of fan-out. Many different external computers are downloading information from a server on the home network. [Ball 2004: 59]

VISAUL provides some interactive filtering options where users can select single internal hosts, or ranges of hosts, to only view traffic connecting to that host. Similarly, the same can be done for external hosts. Time filtering is also available via a timeline. All the traffic over a range of time is drawn by default. The user can enable the time line, and show only the connections occurring within a one second window around the selected time. An obvious extension would be to have a second time window slide bar that allows the user to select the size of the time window. As an example, VisFLowConnect (reviewed in section 3.2.5) employs this time-window technique. To help track the chronological flow of events, VISUAL draws shadows that linger 200 seconds after the connection was terminated (Figure 3-4)¹⁵. Again, allowing this value to be variable, and varying the shade of grey according to the time lapsed would be two improvements.

Connection flow lines can exhibit some interesting fan-in and fan-out effects (as shown in Figure 3-6). The visualization allows the viewer to see external hosts mapping the internal network with a ping sweep for example. It also allows the viewer to see all the external hosts connecting to servers, and a multiply co-

¹⁵ This is similar to my suggestion of decaying transparency for the “Spinning Cube of Potential Doom” (3.2.1)

ordinated denial of service attack by numerous external hosts may be exhibited in this way.

Details on demand is provided by the visualization when a host marker is selected, namely, the host's IP address, IP addresses of all connected computers, the TCP/UDP destination and source ports used, and the percentage of overall traffic the particular host contributes.

Overall, despite my scepticism about the scalability VISAUL purportedly has, this visualization provides a good overview of internal vs. external communication and would serve well in monitoring an internet connection. As mentioned, variable time window and shadow-time would be useful extensions. It would also be useful to have the visualization animated for playback, and Ball et al noted their intent to adapt the visualization for real-time monitoring. At the time of the paper by Ball et al., VISAUL was a static visualization only used to review previously recorded data.

3.2.3 Scanmap3D by Daniel Clark

Scanmap3D (Figure 3-7) is developed by Daniel Clark as open source visualization intended for exploring the snort¹⁶ database logs [Scanmap3D]. It is written with java using the Java3D library. Scanmap3D-2.1b, and the newer version,

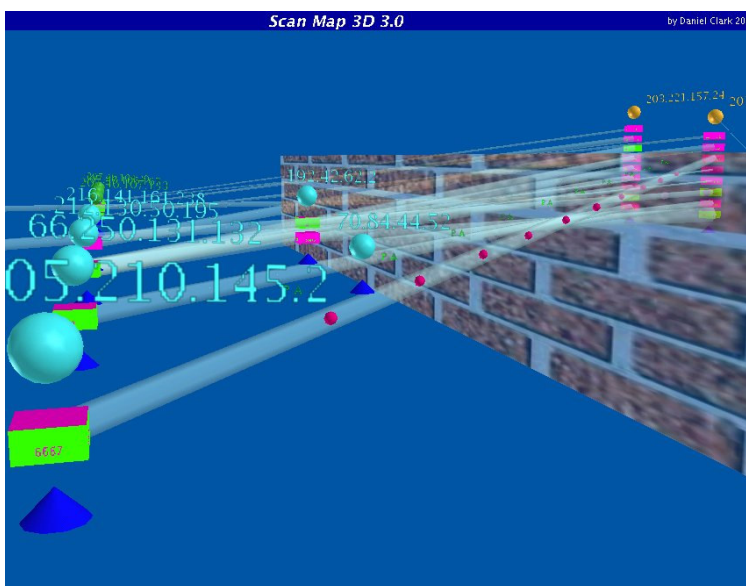


Figure 3-7 : Scanmap3D 3.0 [Scanmap3D]

¹⁶ <<http://www.snort.org/>>

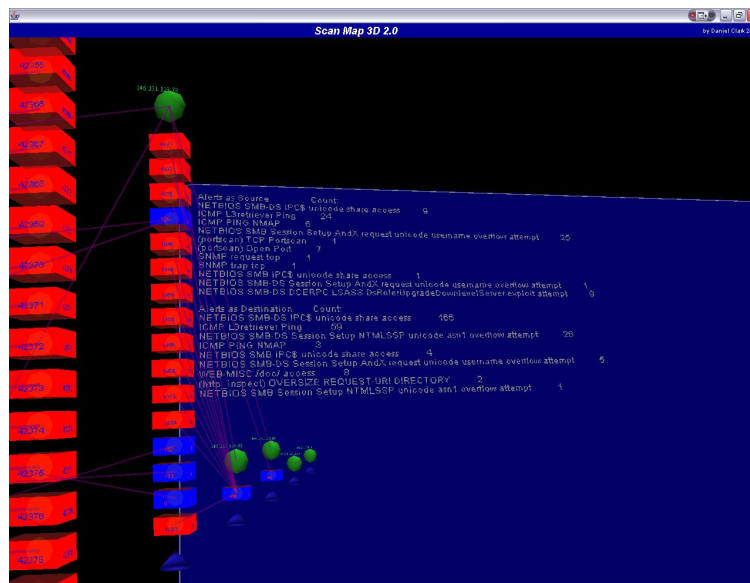


Figure 3-8 : Scanmap3D-2.1b Textual Drill Down [Scanmap3D]

Scanmap3D-3.0, were tested on my lab computer¹⁷ with network traffic captured by snort and logged to a MySQL database.

Taking a look at Scanmap3D-3.0 (Figure 3-7), the light blue spheres signify hosts. Internal hosts and external hosts are drawn on opposite ends of a firewall (which can in practise represent a local gateway, router or actual firewall). The stack of boxes beneath the host sphere denotes ports ordered by event occurrence (for TCP and UDP), with two different colours to distinguish source and target ports. Connections between hosts are drawn as transparent blue lines between port boxes of varying width according to the amount of data sent or received. Protocols without the concept of ports (i.e. ICMP) are drawn to and from the host spheres. Small packets are drawn inside the connection lines where UDP is drawn as a cylinder, TCP a sphere, ICMP a box, and other types as cones. Once a viewer identifies the different shapes with what they represent, a picture of the connectivity relationships between hosts can be seen.

The visualization can be played back varying the rate of playback and applying filtering according to time, protocol, IP address or port. A control panel in a separate window allows the user to manipulate the playback rate. The speed at which packet events are animated is controlled by a packet speed slide-bar. The packet interval controls the delay before rendering the next event. Effectively, these controls determine the speed of the playback, and for this reason, the

¹⁷ Pentium-4 3.0GHz, 1GB RAM, 128MB Nvidia graphics adapter.

playback rate is not a natural, nor is it an accurate reflection of the time elapsed between packet events – only the order of occurrence is conveyed. Via the control panel, scanmap3D provides filtering according to address and port ranges, as well as protocol type. It also allows the user to toggle the rendering connection lines and packets on and off.

Selecting the cone underneath a particular host brings up the textual alert information for the host, as logged by snort (Figure 3-8). The textual drill down mechanism draws the text in an appropriately oriented blue transparent pane within the visualization. This technique can add to the visual clutter and due to the transparency, text can be difficult to read when many connections were being drawn in the background. Instead, opening up a separate textual dialogue window would avoid these issues. Using a separate textual dialog window would also make cut and past operations possible. Furthermore, rendering the text in 3-D space may impact on performance when compared outputting text to a dialogue window. In favour of displaying the textual detail within the visualisation, the textual data is localised right next to the object it relates to, and text and visualization are both viewable at the same time instead of having to switch between windows.

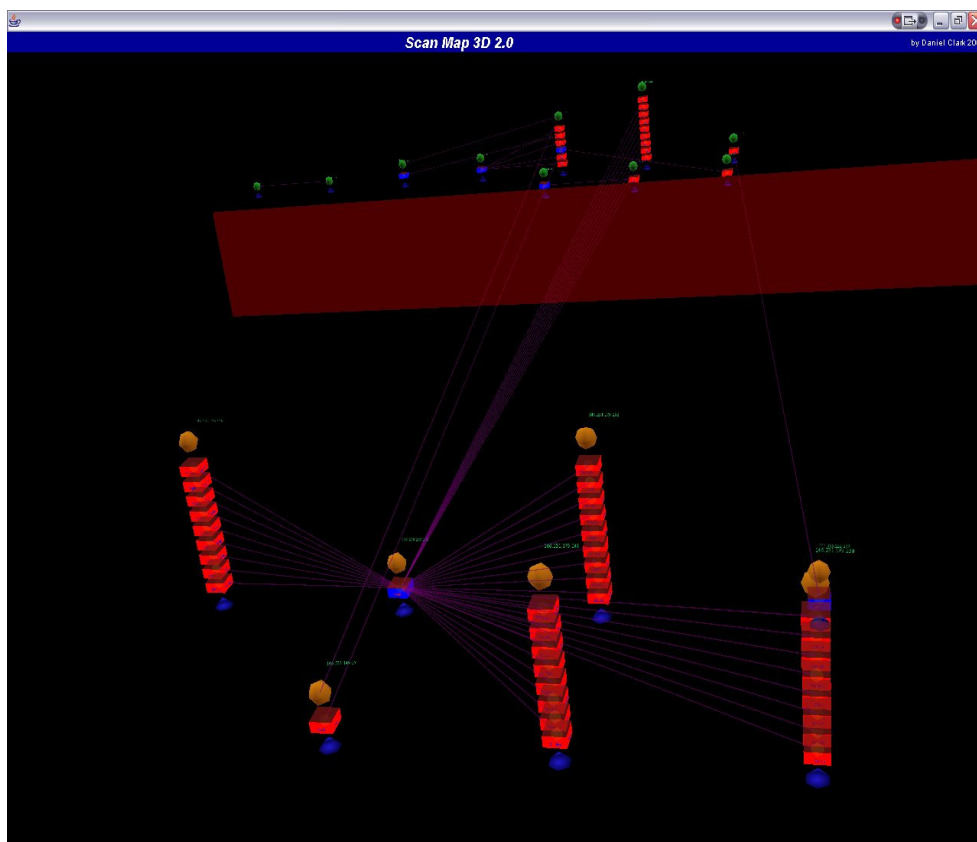


Figure 3-9 : Scanmap3D-2.1b – Connection fans from a broadcast address [Scanmap3D]

In Scanmap3D-2.1b, some intriguing ‘fanning’ affects were observed (Figure 3-9). Although the port stacks are arranged by port number, they are simply stacked on top of each other by order of occurrence without maintaining a regular or uniform spatial relation i.e. there is no consistent scale or height at which ports are drawn according to port number¹⁸. With this visualization, the fan affect is inevitable after numerous UDP or TCP connections, and should not necessarily be misinterpreted as port scans. Port scans need closer inspection of the port numbers drawn on the port box to determine if consecutive port numbers have been attempted.

Unfortunately Scanmap3D-2.1b suffers from performance and scalability limitations. It can only display a fair number of connections without the display becoming overly cluttered and disordered. Performance to begin with is slow, rendering only a few events every second and degrades as more events are drawn. One factor that may contribute to the poor performance is the stacking of these TCP and UDP port boxes. Every time a new TCP or UDP packet was represented, all of the objects above it were shifted upward. Rendering performance became extremely poor if playback of packets was sped up too fast. At maximal packet drawing speed (packet delay set low), the visualisation becomes jerky and

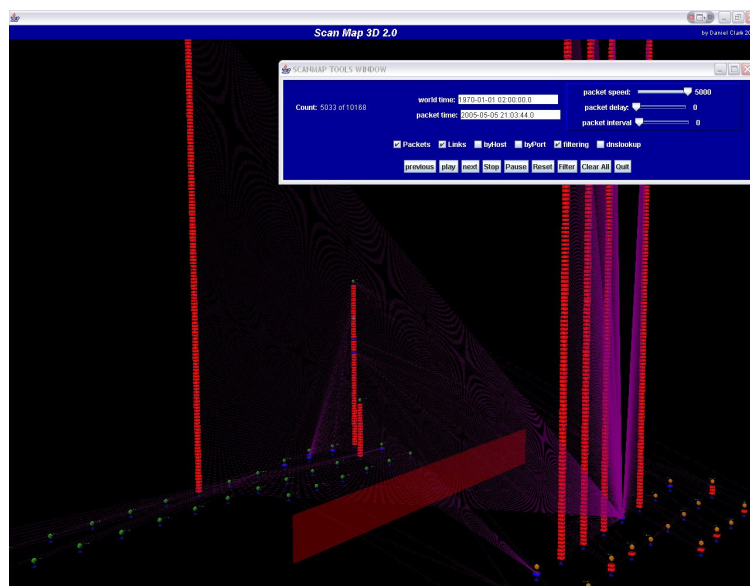


Figure 3-10: Scanmap3D-2.1b – 5000 events between 32 internal hosts and 24 external hosts [Scanmap3D]

¹⁸ For example, were a host connected to on ports 22 then 80 at first, port 80 would be stacked below port 22. On another host, port 10000 and port 65000 could be stacked in the same way at the same height.

unresponsive. The same problem occurs after rendering about 4000 connections. Even the maximum rate at which it does playback packet events requires some patience. After running more than an hour and a half, it had gone through less than 5000 events between 32 internal IP addresses and 24 external IP addresses. It had only covered the time range from 20:10:00 to 21:00:00 (50 minutes), which is a rate slower than real-time playback.

From the testing performed, it would not be suitable for over viewing 5000 connections between more than 50 hosts. Using scanmap3D-2.1b, 5000 events between 32 internal and 24 external hosts was viewed (Figure 3-10). Aside from performance issues, the display could no longer accommodate all the visualized hosts within a single view. Some hosts were cut out by the back clipping plane (the backward edge of drawing space in the graphics environment), and some port stacks grew too high to fit. A large concentration of connection lines obfuscated lines drawn behind them, making it hard to see the connections between some hosts. Added to this, hosts are placed in a grid-like manner, and two distant hosts in the same row can result in connection lines going through and past the hosts placed in between. This causes the connection lines to be redrawn over each other and makes it confusing to spot at which host the connection line originates or terminates. As can be seen, the thin transparent purple connection lines are very faint and difficult to pick up against the black background, especially when rendered in the distance.

The newer version, Scanmpa3D-3.0, has notably improved performance, and a different colour scheme¹⁹ (refer back to Figure 3-7). As a visual enhancement, the width of communication lines reflects traffic volume; this adds more information to the visualization, but may worsen occlusion problems when it comes to visualizing larger data sets. Along with the new colour scheme, the thicker lines make the connections more salient, and in this way it is an improvement over the previous version which drew faint thin feint purple lines. As with version 2.1b, the positioning of hosts in version 3.0 has no obvious ordering, but one difference is that Scanmap3D-3.0 radically relocates hosts from time to time, which can be little disorientating when hosts get jumbled around – as Clark concedes, its something his working on, and is experimenting with various ways of laying out the hosts. I assume the relocation of hosts is in order to reduce the amount crossover between

¹⁹ Unfortunately, when running scanmap3D-3.0 on my system, it failed to render the connection lines. Therefore a fair amount of my observation was based upon the older version, Scanmap3D-2.1b, and I could not check if version 3.0 solved some of the placement problems. Furthermore, I was unable to properly see packet animation flow in both versions.

linked lines, as well as minimizing the distance between regularly communicating hosts. This would help prevent clutter when drawing a high number of connections. However, the problem with moving hosts around is that the viewer may lose the ability to readily recognise repeated communication patterns between particular hosts over time (this was also one of the concerns that VISAUL attempts to avoid with its external host placement – section 3.2.2)

In summary, Scanmap3D is useful for conveying a close and detailed depiction of network connectivity, and the communicative relationships between hosts. It is however limited by the number of hosts and connections it can represent before the display becomes overwhelmed with many crossed lines between hosts, as well as the degrading playback performance. As a more positive testament, the use of transparency in Scanmap3D is most noteworthy. The transparent connection lines and transparent ‘firewall’ help alleviate occlusion problems by letting the viewer see through the obstructions.

3.2.4 The ‘Space Shield’ – Fisk et al, “Immersive Network Monitoring”

The paper, “Immersive Network Monitoring” [Fisk 2003] describes a visualization that looks something like a space shield of sorts with a number of lines radiating from within (Figure 3-11). The visualization is “designed to compliment existing intrusion detection systems used by network operators”, and the primary goal is “to enable detection of uncharacterized attacks” [Fisk 2003: 1]. The visualization boasts a ‘real-time data gathering module’, and data filtering. Their visualization is built using the Flatland development environment²⁰, and has a system for mapping the data domain into a ‘metaphorical representational domain’, along with a system for interaction with the metaphorical world. Drill downs into web-based reports can be done through interaction with the visualization.

²⁰ “Flatland provides functions abstractly similar to a window manager in that it handles the placement of multiple visual objects in a virtual space and mediates user input to those objects and movement around the space” [Fisk 2003: 5] <http://www.hpc.unm.edu/research/scientific_visualization/homunculus_project.htm> (30-05-2005).

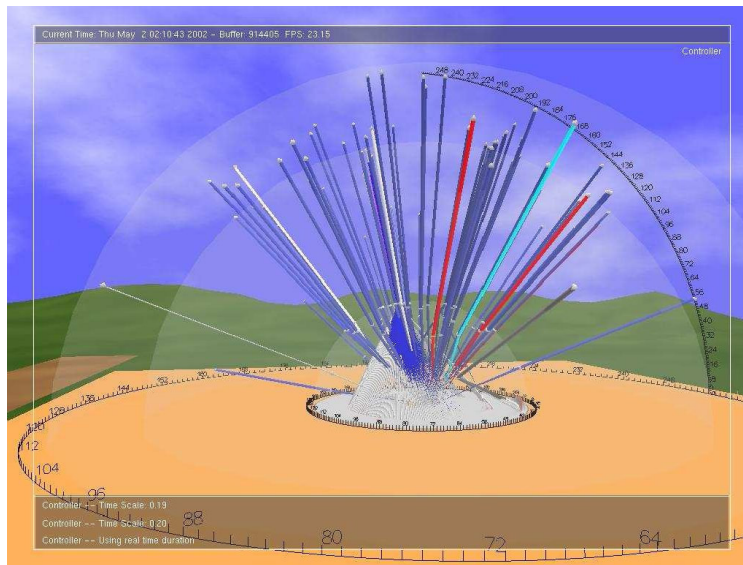
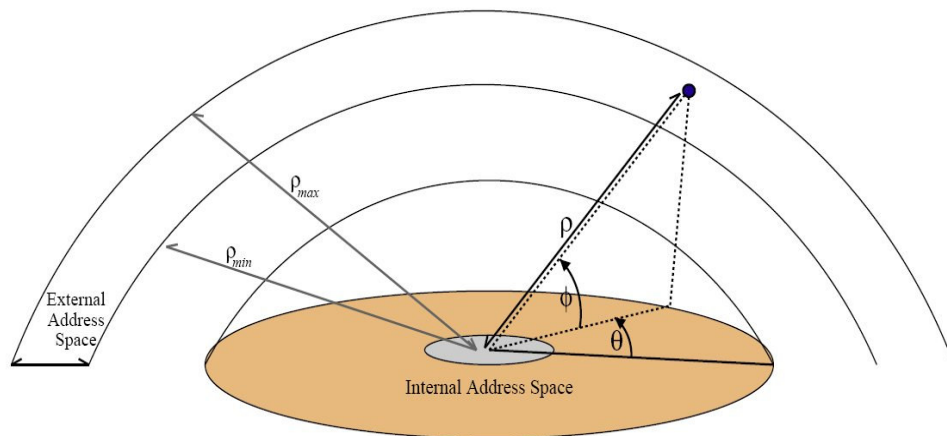


Figure 3-11 : Space shield with heads up display [Fisk 2003: 6]

The visualization takes raw packet source data and processes the data into connection flows. “A flow is defined purely by a sequence of packets with the same identifying 5-tuple (source IP, destination IP, source Port, destination Port, protocol) and no gaps between packets longer than some timeout value *k*” [Fisk 2003: 3]. The metaphorical framework for representing the data is “rooted in the primitive concept of *self vs. other* which relates to concepts such as *attack* and *defend* and *territory* and *shield*” [Fisk 2003: 3].

The internal network space is mapped into the central circular region. This central disk-like region maps out a class B subnet (/16). For the internal address space, each class C (/24) subnet is mapped as a concentric ring and the 256 IP addresses distributed throughout the ring. Given this mapping, addresses on inner rings will be more clustered than those on outer rings.

The external address space (i.e. the internet) is mapped into a hemispherical region on the outer bounds (Figure 3-12). The first three octets are used to map the location of an external host by altitude (vertical angle), azimuth (rotation angle), and radius (distance from center). External internet addresses are effectively clustered into class C subnets because the last octet is not used in mapping the position of an external address; there would be no visible distinction between two external hosts located in the same class C subnet. A ‘shied’ separates the internal network space the outer external network space – it represents the network boundary and appears as a semi-transparent dome in between the internal and external address space.



Coordinates for external IP address *A.B.C.D*:

Altitude $\phi = A \times 90^\circ / 255$
 Azimuth $\theta = B \times 360^\circ / 255$
 Radius $\rho = C \times (\rho_{max} - \rho_{min}) / 255 + \rho_{min}$

Figure 3-12 : Layout of external address space [Fisk 2003: 4]

The spherical and circular arrangement of address spaces seems appropriate considering the self vs. other concept behind the metaphorical components. However, one drawback is that relating the visual position of an address to an actual IP address is not straightforward. Contrast this scheme to the mapping scheme of the “Spinning Cube of Potential Doom”, which when viewed appropriately (side-on), allows a viewer to readily gauge the internal or external IP range of a connection (as IP addresses linearly increase in one direction). Furthermore, the concentric ring arrangement of the internal address space is not evenly distributed. This may result in visual artefacts, where artificial visual distinctions are induced by the mapping scheme. For example, a higher density and clustering of rays will go into a ring closer to the centre since the addresses are mapped more tightly. By comparison, a similar connection flow scenario of rays going to an outer ring address range will appear less dense.

Connections are drawn rays connecting the originating host to the destination through the shield. The rays have external and internal segments. The external segments for a given external class C subnet are aggregated together in a single ray that is always directed to the centre of the internal address space. At the shield, the ray can split up and deviate in the direction of the respective internal source of the data flow. So while multiple flows from the same external class C source share a common ray up to the shield, they will fan out into individual rays once past the shield. The ray extrusions are animated to reflect the direction and intensity of network traffic. One curiosity is that the external rays appear to be drawn all the

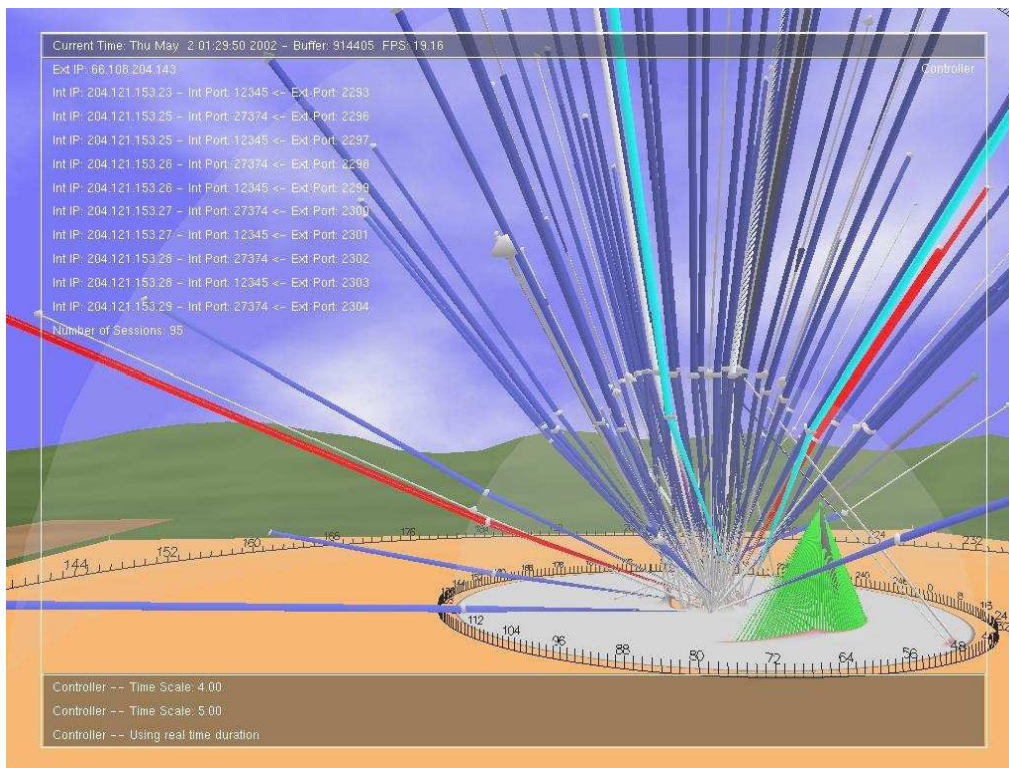


Figure 3-13 : Heads up Display of a Selected Attacker [Fisk 2003: 8]

way into the centre; to avoid unnecessary clutter; it may make sense to terminate the arrays at the edge of the internal network boundary (the inner spheres). This could also be used to convey what got thought the perimeter defence and what didn't.

The green rays within the Figure 3-13 reveals intrusive activity as several ports are scanned whilst hopping across host addresses as it goes along (this is much the same as the 'barber pole' scan mentioned in 3.2.1 with the Spinning cube of potential doom). Because different ports for a given host are not distinguished spatially, a port scan on a single host is less likely to show up. Therefore, spatial distinction of ports is preferable as it would be more effective at revealing port scans, especially those restricted to a single host, but this doesn't seem a feasible option in this particular visualization.

Each ray is coloured according to the port number the originating host used to communicate, and the fan of green results (as seen in Figure 3-13). Port scans may be better revealed using the target port which is normally of more concern in security applications. Were the Space Shield's port representation appropriately colour encoded according to destination port numbers, it would show up as a rainbow fan of colours instead of the green fan. Once again, supposing a port scan is focused on only one host, using the origination port to encode colour would

result in a single green ray which may not appear anomalous in contrast to normal traffic flows. At least if the destination port rainbow colour scheme were used, an animated change in colour would be noted for a single host port scan. (The “Spinning Cube of Potential Doom” is a good example that distinguishes ports both by colour and spatial location – section 3.2.1).

In order to select objects and drill down, glyphs in the form of simple spheres are placed at the endpoints of flow rays and where the ray deflects through the shield. A ‘heads-up-display’ (as seen in Figure 3-11) is an overlay that allows users to interact with the controls of the display and manage the data flow. As users examine the visual environment, they can select objects and instigate database lookups and queries. I found this more sensible than the drill down technique used in Scanmap3D (section 3.2.3). The heads up display could be rendered through a stencil buffer²¹ which would be more efficient than orientating and rendering the information in the 3-D environment. It also avoided the hassle of having to zoom in close enough to read the information (as was the case in Scanmap3D).

The animated playback of the visualization allows for time scaling, where the user may speed the playback up as fast as 60:1 (effectively one minute viewed in one second, or 24 hours viewed in 24 minutes). If the user navigates up close to the endpoint of a flow, more detail becomes visible in the form of a textual IP address and port – this illustrates the concept of proximity based detail, which remains hidden until up close.

The ‘Space Shield’ network visualization as such, conjures a rich and natural metaphor for viewing and exploring network traffic in terms of flows. Two features that the developers of this visualization consider very important is the drill down capability and the time scaling. (I did not note any explicit indication about the scalability of the visualization.)

3.2.5 VisFlowConnect – Yin et al, “NetFlow Visualizations of Link Relationships for Security Situational Awareness”

²¹ A buffer that is used as a 2-D overlay.

The VisFlowConnect visualization system [Yin 2004] is comprised of several 2-D visualizations that abstract between levels of detail, allowing the user to drill down from a higher level into more detailed visual representations. The upper level is the ‘global view’, followed by a similar ‘domain view’, and then an ‘Internal network view’. At the lower level of detail, a ‘Host stat view’ reports data in a textual format. The ‘global view’, ‘domain view’, and ‘internal network view’ make use of a ‘parallel axes’ view to portray the flows of network traffic between external domains and an internal network. A parallel axes “representation plots data from an arbitrary number of axes onto a two dimensional view... as a set of parallel lines with each line corresponding to an axis of the data. Each data point is then represented by a chain of line segments across these axes”. [Yin 2004: 28-29]. Typically, VisFlowConnect could be used to view connections between the *internet* and an organizational *intranet* (external vs. internal).

The visualizations utilize NetFlow logs, a popular log format introduced by Cisco. For each of the visualizations, static data is played back, and events drawn when their time of occurrence corresponds to the ‘current’ reference time (or rather, selected playback position). Since some connection events can be short lived, a ‘time window’ denotes the amount of time beyond its actual occurrence

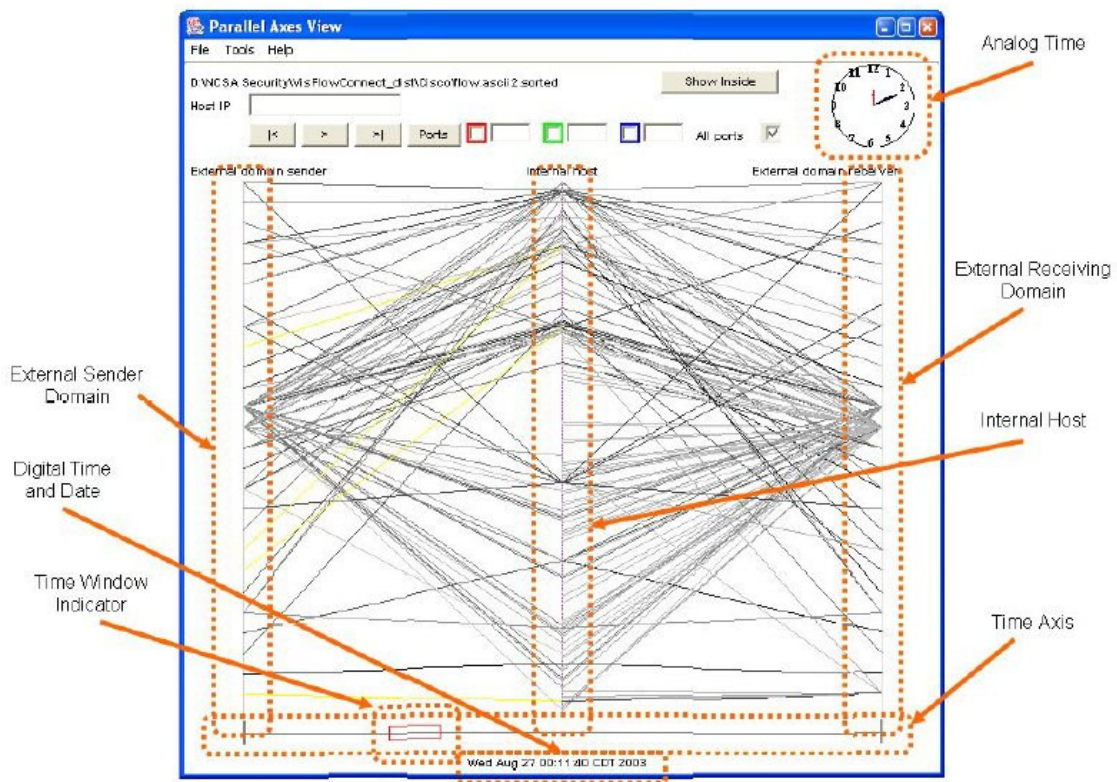


Figure 3-14 : global view [Yin 2004: 30]

that a connection remains displayed. Only connections with timestamps within this range are drawn. The user can dynamically adjust the size of time frame which adjusts the ‘time capsule’ of connection history that is conveyed by these lingering connections. (This is similar to the time window mechanism in VISUAL – section 30).

In the ‘global view’ (Figure 3-14), the left axis maps the external domain of incoming traffic, the middle axis maps the individual address space of the internal network, and the right axis maps the destination of outgoing traffic to the external domains. Lines indicate connection flows where lines on the left relate incoming traffic as, and lines on the right relate outgoing traffic. In order to handle a high volume of connectivity, external networks axes represent aggregated domains rather than individual machines. This avoids overcrowding the display with too many lines which would result in occlusion. Separating the display into incoming and outgoing sections helps avoid line cross-over.

By selecting a particular domain in the global view, the viewer can then drill down into more detail with the ‘domain view’. This narrows the view down showing only connection flows between a single external domain and the internal domain. So it drills down from an n-1 domain view to a 1-1 domain view. As mentioned, the metaphors for the domain view is much the same as the global view with one exception; instead of the aggregation of multiple external domains, it maps the individual external machines of a selected domain. In the domain view,

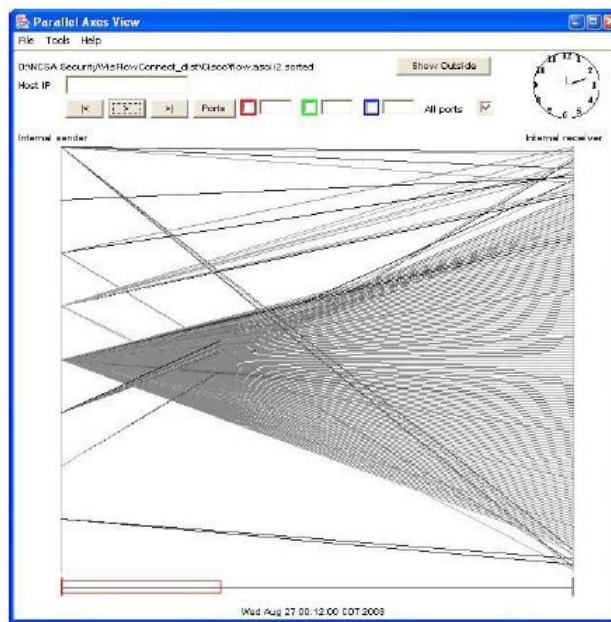


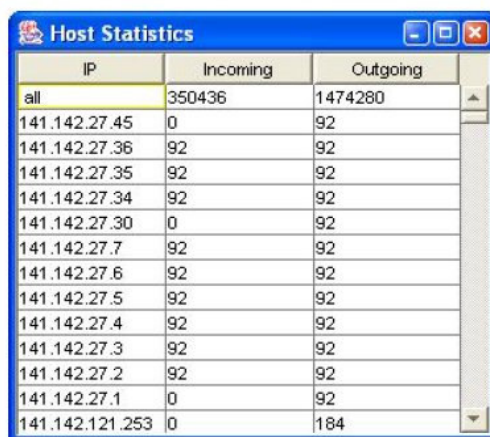
Figure 3-15 : Internal network view [Yin 2004: 30]

the centre axis still represents machines on the internal network, but the left and right axis now represents individual machines from a particular external domain.

This drill down technique does allow the viewer to investigate the external to internal domain traffic in more depth, and is useful for gleaning more insight into the nature of connections between the two domains. However, by excluding other domains, the domain view visualization loses the ability to pick up coordinated attacks from multiple domains. For example, some denial of services attack strategies compromise machines all over the internet (across multiple network domains), and then all target a specific domain. Whilst the global view would cover this, albeit with less limited detail, the narrower domain view would not.

The ‘internal network view’ (Figure 3-15) requires only two parallel axis to illustrate connection directions between hosts. As in the other views, the traffic flows are directed from left to right. The ‘host statistics view’ (Figure 3-16), is the fourth level textual drill down that allows the viewer to see detailed connection information of a selected host. Filters can also be applied to the visualization to reduce the ‘noise’, and help eliminate what is presumably innocuous traffic. They offered four filtering options, port filtering, protocol filtering, transfer rate filtering and packet size filtering.

A virus outbreak is depicted in the global view (Figure 3-18). By selecting the domain with unusually high connections, more detail and a clearer depiction of what is happening can be seen in the domain view (Figure 3-18). Drilling down further into the host statistics view (Figure 3-16), more details are provided in textual format and can be used for further investigation (i.e. note the size of many



IP	Incoming	Outgoing
all	350436	1474280
141.142.27.45	0	92
141.142.27.36	92	92
141.142.27.35	92	92
141.142.27.34	92	92
141.142.27.30	0	92
141.142.27.7	92	92
141.142.27.6	92	92
141.142.27.5	92	92
141.142.27.4	92	92
141.142.27.3	92	92
141.142.27.2	92	92
141.142.27.1	0	92
141.142.121.253	0	184

Figure 3-16 : Host statistic view - virus attack
[Yin 2004: 32]

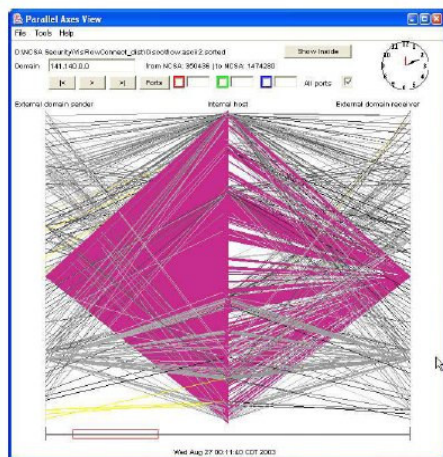


Figure 3-18 : Global view - virus attack [Yin 2004: 32]

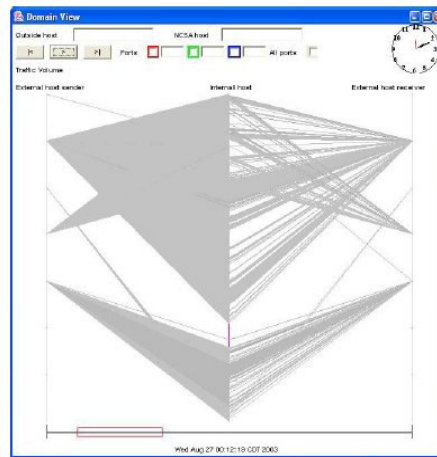


Figure 3-18 : Domain view - virus attack [Yin 2004: 32]

connections are 92).

Yin et al did not offer as yet any indication of the scalability of VisFlowConnect, but suggested that it was forthcoming. “A more detailed description of the internal implementation of VisFlowConnect, including a performance and scalability analysis, has been performed and will be published in the near future.” [Yin 2004: 28]. VisFlowConnect did offer several useful filters which could help reduce visual clutter (‘noise’). Filtering could be done according to time, source and destination hosts and ranges, protocol, port number (and range), and packet size (not to be underestimated in its usefulness). Packet size can be a characteristic of certain forms malicious traffic, i.e. a packet size of 92 (as in Figure 3-16).

3.2.6 OASC Visualizaions – Teoh at al,“Detecting Flaws and Intruders with Visual Data Analysis”

The paper by Teoh et al introduces several differing types of internet security visualizations. The first visualization is two-dimensional and involves line plots to illustrate OASCs – origin AS (autonomous agent) changes. As eloquently described by Teoh et al,

The Internet represents a set of clusters, with each cluster representing an organization’s network. These autonomous systems (ASs) manage traffic within AS clusters. To communicate between systems, routers on an AS edge use the Boarder Gateway Protocol (BGP). [Teoh 2004: 29]

Each organizational network has an IP prefix describing the network. BGP assigns each AS a unique identifier and set of IP prefixes, identifying which subnet of IP addresses correspond to hosts belonging to the AS.

An OASC event consists of the IP prefix affected, a list of ASs associated with the change (generally the prefix's new origin AS), the change's date, and the change type. A change can narrow the mask of addresses an AS already owns (a B type) or another AS owns (an H type), claim ownership of another AS's prefix (C type), or claim ownership of an unowned prefix (O type). Further classification of the last two changes depends on whether a single AS or multiple ASs claim prefix ownership. Because usually only one AS should claim ownership, multiple origin AS conflicts may indicate faults or attacks. Some OASCs are complementary: A CMS event (a C-type change from multiple ASs to a single origin AS) could correct a CSM event (a C-type change from a single AS to multiple origin ASs). Eight OASC types (OS, OM, CSM, CMS, CMM, CSS, H, and B) exist. [Teoh 2004: 29]

They extended the concept to display to a multiple wheel of separate sub displays for different origin AS changes events (2-D line visualization of OASCs (Figure 3-20). Dividing the display up into classes of OS changes helped deal visual obscurity caused by too many lines and crossover. The selected sub display appears in the middle. For given OASC event types, show on their own, they were

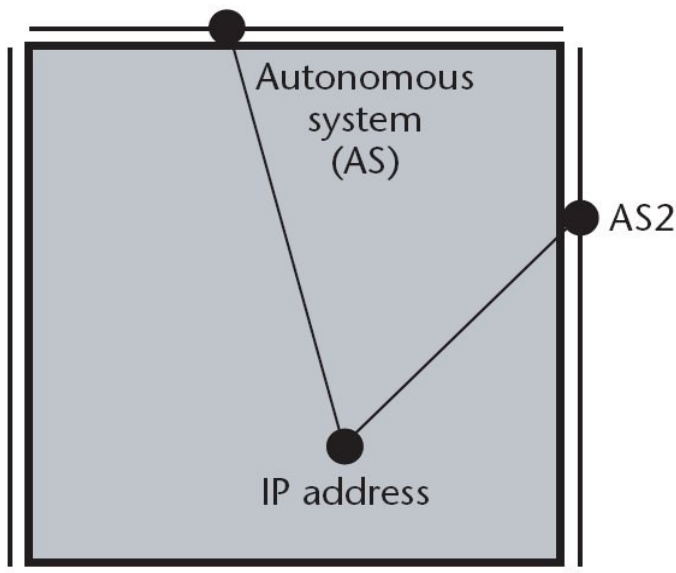


Figure 3-19: AS changes mapping scheme where two lines connect the IP from the previous AS owner to the new owner
[Teoh 2004: 30]

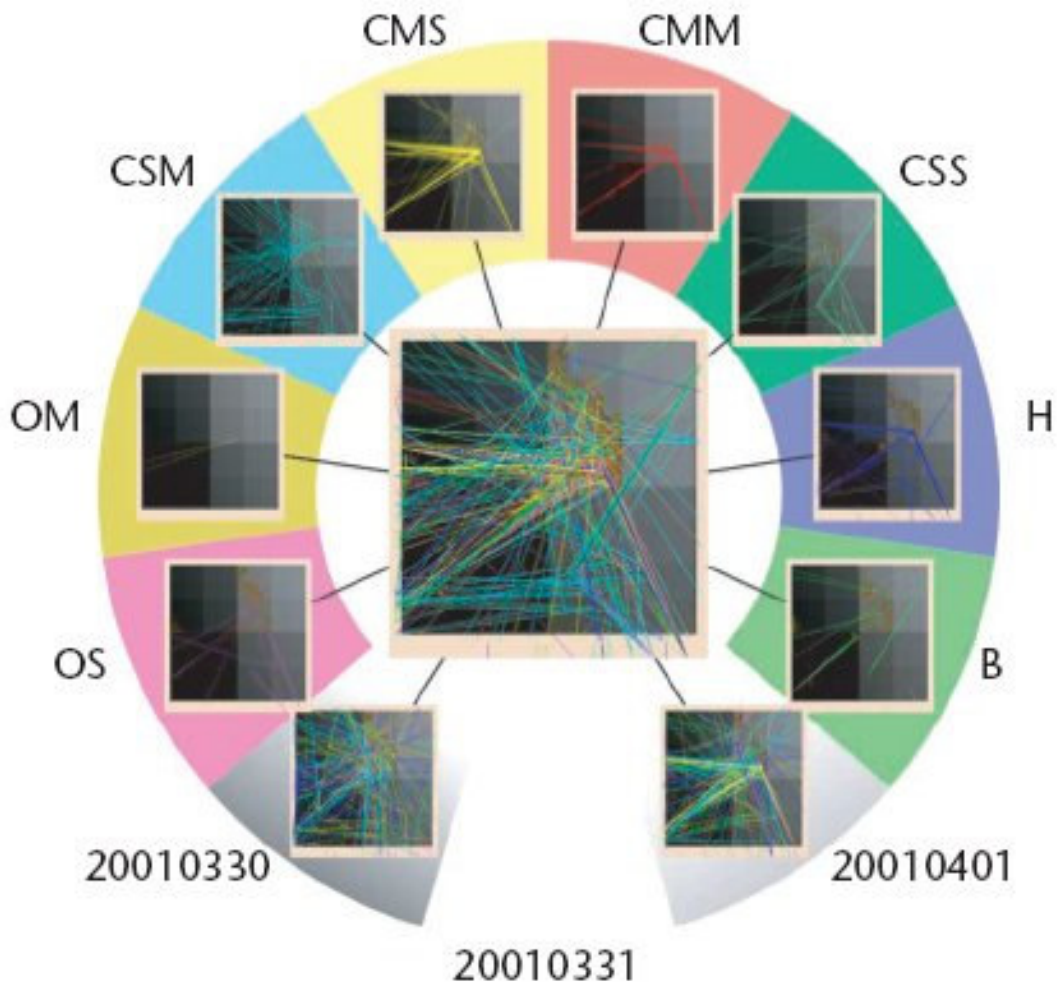


Figure 3-20 : 2-D line visualization of OASCs [Teoh 2003: 30]

able to discern normal from abnormal behaviour by noting clusters of lines, and it helped them pick out coordinated changes through the different OASCs types. In Figure 3-20, one can already note the difficulty in interpreting the visualization caused by many overlapping lines cause.

A second complimentary 3-D visualization (Figure 3-19) showed clusters of overlapping events by mapping vertical AS ranges to an IP address base of a 3-D cube. The overlapping events meant that several OASCs were associated with the same IP address. Multiple origin AS conflicts for an IP address could be a result of faults or attacks.

The suite of visualizations created by Teoh et al are considered to be abstract representations of network events, rather than concrete (as Ball et al. would put it – section 3.2.2). All the other visualizations reviewed in this survey are well suited

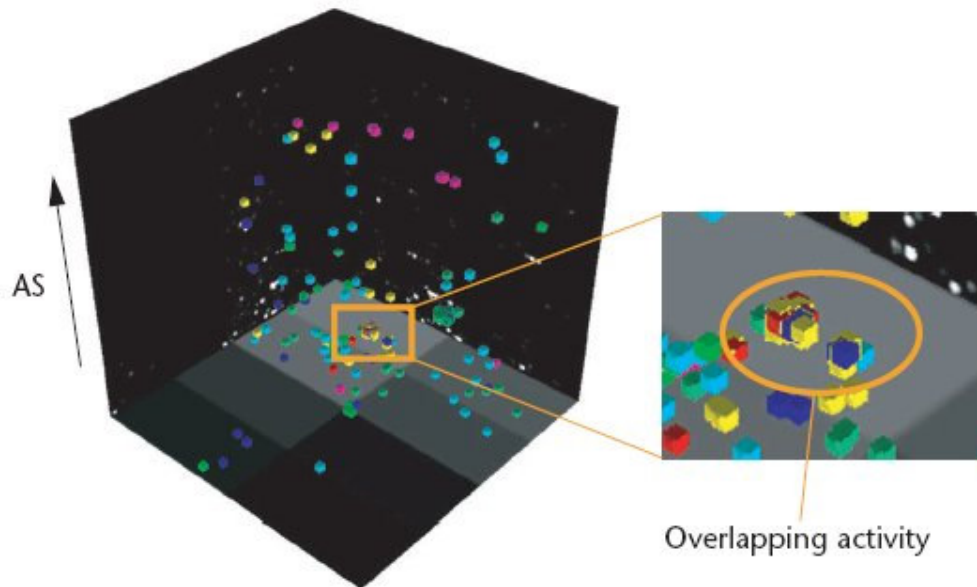


Figure 3-21: 3-D box visualization of OASCs [Teoh 2003: 30]

to internal vs. external network monitoring scenarios. Unlike the other visualizations reviewed in this survey, the OASC visualizations attempt to view occurrences on the internet as a whole.

3.3 General Review and Comparative Analysis

3.3.1 Points versus Lines as a Visual Metaphor for Representing Connections

Unlike many of other visualizations that represent connections with lines, “The Spinning Cube of Potential Doom” plots points, and can be considered a 3-D scatter-plot. One argument for using lines is that they are a better natural metaphor for representing connections and links – one can better imagine a connection as a long wire or link between two separate places. Points on the other hand are obviously far more economical in terms of spatial real-estate, and rendering performance. This point is apparent from noting the obstruction difficulties with visualizations that use line representations resulting in limitations on the number of connections they can represent without too much clutter. Furthermore, since a line entails rendering many points, the performance cost per line connection drawn is higher. Therefore, in terms of scalability, points are more promising.

While it stands to reason that more connections could be represented with the use of points, visual obstruction can still be an issue where one point occurs in front of another fully obscuring the point behind it. Lines can only obscure each other if viewed parallel, and will otherwise cross over each other avoiding fully obscuring the data behind. However, it stands to reason that in general the possibility of a line crossing over is far more often than the occurrence of one point visually obstructing another. Excessive cross-over of lines tended to be a common problem for the line-based visualizations. Too many crossed over lines, especially at small angles of incidence, make it hard to decipher and follow the line from its one end point to the other; consequently, the viewer loses track of what is connected.

A significant problem with points is that they are one-dimensional in the strict sense (if not represented by squares or spheres). In a three-dimensional environment, perceiving the depth of a point can be difficult since a point rendered near or far maintains the same size. Added to this there are no ‘side-on’ edges or hints from orientation to help decipher the exact depth of the point in space. The one-dimensionality of a point also precludes a visualization from conveying added information in terms of size. Lines on the other hand could have varied thickness to indicate traffic volume for example. A work around for these problems with points is to use small cubes or spheres as an underlying representation of a point (e.g. the 3-D OASC visualization in Figure 3-21). However, this may detract from some of the merits of using points in the first place. Visual obstruction is more likely to occur when points become bigger than one pixel and scalability reduced as a result. Furthermore, whilst rendering points is more efficient than rendering lines, rendering cubes or spheres is less efficient than rendering lines.

Making lines thicker entails drawing something other than a line, especially if rendering in three dimensions. In the 3-D case, lines of varied thickness can be represented by long narrow cylinders or rectangular polygons (and therefore not any more efficient than sized cubes or spheres). An extension to this idea involves a long thin conic section between two points. The radius of at a particular end-point could be sized according to traffic sent along the connection by the host at that end. This would reflect not only the volume of the traffic sent across the connection, but also the ratio of contributions from each host.

An obvious and important thing to realise about using thickness of lines, or any other size based metaphors is that it can compromise scalability. Colour on the

other hand is very useful for extending the data dimensionality that visualization can handle, and is more likely to improve scalability than diminish it.

3.3.2 Use of transparency and colour

In all the visualizations, the use of colour to encode information is notable, though some use colour to a lesser or greater degree. The 2-D OASC visualization (Figure 3-20) and the ‘Space Shield’ Visualization (Figure 3-11) utilize different colours for the connection lines to encode extra information. VISUAL (Figure 3-3), and Scanmap3D (Figure 3-7) on the other hand use only one colour for connection lines. VisFlowConnect has selective use of colour for highlighting connection lines (Figure 3-3 and Figure 3-18). Ports are commonly encoded by colour as the “Spinning Cube of Potential Doom”, VISUAL, and ‘Space-Shield’ use varied colours to indicate port numbers.

VISAUL and Scanmap3D make notable use of transparency to help alleviate occlusion problems (section 3.2.2 and 3.2.3 respectively), and ‘Space Shield’ uses transparency to convey the separable hemispheric regions as used in its visual metaphor. VISAUL employs shadows to convey historic information about past connections. A similar idea for the “Spinning Cube of Potential Doom” would be to convey chronological history about the occurrence of connection attempts by making older connections more transparent (section 3.2.1). This would enhance the views ability to pick up patterns that occur over time. Another facet to this idea is the dynamic varying of a time window for which the connection history as such lingers on in the display (the control for this could be implemented as a slide-bar for example). VisFlowConnect (3.2.5) exhibited this feature of a variable time window that controlled how far back connections were to be represented, but to my knowledge did not go further to use colour or transparency to visually encode chronological information.

3.3.3 Animation and Real-Time playback

The “Spinning Cube of Potential Doom”, VisFlowConnect and the ‘Space Shield’ all provided animated playback of recorded data according to conventional time. Scanmap3D on the other hand had non-conventional animation based on packet events and custom delays between packets. This kept the chronological order of events, but animated them in a disjoint manner without directly relating the events

to the actual time they occurred; something I consider an undesirable trait because it would not allow the viewer to accurately perceive the timing of events (3.2.3). The ‘Space Shield’ and VisFlowConnect also had the ability to do real-time monitoring; a feature noted as desirable extensions for “Spinning Cube of Potential Doom” and VISAUL. The ‘Space Shield’ had the ability to scale and compress time, allowing viewers to review traffic in a ‘quick search’ fashion (3.2.4). This was noted by Stephen Lau as a desirable extension for the “Spinning Cube of Potential Doom” [Lau 2003].

3.3.4 Filtering

Scanmap3D, VisFlowConnect, and VISUAL all offered interactive filtering methods. Filtering was done according to various properties of network traffic, namely time, source and destination IP addresses and ranges, protocol, port number and range, and packet size (not to be underestimated in its usefulness). Filtering according to the type of packet for a given protocol can also be useful, i.e. the “Spinning cube of Potential Doom” relies on TCP handshakes to discern connection attempts. Filtering by type ICMP packet would be another example. Some visual filtering techniques can also be employed whereby textual information only becomes evident and substantial upon zooming close in – ‘proximity based detail’. In this manner, the overview of a display remains uncluttered.

Filtering techniques are very useful for reducing noise, performing directed investigation and isolating suspect events; therefore all the more need for the filtering implementation to be interactive and dynamic. Although filtering can improve visual obstruction problems under large data display loads, it is precarious to depend on filtering for scalability, since this may increase the possibility of false negatives – the failure to alert the viewer of malicious activity in the network. If any filtering is done by default, it ought to be done with confidence that only innocuous traffic will be filtered.

3.3.5 Drill downs and details on demand

The need for drilling down into the data is stressed by Fisk et al; to quote them, “Earlier tests with real use of the system demonstrated the absolute necessity of drill-down capabilities” [Fisk 2003: 8]. Scanmap3D and the ‘Space Shield’ have similar methods of supplying data on demand within the visualization. Aside from

proximity based detail (a visual mechanism of drilling down by zooming in), both visualizations drew transparent planes with textual information. As discussed in their respective sections (3.2.4 and 3.2.3), the overlay method of the ‘Space Shield’ by in large was preferable to Scanmap3D’s method of drawing the pane within the 3-D visual world of the visualization. VISUAL and VisFlowConnect provided textual drill down details in separate windows. As an advantage this would make cutting and pasting information possible. A disadvantage is that one cannot easily keep an eye on both the visualization and the textual data at the same time. The overlay method of the ‘Space Shield’ would be ideal for real-time monitoring where one would preferably be able to keep an eye on the happenings inside the visualization whilst looking at the textual data.

3.4 Chapter Summary and Conclusion

This survey has provided some brief background information on the challenges faced in network security, critically reviewed and compared six recent network visualizations, as well as summarised and consolidated some important visualization concepts encountered in the reviews (this section).

A major challenge of monitoring networks is the volume and complexity of network data generated; and as network become faster, this challenge will grow. At present, majority of visualizations appear to suffer from scalability problems. Of the visualizations reviewed in this survey, the most promising scalable visualization was the “Spinning Cube of Potential Doom” – a simplistic demonstration suggested that it would be capable of representing 100,000 connections or perhaps a little more. Its advantage in this regard came from the use of points as a visual metaphor for representing connections, whereas majority of the other visualizations employed connection lines.

Another particular strength of the “Spinning Cube of Doom” was its ability to spatially relate and associate port information with connections. Only two of the reviewed visualizations attempted a spatial representation, namely VISAUL and Scanmap3D. Visual did so in external host markers which were fairly small squares – a bit of a squeeze for the information (3.2.2). Scanmap3D’s port stacks did not consistently position ports at regular heights (as detailed in 3.2.3). Therefore in both cases port scans would show up as less obvious by contrast to how they would appear in “The Spinning Cube of Potential Doom”.

In closing this, one may note that the “Spinning Cube of Potential Doom” lacks some of the advanced features of other visualizations, such as variable replay rate, variable time window, real-time playback, drill down capabilities, and interactive filtering – all feasible extensions for the visualization.

Chapter 4

Design and Implementation

In accordance with the proposed thesis and investigative approach (as per Chapter 1), the design goal of this project is to develop an animated 3-D scatter plot visual display of network traffic, and determine its effectiveness in detecting network anomalies indicative of intrusive activity. In taking the lessons learnt from chapter 3, this chapter describes the conceptual design, system design, and implementation. It includes descriptions of visualization features, the graphical user interface, and performance considerations for underlying system components.

The review of contemporary work in Chapter 3 serves as a comparative benchmark, and identifies scalability as a key concern, suggesting that visualizing large amounts of traffic is to be problematic for many visualizations. In this regard, the review identifies the 3-D scatter-plot concept behind the ‘Spinning Cube of Potential Doom’ to have promising scalability. As is detailed in Chapter 3, the analysis in Section 3.3.1 reasons that using points should offer superior scalability and performance in comparison to lines.

At the time of commencing this project, the source code for the Spinning Cube of Potential Doom was not publicly available. ‘InetVis’, a shortened title for Internet Visualization, is a reimplementaion of the animated 3-D scatter-plot concept, based largely on descriptions found in two of Lau’s articles [Lau 2003, Lau 2004].

4.1 Conceptual Design and Feature Specification

The intended purpose for developing InetVis is to build a visualization that confirms and demonstrates the effective use of an animated 3-D scatter-plot, as well as evaluating features that enhance the concept. Taking the Spinning Cube of Potential Doom as its primary basis, InetVis modifies and extends Lau’s original concept with several ideas, some gleaned from the review of related works (as covered in Chapter 3). Each design goal and idea is motivated in the sub-sections that follow, and where appropriate, due credit is given to its origin.

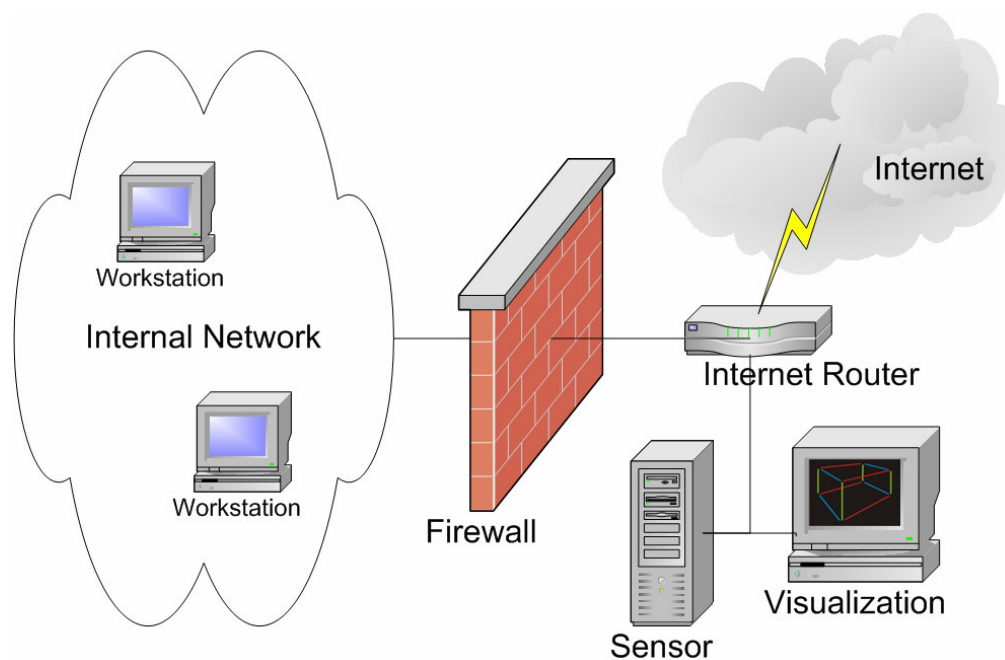


Figure 4-1: Typical network-monitoring scenario

4.1.1 The Perspective of an Internal Domain versus the Internet

The intended application of the network security visualization is to monitor and review traffic traversing a network boundary. Figure 4-1 depicts a typical network-monitoring scenario. An internal network, also called a ‘home’ network by some authors, is a subnet that comprises of a logical IP address range that an organization wishes to protect. As a common convention, a firewall establishes a boundary that protects the internal network from the Internet at large by filtering traffic. The Internet is comprised of the full IP address range, and is subdivided into network domains, where the perspective considers every domain other than the internal domain as external.

A ‘sensor’ outside the firewall facing unfiltered Internet traffic will be in a position to capture unfiltered Internet traffic. This provides a perspective of all intrusive scanning activity that a firewall is intended to block. To perform real-time live traffic monitoring, the visualization application runs directly on the sensor. For review of (non-live) recorded traffic, the visualization can run on any system with capture files and a supporting operating platform.

4.1.2 Points as a Visual Metaphor for Network Events

As is the case for the ‘Spinning Cube of Potential Doom’, the visualization plots points within a cubic reference frame, and is effectively a 3-D scatter plot. The simple points are a metaphor to represent network events. As dealt with at length in 3.3.1, although lines are a suggestive metaphor for connection and points are not, points minimize the use of display space and have faster rendering performance. This concept should therefore be more scalable in comparison to visualizations that make use of lines to represent network events.

The term ‘network event’ is intentionally vague for design purposes, and relates an intention to represent a variable range of network events (in future extensions). Packets are elementary network events, but complex events can also be represented as points in a 3-D scatter-plot scheme. For example, TCP connections, or series of packets modelled as connection flows, could suitably be represented as a point that lasts the duration of the connection.

The current implementation of InetVis only represents network packets. The visualization supports IP, TCP, UDP and ICMP, protocols. They cover a large proportion of Internet traffic, and it is possible to perform intrusive network probing with all of them. Lau’s work is therefore extended by catering for UDP and ICMP traffic in addition to TCP traffic. However, although a broader data source can be visualized at once, including connectionless protocols (UDP and ICMP) complicates the ability to distinguish between successful and unsuccessful connections.

Due to the addition of UDP and ICMP, InetVis omits the ability to distinguish between successful and successful connections. Another reason for this omission is that its chosen data source, LibPcap, only offers elementary packet capture (as is discussed later in Section 4.2.4.1).

4.1.3 3-D Plotting Scheme

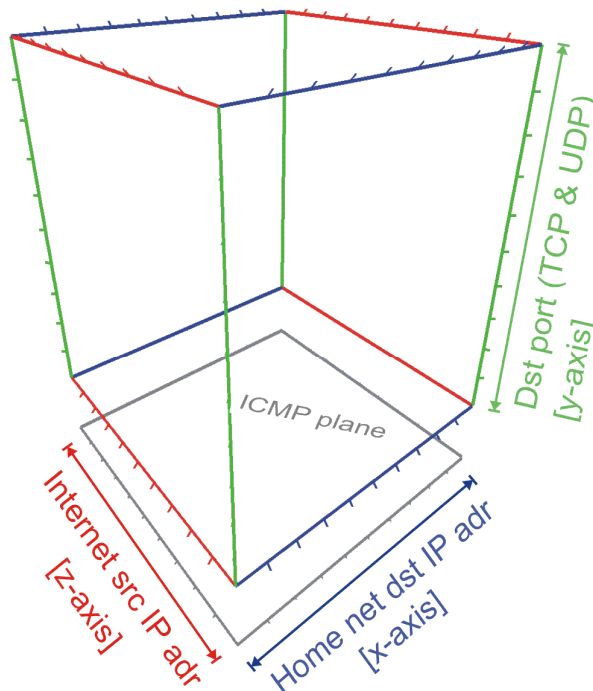


Figure 4-2 InetVis plotting scheme

Figure 4-2 illustrates the plotting scheme and reference frame. It is nearly identical to the scheme used by the ‘Spinning Cube of Potential Doom’, but with some subtle modifications. The scheme associates red with danger and so Internet addresses relate to the red axis while the internal network addresses relate to the blue axis (blue is considered a more calming colour than red). This is contrary to the colour-scheme of the Spinning Cube of Potential Doom, where the blue axis related the Internet address and the red axis related the internal network.

The InetVis plotting scheme is extended to cater for UDP and ICMP traffic in addition to TCP traffic. As with TCP traffic, the visualization also plots UDP traffic within the cube. The ICMP protocol has no port attributes, and one possible choice would be to plot ICMP packets at the bottom of the cube to port 0; use of port 0 for TCP and UDP is not a legitimate port assignment. This solution is unfavourable, because well known services operate on lower ports. Port ranges of 65535 values scale down a graphical distance of a few hundred pixels. In closing this line of reasoning, the concern is that with a wealth of TCP and UDP traffic below port 100 could obscure the ICMP traffic, were it plotted at the bottom of the cube. Furthermore, some exploits use malformed TCP and UDP packets that target port 0, and a viewer might falsely think this be ICMP traffic. To distinguish ICMP

traffic from TCP and UDP traffic, the visualization plots ICMP traffic in a separate plane below the cube.

Defined precisely, the scheme plots points according to three perpendicular axes as follows:

- **Blue x-axis:** The position of a point with reference to the x-axis direction relates the destination IP address of a packet. The plot scales out the destination address data along this direction according to a specified IP address range (a subnet). The scale will vary according to the number of addresses that the internal network IP range specifies.
- **Red z-axis:** The position of a point with reference to the z-axis direction relates the source IP address of a packet. The space within the cube along this direction relates the entire Internet address range of IPv4 (0.0.0.0-255.255.255.255). Therefore, the spaces between Internet addresses are far more condensed than the spaces between internal network addresses.
- **Green y-axis:** The position of a point with reference to the y-axis direction relates the destination port of a packet, and is applicable for TCP and UDP traffic.
- **Grey ICMP plane:** Points positioned in the ICMP plane have no port attribute, and therefore all lie on the ICMP plane below the cube. Their positioning in the x-axis direction and y-axis direction is analogous to the plot used for TCP and UDP.

With this plotting scheme, the blue x-axis reflects which hosts are the targets of incoming Internet traffic, while the red z-axis relates the origin of the traffic, and the green y-axis conveys the destination port used. This should exhibit port scanning on a single target host as a vertical line, and host scanning on a single destination port across addresses as a horizontal line (this is usually for a attempting an exploit that uses a particular port). Horizontal scan lines can also result when network probing is conducted with ICMP, as for example, the address range is mapped out to discover hosts. These line orientations as vertical and horizontal are with reference to a standard frontal view (down the z-axis), and would obviously change where the perspective of the visualization changes. These visual metaphors for the respective scanning activities are analogous to the definitions given earlier, in Chapter 2, Section 2.2.2 – Classification of Intrusive Network Probing Techniques.

4.1.4 Reference Frame

The reference frame includes an optional bounding box that bounds all the data. The intention of this is to assist the viewer with locating a point in 3-D space when projected on a 2-D display medium. Otherwise, the viewer can reduce the frame to its primary axes only, which gives an indication of the orientation of the cube, supposing it were back to front after navigation.

The viewer is also able to add or remove reference markers along the axis, and selectively display grids of adjustable transparency. The seven grids are comprised of the six faces of the cube and the ICMP plane, and the user can display them independently. The number of partitions for grids and markers is also variable, and the user can specify independent values for each direction (x, y, and z respectively). The user can also hide reference frame components to allow for an unobstructed view of the data.

The purpose of these features is to provide a flexible reference frame and assist the viewer in interpreting the value of attributes. The Spinning Cube of Potential Doom does offer some transparent grids, but adjustable markers and flexible control of all reference frame components is presumably an extension.

4.1.5 Colour Schemes

Colouring by destination port will create a rainbow effect (as is done in the Spinning Cube of Potential Doom). This reinforces the spatial relation of destination port, and does not extend the dimensionality of the visualization. Section 4.1.7 describes how InetVis animates the occurrence of packets according to timestamps. Colouring by packet age can compliment the animated pattern in which the packets occur, giving an instantaneous graphical representation of time ordering – this also reinforces the representation of time, without requiring an added dimension. A few alternative attributes for extending the dimensionality of InetVis by the use of colour are the source port, protocol type, and packet size.

With the objective of observing the characteristics of network scanning, colouring by source port may arise in interesting colour patterns that relate the sequence of ports employed by external hosts when scanning. The viewer will be able to notice whether a scanning method uses only one port, or randomly chooses different ports, and how many different ports are used. Colouring by protocol type

could reveal if scans utilize more than one protocol. Colouring by packet size can also be useful in revealing suspicious patterns that result from scans that use packets of the same size. Currently, the implementation of InetVis only caters for colouring by destination port, source port, or protocol.

4.1.6 Variable Point Size

As an added feature, the viewer can adjust the size of all the points at once. In some instances, larger point sizes can suit small amounts of data and make patterns more visible. For larger amounts of data, smaller point size can help alleviate visual congestion that clutters the display. Given that a primary design goal of InetVis is scalability, point size adjustment is kept within to a range of relatively small values.

Size can be used to extend the dimensionality of the data represented (e.g. size could relate packet size), as is noted in the review of some visualizations. However, for points this is not a suitable idea because points are one-dimensional, lack geometry, and poorly reflect variations in size (unless the variations are large and discrete). Furthermore, in a diminishing perspective view, the size of a point will relate its depth. Therefore, the visualization should maintain the same conceptual size for all points; otherwise, a mix of varied sizes and diminishing perspective effects will distort the spatial locality of points.

4.1.7 Animation and Time Scaling

The user can view network traffic in two ways: monitoring a live network interface for real-time traffic analysis, or retrospectively reviewing traffic previously captured and recorded. The display of events is in chronological order as they appear according to the time of their occurrence. For the replay of captured files, the replay rate is adjustable, allowing the user to speed up or slow down the animation of events according to time. Obviously, this is not possible for the live mode, where the timing is fixed to real-time (1.0x).

The maximum replay rate for InetVis is 86400x (1 day per second) and the minimum is 0.001x (1 millisecond per second) – it is presumed that any faster or slower would not be of practical use. Like the Space Shield [Fisk 2003], the ability to scale time allows viewers to review traffic in a ‘quick search’ fashion, although

the Space Shield only offered a maximum rate of 60x. The viewer can skip past periods of inactivity while searching for glimpses of interesting events where anomalous traffic flashes by. When something of interest is noticed, a seek bar can be used to jump back to an appropriate replay position, and the replay speed can then be slowed down for meticulous inspection.

Stephen Lau noted this as a desirable extension for the Spinning Cube of Potential Doom [Lau 2003]. He also suggested extending the concept's features to be capable of displaying real-time traffic capture from a live network interface.

4.1.8 Time Window

The idea of a variable 'time window' is adopted from VisFlowConnect [Yin 2000]. A time window specifies the period the visualization will continue to display an event after its occurrence. This effectively amounts to a form of time filtering, as it excludes events that fall outside the time window. It is useful because a narrow time window will reveal rapid scans while a larger window allows for the observation of slower scans.

This feature also allows the user to manipulate how much data is under review, as a larger the time window will display a greater number of events. Conversely, narrowing the time window can relieve situations of excessive visual clutter. That said, a wider time window is preferable for observing traffic trends and patterns that occur over long periods. With this in mind, the user can balance the amount of data to comfortable levels so that subtler patterns remain visible when engulfed in other data.

4.1.9 Filtering Techniques

The Spinning Cube of Potential Doom did not implement a data filter. Filtering is a valuable tool to focus on phenomena of interest. By applying a filter that excludes uninteresting traffic, it allows the user a clearer, isolated view of events they wish to investigate. Filtering features should be highly flexible, allowing inclusive and exclusive capabilities, and should work in unison. For example, if multiple scanning patterns appear similar, but originate from different sources, the ability to include a list of disjoint address ranges, port ranges, and protocol type would help confirm the similarity of the scans. BPF (Berkeley Packet Filter)

expressions offer this flexibility at the expense of understanding the technical syntax involved in their use [LibPCap, McCanne 1992].

The visualization should provide interactive processing of filters. This allows the viewer to experiment with filtering by applying a filter expression, assessing its effects, and adjusting it to improve the view of the phenomenon under investigation. Pre-processing the data before visualization is ill advised. As Fisk *et al* mention, “*making automated, error-prone filtering decisions like traditional intrusion detection systems*”, could potentially exclude unanticipated patterns of interest [Fisk 2003: 2].

4.1.10 Navigation and Exploration

Similar to the Spinning Cube of Potential Doom, the Space Shield visualization [Fisk 2003], and Scanmap3D, navigation in InetVis allow the viewer to immerse himself, or herself, within the visual environment. Fisk *et al* suggest that an immersive environment allows viewers to explore and perceive more data than would ordinarily be possible with a static view [Fisk 2003: 2].

The use of graphical controls would consume display space, and instead, mouse movement directly controls the perspective of the visualization. The ability to rotate, translate (move), and zoom (in and out), facilitates closer investigation of interesting phenomena. Navigation is constricted so that the user cannot lose sight of the visualization.

4.2 System Design and Implementation

The primary design objective of InetVis is to build a high performance, scalable tool suited to performing network traffic analysis and identifying probing activity. Its second role is to be an experimental visual tool for evaluating visualization concept, and the design intent is general and flexible enough to accommodate extensions. With flexibility and extensibility as a secondary ambition, the initial design of the system takes a modular approach. However, in the pursuit of high performance and the pressure of rapid system development, objects directly manipulate or rely on other objects, somewhat compromising the goal of modularization.

Performance considerations can conflict with an object orientated design methodology. Objected orientated programming methods encourage encapsulation, abstraction, and well defined interfaces that protect access to object attributes (members), but cause object interaction to be more time-consuming. In some critical performance cases, optimising code requires foregoing an object-orientated approach in favour of avoiding excessive function calling, and allowing public access to variables so that other objects may directly manipulate them. Examples of performance-critical code segments are graphical rendering loops, and data extraction processing loops. These can involve numerous iterations, as is the case when plotting several hundred thousand points every frame at several frames per second.

4.2.1 Component Model

A component model serves to establish logical separation, and the intent is to allow for generality and adaptability for future extensions to the application. The system is comprised of distinct components (objects) as follows:

- A **Visualization Pane** is the graphical and interactive viewing window that draws and plots the data.
- A **Control Panel**, interfaces between the user and the Data Processing by providing controls to manipulate playback and filtering.
- A **Data Processing** component interprets and maps the data to its visual metaphors, abstracting this task from the graphical cube visualization.
- A **Data Extractor** component interfaces and abstracts the task of managing an underlying packet capture library, thereby providing services to the Data Processing component.
- A **Packet Event** is a simple underlying representation structure that binds packet header attributes and their graphical point representation.
- A number of support class components, namely the **Packet Header** abstract definition class, and static classes, **Time Utility**, **Plotter**, provide functions and services to the other components.

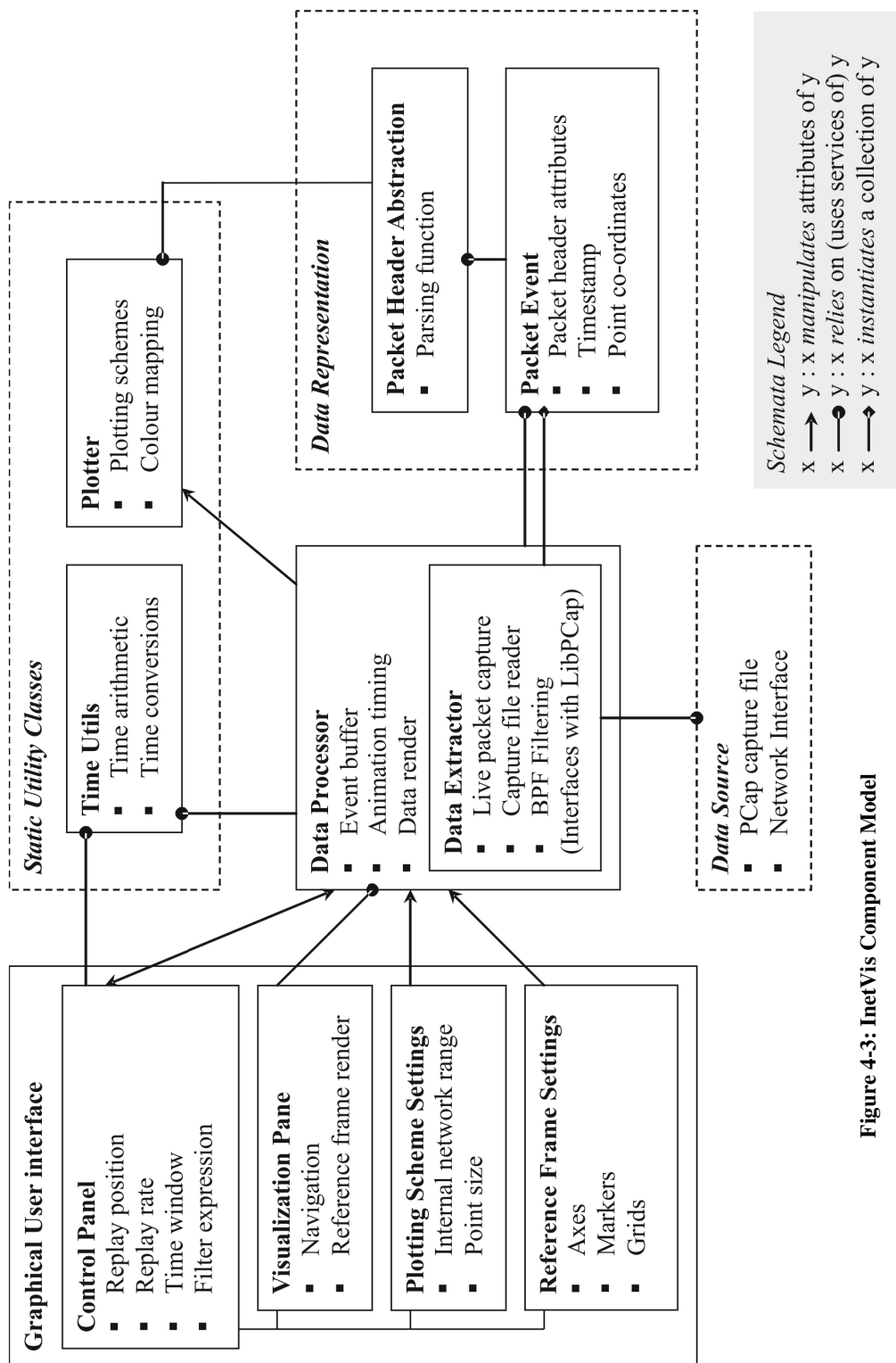


Figure 4-3: InetVis Component Model

Figure 4-3 provides a depiction of the object relations depicting the interactions and dependencies between components. This Figure provides an overview of how the component model meshes. The bullets under each component state its core roles in the system.

4.2.2 Development Platform and Software Libraries

InetVis makes use of the OpenGL graphics library and the LibPCap packet capture library. OpenGL provides high performance 3-D accelerated hardware rendering which allows graphical applications to be interactive. LibPCap performs efficient interfacing with an operating system kernel to retrieve packets from a network interface. Well known open source applications such as TCPDump, Ethereal, and Snort, utilize the LibPCap library to capture network traffic and support the LibPCap file format. This can be useful for comparing results in the visualization with the results offered by these tools. OpenGL and LibPCap implementations are available in the C programming language.

The chosen programming language for development is C++, which extends the procedural paradigm of C to offer an object orientated programming features. C++ is backward compatible with C, allowing it to link with the OpenGL and LibPCap libraries. Furthermore, it is possible to compile optimised C++ code into native code that can run directly on a given platform. This gives C++ performance advantages over interpreted languages like Java. As mentioned in Chapter 3, Section 3.2.3 the poor performance of Scanmap3D is, in part, attributable to its choice of Java as an implementation language.

The C++ STL (Standard Template Library) provides efficient abstract data types. The graphical user interface (GUI) is created with the Qt C++ application framework, which readily supports integrating an OpenGL display container into a window.

All of the above development choices accord with the open source software development philosophy. OpenGL, LibPCap (called WinPCap), the C++ STL and Qt can all run in cross-platform environments, namely Unix-based platforms (i.e. Linux), and Microsoft Windows platforms. With appropriate configuration and linkage to the libraries, the source code of InetVis should port across from its Linux implementation to a Microsoft Windows platform. Therefore, all the software used in this project was freely available at no cost.

4.2.3 User Interface

The user interface is modularized into functional components that form separate windows (as shown previously in Figure 2-1), which are all accessible from a main control panel (via a menu); the control panel itself houses replay controls. Keeping the control panel separate from the visualization pane is an idea taken from scanmap3D. This allows a maximal view and size of the visualization for full utilization of the screen space. To access controls, the user can simply view them over the display. The control panel is small to avoid obstructing the display in the background (given the use of only one screen).

Ideally, this user interface setup suits a multiple screen environment. The viewer can control the visualization on one screen whilst keeping a maximized and unobstructed view of the display on another display device. For example, this is convenient for presentation scenarios where the presenter can assign the visualization pane to a digital projector whilst controlling the display from another screen. One compromise is that the separation of controls from the display diminishes the correlation between making changes, and the resultant effects the changes have on the visualization.

4.2.3.1 Control Panel

The menu allows the user to open a capture file, set the replay mode (live monitoring or replay), control playback, and provides access to all other windows via the view menu option. Its window contents consist of four components to manipulate playback and filtering: the replay position, the replay rate (time scale), the time window, and the BPF filter expression. Slide bar controls allow quick adjustment of the values whilst edit boxes display current values and are editable to facilitate exact specification of values.

4.2.3.2 Visualization Pane and Navigation

The visualization plane is solely dedicated for display, and offers no graphical controls. Navigational control is provided via the mouse. With the left mouse button pressed on the display, the user can rotate the view in a way analogous to mouse movement. Similarly, the user can translate (move) up, down, left, and right with the right button pressed, and zoom in and out with the middle button pressed.

4.2.4 Data Processing, Extraction and Representation

A Data Processor object forms the core backend for InetVis. It performs the following services:

- Controls and receives signals from an animation timer.
- Maintains and updates timing information such as the current replay position, replay rate, and time window.
- Manages data extraction through a data extraction child object and stores packet event data in buffer.

The Data Extraction object, along with classes and structure definitions that represent the data, were subject to considerable efforts to optimise their underlying performance in the system. As specified earlier in Section 4.1.7, rapidly sped up replay is a design specification. Data through-put at high replay rates (such as 86400x) can result in significant real-time processing requirements. The sections that follow discuss the functioning of the components identified as performance critical sections of the system.

4.2.4.1 Data Extraction – The LibPCap Library

The Spinning Cube of Potential Doom leverages information from Bro-IDS logs [Bro]. Presumably, the Bro NIDS identifies successful and unsuccessful TCP connection information and logs of this processed information that simplifies the task of visualizing the information. One drawback is that the visualization is limited to using Bro logs as its only data source, and presumably cannot view data captured by other popular network tools (i.e. TCPDump, Ethereal, and Snort). It also lacks the capability to display live traffic. The Bro-IDS is also Unix-based, and has yet to be ported to run on Microsoft Windows.

The LibPCap library facilitates efficient low-level packet capture from a network interface, and can log packet captures to file, as well as playing back the capture file. This makes LibPCap a suitable choice for designing a visualization that is capable of both live network monitoring and recorded log file replay.

The LibPCap library delivers packet capture data to an application via a callback function. The packet capture data includes header information, not just the packet payload. Via passing parameters by reference, the callback function returns a structure that holds raw packet capture data as a byte string and includes a timestamp and packet length value. From the callback function, a Packet Event

object is constructed. It holds raw packet data, a packet structure to reference the data, the packet length, the time of occurrence, and a graphical element that represents the packet according to a plotting scheme and colour scheme.

4.2.4.2 Packet Representation

The Packet Event object contains the timestamp, length and a ‘raw’ packet capture data as a byte string, and an abstract OSI Packet structure for rapid access to the packet capture data. The OSI Packet structure is an abstraction and collection of respective protocol headers according to the OSI (Open System Interconnection) protocol stack [Tanenbaum 2003]. For InetVis purposes, three OSI layers are of concern follow in the list below, with corresponding protocols:

- Layer 4: Transport (TCP, UDP)
- Layer 3: Network (IP, ICMP)
- Layer 2: Data-Link (Ethernet)

The OSI Packet structure encapsulates defined header structures for Ethernet, IP, ICMP, TCP and UDP packet headers. It provides direct access to the packet data attributes of respective protocol headers, as is detailed in due course, supports performance considerations.

4.2.4.3 Raw Packet Data Parsing

For the construction of a Packet Event object, a single parse will set initialize OSI Packet structure and respective header structures references to directly reference header attributes within the raw packet capture data. As an example of its use, this mechanism improves the efficiency of re-plotting points by avoiding the need to re-parse the raw packet data, as direct access via the header structures is possible.

The design of the parsing mechanism is efficient since capture files can be several hundred megabyte large. As the next section explains, a buffer only processes sufficient information for the current replay position and time window. If the replay position is adjusted or time window increased, a substantial amount of data would have to be extracted from file and parsed. The display needs to wait for this to finish before updating. Therefore, an efficient parsing mechanism improves these delays.

Table 4-1: Ethernet Header

Destination	Source	Length/Type	Data & Pad	FCS
6 bytes	6 bytes	2 bytes	46-1500 bytes	4 bytes

The parser simply casts the first few bytes to an Ethernet Header structure (Table 4-1), and assigns direct references to the attributes of the header. Without having to traverse the packet data byte-string, the Ethernet Header structure references the 12th byte in the header, which points to the Length/Type field. From the code in the Length/Type field, the next encapsulated header can be determined.

If there Length/Type field contains an IPv4 header code (indicated by hex value of 0x0800), then the same type casting method is applied to cast an IPv4 header upon the raw packet data. This initializes the IP Header structure and its Protocol field informs the parser which type of header to expect next, so that the same cast procedure can be repeated for the next encapsulated data (either TCP, UDP or ICMP).

While parsing the raw packet data to construct the Packet Event, the parsing function heeds the plotting and colour schemes by utilizing static Plotter functions to construct a Point and its position and colour from the packet data. The Points are then and ready for rendering to the display.

4.2.4.4 Packet Event Buffering

The Data Processor only buffers a necessary portion of packet capture data in memory because large capture files sizes would result in excessive memory consumption. The Data Processor, controls extraction and buffers Packet Events according to the replay position and time window, which includes some pre-buffering (buffer-ahead). Since traffic can occur in a ‘bursty’ fashion, a pre-buffer guards against buffer under-runs where data processing demands spike, and is intended to keep annihilation smooth.

The buffer-ahead caters for a multithreaded solution, where the application employs threads to separate data extraction into the buffer from rendering data out to display. However, this approach would require implementing a mutex (mutual exclusion control) for the buffer to avoid concurrent access issues. The system would be penalized by an added performance cost of locking and unlocking the

mutex during buffer access and writes. Since the application is not multi-threaded for this reason, the buffer-ahead future is superfluous. As a compromise, sudden spikes may cause momentary frame rate lag.

A deque (double ended queue) was chosen as an underlying data structure for the Packet Event buffer. The STL implementation of the buffer allows constant time cost ($O(1)$) insertion at the beginning and ends of the queue. Insertion anywhere within the deque other than at the ends incurs a linear cost ($O(n)$).

4.2.4.5 Event Buffering and Animation

As the replay position and time window shifts with animation, the buffer constantly receives new input at the front of the deque and removal of old events at the end. This process routinely maintains the chronological order of events in the buffer. During animation and updating the buffer, the Data Processor only compares the timestamp of Packet Events at the beginning and end of the buffer to current values for the replay position and the end of the time window respectively. As is now obvious, this update procedure complements the choice of a deque data structure, since updating is independent of the buffer's size, and only dependent on the number of events that the Data Processor must add or remove.

4.2.5 Rendering and Timing

4.2.5.1 Animation Timing

25 times a second, a timer object signals the Data Processor, which then updates the buffer and renders the data to the visualization pane. The Data Processor advances the replay position and time window frame according to a time factor determined by the replay rate.

4.2.5.2 Graphical Rendering

The OpenGL graphics engine is a state machine. Changing states incurs a performance cost. Therefore, minimizing the number of state changes improves performance. The OpenGL call to specify point rendering is called once for rendering all points within the time window. Conversely, the majority of colour mapping schemes have no correlation with time, and so every point requires a colour mode call.

Given that the visualization renders numerous points of varied colour every frame, ordering the Packet event buffer by colour is a consideration. If there are a limited number of discrete colours, the ordering by colour can reduce OpenGL state changes. However, if points are coloured by a continuous colour gradation, this results in many colours, nullifying the performance advantage of ordering by colour.

Suppose that a limited indexed mode of a few colours is used to colour ports. Even if an alternate STL container is chosen, such as a Map or Hash which could use colour as a key to order events, other performance compromises would counterbalance the advantage gained by reducing colour state changes. The buffer is then no longer be ordered by the time of events. As the buffer is updated, every single event in the buffer has to be evaluated to check if it is still within the scope of the time window, or else removed. At best this entails a linear time performance cost ($O(n)$) dependant on the number of elements in the buffer.

4.3 Chapter Summary and Conclusion

This Chapter offers a concept and feature specification motivated by the evaluation of other network visualizations in Chapter 3 – Related Work. InetVis implements several extensions and modifications to the 3-D scatter-plot concept originally demonstrated by the Spinning Cube of Potential Doom. A summary of the modifications and enhancements to the concept is as follows:

- InetVis visualizes packets, not TCP connection attempts.
- UDP and ICMP traffic is supported in addition to TCP traffic
- The reference frame for InetVis includes an ICMP plane, and swaps axes colour to red (danger) for the Internet and blue for the internal network.
- Flexible reference frame controls, such as adjustable markers and grids, add to the viewer's ability to approximate the attributes of the data visualized.
- More colour schemes are offered, namely, by source port, by protocol, and by packet size. This is over and above colouring by destination port.
- Variable playback rates can slow down or speed up the replay of data.
- InetVis can replay traffic capture files, or display live traffic captured from a network interface.
- A variable time window can adjust the amount of data under review
- BPF filter expressions provide a powerful and flexible method to filter data

Apart from implementing extensions, the design and implementation of InetVis has undergone several optimization measures to attain performance and scalability. The following chapter will offer results to evaluate the fulfilment of the concept, give a measure of the enhancements, and ultimately, visualize intrusive Internet activity.

Chapter 5

Results and Analysis

The results of this research are multifaceted. Documented visual signatures of known network scanning techniques demonstrate that the 3-D scatter-plot concept saliently conveys intrusive network probing activity as anticipated. Following this, Internet traffic is reviewed from the perspective of a darknet, and the discovery of peculiar anomalies is presumed to be the result of novel network scanning methods. Through this exploration of darknet traffic, the experience of using InetVis affords an evaluation of some of its features; features which assist the user in identifying intrusive activity. The performance of the visualization is related by monitoring processor usage under varied loads of visualizing packet capture data. The responsiveness of controlling and manipulating the visualization also attests to its performance.

5.1 Visual Signatures of Network Scanning

To show that InetVis conveys intrusive activity as intended by its design, known scanning techniques are viewed in isolation. These documented ‘visual signatures’ serve as a proof of concept, attesting that the visualization is capable of exhibiting the known forms of intrusive activity. They also indicate that the visualization is implemented correctly.

5.1.1 A Method for Capturing Network Scans in Isolation

To ensure the signatures are clean, the data must be simulated or generated in an isolated, and fully controlled, network environment. To generate capture files of scanning activity, NMap was used in an isolated test network environment. The test network is comprised of two hosts connected by a crossover Ethernet cable, creating a closed and controlled network environment. One host is used to conduct a variety of nmap network scans, and is termed the ‘scanner’. It is configured to view the other host as a gateway. The other host acts as a ‘sensor’, with its network card placed in promiscuous mode to capture all the traffic generated. The

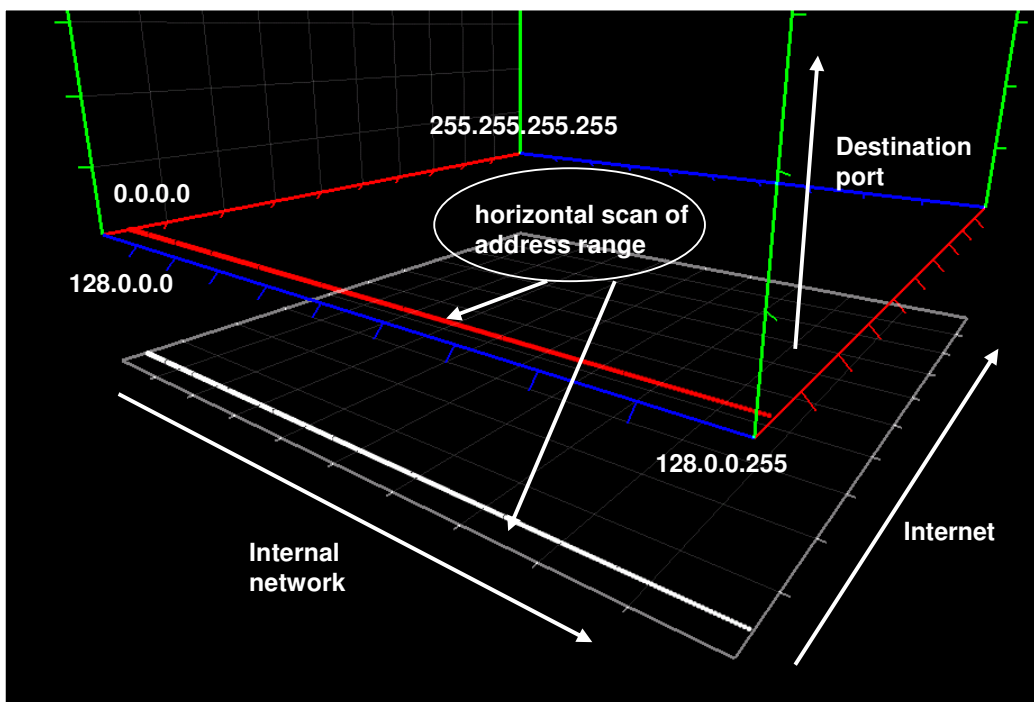


Figure 5-2: A network scan visual signature using ICMP and TCP

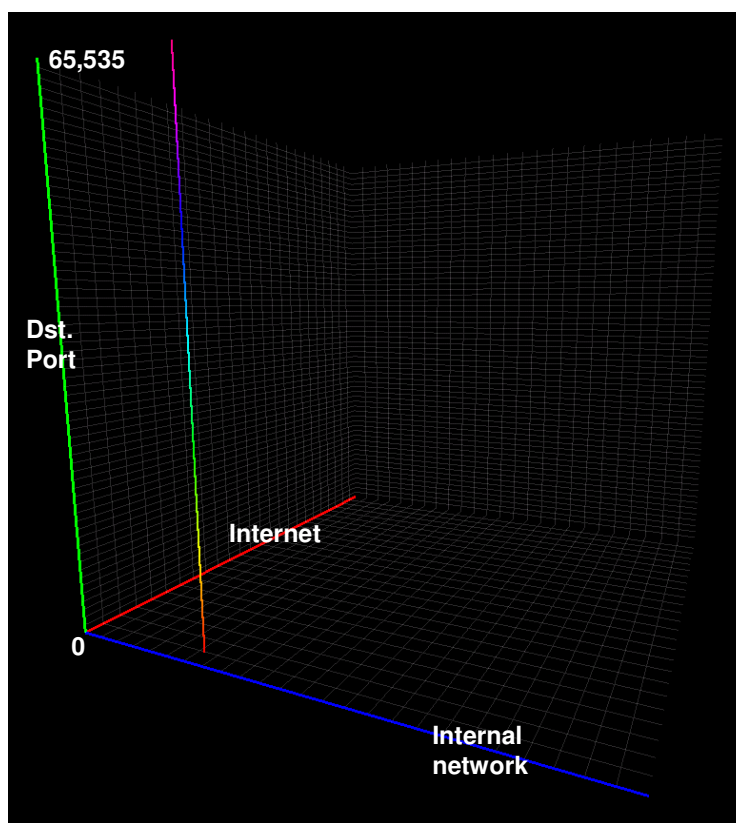


Figure 5-1: Visual signature of a complete port scan

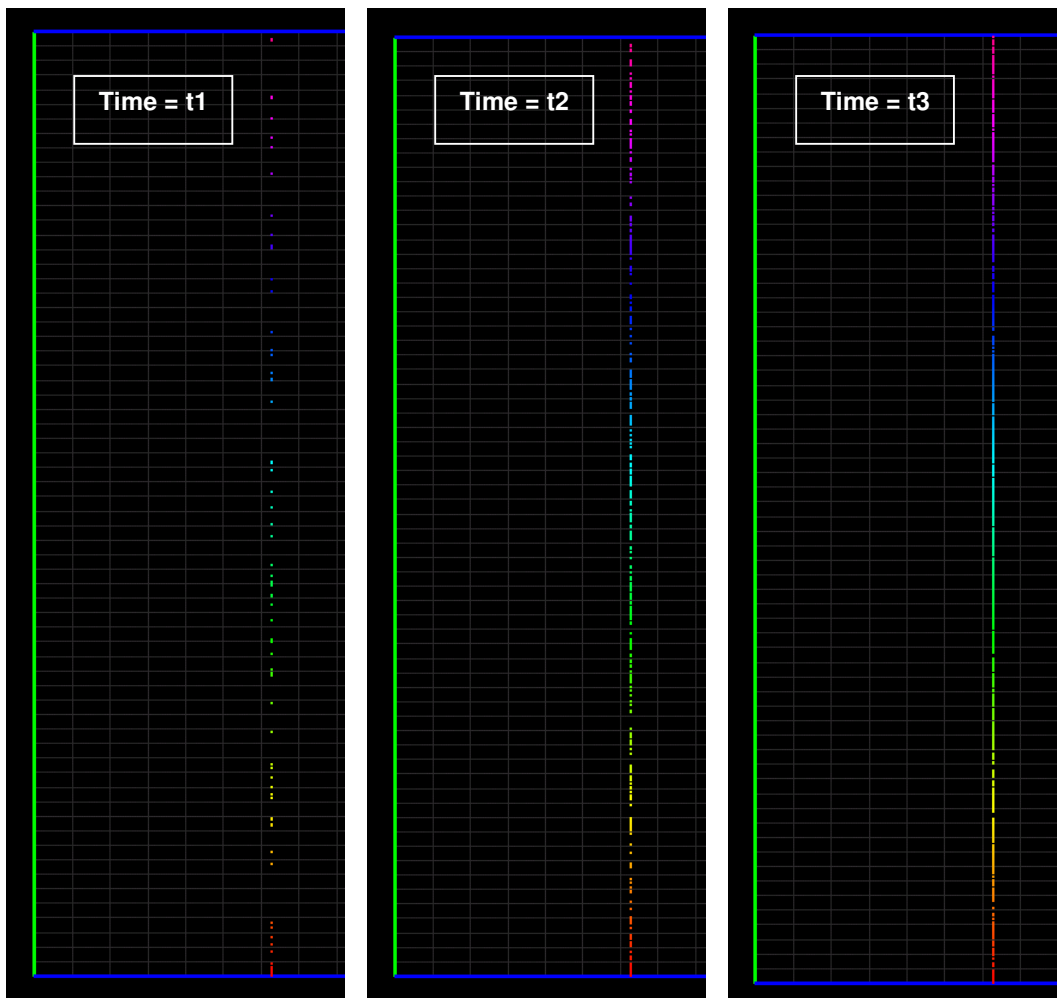


Figure 5-3: The random time sequence of probing ports

As covered in Section 2.2.3.1, network scans attempt to discover hosts by scanning across an address range. Using NMap to perform a ‘Ping Scan’ targeting the 128.0.0.0/24 network, a horizontal line can be seen appearing across the ICMP plane in Figure 5-2. The line above is an example of network address range scanning using TCP.

Although the plotting scheme is defined according to colour axes (as per Section 4.1.3), the labels, and arrows are annotations to assist in interpreting the image. (As the plotting scheme becomes familiar, annotation will be reduced in favour of a cleaner image.). The adjustment of markers and grids thought the figures are intended to help the viewer perceive a gauge of value. Often IP space (red and blue axes) is set to 25 divisions and port space (green axis) set to divisions set to 65 divisions, approximately scaling to the ranges of 255 and 65,535 respectively.

5.1.3 Vertical Scan Signature – Port Scanning

The concept of a port scan is introduced in Section 2.2.3.2. The visual signature of a complete port scan target at the host address 128.0.0.1, forms a vertical line in Figure 5-1. The default for NMap is to scan ports in a random order, as convey in Figure 5-3.

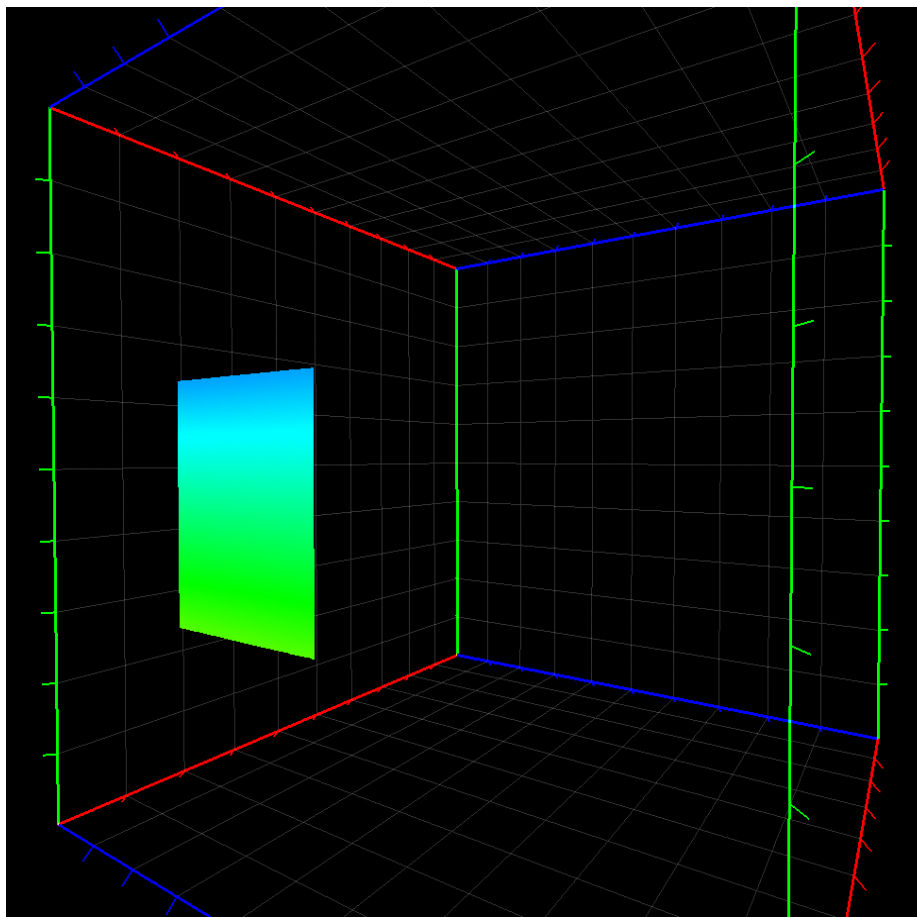


Figure 5-5: Block scan visual signature

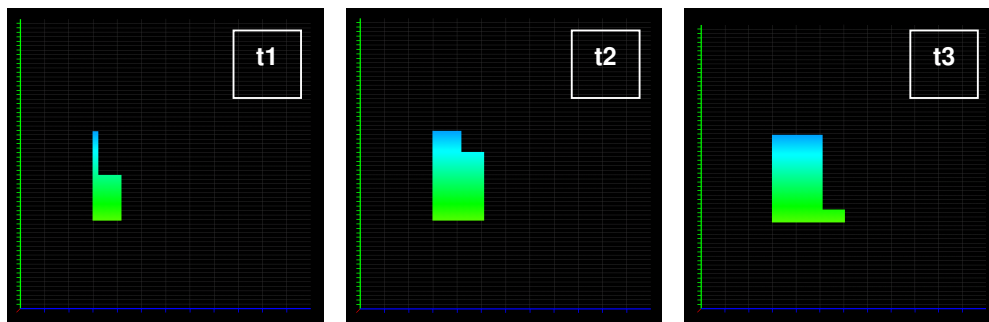


Figure 5-4: Ordered sequence of performing a block scan

5.1.4 Block Scan Signature

Both port scanning and network scanning can be conducted in a vertical and horizontal sweep of ranges. Figure 5-5 shows a complete scan conducted across the network from 128.0.0.64 to 128.0.0.128, on the port range from 20,000 to 40,000. This amounts to probing 1,280,000 distinct ports (and results in 1,280,000 points). Figure 5-4 shows the result when NMap is explicitly set not to scan ports in a random sequence. Stephen Lau dubbed this type of scan, a ‘lawnmower’ scan (as discussed in 3.2.1.2) Scanning thoroughly like this takes a long time, and in

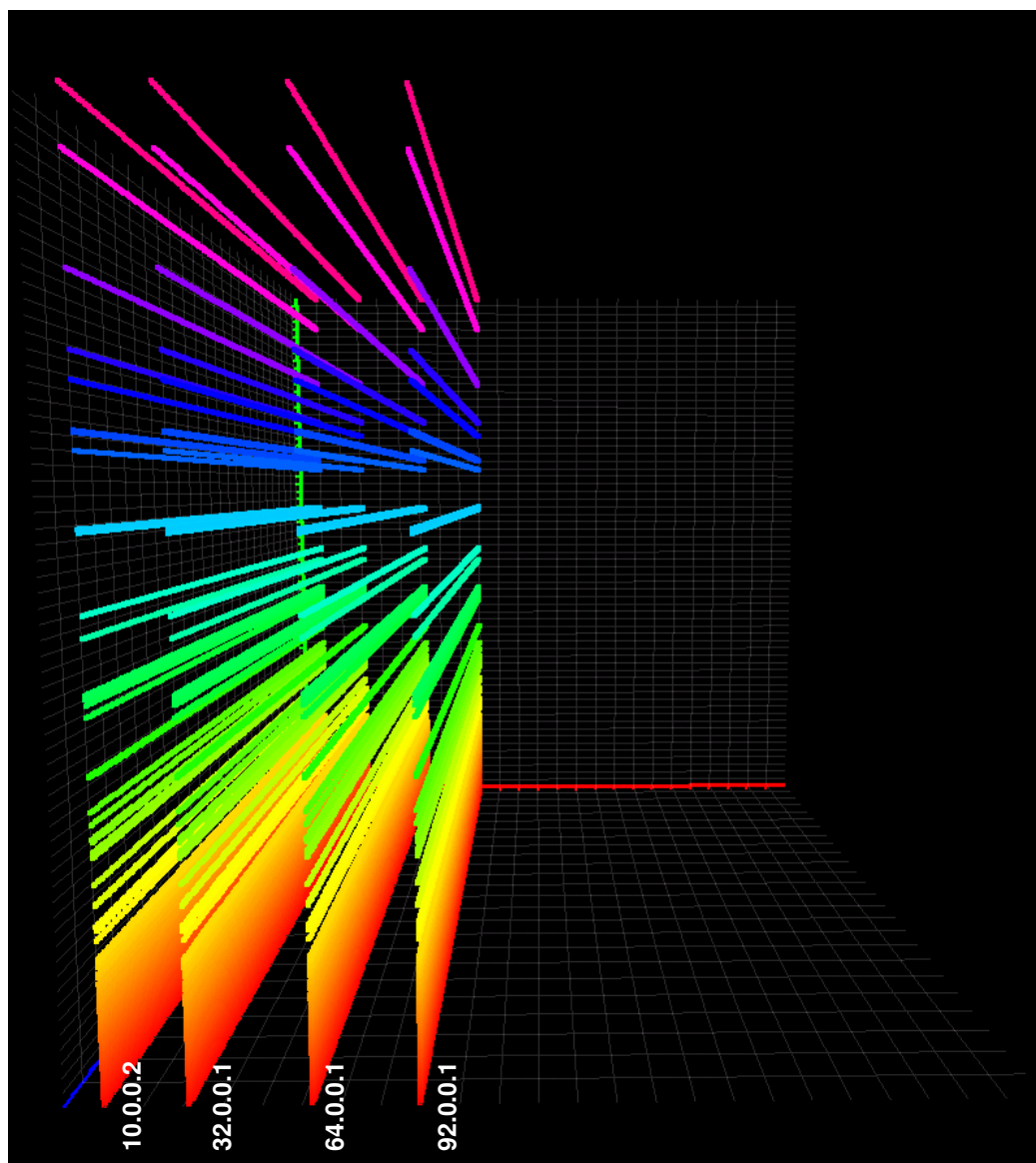


Figure 5-6: A visual signature of a grid scan with decoys

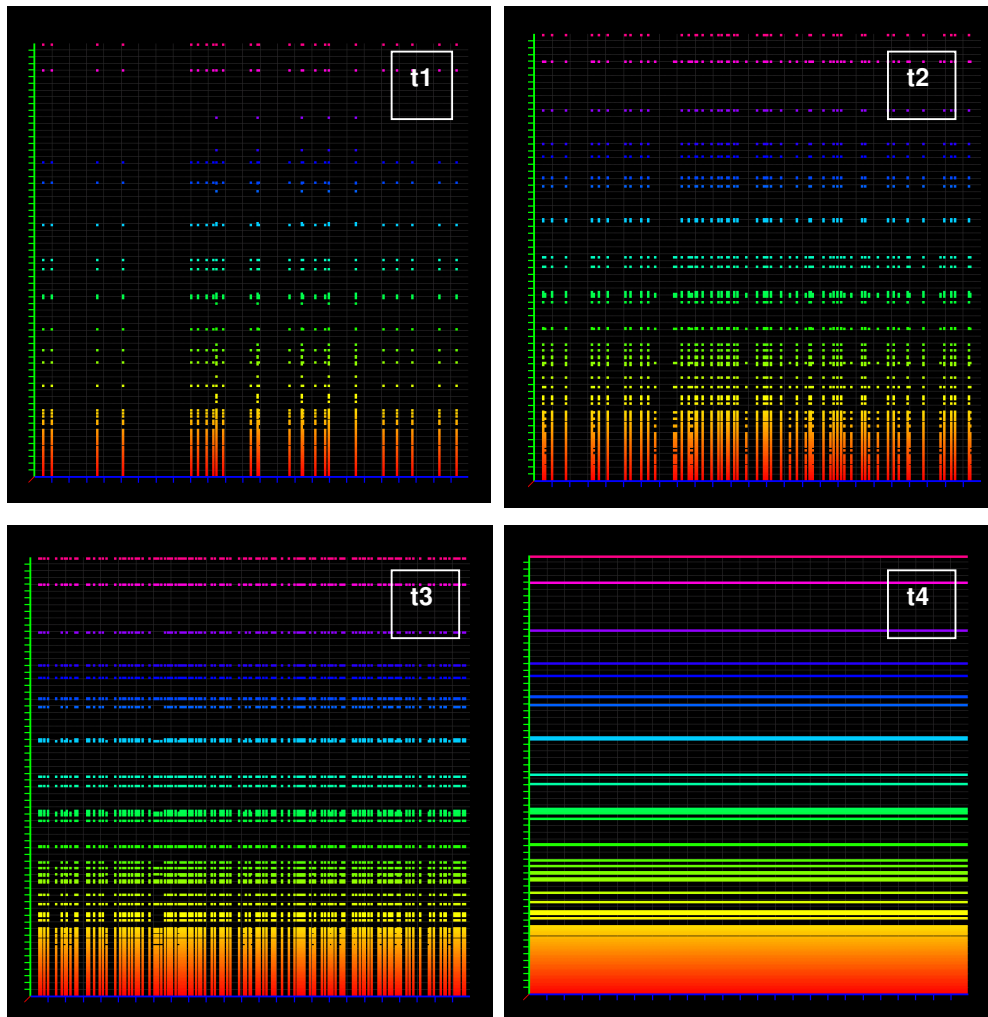


Figure 5-7: Random common service port sweep of entire network

practise yields a low average of successfully locating open ports for each connection probe. The term connection probe refers to a connection attempt aimed at a port.

5.1.5 Grid Scan Signature with Decoys

NMap offers the ability to only target well know service ports, which increases the probability of finding an open port for each connection probe, and reduces the time needed to complete the scan. This method of scanning can effectively map out all the open ports of every reachable host in the target network in a broad, fast sweep of the entire network.

Figure 5-6 shows how the method of sweeping a network range for open ports results in a grid-like pattern. All the hosts on 128.0.0.0/24 network were targeted.

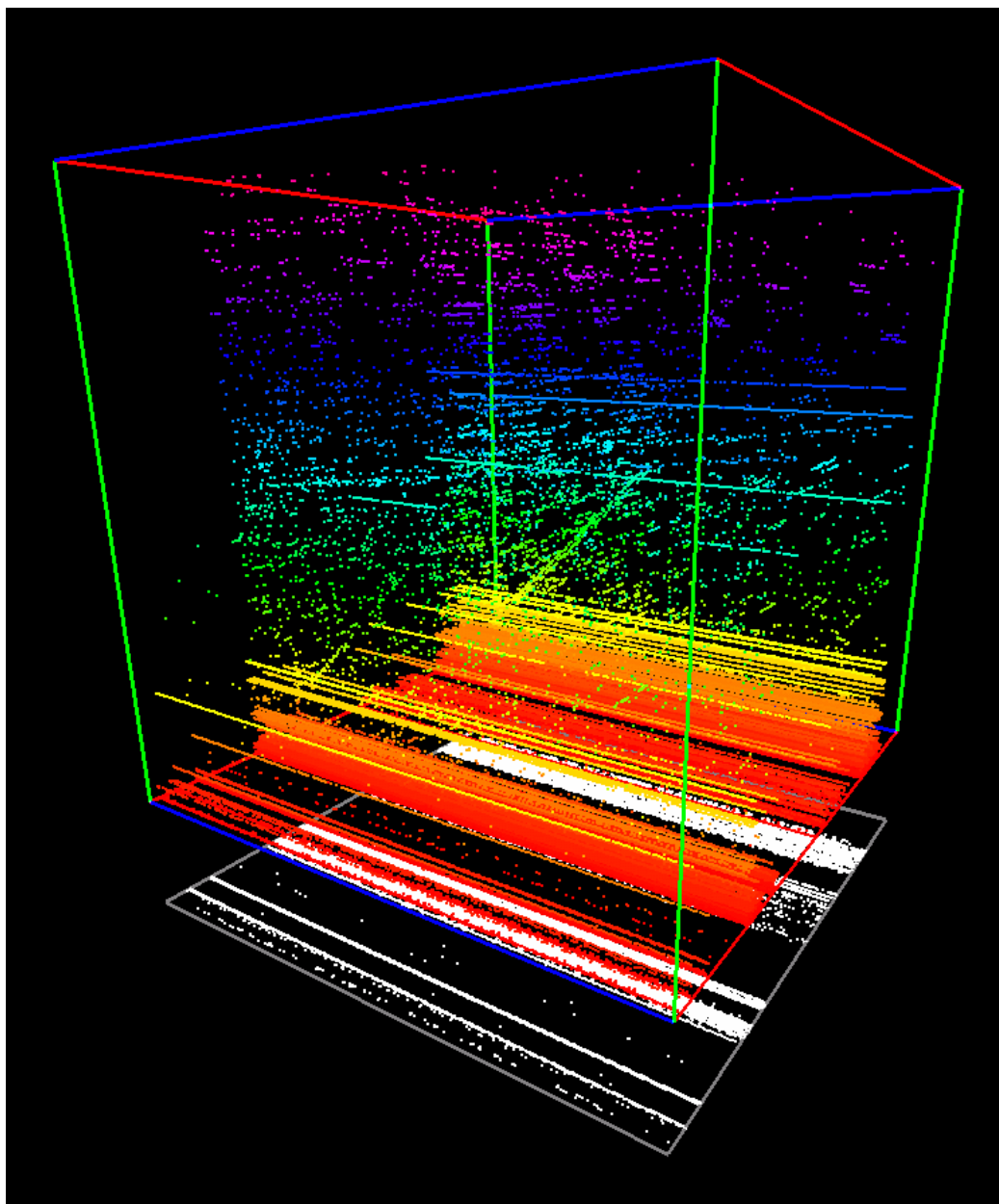


Figure 5-8: All darknet traffic for August 2005

There are four grids because the decoy option for scanning was enabled, where 32.0.0.1, 64.0.0.1, and 92.0.0.1 were decoys, with 10.0.0.2 being the actual source of the scan. Note the clutter at the bottom of the port range. This makes sense, since most well known service ports are assigned to low port numbers (below 1024).

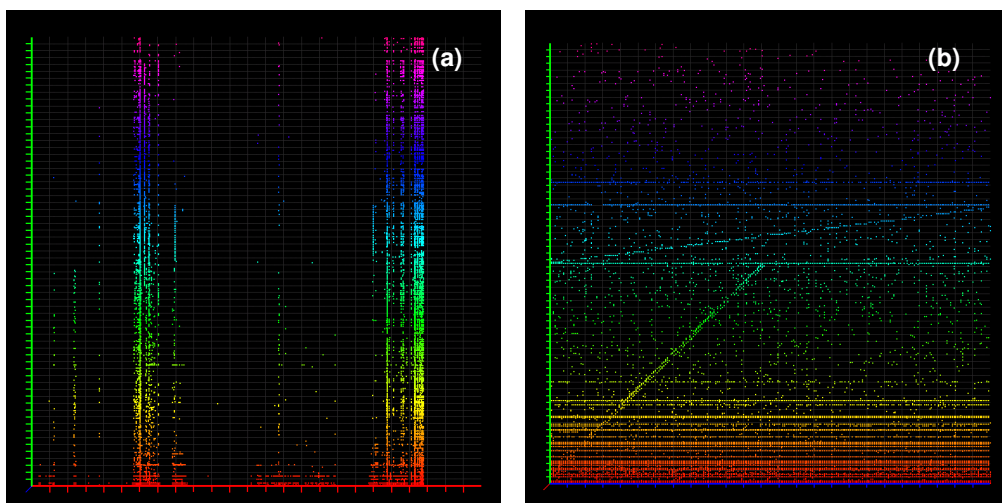


Figure 5-9: Bands of clustered darknet traffic (August 2005)

5.2 Darknet Traffic Analysis

The visual signatures presented in the previous section serve as a reference for identifying intrusive network activity when analysing actual Internet traffic. InetVis was put to practical use in reviewing traffic captured from a class C darknet during August and September 2005. As discussed in Section 2.3.1.4, a darknet is an empty address range where no Internet services are offered, nor are there any hosts in the darknet to use Internet. Therefore, there ought to be no traffic, and the InetVis display should be dark and empty. As clearly shows, this is far from the case. All 867085 packets captured during August 2005 are displayed in the figure. Network scanning across the address range is very common, as a wealth of horizontal lines attest.

5.2.1 August 2005

5.2.1.1 Bands of activity

Figure 5-9 (a) is an orthographic side-on projection down the blue x-axis providing a 2-D perspective of the source address and destination ports in a plane. The vertical bands relate the Internet ranges that contribute heavily to the unwarranted Internet traffic, also showing how heavily the port range is blanketed. This is a useful way of identifying hostile IP ranges of the Internet. Figure 5-9 (b) is a

frontal planar view of the internal network range and destination ports. It gives a clearer depiction of port scanning activity than 3-D views at the expense of the source address dimension. The view is useful, because the accurate geometrical projection exhibits some anomalous diagonal patterns that are harder to see in the 3-D view in Figure 5-8. Less visual clutter is seen in Figure 5-9 (a) and Figure 5-9 (b), because the 2-D view causes many of the 867,085 points to overlap.

5.2.1.2 Anomalous Diagonals

Inspecting Figure 5-8 carefully, three semi-incomplete similar diagonal lines are just noticeable. In the actual use of InetVis, by changing perspective with rotation, translation and zooming, these diagonal line patterns become more apparent. The filtered image, Figure 5-10, shows a clearer view of these diagonals,

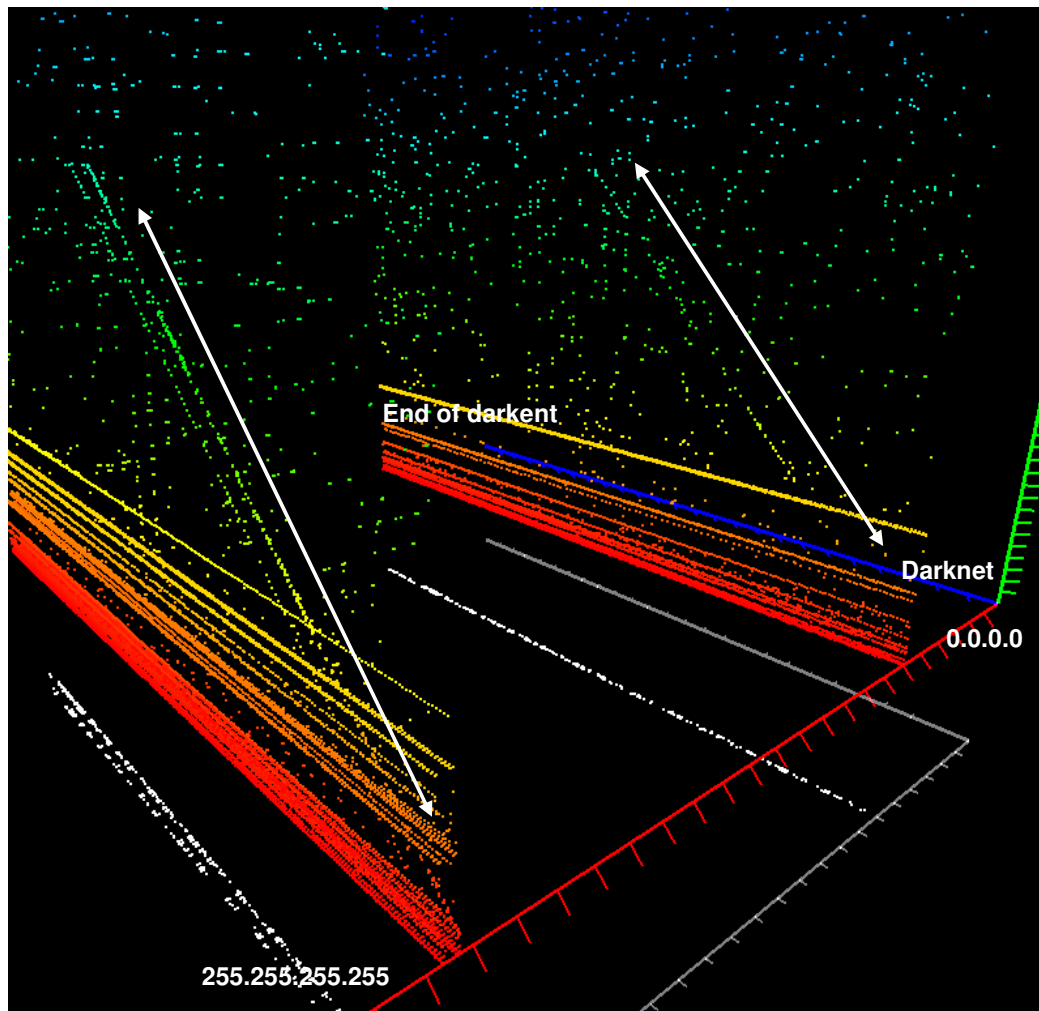


Figure 5-10: Partial diagonal lines viewed with filtering (August 2005)

with annotated arrows provided as a hint for their location. The specified BPF filter is:

```
"not udp and (src net 61.0.0.0/8 or src net  
220.0.0.0/7 or src net 218.0.0.0/8)"
```

This simply removes unwanted traffic from the display by identifying the three separate network ranges from which diagonals originate; an example of the flexibility of BPF filter expressions which offer many other attributes for filtering. A review of the 2-D Figure 5-9 (b) shows a visible 45-degree line comprised of the three separate lines, indicating that they all lie in the same diagonal plane. Therefore, three near identical traffic patterns arise from three different origins.

The possibility that this is a simple decoy scan is unlikely, since the time that the lines form during replay does not appear to be related. Therefore, it is probable that the same scanning tool, malicious code, or network mis-configuration explains the similarity. The scale along y (green) versus the scale along x (blue) is a ratio of 256:65535. To create a 45 degree line, as is the case with the anomalous diagonal, requires scanning a host on one port and then skipping several ports to scan the

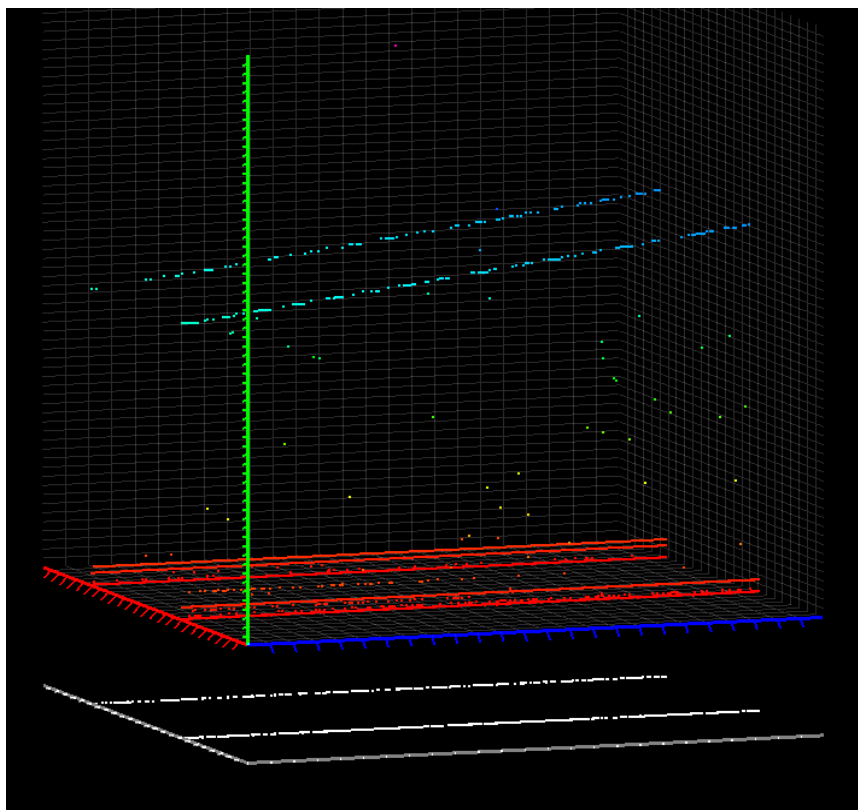


Figure 5-11: 'Step' scanning activity (August 2005)

next host on a substantially higher port. Therefore, to form the 45 degree line requires, approximately, jumping up (an uncanny) 255 port values as it traverses from host to host.

5.2.1.3 The Step Scan

In Figure 5-9, Instead increasing the port number for every host traversed, the 'step', as scan shown in the orthographic view of Figure 5-11, first traverses a number of hosts on a destination port before stepping up to a higher port. Like the diagonal anomalies, it has more than one origin, but unlike them, it covers the entire network range. The scan is very slow, and continues in from August into September.

5.2.1.4 The 'Creepy Crawly' Scan

A novel horizontal network scan covers the entire address range in continuous progressive segments, as shown in Figure 5-12. This is hidden in the depths of the display, certainly not noticeable in Figure 5-8. Its discovery was made possible by experimenting with time window values and immersive navigation within the display. The line appears as a solid normal network scan if the time window is set too large, and conversely, forms no segments if the time window is set too small. A time window of 36 hours was found to be appropriate. Figure 5-14 relates the slow progression over time, where $t_1 = 2005-08-12:00h00$, and t_2 follows 18hours later, with t_3 36 hours after t_1 . Given the slow and segmented nature of this probe, it is most probably devised to go undetected by NIDS.

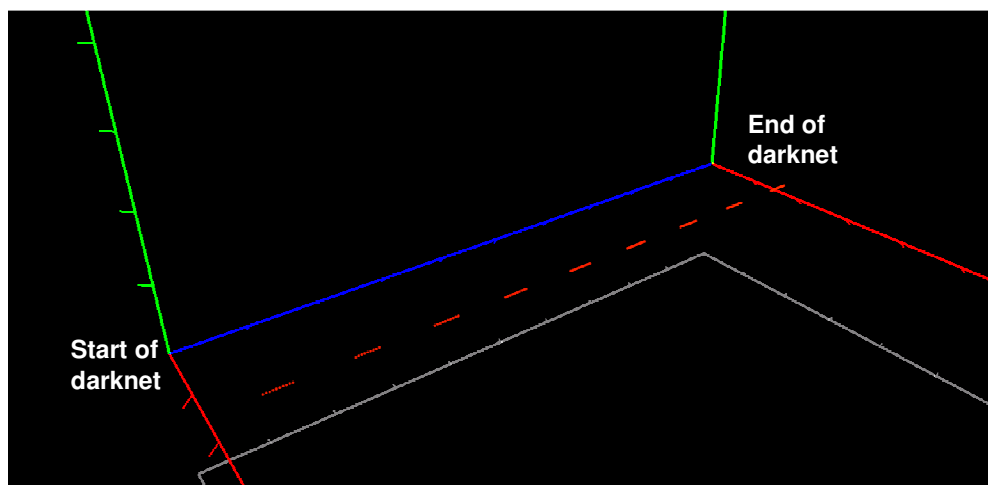


Figure 5-12: The 'creepy crawly' scan (August 2005)

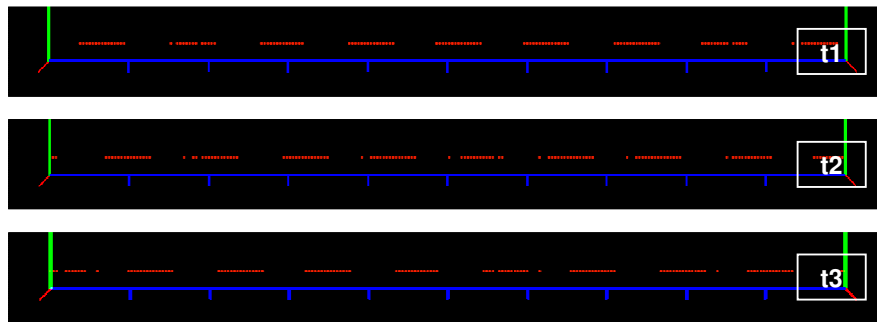


Figure 5-14: Slow progression of 'Creepy Crawly' scans

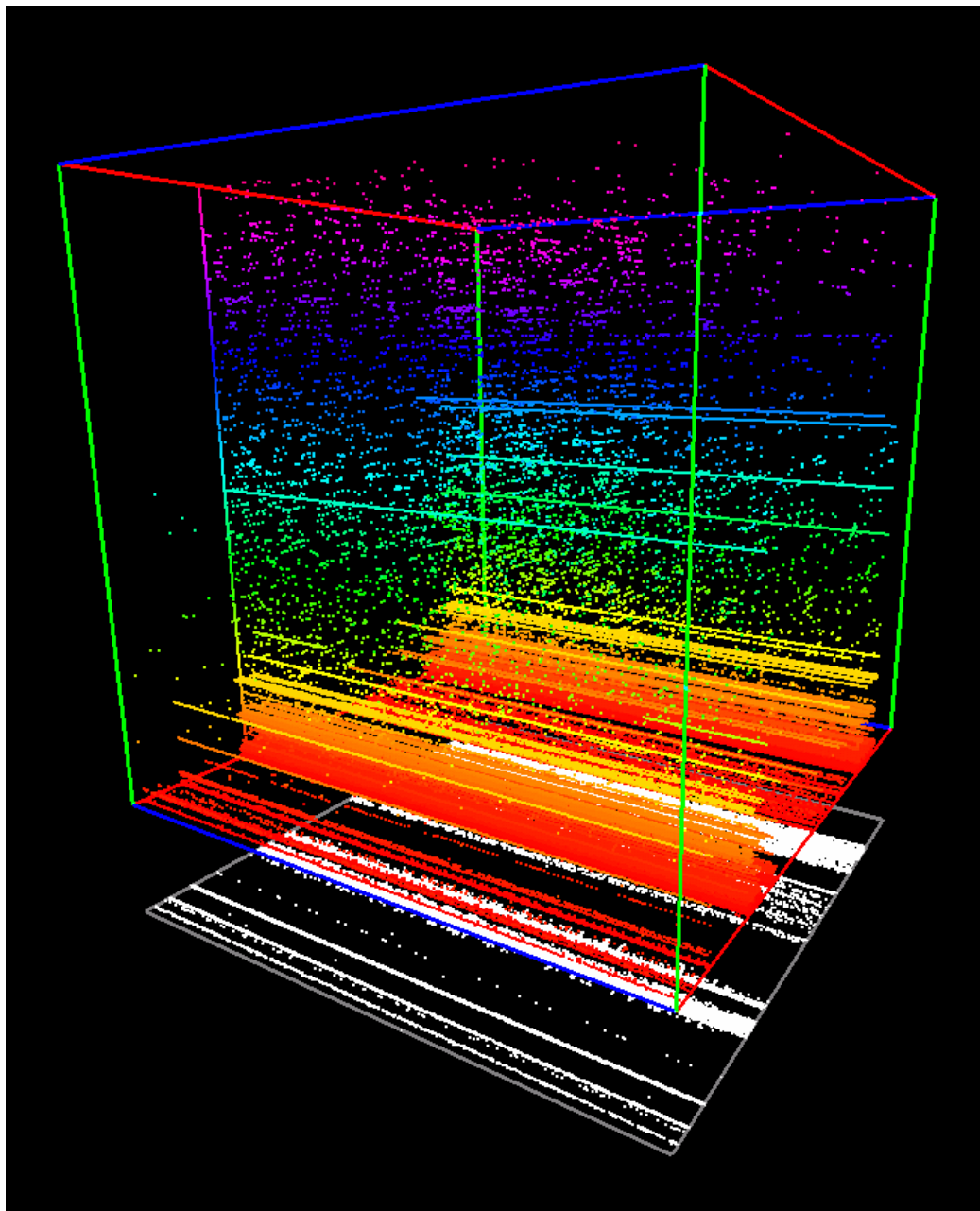


Figure 5-13: All darknet traffic for September 2005

September 2005

The capture from September results in a similar account of very many horizontal scans, and in particular a comprehensive port scan of what appears to be the darknets gateway in Figure 5-13. One of the reasons not many port scans are observed is that scanners first establish the presence of a hosts instead of expending time trying to scan the port of a non-existent host. Given a mostly empty darknet,

5.2.2.1 Diffuse diagonals

By twisting and reorienting the vantage point to below the cube, anomalous traffic that looks like 'noise' from most perspectives turns out to have a diagonal, but diffuse pattern, as is shown in Figure 5-15. The pattern is suspicious as it covers a

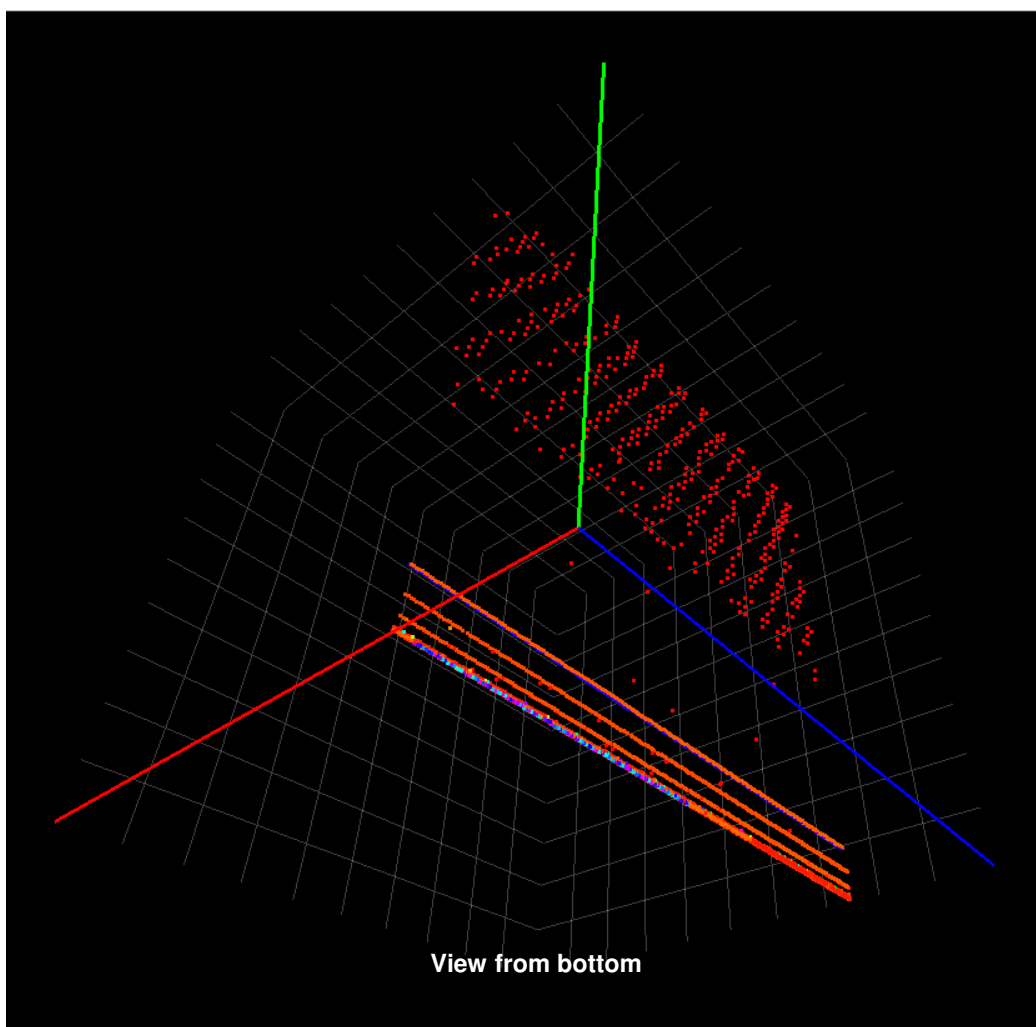


Figure 5-15: Diffuse diagonals (September 2005)

broad range of host is a poor pseudo-random fashion. The possibility of finding an open port on the higher ports that it covers is less likely than scanning lower down.

5.3 Performance and Scalability

The resource use of IneVis was monitored, since the frame-rate for InetVis is capped at 25 frames per second to avoid unnecessarily consuming resources that could be better utilized for data processing. As can be expected, replaying traffic at a rate of 1 day per second may require a fair proportion of resources for capture file extraction and processing.

While replaying the August darknet capture file, the visualization was able to maintain a replay rate of 86400x (1 day per second) up to a threshold of 500,000 points, whereby processor usage reached 100%. Since the buffer is a deque data structure offering constant time insertion and removal at its ends, it is inferred that rendering 500,000 points began to impact on the overall systems performance, where the system was unable to maintain the replay rate and navigation controls became somewhat jerky, but not unresponsive. At more conservative replay rates, the system was capable of comfortably handling over 800,000 packet events, maintaining acceptable frame-rate and interactive control.

5.4 Summary and Conclusion

The results demonstrate that the animated 3-D scatter-plot concept is effective at conveying intrusive network probing, and that the InetVis visualization functions as designed. The documented ‘visual signatures’ exhibit known network scanning activity as anticipated, and confirm that the 3-D scatter-plot concept saliently conveys intrusive network probing methods. The visual signatures also inform the visual review of Internet traffic using InetVis.

From the perspective of a darknet, large volumes of suspicious activity are observed as rouge Internet traffic. Some of the unconventional scans and anomalies receive closer inspection, and the discovery a creepy crawly scan, a step like scan, and suspicious diagonal phenomena is documented. The unconventional scans are presumed be measures at evading detection. These discoveries attest that

visual analysis can allow humans to detect intrusive activity intended to evade NIDS.

Three extensions that markedly improve Lau's 3-D animated scatter-plot concept are the variable replay rate, the adjustable time window, and BPF packet filtering. Without these, the results from darknet traffic analysis would be far less revealing. As a testament to its performance and scalability, InetVis proved to be capable of handling over 800,000 packet events within the display.

Chapter 6

Conclusion

The 3D scatter-plot concept proves to be impressively scalable in comparison to the more elaborate visualizations with metaphorically complex representations. Features like variable replay rate, adjustable time window, bounded navigation (for deeper exploration), and dynamic filtering enhance the ability to pick up and detect a varied range of scanning activity. These are all improvements of Lau's original animated 3-D scatter-plot visualization – the Spinning Cube of Potential Doom.

A number of possible extensions to InetVis that could further improve the concept are listed below:

- Logarithmic port axis to distribute the clutter seen in the lower port region
- Port range drill down by specifying any arbitrary range smaller than the full 65,535 ports would make smaller port scans more evident, and allow inspection that is more detailed, i.e. only looking at the bottom 1024 ports.
- Likewise, the ability to specify an External Internet range would allow an inter-domain drill down. This would be particularly useful for further investigating Internet ranges responsible for a significant proportion of intrusive activity.
- Labels at the ends of axes, have not been implemented (as yet), and would certainly assist the viewer in grasping a more referential detail.

As seen in Chapter 5, the results displayed in captured images show that InetVis saliently conveys traffic patterns and reveals suspect network probing – intrusive reconnaissance that often precedes an attack. The images presented from the darknet traffic capture clearly attest that the Internet is a hostile a network environment, and that unconventional probing methods are present, such as the 'creepy crawly' scan.

In conclusion, this research illustrates the deft application of visualizing network traffic by making intrusive activity intended to be invisible visible.

References

[Ball 2004]

Ball, R., Fink, G.A., North, C. “*Home-centric visualization of network traffic for security administration*”, Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. p. 55-64. ACM Press, New York, 2004.

[Bro]

Bro Intrusion Detection System. <<http://www.bro-ids.org/>> (03/06/2005).

[CERT]

CERT/CC – CERT® Coordination Center. Carnegie Mellon Software Engineering Institute. Carnegie Mellon University. <<http://www.cert.org/>> (27/05/2005).

[Ethereal]

Ethereal – network protocol analyser. <<http://www.ethereal.com/>> (07/11/2005)

[Fink 2004]

Fink, G.A., Ball, R., North, C., Jawalkar, N., Correa, R. “*Network Eye: End-to-End Computer Security Visualization*”. 2004.
<csgrad.cs.vt.edu/~finkga/downloads/Fink-et-al-VizSec2004.pdf> (02/06/2005).

[Fisk 2003]

Fisk, M., Smith, S.A., Weber, P.M., Kothapally, S., Caudell, T.P. “*Immersive Network Monitoring*”. PAM2003 – Passive and Active Measurement 2003), NLANR/MNA (National Laboratory for Applied Network Research / Measurement and Network Analysis Group).
<<http://public.lanl.gov/mfisk/papers/pam03.pdf>> (25/05/2005).

[Lau 2004]

Lau, S. “*The Spinning Cube of Potential Doom*”, Communications of the ACM archive, Volume 47, Issue 6. p. 25-26. ACM Press, New York, 2004.

References

[Lau 2003]

Lau, S. “*The Spinning Cube of Potential Doom*”, article, November 10th, 2003. National Energy Research Scientific Computing Center (NERSC) website. <<http://www.nersc.gov/nusers/security/TheSpinningCube.php>> (25/05/2005).

[Lyman 2003]

Lyman, P., Verain, H.R. “*How Much Information 2003*”, Section 8 – Internet. School of Information Management and Systems, University of California at Berkeley. Regents of the University of California, 2003. <<http://www.sims.berkeley.edu/research/projects/how-much-info-2003/Internet.htm>> (25/05/2005).

[McCanne 1992]

McCanne, S., Jacobson, V. “*The BSD Packet Filter: A New Architecture for User-level Packet Capture*”. *USENIX Winter 1993 Conference Proceedings*, San Diego, California, 1993. <www.tcpdump.org/papers/bpf-usenix93.pdf> (27/06/2005), or <<http://www.usenix.org/publications/library/proceedings/sd93/mccanne.pdf>> (06/11/2005).

[Metasploit]

Metasploit – penetration testing and exploit research tool. <<http://www.metasploit.com/index.html>> (07/11/2005).

[Nessus]

Nessus – vulnerability scanning tool. <www.nessus.org> (07/11/2005).

[NMap]

NMap – network mapper, scanning utility. <<http://www.insecure.org/nmap/>> (07/11/2005).

[Northcutt 2001]

Northcutt, S., Cooper, M., Fearnow, M., Frederick, K. *Intrusion Signatures and Analysis*. New Riders, Indianapolis, 2001.

[O’Neill 2003]

O’Neill, E.T., Lavoie, B.F., Bennett, .R. Web Characterization project, OCLC (Online Computer Library Center). <<http://www.oclc.org/research/projects/archive/wcp/>> (25/05/2005). “*Trends in the Evolution of the Public Web*”, <<http://www.dlib.org/dlib/april03/lavoie/04lavoie.html>> (25/05/2005). Size and growth statistics,

References

<<http://www.oclc.org/research/projects/archive/wcp/stats/size.htm>>
(25/05/2005).

[LibPCap]

“LibPCap - Packet Capture library ” man page.
<http://www.tcpdump.org/LibPCap3_man.html> (06/11/2005).

[Putton 2001]

Putton, S., Yurik, W., Doss, D. “*An Achilles’ Heel in Signature-Based IDS: Squealing False Positives in SNORT*”. Proceedings of the 4th International Symposium on Recent Advances in Intrusion Detection (RAID), 2001.
<<http://www.ncassr.org/projects/sift/papers/raid01.pdf>> (03/06/2005).

[Rehman 2003]

Rehman, R.U. Intrusion Detection with Snort, Advanced IDS Techniques Using Snort, Apache, MySQL, PHP and ACID. Pearson Publishing, New Jersey, 2003.

[Scanmap3D]

Clark, D. Scanmap3D open source visualization for Snort. Scanmap3D-2.1b and Scanmap3D-3.0. Source code INSTALL file included with distribution.
<<http://scanmap3d.sourceforge.net/>> (29/05/05).

[Symantec 2005-Sep]

Turner, D. (exec. Ed.), Entwisle, S. (Ed.), et al (Symantec). “*Symantec Internet Security Threat Report, Trends for January 05–June 05*”. Volume VIII, September 2005. Symantec Enterprise Solutions Website:
<<http://enterprisesecurity.symantec.com/content.cfm?articleid=1539>>
(03/11/2005).

[Symantec 2005-Mar]

Turner, D. (exec. Ed.), Entwisle, S. (Ed.), et al (Symantec). “*Symantec Internet Security Threat Report, Trends for July 04 – December 04*”. Volume VII, March 2005. Symantec Enterprise Solutions Website:
<<http://enterprisesecurity.symantec.com/content.cfm?articleid=1539>>
(20/04/2005).

[Symantec 2004-Sep]

Turner, D. (exec. Ed.), Entwisle, S. (Ed.), et al (Symantec). “*Symantec Internet Security Threat Report, Trends for January 1, 2004 – June 30, 2004*”. Volume VI, September 2004. Symantec Enterprise Solutions Website:
<<http://enterprisesecurity.symantec.com/content.cfm?articleid=1539>>
(16/03/2005).

References

[Tanenbaum 2003]

Tanenbaum, A., S. Computer Networks. Prentice Hall, New Jersey, 2003.

[TCPDump]

TCPDump – packet logging utility. <<http://www.tcpdump.org/>> (07/11/2005).

[Teoh 2004]

Teoh, S.T., Ma, K-L., Wu, S.F., Jankun-Kelly, D.T.J. “*Detecting Flaws and Intruders with Visual Data Analysis*”, IEEE Computer Graphics and Applications, Volume 24, Issue 5. p. 27-35. IEEE, 2004.

[TCPDump]

“tcpdump - dump traffic on a network” man page.

<http://www.tcpdump.org/tcpdump_man.html> (06/11/2005).

[Yengeswaren]

Yegneswaran, V., Barford, P. Ullrich, Y. “*Internet Intrusions: Global Characteristics and Prevalence*”. ACM SIGMETRICS Performance Evaluation Review archive Volume 31, Issue 1. p. 138-147. ACM Press, New York, 2003.

[Yin 2004]

Yin, X., Yurcik, W., Treaster, M., Li, Y., Lakkaraju, K. “*VisFlowConnect: netflow visualizations of link relationships for security situational awareness*”, Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security. p. 35-44. ACM Press, New York, 2004.