A framework for the application of Network Telescope Sensors in a global IP Network

Submitted in fulfilment
of the requirements of the degree of

Doctor of Philosophy

of Rhodes University

Barry Vivian William Irwin

Grahamstown, South Africa
January 2011

Abstract

The use of Network Telescope systems has become increasingly popular amongst security researchers in recent years. This study provides a framework for the utilisation of this data. The research is based on a primary dataset of 40 million events spanning 50 months collected using a small (/24) passive network telescope located in African IP space. This research presents a number of differing ways in which the data can be analysed ranging from low level protocol based analysis to higher level analysis at the geopolitical and network topology level. Anomalous traffic and illustrative anecdotes are explored in detail and highlighted. A discussion relating to bogon traffic observed is also presented. Two novel visualisation tools are presented, which were developed to aid in the analysis of large network telescope datasets. The first is a three-dimensional visualisation tool which allows for live, near-realtime analysis, and the second is a two-dimensional fractal based plotting scheme which allows for plots of the entire IPv4 address space to be produced, and manipulated. Using the techniques and tools developed for the analysis of this dataset, a detailed analysis of traffic recorded as destined for port 445/tcp is presented. This includes the evaluation of traffic surrounding the outbreak of the Conficker worm in November 2008. A number of metrics relating to the description and quantification of network telescope configuration and the resultant traffic captures are described, the use of which it is hoped will facilitate greater and easier collaboration among researchers utilising this network security technology. The research concludes with suggestions relating to other applications of the data and intelligence that can be extracted from network telescopes, and their use as part of an organisation's integrated network security systems.

Acknowledgements

Many people have assisted and supported me over the period of this research. Foremost has been my wife Yoland who has had endless patience for what has been a lengthy process. Much appreciation is also due to may parents Pat and Anne, not only for their support in numerous way and ongoing encouragement, but also for the proofreading done on multiple drafts.

My supervisors, George Wells and Peter Clayton deserve thanks for embarking on the journey with me and providing appropriate nudges regarding my progress along their way and for the general guidance in the compilation of this work. My students also need recognition as each of them have chipped away at smaller parts of the problem. Particularly gracious thanks to Nicholas Pilkington, Jean-Pierre van Riel, Richard Barnett, Blake Friedman and Bradley Cowie for their work in bringing the initial analysis framework and utilities to life from the whiteboards full of design specs. Francois Jacot-Guillarmod and Guy Halse of the Rhodes University Information Technology Division are thanked for their efforts in securing me the requisite IP address space needed for this project, and having a sympathetic ear to my requests for massive volumes of scarce Internet bandwidth needed for obtaining comparative datasets.

Randall Munroe is also due credit for providing the initial inspiration with his *Map of the Internet*[1] for the Hilbert Curve based representation of IP address space which was refined and adapted for the data analysis and visualisation performed in this work.

This work was performed in and funded by the Centre of Excellence in Distributed Multimedia at Rhodes University with financial support from Telkom SA, Comverse, Verso Technologies, Tellabs, StorTech, EastTel and THRIP. Funding was also received from the National Research Foundation Thutuka Program Grant number 69018 and the Rhodes University Joint Research Committee (JRC). This work has also made use of the GeoLite Geolocation library from MaxMind, and of the CAIDA Backscatter datasets from 2004-2008.

[1] http://xkcd.com/195/

ACM Computing Classification System Classification

Primary Classification:

C. Computer Systems Organization

C.2 COMPUTER-COMMUNICATION NETWORKS

C.2.0 General Subjects: Security and protection

Additional Classification:

C. Computer Systems Organization

C.2 COMPUTER-COMMUNICATION NETWORKS

C.2.3 Network Operations Subjects: Network management

K. Computing Milieux

K.6 MANAGEMENT OF COMPUTING AND INFORMATION SYSTEMS

K.6.5 Security and Protection (D.4.6, K.4.2)

Contents

List of Figures

List of Tables

List of Code Listings

Part I

Introduction

Science fiction does not remain fiction for long.
And certainly not on the Internet.

Vinton Cerf: Co-creator of the TCP/IP Suite

1 Introduction

In today's increasingly interconnected world, threats against data and infrastructure continue to evolve. This work intends to discuss the identification, mitigation and defence against such threats within the context of the globally interconnected network that we have today — the Internet. Organisations are driven to attain Internet connectivity as a basic tenet of modern operation — much as the facsimile machine became an essential business tool during the 1980s. This process, however, results in increased exposure of the organisation and its computing systems to risk. Many of the organisations connecting and transacting online today also have limited technical support internally, and lack the ability to be able to employ full-time network and operational security staff; further exacerbating this risk. With the current trend of outsourcing, off-shoring and multinational footprints, and consumption of services from and deployment of solutions into the computational cloud, this risk is multiplied considerably due to the markedly expanded attack surface provided by the increased number of points and systems connected to the global Internet. The rise and prevalence of 'smart' mobile devices such as highly functional music players and communication devices — most commonly mobile phones, but also gaming devices, e-book readers and tablets — provide alternate paths into an organisation. These often make use of cellular communications or wireless broadband technologies. In most cases such devices are able to trivially bypass the traditional bastions of network security in which near absolute trust is placed — the border firewall. Bill Cheswick's often quoted 1990 description of the traditional network security model as "a sort of crunchy shell around a soft, chewy center" (Cheswick, 1990a), has probably never been more apt. Considering this model, one can view the current network security situation as a hard shell that, while still strong, has become increasingly porous, with the holes becoming larger and more numerous.

1.1 Problem Statement

Over the last ten years, users of the Internet have experienced the devastating effects of network worms. These include the SQL Slammer (Microsoft, 2002; CERT/CC, 2003; Moore *et al.*, 2003) and Witty (Shannon and Moore, 2004a; Paxson, 2005) worms that spread at unprecedented speeds; CodeRed (CERT, 2001) and CodeRed II variant which gained widespread press coverage. The 2003 emergence of MS Blaster (Schultz, 2003a; Bailey *et al.*, 2005c) and Welchia (Bailey *et al.*, 2005c) — worm and anti-worm were both problematic with many researchers considering the side effects from the Welchia spread to be worse than the ill effects experienced due to MS Blaster (Bailey *et al.*, 2005c). Appendix A contains a more detailed timeline of these major network events.

In recent years, the differentiation between the computing terms of Virus and Worm have also blurred, with many instances of what would traditionally be regarded as viruses now having a significant level of network awareness and multipartite payloads to enable multi-vector spreading. Examples of this are shown by the Zotob (Schneier, 2005; McGraw, 2007) and SobBig (Levy, 2003; Schultz, 2003b; Berghel, 2003) virus families which propagated via multiple vectors such as email, file sharing, peer to peer networks, network shares, and some variants by direct remote exploit of systems on a network.

With this shift to malicious software (commonly referred to as 'malware') becoming increasingly network aware, we have seen a significant increase in the volumes and intensity of what can be deemed to be traffic with malicious intent arriving at Internet hosts. There has been a concurrent increase in the levels of human driven scanning and attack activities. The net result is that systems and organisations exposed to the Internet are facing a greater level of risk than they have before. At the same time many of the organisations, and even individuals embracing the new technologies offered by this form of communication, lack the skills required to suitably secure their systems.

As the number of connected nodes on the Internet has increased since its commercialisation in the early 1990s, the volume of what is often referred to as backscatter or Internet Background Radiation (IBR) (Pang *et al.*, 2004; Pemberton, 2007; Wustrow *et al.*, 2010) traffic has increased. Although this is partially due to what is probably the result of system misconfiguration, a significant portion of such traffic can be shown to originate from malicious agents, both programmatic and human in nature. A detailed discussion of the characteristics and classification of this traffic can be found in Sections 2.3 and 5.5.

While it is recognised that no automated system can catch every malicious datagram entering a network (although the vendors of Intrusion Prevention and Intrusion Detection Systems make claims and counter-claims to this effect), solutions relating to the use of interactive sacrificial systems including honeypots and simulated honey nets, or more controversial techniques such as 'bait and switch', bring about their own set of risks and complications in their operation and deployment within an organisation's network. While systems such as these have a definite value, the techniques and tools presented in this work are designed to function primarily on data collected using passive, low interaction means. As such it is felt that the risk for an organisation to deploy such a system, whether as a single node or as part of a collaborative distributed sensor network, is minimal in comparison to more interactive methods. This does, however, come at the cost of decreased access to certain types of data – most notably the payloads of TCP data connections. The value proposition and trade-off such solutions represent needs to be carefully evaluated by researchers and organisations contemplating the use of such systems.

1.2 Research Outline

This research has been conducted with the following research objectives in mind:

- The assessment of distributed and collaborative monitoring systems as a means of early detection, and more importantly as mechanisms to aid in the discrimination between background network traffic (backscatter/radiation) and more malicious traffic, and further whether this malicious traffic is widespread or has targeted a specific organisational network.

- The development of suitable tools for the analysis and visualisation of data at the scale of a global network. By their nature these data sets are very large, often comprising millions of individual packets or network sensor events. Image-based analysis is one of the more viable means to produce rapid and meaningful overviews of the data when utilising datasets of this magnitude.
- The final objective is the development of a framework based on the previous two goals which allows for the easier and more structured application of network telescopes as a part of an organisation's security strategy. Information sharing for operational and research purposes is also to be considered.

1.3 Research Method

It is the intention of the researcher to address the research objectives above through a combination of experimental work to validate hypotheses, and synthesis and discussion around existent research in the fields. The field of network telescope usage is fairly young, with David Moore's 2002 presentation at USENIX (Moore, 2002) being one of the earliest mentions of the practical implementation and application of the technology. The reports on the SQL Slammer (Moore *et al.*, 2003) and Witty (Shannon and Moore, 2004b,a) worms were amongst the earliest published information security research works making use of network telescopes. Details of the operation and configuration of such sensors was first detailed in the 2004 CAIDA[1] technical report by Moore *et al.*. Even some six years on from these publications, there is still much in the field of Network Telescopes, also known colloquially as 'darknets', that remains loosely defined and open to interpretation. This research aims to clarify some of these aspects. During the course of conducting this research, several new tools and methods for analysis were developed and metrics arrived at. An analysis of the Conficker worm outbreak and spread from late November 2008 is also presented, as observed on the researcher's own network telescope.

[1] Cooperative Association for Internet Data Analysis - `http://www.caida.org/`

1.4 Document Structure

This document is comprised of three parts structured as follows:

Part I – Contains introductory material as well as details on the establishment of a Network Telescope Sensor at Rhodes University and the data collection process.

- Chapter 2 provides a brief introduction to network security, with a particular focus on network aware malware, malicious activity and current monitoring strategies.
- Chapter 3 discusses the data sets used in this research, and in particular the setup of the Rhodes Network Telescope from an organisational and configuration perspective. Details of the data storage and processing framework are presented.

Part II – Constitutes the bulk of the work, and discusses the analysis process, tools developed, and the results of the analysis on the collected dataset along with a case study of the Conficker Worm outbreak.

- Chapter 4 introduces the tools and techniques that were utilised and developed for processing the collected data.
- Chapters 5 and 6 present the results of the analysis of the data collected using the process described in Chapter 3. The methods used are based on the discussions in Chapter 4. The analysis follows the lines of basic protocol focussed analysis discussed in Chapter 5, followed by higher level analysis of related data in Chapter 6.
- Chapter 7 presents a detailed case study focused specifically on the spread of the Conficker worm and the resultant traffic as interpreted via the Network Telescope data.

Part III – Reflects on the analysis performed in the previous section. Some consideration is given to the application of Network Telescopes in operational and research roles.

- Chapter 8 builds on the analysis performed in the previous Chapter and proposes a number of base metrics which can be used for describing the network telescope and enumerating aspects of collected datasets.
- Chapter 9 reflects on the preceding four chapters and provides recommendations for the use of Network Telescope Technologies.
- Chapter 10 revisits the research goals as stated in this chapter and reflects upon the research performed.

The document concludes with a number of Appendices containing supplemental information. These are referred to within the main body of the text.

1.5 Document Conventions

In the remainder of the document, as a general rule, URLs pertaining to websites, organisations or software mentioned, are provided as a footnote. The rationale behind this is to minimise the break in the flow of the document, and to allow readers quick access to the information, rather than having to look up the relevant information in the References section of this work. The electronic version of this document contains click-able hyperlinks for section references as well as all URLS. Citations in the text body are also hyperlinked to the appropriate entries in the References section.

Where port numbers are referred to the notation of port number/protocol is used as in 22/tcp to indicate a connection to port 22 (commonly used for the SSH protocol) making use of the TCP transport. Similarly IP addresses and networks are cited by making use of Classless Inter-domain Routing (CIDR) notation (Fuller and Li, 2006). IP Networks are referenced by their address, a '/' and then the number of bits that constitute the netmask. Values of /8, /16 and /24 correspond to the older Class A, B and C networks as described in (Clark, 1985).

The beginning of knowledge is the discovery of something we do not understand.

Frank Herbert - Science Fiction Author and Writer

2

Literature Survey

DURING the last 25 years, the field of computer networking has emerged and flourished. The history of the development of this art and science is closely intertwined with the evolution and development of the ubiquitous global network — 'The Internet'. As with any new frontier, it has been filled with potential and risks. From its initial beginnings as a project funded by the United States of America's Defense Advanced Research Projects Agency (DARPA) – itself the result of man's entry into space travel, having been formed in 1958 in response to the launch of Sputnik by the then Soviet Union – the resultant ARPANET has evolved into the Internet as we know it today. This chapter serves to provide context to the research and related topics within the scope of the global network.

A brief introduction to the history of Network security is provided in Section 2.1. The concept of Internet Background Radiation (IBR), or Backscatter, is introduced in Section 2.2, with a more detailed discussion of its composition in Section 2.3. Discussions around the need to monitor the IBR and means for doing this are presented in Section 2.4. An introduction to the conceptual aspects of Network Telescopes is followed by a taxonomy of the modes of operation in Section 2.5. Details relating to the actual operation, and some considerations that need to be kept in mind when establishing a sensor are provided in Section 2.6. Honeypot and Honeynet technologies, and their relationships to network telescopes are briefly addressed in Section 2.7. The Chapter concludes with Section 2.8 providing a brief overview of some of the related work done using Network Telescopes.

2.1 Evolution of Network Security

Operational Network Security and the closely related field of Network Security research, can probably be traced back to the events surrounding the self propagating code found on the then fledgling Internet on 2nd November 1988. The incident became known as the Internet Worm, or Great Worm[1], although this was later renamed the Morris Worm[2] after its author Robert T Morris Jr, then a student at Cornell University (Gardner, 1989; Eisenberg *et al.*, 1989). Gene Spafford documented much of the initial response and subsequent analysis in (Spafford, 1989b). This was followed by a spate of other research relating to the incident itself, and the impact on the Internet at large (Highland, 1989; Spafford, 1989a; Denning, 1990).

Many security researchers consider this the 'Sputnik moment' that catalysed researchers and administrators to take action to secure, what at that time was largely an open network. As a direct result of the efforts to remediate the threat posed by the worm, Carnegie Mellon university established a Computer Emergency Response Team (CERT) with DARPA funding. This organisation later be came known as CERT/CC[3] acting as a Co-ordination Center for CERTS around the world. Other responses to this were the establishment of the Phages Mailing list[4], used to discuss network threats for several years to come. The IETF also published RFC 1087 entitled *Ethics and the Internet* (DARPA, 1989).

A detailed history of the evolution of network security is beyond the scope of this research, and readers are referred to *A history of Internet Security* (DeNardis, 2007) and *Defense and Detection Strategies against Internet Worms* (Nazario, 2003). A timeline of major worms is shown in Appendix A.

[1] http://www.catb.org/~esr/jargon/html/G/Great-Worm.html

[2] http://en.wikipedia.org/wiki/Morris_worm

[3] http://www.cert.org/

[4] http://securitydigest.org/phage/

The relevance of this to network telescopes is that much modern malware is both network aware, and programmed to actively propagate over the Internet. Network Telescopes provide a means of observing the propagation (and potentially any attack activity) allowing for detailed characterisation and analysis of these evolving network threats. Several worms that exploited vulnerabilities in the Microsoft Windows RPC/DCOM stack (via 445/tcp) are discussed in Chapter 7. Brief mention is also made of the SQL Slammer and Witty worms in Chapter 5.

2.1.1 Mitigation and Defense

In response to the increasing threat levels experienced by Internet connected hosts, many organisations operate fairly tight firewalls, filtering, and subsequently rejecting significant portions of internet traffic. Yet many of these same organisations allow for unrestricted outbound communications — as evidenced by the mass of traffic to 445/tcp after the advent of the Conficker worm. This research proposed the use of network telescopes, in two complementary roles. The first of these is to allow researchers to understand the propagation, and provide an early warning system regarding emerging threats. The second is to apply what can be learned though the deployment of a network telescope to augment the current security solutions commonly in place within an organisation.

One of the biggest issues facing both vendors and operators of Intrusion Detection (IDS) and Intrusion Prevention (IPS) systems, is the increasing volumes of network traffic. The application of Network telescopes (particularly those offering low interaction as discussed in Section 2.5) can provide a means of dealing with the deluge of information.

2.2 Incoming Traffic

The primary advantage of using a network telescope as a means of capturing traffic for analysis over standard methods is that, due to the fact that no legitimate services are running in this address space, we can assume that all traffic being seen by the telescope can be classified as a basic level as being both unwanted, and therefore potentially hostile.

Working on the basis of above classifications, we are able to focus on the varied attributes of the data, rather than having to first perform the somewhat involved separation of legitimate traffic as would be required if data was being collected from a 'live' or production network or host. For the purposes of this paper, all this traffic will be processed together and no further differentiation will be made, although this separation (and the process in order to achieve this) may form the basis of future work.

Traffic originating from Internet hosts and arriving at a network telescope can be classified as one of the following three broad categories:

- BACKSCATTER - Traffic resulting as the monitored address space being used for spoofing elsewhere, most often as decoy scans (Komarnitsky, 2000; Pouget *et al.*, 2008; Lyon, 2009), as Denial of Service (DoS) attacks, or as a result of misconfigured hosts. This traffic consists primarily of certain classes of ICMP traffic and of TCP packets with RST (reset) or SYN (synchronise) and ACK (acknowledgement) flags set.
- MISCONFIGURED -This traffic could be classified as partially backscatter, as well as potential aggressive traffic, and is most often resultant of misconfigured hosts online (Moore *et al.*, 2001; Cooke *et al.*, 2004; Kumar *et al.*, 2005).
- AGGRESSIVE/HOSTILE - The bulk of the observed traffic seen on the network telescope can be classified as aggressive, or potentially hostile. This includes the obvious cases of overt network scans - both via ICMP and TCP scanning and obviously hostile packets with exploit payloads (these only being seen in the case of UDP based exploits due to its connectionless nature). The remainder is made up of traffic that can be grouped as being originated by various automated scanning agents such as Internet Worms and related Malware (Costa *et al.*, 2005; Zou *et al.*, 2005).

2.3 Traffic Classification

While all traffic received at the network telescope monitoring node can be seen to be unsolicited, the collected backscatter can be further classified under a number of categories. Strictly speaking backscatter can be regarded as traffic that is passive, and as such distinct from the active traffic recorded on the sensor. The term is, however, often misused in the sense of referring to all traffic that is not directly associated with communications of hosts on a network. This section builds on the view that traffic can be divided into the two broad classes of active and passive. Further discrimination is performed within these categories.

This classification of and careful discrimination of traffic is important if the data collected by the telescope is to be used for other purposes such as automated countermeasures, or for feeding into larger security management systems. Without the discrimination, one risks further prejudicing victims of denial of service attacks and other networked hosts which have responded to spoofed packets.

An overview of the packet structures for TCP, UDP and ICMP is provided in Appendix C, which are relevant to the following discussions relating to traffic classification.

2.3.1 Passive Traffic

Passive traffic can be defined as traffic from which no legitimate response can be expected from a system's TCP/IP networking stack when received. As such it is unlikely that a potential attacker or instance of malware will be able to determine anything about the target system. The traffic observed can be seen to be the result of the following types of activities which result in the reflection of traffic from the originating machines to the telescope sensor. All of these require that the source address of datagrams be spoofed to be within the IP address range monitored by the telescope sensor.

- Scanning Activity making use of decoy scans
- Denial of Service or Flooding — this can take the form of ICMP 'ping floods' utilising ICMP Type 8, datagrams and potentially generating Type 0 responses (resulting in a bandwidth consumption of both up and downlink). TCP SYN floods are also a common means of attempting exhaustion of resources on a target system.
- Misconfiguration — a machine is misconfigured, using the address range monitored by the telescope, this is most likely to the address field being

entered incorrectly, either on endpoint systems, or in Network Address Translation (NAT) gateways.

- Mangled packets — There is an extremely small portion of traffic that is completely nonsensical and non protocol conformant. The origins of this are unknown but could be due to hardware or software error.

Reviewing this further and focusing on the three primary IP payloads observed the following characteristics can be used to classify traffic as passive. The basis on which these tables were produced was to choose packet configurations that would not result in a response being sent. This does however assume that there is a direct open channel of communication between hosts and that a firewall is not in place that responds to filtering by sending TCP Reset (RST) or ICMP Unreachable (Type 3) datagrams. Also omitted from the following are packet configurations which are undefined.

Table 2.1 provides a summary of the rules that were used for determining passive traffic. The BSD Packet Filter (BPF)[5] (McCanne and Jacobson, 1993) syntax is shown as it is the most commonly implemented cross platform packet filtering language. The use of `tcpflags` and `icmptypes` are simple shortcodes for the numeric byte index into the packets provided by the BPF in the libpcap library in order to make more semantic sense when constructing filters. It is worth noting that numeric values are used for the last three items in the ICMP list as they are not currently supported as shortcode mnemonics in tcpdump[6]. Numeric codes could, however, be used for any of the others.

While this class of traffic can be seen to be threatening to the network in its own right, it could have other negative consequences, particularly in terms of organisational risk to reputation. Alternately it could manifest as a trigger for some kind of strike-back in the event of spoofed addressing. This strike-back could be in the form of malicious attacks (as discussed in *Aggressive Network Self Defense* (Mullen, 2005)), publication in RBL[7] abuse listings (such as SpamHaus XBL[8]), or collaborative filtering projects such as Dshield[9].

[5] `http://www.tcpdump.org/`

[6] As of tcpdump version 3.9.8 and libpcap version 0.9.8 (FreeBSD 7.0-RELEASE)

[7] Real-time Block List - commonly also referred to DNS Block Lists

[8] `http://www.spamhaus.org/xbl/`

[9] `http://www.dshield.org/`

Table 2.1: Passive Packet Configurations

TCP			
Flag		*Name*	*BPF Syntax*
RST		Reset	`tcp[tcpflags]=tcp-rst`
ICMP			
Type	*Code*	*Name*	*BPF Syntax*
0	0	Echo Reply	`icmp[icmptype]=icmp-echoreply`
3	any	Destination Unreachable	`icmp[icmptype]=icmp-unreach`
4	0	Source Quench	`icmp[icmptype]=icmp-sourcequench`
11	any	Time to Live Exceeded	`icmp[icmptype]=icmp-timxceed`
12	any	Parameter problem	`icmp[icmptype]=icmp-paramprob`
13	0	Timestamp reply	`icmp[icmptype]=icmp-tstampreply`
16	0	Information reply	`icmp[icmptype]=icmp-ireqreply`
18	0	Address mask reply	`icmp[icmptype]=maskreply.`
31		Datagram conversion error	`icmp[icmptype]=31`
34		IPv6 I-am-here	`icmp[icmptype]=34`
36		Mobile registration reply	`icmp[icmptype]=36`

2.3.2 Active Traffic

Active traffic is defined as traffic which is expected to elicit a response of some kind when processed by a target system's TCP/IP stack. A summary of the primary active traffic configurations for TCP and ICMP can be seen in Table 2.2. The TCP scanning techniques attempt to elicit either a RST packet in response from posts not listening, or a packet as part of phase 2 of the TCP 3-way handshake (Postel, 1981d) to result in a SYN+ACK packet in case of SYN scanning on ports that are listening. The ICMP types discussed all elicit response of the types detailed in Table 2.1, with ICMP commonly making use of request and response type pairing such as that used by the echo functionality utilised by 'ping'. This is by no means an exhaustive list of possible scanning techniques, but represents the most common options offered by scanning tools such as Nmap[10]. Related work on the responses able to be elicited from remote hosts running various operating systems has been presented in Irwin (2009). This work presented means of being able to identify remote host operating system families though the use of a single packet. Further details on advanced aspects of ICMP scanning can be found in (Arkin, 2000).

[10] `http://www.nmap.org/`

Table 2.2: Active Packet Configurations

TCP		
Flags	**Name**	**BPF Syntax**
NONE	NULL Scan	`tcp[tcpflags]=0`
FIN	FIN Scan	`tcp[tcpflags]=tcp-fin`
SYN	SYN Scan	`tcp[tcpflags]=tcp-syn`
PSH	PSH Scan	`tcp[tcpflags]=tcp-psh`
URG	URG Scan	`tcp[tcpflags]=tcp-urg`
URG\PSH\FIN	'XMAS' Scan Variations	`tcp[tcpflags]= tcp-urg&tcp-psh&tcp-fin`
PSH\FIN		`tcp[tcpflags]=tcp-psh&tcp-fin`
URG\FIN		`tcp[tcpflags]=tcp-urg&tcp-fin`
URG\PSH		`tcp[tcpflags]=tcp-urg&tcp-psh`

ICMP		
Type	**Name**	**BPF Syntax**
8	Echo request (ping)	`icmp[icmptype]=icmp-echoreq`
13	Timestamp	`icmp[icmptype]=icmp-tstamp`
16	Information request	`icmp[icmptype]=icmp-ireq`
18	Address mask request	`icmp[icmptype]=maskreq`
30	Traceroute	`icmp[icmptype]=30`
33	IPv6 where-are-you	`icmp[icmptype]=34`
35	Mobile Registration Req	`icmp[icmptype]=36`

2.3.3 Other Traffic

Other than the two classes of traffic mentioned above there still remains a portion of traffic which cannot be classified with certainty as being definitely active or passive without further context based analysis. The major constituent of this is TCP transfer traffic, that appears to be part of an established session, and as such has the acknowledgement (ACK) flag set. Of the TCP traffic with this flag set, probably the most contentious portion is packets containing both the synchronise and acknowledgement flags - so called SYN+ACK packets. These packets have long been used as a means of trying to subvert firewall and IDS rules when scanning by tools such as Nmap. They can also be used in the active determination of remote operating systems.

The scenario where they may actually constitute random scatter is when they are being sent in response to SYN packets with forged IP source addresses within the monitored network range. In such cases the resultant packet is simply following the second step of the TCP 3-way handshake (Postel, 1981d; Stevens, 1993). This can be as the result of a misconfigured device connecting or, more likely, that the monitored network address range is being spoofed as part of a scanning decoy or denial of service attack. Packets appearing with only the ACK flag, could also be used as means of scanning to determine the filtering state of a TCP port (such as by using the -sA option to Nmap[11]). A network telescope should not observe traffic of this type resulting from any kind of normal communications. Consequently TCP packets with the ACK flag set cannot be easily classified, although those containing SYN+ACK are more likely to be reflected traffic due to source address spoofing. A detailed discussion of network scanning techniques can be found in *Nmap Network Scanning* (Lyon, 2009).

UDP traffic also forms a component of this traffic that is difficult to classify. Since it is stateless by design (Postel, 1980), no initiation or response can be inferred from the packet headers, necessitating deeper inspection of the payloads. What has been observed is that UDP datagrams to the high ephemeral port ranges tend to be traceroute packets, DNS responses or vagrant traffic from P2P file sharing applications. By contrast those targeted to lower order ports tend to be malicious such as in the case of those used by the SQL Slammer worm and the Winpopup messages described in Sections 5.2.2 and 5.3.3 respectively. UDP packets directed

[11] A good overview of supported scanning methods can be found at `http://nmap.org/book/man-port-scanning-techniques.html`

to ports with no service listening, are likely to result in an ICMP Type 3.3 (Port Unreachable) message being generated. Listening services may return a UDP datagram, depending on the service, and whether the Initial packet used to probe, contained a valid payload in the context of the service.

Similarly ICMP Type 3 messages with codes 0,1,2,6,7,9,10,11 and 12 can also be emitted by routing equipment, or other devices performing network filtering, which may indicate the port is filtered. This information can be used for scanning, enumeration and reconnaissance of vulnerable services.

In order to be able to accurately determine the intention of UDP traffic, a researcher would need to perform analysis on the payloads contained within the datagram. Some of this has been performed in Sections 5.2.2 and 5.3.3.

2.4 The case for monitoring

In order to be able to understand and react to emerging threats, information security researchers both in academia and at security vendors first need to be able to isolate and study both the behaviours of the actual binary files associated, but also the behavioural aspects relating to the global network — the Internet. While security vendors in particular have a number of ways in which samples of potential and emerging threats are collected, and other projects such as VirusTotal[12], provide a means for other researchers to submit data, these lack any means of assessing the impact on the network, rather being utilised (and specialised) towards the analysis of the underlying malware.

2.4.1 Passive Monitoring

Building on the need to perform network monitoring, one can refine the concept. The first of the two biggest issues with traditional monitoring schemes is the problem of having to differentiate between legitimate or production traffic and that which is not intended. The second issue is dealing with the increasing volume of traffic on modern networks.

[12] http://www.virustotal.com/

The application of network telescopes to the monitoring issues above can in some ways mitigate the growing challenges. This can be achieved, by monitoring only the traffic that is not part of production traffic, and therefore already potentially suspect. Systems such as those described in this document that do not emit any response are very difficult if not impossible for remote persons to differentiate from unallocated address space, with no hosts listening.

2.4.2 Distributed Monitoring

The principle of monitoring networks can be extended to implement distributed monitoring. This is one of the main arguments behind the greynet variant of Network Telescopes, where multiple smaller sensors are used, or alternaltey a single sensor has multiple smaller address allocations to monitor. If an organisation has larger address space, or is lucky enough to have numerically non-contiguous address space (such as at multiple geographically dispersed offices), distributed monitoring makes sense. This monitoring can either be passive as described above, or possibly offer higher levels of interaction as introduced in the Telescope Taxonomy section following. One advantage of distributed monitoring is that even with relatively small allocations, one has a higher likelihood of observing events of interest. This is especially true if the address blocks being monitored are numerically dispersed. An example of the advantage given by distributed monitoring is given in Section 7.4.

2.5 Telescope Taxonomy

While the concept of a network telescope has been discussed since 2000 (Moore, 2002; Nazario, 2003), there are varying definitions as to what exactly the concept means. The following draws together three primary classes of systems that fall under the umbrella term of a network telescope, each category being regarded as progressively more interactive or 'live'. The logical progression of increasingly interactive monitoring is the use of honeypot and honeynet systems as discussed in Section 2.7.

2.5.1 Darknets, Blackholes and Sinks

These terms are used to describe a system with no live hosts, that acts as a completely passive packet sink. This type of network is described as a *'blackhole'* by (Cooke *et al.*, 2004; Bailey *et al.*, 2005a; Cooke, 2007). The naming to some extent relates to the likeness of this form of operation to the astronomical entity which sucks in all matter and does not emit anything. The term *darknet* as used by Bailey *et al.* (2005b); Ford *et al.* (2006); Cooke (2007); Sinha *et al.* (2007); Oberheide *et al.* (2007) and Oberheide *et al.* (2007), itself stems from the fact that the network block is not populated by live systems (other than possibly the capture system) and hence is 'dark'. These terms along with 'sink' are most commonly used interchangeably with that of 'network telescope', and refer to the type of sensor implementation used in this research, the setup of which is described in Chapter 3.

The term darknet[13] is also used in the mainstream media in a more pejorative sense, referring to hidden distribution networks, or means of distributing illicit content. This widespread acceptance followed the publication of T*he Darknet and the Future of Content Distribution (Biddle* et al.*, 2002).* This terminology has now become used with in the Digital Rights Management(DRM) research space as well such as in Bethencourt *et al.* (2007).

2.5.2 Dimnets

While no canonical definition of this term exists, this form of telescope would appear from mention in mailing lists and other discussions to be an implementation of a network telescope where the IP network address block is sparsely populated by live systems (conceptualised as being 'on' or 'light'). Hence the network could appear to be dimly lit from an 'illumination' perspective. This term could be used to describe the network telescope operated by CAIDA (2005; 2006; 2007) since there are some active hosts on the /8 network address block. Other Telescopes using very large address spaces are likely to fall into this category. Traffic from active hosts is either stripped out from captures as part of post-processing, or the subnets in which these hosts reside are excluded from the capture.

[13] `http://en.wikipedia.org/wiki/Darknet_(file_sharing)`

2.5.3 Greynets

This term seems to have also developed a pejorative connotation in popular understanding, being associated with botnets and other malware command and control (C&C) channels[14]. In the context of network telescopes, the term was coined by Harrop and Armitage (2005a,b). Their definition of the term is (Harrop and Armitage, 2005b, p1):

> Greynets are collections of non-contiguous blocks of IP addresses that are 'dark' in the classical darknet sense, but interspersed between groups of 'lit' IP addresses.

This can be extended to be understood to be the use of smaller 'shards' of IP address space, rather than the traditional telescope implementation where Larger contiguous blocks (usually of size /16 or greater) are used. The IETF has published an informational note in RFC 6018 (Baker *et al.*, 2010a) detailing the application of these to both IPv4 and IPv6 address space.

Alternately, continuing with the concept of a dimnet above, it can be understood to be a network telescope which has a higher proportion of live hosts such as in the case where a small percentage of spare operational address space in a data centre is being used for monitoring. Greynets could also used to describe a telescope running a low level of interaction in order to capture more traffic than would otherwise be possible using traditional capture configurations. The most likely type of low interaction activity would be an implementation of a TCP SYN/ACK response spoofer. Such a system allows for the completion of the TCP 3-way handshake after which data payloads can be seen, something not possible on a completely passive sensor. This shortcoming of traditional telescope operation is discussed in detail in Section 3.2, and in the Chapters constituting Part II of this work. While technically this could be considered to be an active sensor, the extremely low level of interaction allows it to be included here.

2.6 Network telescopes

The concept of Network Telescopes as a means of monitoring activity on the Internet at large, has developed progressively since early 2000. They have become

[14] `http://en.wikipedia.org/wiki/Greynet`

increasingly popular over recent years as seen in Section 2.8 and are on occasion referred to via other terms such as darknets. The taxonomy of terms is discussed in Section 2.5 above.

In essence a network telescope is a passive sensor system that collects incoming traffic or 'radiation' from the Internet. This radiation is constituted from multiple source systems and traffic types. The analysis of this collected data can provide useful insight into the operation of the Internet, or even particular events such as worms or distributed denial of service (DDoS) attacks. Over the last few years researchers have focussed on using telescopes for DDoS analysis as discussed in Moore *et al.* (2001). Data collected has been successfully utilised for worm analysis, particularly that of Code Red (the first worm observed on a Network Telescope) (CERT, 2001; Moore and Shannon, 2001; Moore *et al.*, 2002); and Witty (Shannon and Moore, 2004a,b; Kumar *et al.*, 2005).

In the case of Kumar *et al.* (2005) the researchers were able to perform detailed analysis of the Witty worm based on the traffic observed, to the extent of evaluating the number of physical drives present in infected systems and the probable identification of 'patient zero'. This was achieved through the analysis of data collected by a network telescope.

There is a small portion of other traffic (observed to be less than 1%) that makes up the collected total. These packets have no obvious purpose and generally consists of datagrams with strange and non protocol conformant values. Currently very little research has been done on this type of traffic. A review of some of the anomalous traffic observed in this research is reported in Chapters 5 and 6.

The operation of a network telescope requires careful planning involving the system operator, the potential data users, and the host organisation. The simplest telescope can be constructed from a system with a single network interface card and relatively low CPU specifications. The design of a basic telescope is illustrated in Figure 2.1. In this configuration, the router routes an assigned block of addresses to the telescope system. Possibly the most significant consideration from a hardware perspective is to provide sufficient disk storage for captured data on the telescope system.

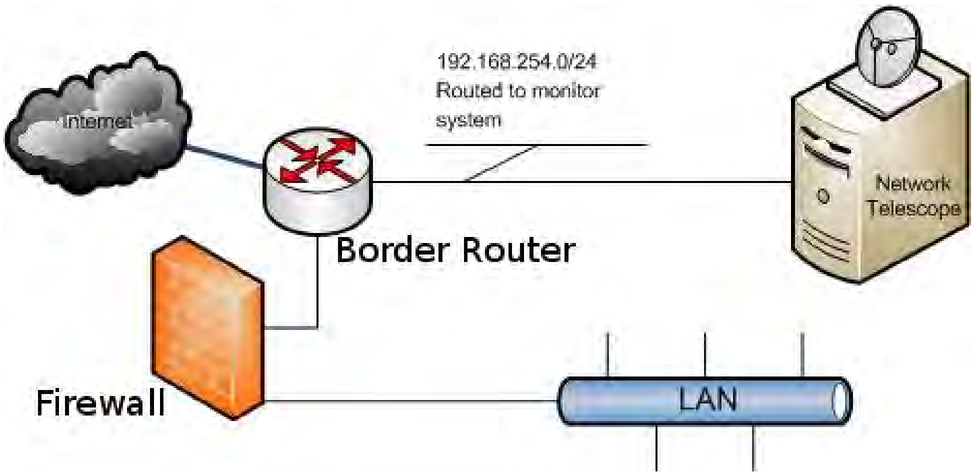


Figure 2.1: Basic Network Telescope

2.6.1 Disk space

Consideration needs to be made when sizing the storage of the telescope system. For long term storage, data is generally best kept on a system other than the operational telescope sensor itself. The volume of system storage is dependent on a number of factors, the most important being the anticipated volume of traffic in terms of the rate of arrival: how much data is to be to be logged per packet and the size of the network block being monitored by the telescope.

Average packet rates vary between 1 213 packets per hour[15] on a small (/24 sized telescope) to several million on a larger telescope such as that used by the CAIDA project (Shannon *et al.*, 2005, 2006, 2007). A more accurate calculation should be based on the packet rate per IP and the average packet size. The average size of the 40 million packets recorded on the Rhodes telescope is 102 bytes. Packets are recorded as observed 'on the wire', thus including the Ethernet level headers as the datagrams are routed to this system over an Ethernet network. As discussed in Section 5.4, the observed average is biased toward smaller packets resulting from the high proportion of TCP connection attempts observed, particularly following the advent of the Conficker worm in November 2008. The average size of packets recorded prior to this outbreak was 126 bytes. By virtue of the fact that a network telescope operates in a passive mode, one would not expect to see many packets with large payloads, other than UDP and ICMP datagrams which tend to be less prevalent than TCP. Payloads are possible for these two protocols, since no handshake or connection setup is required. Despite the lack of a capability to

[15] This is as averaged over the entire 50 month capture period.

Listing 1 Telescope Storage Sizing

$$Storage_{interval} = (PacketRate \times AvgPacketSize) \times TelescopeSize$$
$$Storage_{total} = Storage_{interval} \times IntervalCount$$

complete the necessary 3-way handshake, vagrant TCP datagrams are occasionally observed, complete with payloads.

While the above provides a guideline, one may also need to factor in the overhead imposed by the capture storage mechanism. A safe baseline to work on is to assume that packet sizes captured will most likely have a maximum of 1 500 bytes, based on the maximum MTU for IP packets on most Ethernet networks. Although particularly unlikely in the context of a network telescope, which is not intended to observe packets with payload as part of its normal operations, it serves as a value that is likely to cater for a worst case scenario. This sizing will also allow for some headroom should the packet rate assumption be incorrect. Modern hard-disk based storage is relatively inexpensive, and the price per megabyte has continued to drop over recent years. Regular rotation and archiving of packet captures on the operational system and migration to a long term data-store should also ensure that the storage resources are not exhausted.

Observed rates per monitored IP address averaged at four packets per IP per hour, but spikes in excess of 9 000 have been observed. Using the algorithm detailed in Listing 1 a rough estimate of sizing can be made, dependant on the period data is intended to remain on the collector device. It is recommended that the resultant value be rounded up to the next largest drive size to allow for some growth and flexibility. This calculation assumes that a completely passive telescope will be used and as such the majority of packets will be small (typically <64 bytes). Should a telescope with a higher level of interactivity be used, packet sizing and rates may need to be adjusted accordingly. An example of sizing for a passive telescope as described is provided in Listing 2.

The actual bandwidth available to the telescope should also be considered when making sizing decisions as the bandwidth itself may limit the volume of data that can be captured. While this may not be a serious consideration in first world networks, in the developing world where links in the sub 2Mbit range are common, congestion could be a problem (Wei and Mirkovic, 2008; Wustrow *et al.*, 2010). This is particularly important when a new telescope is being established, with additional network address space being allocated to an organisation. Reusing

Listing 2 Example Telescope Sizing

This example makes the following assumptions:

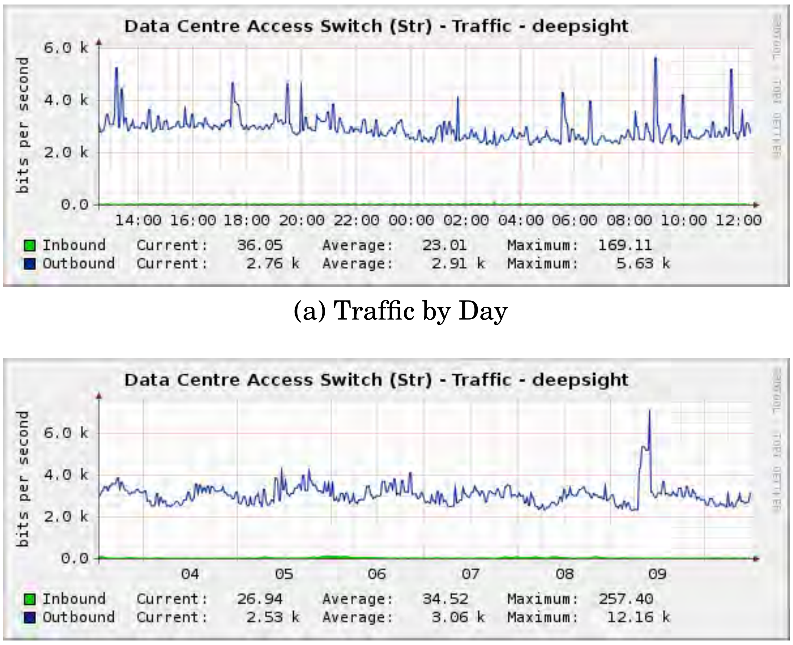
- That a telescope will be established using the upper half of a class C address block that an organisation has available, providing 127 addresses (since the broadcast address cannot be used).
- Packet size is determined to be $AvgPacketSize = 1500bytes$.
- Packet Arrival rate will be 10 Packets/IP/Hour.
- The interval period is one hour.

Therefore the storage required per Interval can be calculated as:
$Storage_{interval} = (10 \times 1500) \times 127 = 1905000bytes \backsimeq 1860Kbyte$

If a /16 telescope is used, even with $AvgPacketSize = 150bytes$, the requirements increase significantly:
$Storage_{interval} = (10 \times 150) \times 65535 = 98302500bytes \backsimeq 93.7Mbyte$

If a week of data is required to be stored (168 hour intervals) in the case of the larger system
$Storage_{total} = 93.7 \times 168 \backsimeq 15.37Gbyte/Week$

This can be reduced to $\backsimeq 3.84\,Gbyte/Week$ when compressed using lzop.
Factoring in compression this could be reduced by between 75-85% giving a storage sizing of 3.8-2.3Gbyte per week, dependant on payloads and compression algorithm. Extrapolating this to a year would result in storage needs for compressed data of approximately 200 Gigabytes, assuming the packets were of this size. Sizings would increase significantly if systems with higher levels of interaction were used, as these would complete the TCP three-way handshake, increasing both the packet count, and the potential for capturing packets containing payloads.



(a) Traffic by Day

(b) Traffic by Week

Figure 2.2: Example traffic graphs for the researcher's network telescope

address space already allocated to an organisation is likely to have a minimal impact in bandwidth utilisation (in the case of a passive monitoring solution) as the traffic now destined to the telescope has already been traversing the upstream link. Further bandwidth may have been consumed due to the traffic destined for unallocated address space being discarded by the border router as unroutable, which in itself may have resulted in ICMP error messages of Types 3 (Unreachable) or 11 (TTL exceeded) being sent back out onto the Internet. Example traffic graphs of the researcher's telescope system are shown in Figure 2.2. showing the fairly consistent level of traffic received.

2.6.2 Logging

The choice of what to log on the telescope system also has a strong bearing on the disk sizing already discussed. Current recommendations are to record the entire packet as observed 'on the wire' as this allows for the most flexibility in future research utilising the collected data. In some situations it may be worth only recording packet information such as source and destination systems and port information. This can be captured using only the first 64 bytes of an IP datagram, and still have some space available for payload content where applicable. Specific clean-up can be done as post processing or with independent protocol specific capture streams. Certain high volume environments, such as the telescope operated by the University of Wisconsin, only captures every 10^{th} packet (Kumar *et al.*, 2005). Such action is useful for trend analysis, but has shortcomings if specific scanning techniques are wanting to be studied such as in the work done by van Riel and Irwin (2006a) and Barnett and Irwin (2008).

Care should also be taken relating to time and time zones. System time should be kept as close to 'true' as possibly using solutions such as NTP, with suitable upstream servers. The choice of time zone that a system may reside in can also be important, particularly in areas where daylight saving time is used, as this could result in certain hours 'doubling up'. Where possible such changes should be noted. One approach in such situations is to set the system to UTC/GMT, rather than to a local time zone. Whatever approach is decided on, it is important that it is suitably documented, and communicated to others that may use the datasets. The circadian traffic patterns discussed in Sections 6.2 and 7.2 rely on having correct time zone information, particularly when wanting to perform adjustments to timestamps based on geolocation of source countries or cities.

Once matching packets have been captured to file, these can in most cases efficiently be compressed for archiving. This operation is usually performed as part of the rotation and subsequent compressions of the capture files after a given interval. The interval selected for rotation is likely to be dependant on traffic volumes, or based on common temporal intervals such as daily, weekly or monthly. Certain high volume nodes may want to rotate data logs hourly, such as done with the datasets produced by the CAIDA telescope (Moore *et al.*, 2004), or even more frequently if needed. Telescopes monitoring large (>/24 sized) netblocks, may also use separate capture files to divide up the network address space being monitored. Common compression utilities used are gzip[16] and bzip2[17] which are present on most Unix-like systems. The CAIDA project uses the lzma compression as implemented in the lzop[18] and 7zip[19] utilities, the latter providing convenient cross-platform support. One advantage of this compression algorithm is that while not offering the best overall compression ratio, the trade-off is that operations to compress and decompress files are quite quick.

[16] `http://www.gzip.org/`

[17] `http://bzip.org/`

[18] `http://www.lzop.org/`

[19] `http://www.7-zip.org/`

Table 2.3: Timing of Compression Algorithms

Tool	$Time_{compress}(s)$	**% Size decrease**	$Time_{decompress}(s)$
gzip	18.916	81.10	3.588
bzip2	122.562	82.50	31.995
lzop	2.395	77.20	3.617

The system used was a Intel Pentium 4 3.2 Ghz CPU with hyper-threading disabled, 1Gig DDR Ram, and 2x320Gig SATA 7200rpm drives. libpcap files of 256MB were used for testing. Average results reported.

Timing results obtained from compressing sample pcap files using these three utilities are shown in Table 2.3. From these results it is evident that there is a trade-off between compression ratio, in which case bzip2 is appropriate, and speed both for compression and decompression, for which lzop performs best. The gzip algorithm performs well and is widely supported across a number of platforms. The difference in compression between lzop and gzip may be small, but on large datasets can prove significant over time. It is also recommended that suitable cryptographic hashes such as as those from the MD[20] and SHA[21] families be applied to the data files and logged to ensure their integrity when processing in the future. These checksums can be generated using specific utilities present on most unix platforms, or using the cross-platform OpenSSL cryptographic toolkit[22].

2.6.3 Security

There may be some security concerns within an organisation when a network telescope is established. Section 3.2 discusses some of the specific issues that the researcher had in establishing the network telescope at Rhodes University. The security risks of adding a telescope monitoring system to a network are less than that of adding any standard Internet facing service for a number of reasons:

Reduced attack surface — The ideal behind a passive telescope implementation is that it will not respond to any incoming traffic. While there may be the need for some administrative traffic, this could either be tightly firewalled, or preferably routed via a second dedicated administrative interface. A system level firewall or packet filter should be employed to ensure that no responses to incoming traffic are generated. Alternatively, hardware solutions

[20] http://en.wikipedia.org/wiki/MD5

[21] http://en.wikipedia.org/wiki/SHA

[22] http://www.openssl.org/

are available in the form of various specialised network taps or appropriately configured switch ports which ensure that the network is *read only*. This is in contrast to a standard host which is offering active services to the Internet. Putting these measures in place is important in ensuring that the telescope remains passive. Appropriate changes will be requires if higher levels of interactivity are to be implemented.

Denial of Service — Concerns relating to the possibility of denial of service against the telescope system can also be addressed. Since the system is configured to not respond to any incoming traffic there is no real difference between the system logging traffic, and then discarding it, and the outer gateway discarding it, due to the address space being unallocated. Depending on how the border router is configured, it may not even be possible to determine remotely that there is any difference, if no ICMP Type 3 or 11 datagrams are emitted when traffic is discarded. There is also a very low likelihood of attracting specifically targeted attacks since in the case of a passive sensor, it does not actively partake in any Internet communication. Should any information be published publicly relating to the telescope, care should be taken to suitably anonymise or redact information. This is important to ensure that the address space used is not targeted in the future, potentially resulting in undue bias in the data collected.

Vulnerability to Malware — A telescope system on its own is probably less vulnerable to malware and exploitation than any other Internet facing system. Care should, however, be taken if an Intrusion Detection System (IDS) is being run on the platform as well as there are well documented cases of passive infection of monitoring devices, particularly with the Witty Worm (Shannon and Moore, 2004a; Paxson, 2005; Kumar *et al.*, 2005) which exploited a buffer vulnerability in the decoding of ICQ packets in ISS security software (eEye Digital Security, 2004). Exploits for Snort (Roberts, 2003; Internet Security Systems, 2003), tcpdump (The Electronic Souls Crew, 2002; SecuriTeam, 2003, 2005) and WireShark protocol decoders (SecuriTeam, 2007, 2008; VUPEN Security, 2010) have also been published. Operators should ensure that systems are suitably patched. Appropriate firewall rules on the capture interface can also aid in the mitigation of potential threats. Care also needs to be taken when processing datagrams, as these may potentially contain exploit code in the packet payloads such as in the case of the SQL Slammer worm. Antivirus scanners on systems being used for analysis should be suitably configured so so as not to interfere with capture files containing potentially hostile code.

Bearing the above in mind, a correctly configured system should prove an asset to an organisation's Information Security toolset, rather than a liability. The application of a network telescope in both research and operational roles is discussed in Chapter 9. Some potential threats to passive sensors are discussed in Shinoda *et al.* (2005).

2.7 Honeynets

High interaction data collection systems such as honeynets are beyond the scope of this research, A discussion is included here for completeness sake, as they can be used to complement a network telescope, particularly if a researcher is trying to isolate samples of malware. The concept of a 'honeynet'[23] was proposed in 1999 by Lance Spitzner in his paper *To Build a Honeypot* (Spitzner, 1999) as an extension of the already established concept of running honeypot systems, describing the extension as a means of emulating an entire collection of vulnerable systems, or a simulated network.

A honeypot is a closely monitored sacrificial system that is placed on a network with the intention that it is to be available for compromise. The use of such systems allows for potential early detection of automated malware propagation. The second and possibly more prevalent use of such systems is to allow for the monitoring and analysis of live intrusions in order to understand both how exploitation techniques evolve and to gain understanding of the exploits and toolchaines used by intruders. While the name itself has only recently been in widespread use, the concept of inspecting an intruder's moves has been documented as far back as 1989 in the *Cuckoo's Egg* (Stoll, 1989) and the following year in *An Evening with Berferd* (Cheswick, 1990b). The new replay techniques available in modern virtualization systems such as Vmware potentially allow for even more detailed replay analysis of exploitation.

Honeynets are an extension of a single honeypot system, which can operate either as a collection of such systems, or specialised software such as honeyd[24] and

[23] http://honeynet.org/

[24] `http://www.honeyd.org/`

nepenthes[25] which allows for the emulation of entire virtualised networks.

The fundamental difference between the use of Honeynets and honeypot systems and the network telescope is that by their design they are an active technology and as such they can be detected, and some Malware has active defences against such systems (Chen *et al.*, 2008; Sun *et al.*, 2008). While honeynets are worth mentioning in the context of network monitoring, they are outside the scope of this research. The reader is referred to the following for further detail: Spitzner (1999); Provos (2004); Anagnostakis *et al.* (2005); Andreolini *et al.* (2005); Portokalidis and Bos (2007); Vanderavero *et al.* (2008)

An excellent detailed instructive document on setting up a full featured honeynet system using the nepenthes medium interaction software, to capture and process malware can be found in the report provided by Beck *et al.* (2007).

2.8 Related Work on Network Telescopes

A growing body of material has been published relating to the use and applications of Network Telescopes as research tools. While by no means an exhaustive list, this section highlights some of the major areas in which this research tool has been applied, and relevant publications resulting from this. Specific examples are referred to in the remainder of the text in the context of the network telescope data under discussion.

2.8.1 Distributed Denial of Service (DDoS)

The backscatter datasets provided by CAIDA (Shannon *et al.*, 2005, 2006, 2007) are the best examples available. Notable work published utilising network telescopes for understanding this type of traffic are Moore *et al.* (2001); Stavrou *et al.* (2005); Zou *et al.* (2006a); Kompella *et al.* (2007); Pouget *et al.* (2008).

[25] `http://ostatic.com/nepenthes/`

2.8.2 Malware Characterisation

Network telescopes have been used successfully to characterise malware distribution, and propagation. The analysis of the traffic generated by the Witty worm (Shannon and Moore, 2004a; Weaver *et al.*, 2004; Shannon and Moore, 2004b; Paxson, 2005), the work by Kumar *et al.* (2005) is probably the most successful analysis to date. CodeRed ahas also been analysed in Moore and Shannon (2001); Cai *et al.* (2007); Moore *et al.* (2002); Castaneda *et al.* (2004) . More recently data collected on network telescopes has been used by researchers in gaining a greater understanding of the Conficker Worm (Hick *et al.*, 2009; Aben, 2009; Irwin, 2010). This worm is studied in detail from the perspective of the traffic recorded on the researcher's own network telescope in Chapter 7.

More generic network aware malware has also utilised data from network telescopes as in Harder *et al.* (2006); Hu *et al.* (2007); Pouget *et al.* (2008); Vanderavero *et al.* (2008).

2.8.3 Network Traffic Characterisation

The final category where network telescope sensors have been applied is in gaining a better understanding of the total traffic that is received by a sensor. The seminal work in this regard is by Pang *et al.* (2004), with the recent update to this work by Wustrow *et al.* (2010) providing valuable insight. Other notable publications in this are are Moore (2002); Cooke *et al.* (2004); Harrop and Armitage (2005b,a); Harder *et al.* (2006); Pemberton (2007); Vanderavero *et al.* (2008); Irwin *et al.* (2007); Barnett and Irwin (2008). The researchers own work in this regard has been referred to in the body of the this document, primarily in the chapters constituting the analysis portion of the study.

2.9 Summary

This chapter has introduced core concepts relating to network telescopes that will form the basis of later discussion. Evidence has been given as to the varied modes of application for the data collected by a network telescope, in addition to the different ways in which a network telescope can actually be implemented. The next chapter reports on the establishment of the researchers own network telescope, and the surrounding infrastructure required for its operation. The datasets used in the remainder of the study are described.

It is a capital mistake to theorize before one has data.

Sir Arthur Conan Doyle in *Sherlock Holmes*

3

Data Collection

THIS chapter contains further information on the datasets used as the basis for the analysis discussed in Chapters 5 and 7. During the course of this research a number of different datasets were used. The primary source of data was collected using the researcher's own systems. The second major source of data used (although more for validating the tools developed – as described in Chapter 4) were samples taken from data sets made available by the *Cooperative Association for Internet Data Analysis* (CAIDA) Telescope Project[1] (Moore *et al.*, 2004; Shannon *et al.*, 2005, 2006, 2007), located at the University of California San Diego (UCSD) Supercomputing Centre in the United States of America.

The chapter opens with a discussion of the datasets used in the research. Section 3.2 discusses in detail the issues surrounding the establishment of the Rhodes University Network Telescope (RU-TELESCOPE) and the design of this data collection network. An overview of the CAIDA sensor and related datasets is presented in Section 3.3. A discussion of other datasets evaluated is presented in Section 3.4, followed by a description of the data storage methods used in this research project in Section 3.5.

[1] `http://www.caida.org/`

3.1 Data Sources

The principle dataset used in this research was collected from the smaller network telescope established by the researcher in August 2005 which, in the period under study culminating at the end of September 2009, had captured just over 40 million packets. As a comparison, this volume of traffic was present in four hours of traffic from the CAIDA project collected on the 28th of February 2008 (Shannon *et al.*, 2007). The telescope operated by the researcher was $\frac{1}{65536}th$ the size (equivalent to $\approx$0.001525%) of the sensor used by the CAIDA project, consisting of a single /24 network block rather than the /8 used by CAIDA researchers. Conventional wisdom relating to the sizing of network telescopes (Pang *et al.*, 2004; Goebel *et al.*, 2007; Wustrow *et al.*, 2010) has agreed that a large address space is needed in order to obtain meaningful data but, as discussed in the following chapters, comparable results to research done on larger sensors have been achieved on a much smaller scale using the writer's own monitoring system, albeit with the use of a significantly longer temporal baseline. The datasets of significance utilised in conducting this research are:

RU-Telescope: Data collected by the researcher provided the primary source used for the analysis work performed in this project. This dataset comprises over 40 million individual events and spans a period of 50 months from August 2005 to September 2009. It is referred to in this document as the RUSCOPE1 set. This is a completely passive telescope comprising a single contiguous, independently routed /24 network block within the larger allocation of the South African Tertiary Education Network (TENET)[2]. Details of the setup and collection of this data are discussed in Section 3.2. As far as the writer is aware, this is the first operational network telescope to be established in Africa and is one of the longest continuously recorded datasets available. In August 2009 a second /24 netblock from the University's own primary network allocation was added to the monitoring system for comparative analysis in the future (RUSCOPE2). The overlap in data between these datasets was too small to warrant inclusion of a detailed analysis in this report.

The work in this study deals exclusively with traffic received using IP version 4 addressing. An IP version 6 network telescope was operated for a period of

[2] TENET is the Tertiary Education Network which provides Internet connectivity for all higher education and research facilities in South Africa. Further details can be found at `http://www.tenet.ac.za/`.

18 months using a /48 address block. During this time the only traffic received were the probes sent by the researcher to ensure it was still operational. It is believed that telescopes implemented using IPv6 address space will be of limited value, until there is a significantly higher global adoption.

CAIDA-Backscatter: Several samples of data overlapping the time period of the RU-TELESCOPE were obtained from the CAIDA project archives. These datasets come from a large network telescope operated at the University of California San Diego (UCSD) which is capturing a significant portion of their /8 (traditional Class A) allocation. Parts of this network are used for production traffic, so it is not a true blackhole sensor as discussed in Section 2.5, as further filtering is done (Moore *et al.*, 2004) to remove active scans and probes, publishing only the remaining backscatter traffic (as discussed in Section 2.3). Details of the CAIDA project can be found at `http://www.caida.org/`. An overview of this data collection is presented in Section 3.3.

CAIDA-Telescope: Two datasets are available from CAIDA that make use of their network telescope infrastructure, but contain more data than is present in the backscatter datasets previously described. The first of these is the *Two days in November 2008* dataset comprising of two days of telescope traffic recorded by the CAIDA project on the 12th and 19th of November 2008 (Aben *et al.*, 2008). This data is of interest, as it pre-dates the detection of the Conficker A worm (discussed in detail in Chapter 7) first identified on the 21st of November 2008. An additional dataset is also available titled *Three Days Of Conficker* (Hick *et al.*, 2009), but has not been analysed in detail due to its large size (69GB). These datasets differ from the backscatter set above in that they comprise all traffic as recorded by the /8 UCSD sensor, other than identified production traffic which has been removed.

DShield/ISC: Data was obtained from the Dshield project[3] run by the Internet Storm Center (ISC)[4]. Extracts were obtained covering the time periods under investigation. The data on Dshield.org originates from volunteers who report scanning and other malicious activity against production systems — usually gathered from firewall and IDS logs. While not telescope type data, it does provide a means of evaluating trends in port scanning and traffic 'hotspots' from the perspective of a very large, geographically and topologically dis-

[3] `http://www.dshield.org/`

[4] `http://isc.sans.edu/`

persed observation base. The collected data was used to correlate general trends observed in the network telescope datasets used in this research, and to validate the trends identified in the analysis of the RU-TELESCOPE dataset.

While the above datasets were invaluable in development and validation of tools, and data processing techniques, the analysis documented in this study has been focused on the researcher's own network telescope data. The analyses presented in Chapters 5, 6 and 7 focus exclusively on the RUSCOPE1 dataset.

3.2 Rhodes Telescope

The need for a network telescope for the collection of baseline data was identified by the researcher in April 2005. The decision to collect data was due to the potential problems accessing the rather large CAIDA repository (discussed in Section 3.3), particularly around Internet bandwidth considerations. At the time only the CAIDA backscatter data was available, which was perceived to be of somewhat limited value primarily due to not all traffic being included in the available datasets. True telescope datasets, in the sense of reporting all traffic received, (as described in Section 3.1) were only released by CAIDA in mid 2009 (Aben *et al.*, 2008; Hick *et al.*, 2009).

The primary issue that needed to be resolved from the perspective of the researcher's institution, was the design and implementation of the telescope collector system in such a way as to minimise any potential risk to the institution should there be any compromise of or attack against the telescope system once implemented. The primary concern expressed by the institutional IT staff was for the potential of attracting a Denial of Service (DoS) attack directed against the telescope system and associated address space, thereby resulting in a potential compromise of the University's Internet connectivity. As of the time of writing there have been no adverse experiences associated with the operation of the network telescope.

The network telescope sensor was comprised of an independently routed /24 (formerly a Class C) sized netblock that was obtained specifically for use in this research. The implemented system agreed upon by the researcher and the university administration is shown in Figure 3.1. The use of an independent netblock not associated with the existent institutional netblock allocations allowed for the relatively easy removal of the monitored netblock from global BGP advertisements, and thus routing tables, in the event of any problems; effectively causing traffic for this netblock to stop being routed over the institution's Internet link.

The monitored netblock is, however, a component of the aggregated netblock allocated to the TENET by AfriNIC, though which the University obtains its upstream Internet connectivity. For the remainder of this research the dataset collected using this network is referred to as RUSCOPE1. The system was commissioned on the 3rd of August 2005 and in the period under study through to 30th September 2009, has logged 40 801 854 datagrams.

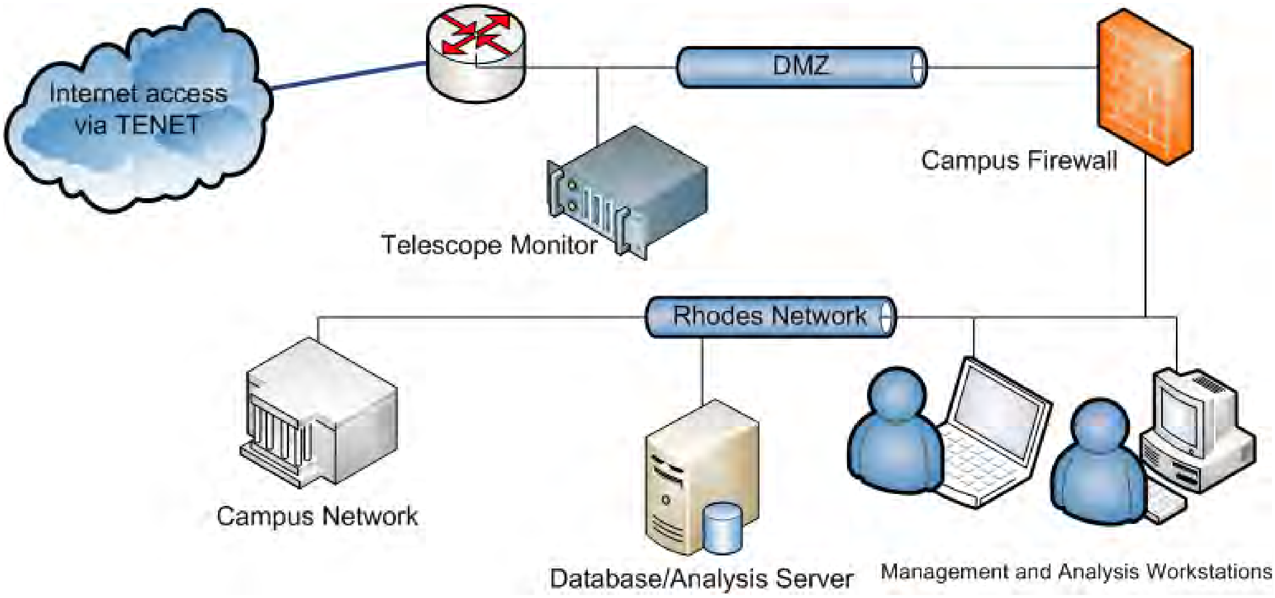


Figure 3.1: Rhodes Network Telescope System

A second network block was commissioned for use in a second telescope sensor in mid August 2009, and comes from the University's own primary address space allocation. Consisting of a similarly sized /24 netblock, this was configured to be monitored by the same physical telescope system as used for the RUSCOPE1 data, but logged to a separate collection of packet capture files. The addition of this netblock was possible due to re-organisation of the logical network and IP address allocations at the University and experience that had been gained in the four years of operation of the original RUSCOPE1 netblock. This is referred to as RUSCOPE2. Of interest to the researcher is the correlation between these similarly sized sensors which have a substantial logical distance between the address blocks (more on distances can be seen in Section 4.3). The short temporal overlap of the two datasets has precluded detailed analysis in this work, but will form part of future research based on this work as discussed in Section 10.3.

3.2.1 System Configuration

The telescope monitoring system was initially based on hardware running the FreeBSD 5.4 operating system and has been upgraded over the period of operation to FreeBSD 7.0. System hardware utilised initially was an Intel Pentium 4, 3.2 Ghz with 1GB Ram in an Intel 1U rack-mount Chassis. Storage was provided by a pair of 160 Gig hard disks configured in a RAID 1 (mirror) group in order to provide resiliency and minimise data loss in the event of disk failure. At the time of writing, the capture component is provided by a Sun Microsystems Sunfire V100 running a 548.00 MHz UltraSparc-IIe Processor with 1GB of RAM. This illustrates the low hardware requirements for data capture. Data processing is, however, significantly more resource intensive as discussed in Section 3.2.2 and is performed on separate platforms.

From the inception of the project, the server was physically co-located in the Computer Science Department data centre until January 2009, when it was relocated to the University's primary data centre. The motivation for this move was largely driven by the need for a stable power connection, which is provided by generator backup at the new location, and flexibility in adding additional address space. During 2009 there were no outages attributable to networking or power failure – in contrast to the series of power and network outages experienced in 2008. Those that did occur were the result of outages on the upstream Internet connection. This relocation also allowed for the addition of the second /24 netblock in August 2009. The primary 100Mbit network interface of the system was logically provisioned to a point on the University's DMZ. With both hardware platforms, the second on-board network interface was used for the periodic retrieval of data and subsequent upload for further processing and archiving on dedicated systems internal to the campus network. During normal operation this port remained disconnected. System configuration on the collector was kept minimal, with the system running a firewall denying all inbound traffic on the DMZ connected interface.

Listing 3 Sample packet capture script

```
#!/bin/sh
DATE=`date +"%Y%m%d-%H%M"`
PATH="/data/darknet"
NH="/usr/bin/nohup"
CMD="/usr/sbin/tcpdump"
FLAGS="-q -n -s 0 -i dark0"
cd $PATH
#IP addresses below have been redacted
$NH $CMD ${FLAGS} -w darknet-${DATE}.cap net 196.xx.xx.0/24 & >/dev/null
$NH $CMD ${FLAGS} -w darknet-ru-${DATE}.cap net 146.xx.xx.0/24 & >/dev/null
echo Capture started
```

3.2.2 Data collection and processing

Network traffic was collected using the tcpdump utility which was utilised to log packets to file. The advantage of this approach is that packets could be recorded as they arrived on the wire and, by operating in promiscuous mode, was able to capture packets with the system firewall set to deny all traffic incoming on that network interface. Files were rotated on a monthly basis. Listing 3 shows a script used for starting the capture process. Capture files were created using a time-stamp in the filename in the form of `YYYYMMDD-hhmm`, based on the start time of the capture process. These files were periodically rotated and compressed. No further processing was done on the capture system, but copies were transferred to the Analysis Server platform for long term archiving and analysis. Each capture file had a corresponding checksum calculated and stored in a separate file when rotated. Checksums were verified after transfer to the analysis platform prior to any further processing, thereby validating the integrity of the file. This was particularly important given that the files were obtained over slow Internet links often requiring multiple attempts to completely retrieve the capture files.

Analysis of the dataset was performed across a variety of systems and platforms. For general purpose processing and exploratory work, a utility was developed to parse and import the libpcap format data files into a central PostgreSQL[5] Database Server. The database was housed on a Dell Poweredge 860 with 4 gigabytes of RAM and an Intel Xeon X3220 Quad core CPU running at 2.40GHz. A terabyte of disk based storage was used, part of which was mirrored for the database partition, providing increased resiliency and performance for read operations. Various client systems and applications could then connect to this database to extract and further process data. The database server also housed a copy of the R Statistical process-

[5] `http://www.postgresql.org/`

ing language[6] which was used for high-level statistical analysis of outputs from the database. Discussion relating to the database design and the programmatic loading of captured data into the database is covered in Section 3.5, with further detail contained in Appendix D.

One important decision made with these initial processing tools was to omit packet payloads when loading datagrams into the database. This was based on the fact that the majority of the packets were unlikely to have payloads, since the TCP 3-way handshake could not complete, and this would also provide a savings on the space required in the database. The original capture files were, however, retained and used directly by tools such as InetVis, the libtrace[7] suite, and Wireshark[8], for performing detailed packet-level analysis and exploration. A discussion of the primary tools used in the analysis of the telescope data is contained in Chapter 4.

3.2.3 Data Overview

This section provides an overview of the characteristics of the traffic observed as part of the RUSCOPE1 dataset used in this research. Figure 3.2 shows the distribution of traffic, with the total number of packets received in each year, over the 50 month period of operation, amounting to approximately 2.5 gigabytes of traffic in total. It is worth noting that the nine months of traffic processed in 2009 amounts to three times as much by volume (by count of packets received) as the preceding three years combined. During the period August 2005-December 2008 20 million events were captured, with just over 20 million packets being recorded in the first three quarters of 2009. Much of this can be attributed to the advent of the Conficker worm (Microsoft, 2008b, 2009) in late November 2008 resulting in the exploitation of the vulnerability patched by the MS08-067 (Microsoft, 2008a) security advisory in October of that year. This is explored in further detail in Chapter 7 as part of the detailed analysis of traffic observed on 445/tcp. Details of annual traffic observations and a breakdown by protocol can be seen in Table 3.1.

During the capture period a number of outages were experienced. These were due to upstream Internet provider outages, routing issues, power supply failures, and networking problems internal to the campus network. Table 3.2 provides a high

[6] http://www.r-project.org/

[7] http://research.wand.net.nz/software/libtrace.php

[8] http://wireshark.org/

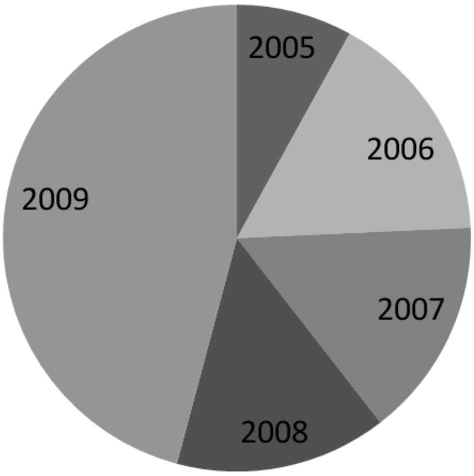


Figure 3.2: Data collected by year

Note: 2005 and 2009 are partial years representing 5 and 8 months of data capture respectively

Table 3.1: Breakdown of traffic composition by Protocol and Year

	Protocol				
Year	**ICMP (1)**	**TCP (6)**	**UDP (17)**	**Other**	**Total**
2005	396 216	2 120 994	754 649	23	3 271 882
2006	1 022 208	4 289 758	1 326 199	20	6 638 185
2007	429 006	4 826 948	937 954	76	6 193 984
2008	220 012	4 833 169	937 648	158	5 990 987
2009	302 121	17 212 374	1 191 517	804	18 706 816
Grand Total	**2 369 563**	**33 283 243**	**5 147 967**	**1 081**	**40 801 854**

Table 3.2: Summary of traffic captured for RUSCOPE1

Year	$Days_{cap}$	$Days_{max}$	%	$Days_{missing}$
2005	141	150	94	9
2006	349	365	95.61	16
2007	314	365	86.02	51
2008	352	366	96.17	14
2009	273	273	100	0
Total	1429	1519	94	90

level summary of the number of days with captured packets present in each year's worth of capture data. A day is counted as present if any recorded traffic exists for the given 24 hour period. A particularly high level of outages were experienced in 2007 due to numerous upstream connectivity problems and a faulty switch to which the monitoring system was connected. South Africa experienced a number of rolling blackouts during the first half of 2008, which resulted in 4-6 hour periods of no data. As mentioned previously, avoiding unnecessary outages such as these was one of the primary reasons for the relocation of the capture device to the University data centre at the beginning of 2009.

Overall, a 94% capture rate (by day) was achieved during the 1 519 day period of observation. This was based on an outage being defined as a period where no traffic was observed in a 24 hour window. At a finer level of detail however, this drops to 89.32% when viewed with minute granularity. At the finest level of granularity only 15.83% of the potential second periods have data; when factoring out the known extended outage periods, this still only accounts for 17.73% of the potential second periods being monitored. These results, when considered, are however unsurprising, as scanning traffic tends to be bursty, and there are observed periods of quiet even when all systems are operational. This is an important factor to bear in mind when calculating packet arrival rates as one possible metric for comparative analysis of network telescopes. Further discussion of proposed network metrics and their application to this dataset can be found in Chapter 8.

3.2.4 Data set summary

A summary of the basic characteristics of the RUSCOPE1 dataset is presented in Table 3.3 along with that of RUSCOPE2. Of immediate interest is the difference in the rates that packets were received for the two network telescopes. A detailed

Table 3.3: Rhodes Telescope Dataset Characteristics

RUSCOPE1			
Timespan	**Start Date**	**End Date**	**Outage**
1519 days 14:07:27	2005-08-03	2009-09-30	±90days/6%
Packet count	**Avg Pkt Rate**		
40801854	18.65 pkts/min		
RUSCOPE2			
Timespan	**Start Date**	**End Date**	**Outage**
42 days 10:40:43	2009-08-20	2009-10-1	0%
Packet count	**Avg Pkt Rate**		
311973	5.4 pkts/min		

Table 3.4: Breakdown of traffic composition by percentage

	Protocol			
Year	**ICMP (1)**	**TCP (6)**	**UDP (17)**	**Other**
Percentage composition by Protocol per year				
2005	12.109	64.825	23.0647	0.0007
2006	15.399	64.622	19.978	0.0003
2007	6.926	77.929	15.142	0.0012
2008	3.672	80.674	15.651	0.0026
2009	1.615	92.011	6.369	0.0040
Percentage composition by year per Protocol				
2005	16.721	6.372	14.659	2.1276
2006	43.139	12.888	25.761	1.850
2007	18.104	14.502	18.219	7.030
2008	9.285	14.521	18.213	14.616
2009	12.750	51.714	23.145	74.375

analysis of the RUCSOPE1 dataset is provided in Chapters 5 and 6. A series of metrics relating to the dataset are presented and discussed in Chapter 8, and a focus on the spike in traffic observed in late 2008 onwards, and attributable to a rise in traffic destined to 445/tcp is presented in Chapter 7.

A summary of the composition of traffic on an annual basis is shown in Table 3.4. This is the same data as previously presented in Table 3.1, normalised as percentages. The first portion of the Table shows the composition of traffic by protocol on an annual basis. The second component shows the relative contributions by each year to the total for the primary protocols observed.

3.3 CAIDA Telescope Datasets

The Cooperative Association for Internet Data Analysis (CAIDA) telescope system, located at the University of California San Diego (UCSD), has been in operation since early 2004 and provides a variety of traffic captures and other datasets[9] relating to Internet monitoring and topology. The datasets that have been used in this research are a selection of samples taken from the backscatter datasets from 2004 to 2007 (Shannon *et al.*, 2005, 2006, 2007) and the November 2008 Telescope Data set (Aben *et al.*, 2008). The fundamental differences between the CAIDA datasets and those captured on the Rhodes Telescope relate to the size of the telescope and the temporal continuity of the data. The CAIDA telescope operates on a /8 network block (comprising some 16.7 million or 2^{24} addresses) and so is 65 thousand times (2^{16}) larger than the Rhodes University telescope system. Consequently the volumes of data captured are expected to be several orders of magnitude larger. The CAIDA telescope is also deemed to have a higher sensitivity level since it can in theory observe $\frac{1}{256}th$ (≈0.39%) of scanning/backscatter activity online, although this may be slightly higher when one factors in the former Class D and E address space (224.0.0.0/4) which is reserved. While the Rhodes system captured 20 million events over a 40 month period (average rate per hour of 2200)[10], there are peaks in capture on the CAIDA system in excess of 50 million events per hour. What is interesting is that despite this high arrival rate, this equates to a lower event per IP address ratio of 3.15/hour compared to the average of 8.92/hour observed on the local system. While the fact that some filtering of active scanning has taken place influences this, it is still significant.

The CAIDA datasets are specifically backscatter datasets, and as such anything that could be construed as active traffic has been removed from the captures. This includes scanning attempts, ICMP probes, and all UDP traffic. Published packets have also been clipped at 64 bytes. Taking this into consideration, this collection of datasets is still of value when performing comparative analysis against the RUSCOPE1 dataset. What these sets lack in diversity, they make up for in volume as would be expected. The backscatter datasets are collectively referred to as the CAIDA-Backscatter collection. The telescope datasets released as discussed in Section 3.1 are referred to as the CAIDA-Telescope datasets.

[9] `http://www.caida.org/data/`

[10] This was between August 2005 and December 2008

3.3.1 System Configuration

The CAIDA data is gathered from the UCSD telescope which is operated on a /8 netblock. The details of the exact network are unknown as all captures are passed through an anonymization and cleaning process prior to release. This process is in terms of operational safety of the telescope as described in Section 2.3. The downside to this operation and the absence of active traffic from the CAIDA-Backscatter sets is that it makes performing passive Operating System fingerprinting using tools such as p0f[11] impossible due to the lack of TCP-SYN packets and the blanking out of some header fields on which the tool depends during the anonymization process. The CAIDA-Telescope data contains more detail, particularly the TCP-SYN packets resulting from TCP connection attempts into the monitored address space. Details of the specific anonymization technique used are unclear, but it is fairly certain that they made use of techniques similar to those described in Xu *et al.* (2001) and Pang *et al.* (2006) which focus on prefix preservation, which in turn is an advancement on the original TCPdpriv anonymization tool (Minshall, 2005) produced by Lawrence Berkeley National Laboratory (LBL).

While sections of the complete CAIDA-Backscatter dataset from 2004-2008 were analysed based on the statistics summary files (an example of which is shown in Listing 4), a selection of captures from 2007 backscatter data were loaded into the same database backed analysis framework that was used for the Rhodes Telescope data. Selected data samples were taken from the February, August and November of 2007 capture sets and amounted to just under 500 million packets. This was used for verification, and load testing of the tools and methods described in Chapter 4.

3.3.2 Data collection and processing

Backscatter data is provided for download with, on average, a seven day window of data made available for each month. The 2006 and 2007 backscatter datasets (Shannon *et al.*, 2006, 2007) have been used for comparative analysis. Due to the very large data sizes, some sub-sampling has been performed, taking a representative sample of three days across each capture period. The datasets were processed in a similar manner to the Rhodes RUSCOPE1 set, being processed by the same loading infrastructure as described in Appendix D. Each capture file has a

[11] `http://lcamtuf.coredump.cx/p0f.shtml`

Listing 4 Sample CAIDA capture overview metrics file

```
Maximum capture length for interface 0:                99999
First timestamp:                       1172692800.000030000
Last timestamp:                        1172696399.997734000
Unknown encapsulation:                                     0
IPv4 bytes:                                        205221149
IPv4 pkts:                                           3291255
Unique IPv4 addresses:                               1723481
Unique IPv4 source addresses:                         548924
Unique IPv4 destination addresses:                   1174558
Unique IPv4 TCP source ports:                          50472
Unique IPv4 TCP destination ports:                     62518
Unique IPv4 UDP source ports:                              0
Unique IPv4 UDP destination ports:                         0
Unique IPv4 ICMP type/codes:                              16
IPv6 pkts:                                                 0
IPv6 bytes:                                                0
non-IP protocols:                                          0
non-IP pkts:                                               0
```

corresponding statistics file (an example of which is shown in Listing 4) and checksum. Checksums were verified after download and prior to any further processing, thereby validating the integrity of the file, something particularly important given that these files were often downloaded in smaller chunks over extended time periods.

The statistics file is useful as it allows for rapid access to several useful metrics without having to process the entire capture file — this is a concept built on in Chapter 8. What is interesting is that in the sample listing shown, only 1 174 558 unique destination addresses were recorded, representing a coverage of only 7% of the address space in this hour. More detailed analysis of 12 hour periods showed average coverage of 56%.

The most likely reason for this is that the majority of scanning algorithms would have some kind of localised scanning, focusing on a /24 or even /16 network at a time. Given that the popular NMAP[12] port scanner takes 230 minutes[13] to completely scan the 65 thousand addresses in a /16 network on a single port[14], it is unlikely that a single scanning host could cover multiple /16 blocks within a single

[12] http://nmap.org/

[13] an effective rate of $\simeq 5$IP addresses/second

[14] nmap -P0 -p80 172.31.0.0/16. The target network was routed via a discard device

Table 3.5: Nmap scan speeds over /16 address space

Mode	Time (*m:s*)	Packets/second
normal	230:42	4.734
aggressive	117:12	9.320
insane	62:19	17.528

Nmap was run against null routed network block using the following command:

`nmap -T x -P0 -p80 172.31.0.0/16.`

x `was varied from 3 to 5 in order to vary the aggressiveness of the scan`

hour period. Run in aggressive mode, which is more likely to be representative of worm or other automata scanning, this same scan takes 117 minutes[15]. It is therefore unlikely that the entire /8 address space could reasonably be scanned by a single system in anything less than 11 days by a single host even at its most aggressive setting. Section 6.2 discusses this further. Details of these timings are given in Table 3.5.

3.4 Other Significant Datasets

In recent years, a number of network telescope datasets of varying types and collected in diverse locations, have become known; those of which the researcher is aware, are listed for interest in Appendix E. Of these only the CAIDA backscatter datasets, already discussed in Sections 3.1 and 3.3, have been accessed and evaluated. This is primarily due to the severe bandwidth constraints currently prevalent in South Africa, where it is not feasible to download such multi-gigabyte datasets. Other data repositories, such as the PREDICT[16] dataset, have highly restrictive access policies which require one to be conducting the research within certain geographical regions (such as within the continental United States of America), or require researchers hold certain citizenship.

In reviewing recently published literature, one becomes aware that while there is a fairly substantial set of data that has been collected by networking and security researchers in the last few years, not all of this is accessible. Should this data be made available, due care will have to be taken with the anonymization of the traffic in order to protect the sensor network's integrity and minimise the likelihood of possible pollution or contamination.

[15] an effective scan rate of $\simeq 35$ IP addresses/second

[16] `https://www.predict.org/`

It is hoped that by encouraging the use of metrics such as those proposed in Chapter 8, a more equitable comparison can be achieved between datasets from different sources, without having to disclose the actual traffic captures – thereby solving the issues of anonymization and privacy. An added benefit of this would be substantially reduced data volume to be transferred among researchers.

3.5 Database Storage

While tools such as InetVis[17] (van Riel and Irwin, 2006b; Irwin and van Riel, 2007), WireShark[18] and the utilities discussed in Section 4.2 operate directly on the packets stored, in the pcap files, much of the statistical and graphical analysis, along with basic data exploration required data in a textual format. Based on this need to have an easy means of accessing and manipulating data, and due to the flexibility it was likely to provide, packet captures were loaded into a relational database system. Details of the loading process used, and issues relating to the operation of the database are included in Appendix D. This Appendix also discusses the actual packet fields selected for storage within the database. One important decision taken at the start of this project was to omit any packet payloads from the database — a decision driven by the fact that the majority of packets recorded actually lacked payload, and if required, this information would be available in the raw pcap files. The majority of the data processing work performed on the collected data was to do with aggregation and summation of the collected data.

The PostgreSQL[19] database system was chosen for this project. One of the primary factors in the initial selection of this DBMS system is its excellent native support within the database software for dealing with Internet addresses[20]. This functionality allowed for IP addresses to be natively used within the database without having to perform explicit IP address to integer type conversions as was needed with the MySQL database engine at the outset of this project. This greatly simplified the development of tools and queries to extract data as the type conversion could be omitted. The `inet` and `cidr` data types used within the PostgreSQL database also allowed for a number of manipulations to stored data that were particularly useful in the context of the network telescope data being stored.

[17] http://www.vizsec.org/applications/inetvis/

[18] http://wireshark.org/

[19] http://www.postgresql.org/

[20] http://www.postgresql.org/docs/8.2/static/functions-net.html

The ability to perform netmask based manipulation of IP addresses, and even arithmetic (as used extensively in Chapter 5) enabled much of the initial data manipulation to be achieved within the DBMS itself, further negating the need to write customised data processing code. In most cases much of the work done in Chapters 5 and 6 using the tools discussed in the Chapter 4, made use of CSV[21] formatted files produced by SQL queries run against the database system. The CSV file format is well supported across numerous existent tools such as graphing packages, spreadsheet systems and within programming languages such as Python, Perl and PHP.

The scalability of this database platform was another consideration, as the data system was envisaged to have a useful life exceeding that of this research project. The tests that were done loading 500 million packets from the CAIDA backscatter datasets as discussed previously, showed that performance levels were still acceptable at this scale. The single largest influence on database performance was found to be available RAM in the DB system. The migration of the data store from the system described to a new platform with 16 gigabytes of RAM in late 2010 resulted in nearly a 10x speedup in some queries. Performance tuning of the database system was not a primary goal of this research, and the tuning that was done was fairly rudimentary. From a software configuration, shared memory segments were increased, and appropriate indexes were created to speed up operations. The reader is referred to documentation specific to the PostgreSQL platform[22] for further details on performance tuning and optimisation.

3.6 Summary

This chapter described the process of data collection for the Rhodes University network telescope and the origins of other datasets used in the validation of results and tools developed during the course of this research. The methods of initial data processing were addressed in Section 3.5. Issues surrounding the storage and initial processing of data were also discussed. These datasets were used in the development of the analysis tools discussed in the following chapter. The application of these on the RUSCOPE1 dataset is presented in Chapters 5, 6 and 7 which

[21] Comma separated value, files are commonly used for data exchange due to their ease of parsing, and compatibility with a large range of applications

[22] `http://www.postgresql.org/docs/current/interactive/index.html`

comprise the second part of the research. This part collects together the analytic work done on the data collected on the Rhodes University Network Telescope.

Part II

Analysis

If you know the enemy and know yourself, you need not fear the result of a hundred battles. If you know yourself but not the enemy, for every victory gained you will also suffer a defeat. If you know neither the enemy nor yourself, you will succumb in every battle

Sun Tzu - The Art of War

4

Analysis Tools

SEVERAL new tools and techniques were developed for the analysis of the dataset collected during the course of this research. This chapter presents an overview of the primary analytic methods used. These are discussed in two logical groupings, initially those of a non-graphical nature, followed by various data visualisations. Implementation details for tools are to a large extent omitted, but suitable references to other works relating to this are provided.

The chapter opens with Section 4.1 presenting a brief introduction to the exploratory methods used against the database systems into which traffic was loaded. This section also explores the iterative analysis process used for examining data. A discussion of the existing tools used for performing analysis of the pcap data files collected by the telescope infrastructure follows in Section 4.2. A proposed metric for analysing the potential localised bias of a telescope is presented in Section 4.3. A bridge between the strictly numerical and visual analysis methods is presented in Section 4.4, looking at the value of analysing the geopolitical origins of the packets.

Sections 4.5 and 4.6 discuss two novel tools for graphical analysis that were developed to aid this research. Both of these tools provide a highly compact graphical means of data representation. These interactive visual display tools allow for rapid interpretation and exploration of millions of data items concurrently. Sections 4.7 and 4.8 explain the use of two other graphical mapping techniques used for performing data analysis, particularly over longer time periods.

The results of the data analysis using the tools and methods described in this Chapter are contained in Chapters 5, 6 and 7 which make up the remainder of this part of the document.

4.1 Query and Analysis

The database storage system as described in Section 3.5 was used for the majority of the analysis both numerical and graphical. Particular use was made of the multitude of network manipulation and native aggregation functions within the PostgreSQL database system. Much of the query development was done within the PgAdminIII[1] environment, which allowed for easy development as well as profiling of queries within a GUI environment and is closely integrated with the PostgreSQL database platform used for storing the data. The cross-platform nature of this tool also facilitated ease of use across a number of different systems. Once suitable queries were developed, these were integrated into appropriate script such as the ones used to produce the Time Series plots described in Section 4.8. Queries were annotated and saved to a central repository for potential reuse.

One of the biggest advantages of having the packet captures loaded into a database system was the ability to rapidly prototype queries, as well as to perform ad-hoc analysis without the need to write a completely new program, with the associated overheads. By extension, other data analysis tools and systems could also be integrated directly with the data without having to be able to parse the pcap file formats, by virtue of them having the ability to talk to database systems. This ease of access allows for the information in the database to be accessed semantically by other researchers without having to first overcome the hurdle of parsing packet headers. Similarly, if need be, suitable measures could be put in place in the database to provide masked information to other individuals or applications consuming data.

[1] `http://www.pgadmin.org/`

Two primary approaches were taken when analysing data in the database. The first of these was to perform a selection of 'top 20' type queries over the range of newly imported data. The results from these queries were then used to identify any particular anomalies which, if found, could be further investigated in the actual packet captures using tools such tcpdump and WireShark described previously. Similarly result sets were also generated to produce Hilbert plots (Section 4.5), Heat Maps (Section 4.7) and Time Series (Section 4.8) plots. These graphical outputs were also evaluated. This was performed as an iterative process, with queries often being repeated in order to analyse specific time periods or portions of IP address space. Where areas of interest were found that warranted further investigation of the raw datagrams, timestamps were extracted. These were used to with the tcpslice utility in order to carve packets out of the raw captures into working sets. Where necessary, this was also combined with suitable BPF based filtering in order to isolate traffic. The resulting working sets of raw packets were analysed in tcpdump and WireShark.

The second approach was to replay data though InetVis (Section 4.6) with varying speed-up factors, and observe for any interesting artifacts such as bursts of scanning activity. Again an iterative process was commonly used in order to narrow down the activity in both temporal space (when it occurred) and in IP address space (from whence it came). This information was then used again to carve out working sets of raw packets from the appropriate capture files. Database queries were used to extract suitable data as well.

The two approaches described so far tended to be used for fairly short periods of data (usually monthly or bi-weekly). It was through performing this kind of analysis, and the iterative nature of it that allowed for the refinement of the tool chain and analysis process. When viewing the dataset as a whole, in addition to the periodic type analysis described, queries were performed across the majority of the fields recorded to look at possible relationships. Some of the analysis performed became increasingly meaningful as the temporal baseline increased, particularly the Hilbert Curve and Time Series plots.

While the structure of the database remained the same from the time of inception and initial data loading, various indexes were added over time in order to optimise queries. This was done both as the dataset grew, as well as incorporating fields into queries that had not been extensively used previously. Due to this process, the database performance was found to increase substantially over time despite the substantial addition of data. When comparative analysis was performed on the data across ports, and varying network aggregation plots, trends and and areas of interest in the data could be identified. These have been included in the discussions contained in Chapters 5, 6 and 7.

4.2 Pcap Manipulation tools

During the course of analysis, several different tools were used in order to process the capture files. This section briefly introduces the tools that were used. Slow processing speeds when dealing with extremely large capture files was one of the main drivers behind the work on implementing a GPU based packet classifier (Nottingham and Irwin, 2009a,b,c, 2010a,b). This, in combination with the experience gained in assessing what type of aggregate queries over data are most meaningful, should allow for high performance analysis without the need to pre-load data into the database system as is currently done. The ability of researchers to perform 'what-if' and ad-hoc queries on large datasets should also be greatly enhanced.

4.2.1 tcpdump

One of the oldest tools used for performing packet analysis is tcpdump[2]. Originally developed in 1987 at Lawrence Berkeley Labs (LBL) by a team in the Network Research Group including Van Jacobson, as a network debugging tool, the tcpdump utility and its associated libpcap library have become the de-facto software on unix-like platforms for performing packet capture and analysis. The BSD Packet Filter (McCanne and Jacobson, 1993), more commonly known as BPF, provides a right set of primitives for filtering and selecting traffic based on various header fields and payload values. This filter is used by tcpdump and the other tools mentioned in this section for selecting packets with attributes matching the given criteria. The file format used to store packets by libpcap (commonly known as pcap format) has become commonly used for the interchange of packet data and is readable by the majority of network diagnostic and analysis tools.

A primary advantage of tcpdump over other capture tools is that it is very lightweight and runs from the command line, allowing for operation without any

[2] http://www.tcpdump.org/

kind of graphical environment. A number of protocol decoders are contained for major IP protocols such as TCP, UDP and ICMP along with some higher level protocols such as DNS. This tool was used for the initial packet capture and the subsequent selection of interesting packets from the capture files along with tcpslice.

4.2.2 WireShark

Gerald Comb's WireShark[3] tool (known as Ethereal prior to May 2006) was developed in 1998 and provides a fully featured packet analyser. It differs in this sense from tcpdump, whose main purpose is the collection and basic decoding of packets. WireShark offers a much fuller featured graphical environment in which one can perform extensive filtering and analysis of the packets recorded as well as higher order protocols. As with tcpdump, WireShark is able to run across a number of operating system platforms.

The biggest issue encountered with WireShark was the inability to load and work with files containing large numbers of packets. As such, capture files were preprocessed using either tcpdump or tcpslice to produce smaller working files which were then used in WireShark. The largest benefit from the researcher's viewpoint in using WireShark was the extensive filtering and drill-down that was available in a far more intuitive manner than the plain hexadecimal based dumps available in tcpdump. Readers are referred to *Practical Packet Analysis* (Sanders, 2007) for more details on WireShark operation.

4.2.3 libtrace

A relative latecomer to the packet and network traffic analysis scene, libtrace[4] is developed and maintained by the WAND research group at the University of Waikato in New Zealand. This is not to be confused with the similarly named LibTrace[5] which is actually a C++ ray-tracing library. This library provides a much faster means of performing analysis of packet capture files in pcap and other common packet capture formats. One particular advantage is its ability to directly

[3] http://www.wireshark.org/

[4] http://research.wand.net.nz/software/libtrace.php

[5] http://libtrace.sourceforge.net/

```
Start time: 1250757208.1208 (Thu Aug 20 10:33:28 2009)
End time: 1254424451.9111 (Thu Oct 1 21:14:11 2009)
Duration: 3667243.7903 (42 days, 10 hours, 40 minutes, 43.7903 seconds)
Total Packets: 311973
Average packet rate: 0.09 packets/sec
Uncompressed trace size: 45045733
```

Figure 4.1: Sample report from libtrace

process pcap files compressed using the gzip compression tool. The researcher became aware of this tool as a result of its use in the work conducted by Pemberton (2007), which built some further analysis tools using the library.

A number of sample utilities which ship with the library were found to be of use. In particular the tracereport[6] tool which was able to produce a number of high level reports on the processed data file. An example of the 'misc.rpt' file is shown in Figure 4.1, which contains some of the high level meta-data produced pertaining to the data file.

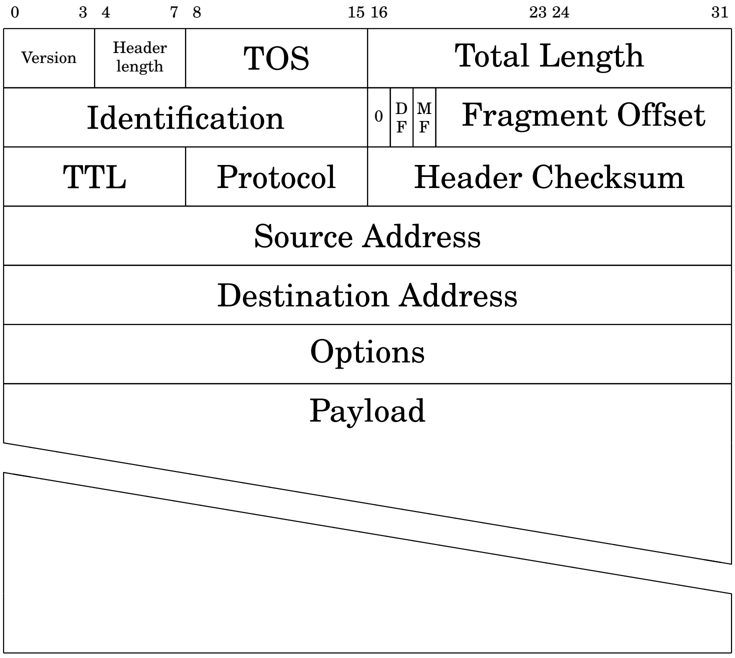
4.2.4 tcpsplice

Tcpslice was originally developed by Vern Paxson while at Lawrence Berkeley Laboratory and is currently maintained by the team at tcpdump.org (Paxson). It provides a means of allowing one to extract portions from one or more pcap capture files. Despite its name, it can also also be used to merge such files together. The strength of this tool is in working with time offsets and ranges, particularly across multiple files. Extensive use was made of this tool in producing unified capture files prior to loading. Particular use was made of the tool during 2007 when, due to power problems, each month consisted of several capture files.

4.3 Distance Score Calculation

All IP addresses can be seen as existing on a discrete continuum of integer values from 0 to 2^{32}. This can be calculated by looking at the integer representation of the 32-bit source and destination addresses used in IPv4, rather than the traditional

[6] http://www.wand.net.nz/trac/libtrace/wiki/TraceReport



after RFC791 (Postel, 1981c) and Stevens (1993)

Figure 4.2: IP version 4 Packet Header

'dotted-quad' notation which segments the address into four 8-bit values presented in the range 0-255. Is is only through common notation that they are most often represented as 'dotted quad'. In their most basic form they can be considered to be a unsigned integer and are processed as such by almost all network stacks. They exist as such in the native packet format (Postel, 1981c; Stevens, 1993). The structure of an IPv4 Datagram is shown in Figure 4.2. Consequently it is possible to calculate a difference between two IP addresses in order to determine their logical or numerical distance from each other within this continuum. This may be, and in most cases is, significantly different from their topological distance (the number of network hops between them) and any physical distance calculated between the locations of their geophysical manifestations.

The value of the computation of such a logical distance lies in its use in quantifying the network locality or proximity of two IP addresses. The ability to provide a quantification of the relative closeness of two addresses is something which is particularly useful in the analysis of network aware malware scanning algorithms and consequently the analysis of network traffic collected by IDS or sensor networks, such as that used in this study. In such cases it has been shown as in Zou *et al.* (2005) and Wei and Mirkovic (2008), that a sensor network is likely to experience a notable bias towards traffic originating from networks that are numerically or logically close. This is largely due to the relatively primitive scanning and propagation algorithms employed by malware automata and naive crackers.

Most automata to date have evolved from the totally random scanning patterns of earlier incarnations and operate in two distinct modes: near and far. The near scanning mode is employed first where networks numerically near to that of the infected host are scanned first. Once this is complete a shift can be seen to a more randomised scanning of far address space. This is particularly well exhibited by worms such as Code Red and Code Red II, as discussed by Chen and Ji (2005); Zou *et al.* (2006b); Chen and Ji (2007); Chen *et al.* (2007). The rationale behind this calculation and subsequent evaluation is that due to the relatively crude scanning and propagation algorithms (Nazario, 2003; Richter, 2008) implemented by much of the malware automata seen online there will be a natural bias of the telescope to see proportionally higher traffic from numerically closer networks. This provides a means of assessing the closeness of these networks. Further detail on the process and initial results obtained using this method to analyse a portion of the RUSCOPE1 dataset was published in Irwin and Barnett (2009).

4.3.1 Application

In order to calculate a distance between two addresses, the two need to be in a format that can be easily manipulated. From a very simple perspective, an IP version 4 address can be converted to an integer value using the process described below, where A.B.C.D represents an IP address written in dotted-quad notation.

$A.B.C.D = A * 256^3 + B * 256^2 + C * 256 + D$

or

$A * 2^{24} + B * 2^{16} + C * 2^8 + D$

This is in effect what is used internally by system calls such as `atoi()` and inversely by `itoa()` in system libraries and networking stacks. These values can then be subtracted to arrive at the numerical difference between the two values. For the purposes of the investigation carried out, the conversion was carried out internally by the PostgreSQL database system. When calculating the distances for the RUSCOPE data the IP address at the midpoint of the monitored range (196.x.x.128) was used. As explained below, this made very little difference to the resultant score and proved much easier to implement within the database processing system.

While the process described gives an accurate conversion, the actual distance between two addresses has little need of the least significant components since they are in most cases on the same network, or at least within the same organisation or logical netblock. As such, the conversion can be refined to being $A{*}2^{24}{+}B{*}2^{16}{+}C{*}2^{8}$, effectively omitting the least significant portion of the address.

At this point it is worth noting that while direct assignments to organisations of smaller than /24 do exist, they are relatively few and are from a legacy period prior to the establishment of the regional registries such as LACNIC and AfriNIC in 2002 (Gerich, 1993; Fuller *et al.*, 1993; Hubbard *et al.*, 1996). Such assignments to organisations from their service provider are, however, much more common with assignments of /28 and /29 being prevalent for broadband connectivity (Tsuchiya, 1991; Gerich, 1993; IANA, 2002; Albanna *et al.*, 2001). The reduced form of the conversion proposed above logically clumps such networks as being part of the same logical higher level assigned network block, which is likely to be a direct assignment from a Regional Registry. Examples of using the baseline and the reduced formulae are shown below.

Example 1: Addresses within the same natural subnet (255.255.255.0)

```
IP Address A : 192.168.149.67 (3 232 273 731)
IP Address B : 192.168.149.254 (3 232 273 918)
This gives a natural difference of 187
```

Using the second method provided above, the difference can be shown to be 0 when the least significant octet is omitted.

Example 2: Differing network addresses

```
IP Address A: 146.231.123.15 (2 464 643 855)
IP Address B: 209.67.212.202 (3 510 883 530)
```

Here the difference can be shown to be 1 046 239 675 or approximately 1 billion (1.046 $\times 10^9$) addresses apart . Removing the least significant byte from the calculation gives 1 046 239 488 which is only 187 ($202 - 15$) different from the answer obtained by the method shown in the first example above. This difference is in itself insignificant as it accounts for less than $\frac{1}{1000}th$ of a percent of the final result. When one takes into account the large values which can be obtained as the result of natural subtraction of the components (using either algorithm) the need to be able to reduce this to a more comprehensible number is of some interest.

The ideal was to be able to reduce the potentially large range of differences from $\pm 0 - (2^{32} - 1)$ to a more comprehensible range. By taking the Log_{256} of the absolute integer difference provided a score in the somewhat reduced range of 0.0≤4.0. The sign however should be reapplied to the result of the calculation. Table 4.1 shows some common values obtained. Broadly, the resultant values can be interpreted as follows:

0 ≤ 1 Addresses lie on the same network (there are 255 or less individual addresses between the two).

1 ≤ 2 Addresses lie within two /16 networks (65535 IP addresses) of each other. This may not necessarily be a block of addresses lying on a contiguous natural boundary.

2 ≤ 4 Addresses lie elsewhere on the Internet, with the values approaching 4.0 as the distance increases.

From a practical point of view, the maximum value lies closer to 3.9759 when one accounts for the maximal distance between 0.0.0.0 and 224.0.0/4, which is the start of multicast address space Albanna *et al.* (2001); Reynolds and Postel (1994); IANA (2002). When interpreting these values as described, it is important to note that they are unsigned, and as such there is no directionality associated with the distance. While this has its possible disadvantages, it obviates the problem of the ordering of the IP addresses in the initial calculation. Thus, when using the score for measurement it in effect is a score of $\pm\Delta$, providing a range on either side of the target address. The resultant formula for the calculation of the IP Distance Score ($IP\Delta$) between two IP addresses can defined as:

$$IP\Delta = Log_{256}(ABS(INT(IP_A) - INT(IP_B))$$

Table 4.1: Common IP∆ values for distances

$\pm IP\Delta(log_{256})$	Raw Score	# of /24 Blocks	# of /16 Blocks
1	1	1	0
1.38	2048	8	0.03
1.5	4096	16	0.06
1.88	32768	128	0.5
2	65536	256	1
2.13	131072	512	2
2.25	262144	1024	4
2.38	524288	2048	8
2.5	1048576	4096	16
2.63	2097152	8192	32
2.75	4194304	16384	64
2.88	8388608	32768	128
3	16777216	65535	256

From a practical perspective, the values were calculated as signed within the database system which allowed for the directionality to be analysed and correctly vectorised.

4.4 Geopolitical Analysis

An analysis of the geographical origins of observed traffic is of use to telescope operators and and network security researchers in general. From the perspective of backscatter and reflected traffic analysis, being able to attribute observed packets to countries allows for one to gauge levels of hostile behaviour directed at computers in that state. Geolocation has gained in popularity in recent years with it being used in such varied roles as load balancing, content distribution, network filtering, access control on web sites, spam scoring and as a component in combating online fraud. It is worth stating that it needs to be accepted that while the geo-location process (that of relating a particular IP address to a geo-political region) is fairly accurate, it is impossible to attribute observed packets to a certain country with absolute certainty, since spoofing of IPv4 datagrams is relatively trivial. This is particularly relevant for network telescopes, where no TCP three-way handshake is completed, ensuring that either the spoofing is relying on a tap on the telescope uplink, or there is a legitimate system communicating. For the purposes of this section all countries are referred to in tables and diagrams by their ISO 3166[7] two-letter short-codes. A selected list of these is presented for reference in Appendix C. Initial results obtained in performing a geo-political analysis of the traffic are available in Irwin *et al.* (2007). These are re-evaluated in Section 6.3.

The actual process of geo-location involved looking up an IP address in databases maintained by organisations, and from this at a minimum, being able to extract a country. In some case a lot more information, such as longitude and latitude, ISP, City and connection speed can be obtained. For the purposes of this research the majority of processing was done at country only level. Some of the results obtained in using co-ordinate based plotting are shown in Section 4.4.2, along with some of the problems encountered. Earlier research on the accuracy and applicability of the techniques used can be found in Carr (2003); Acton *et al.* (2007).

4.4.1 Analysis

Conversion from IP address to country of origin, was done using the GEOIP LITE[8] library from Maxmind. This was cross checked with the open access geo-location database maintained by HOSTIP.INFO[9], and results were found to be in agreement. The choice to use the GEOPIP system was primarily due to its C programming API and fast performance, while the HOSTIP.INFO approach is a database driven system and required iterative queries. The geo-location tool developed was run on the raw pcap files after they had been loaded into the database system as described in Section 3.5, with the results being loaded directly back into the database. While the process is not perfect, it has become increasingly accurate over recent years as it has become increasingly commercialised, with a number of different providers offering solutions to this problem. Of the unique addresses within the RUSCOPE1 dataset, only 1 380 were not conclusively identified, representing less than 0.02%. The majority of these are RFC1981 blocks, and other Bogon addresses as discussed in Section 6.5.

Aggregate information was extracted from the RUSCOPE1 database and stored as CSV files, which were then further processed in order to produce the plots shown in

[7] `http://www.iso.org/iso/country_codes.htm`

[8] `http://maxmind.com/`

[9] `http://hostip.info/`

this section. The second phase of data analysis was to plot the processed information onto a world map in order to be able to generate a quick visual overview of the source of traffic coming into the telescope network. This was achieved in two ways, the first, using the co-ordinate information available to plot the longitude and latitude of each source network (with this data being extracted from the HOSTIP.INFO database). The second aggregated packet information on a per country basis and used the packet totals as a key for shading the countries on a world map.

The ability to attribute a city, region or set of geographic co-ordinates with an IP address is useful in certain situations, and use of this has been made in some of the analysis conducted in Chapter 6. Given that much of the other analysis has been performed by aggregating data, it is also important to be able to plot and visualise this information in a meaningful manner.

4.4.2 Location plotting

The initial approach used for analysing the origins of datagrams was to plot the co-ordinates corresponding to observed source IP addresses on a world map as individual points. While this did show some interesting trends, it was found that outside of the United States of America and Western Europe, most addresses are either resolved as having the geographic co-ordinates of a nation's capital, or that of other major cities. A global view of the total traffic recorded during the period of August 2005–June 2007 is shown in Figure 4.3. This was generated using the longitude and latitude co-ordinates as provided for the network address blocks in the HOSTIP.INFO database, and plotted using a developed tool making use of the python matplotlib[10] library.

While initial results were quite promising, the output from the tool was found to suffer from a number of of issues. One of the most significant of these was that while images are useful for high level interpretation, rapid quantification of the volumes of traffic originating from a given country are difficult to ascertain. Colouring of the plot points was experimented with in order to convey the magnitude of the volume of traffic attributable to a point, but this tended to result in further occlusion of information due to the close packing of points in places such Western Europe and the United States of America. For countries outside of these

[10] http://matplotlib.sourceforge.net/

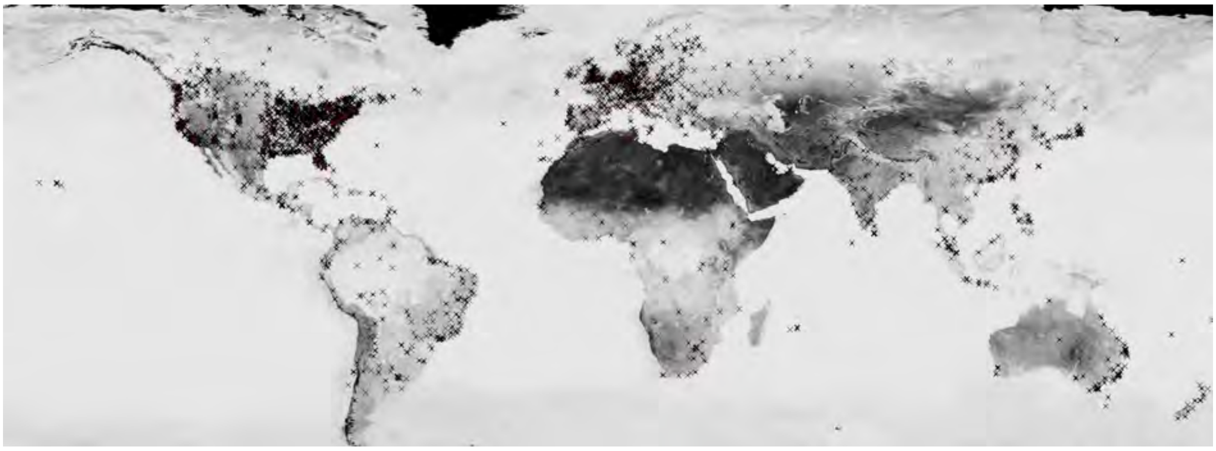


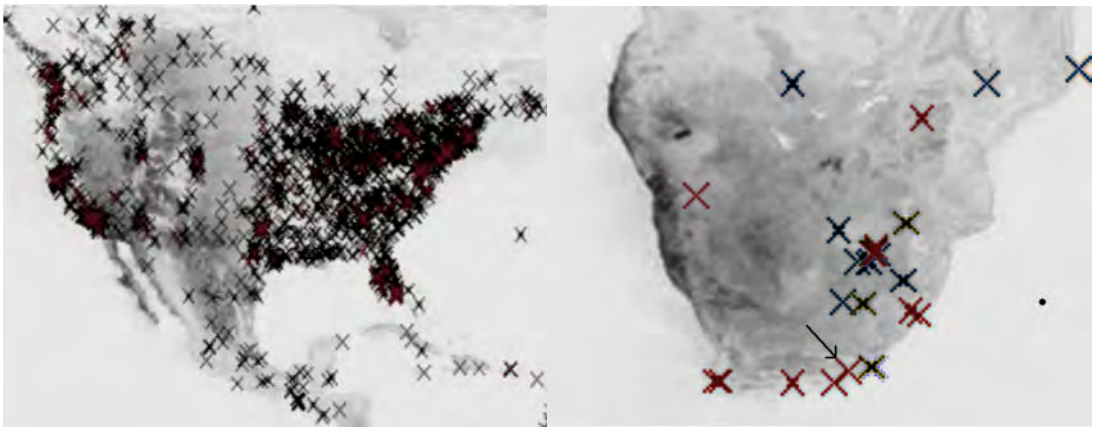
Figure 4.3: Longitude-latitude plotting of traffic by Source Address (September 2007)

geographic regions, points also tended to be concentrated at major cities only, as discussed in the previous section. Performance was also found to be problematic when plotting large numbers of points.

Considering Figure 4.3 in more detail, it is worth noting that the eastern United States (US) looks particularly dense, as is the United Kingdom (UK), Germany (DE) and Austria (AT) due to the finer detail that is available in comparison to locations in countries such as China (CN) and India (IN), where many network blocks resolve to much fewer geographical co-ordinate points despite fairly widespread populations. One other interesting characteristic that can be observed is the placement of points in Russia (RU), which coincide with the path of the trans-Siberian railway.

Figure 4.4 shows cropped images of the United States and of Southern Africa produced using the same data as in Figure 4.3. The clear clustering can be seen round the major centres of South Africa. Grahamstown, the home of Rhodes University, is indicated by the small arrow in Figure 4.4(b). Despite its small size, Grahamstown featured as it has a relatively high IP address density, given the university's /16 (class B) allocation as well as other local Internet Service providers and schools. Some of these local networks were a source of traffic coming into the telescope.

As a solution to the occlusion issues, and due to interest in the aggregate values attributable to countries, an alternate plotting tool was devised and is discussed in the following section.



(a) United States (b) Southern Africa

Figure 4.4: Close up sections of geolocated plots

4.4.3 Geopleth plots

As a solution to the traffic quantification problem mentioned above, a tool was initially developed to generate output such as shown in Figure 4.5, where colour fills could be applied to countries to provide a quick means of quantifying the levels of traffic originating during a given period. This display model suffered from some shortcomings, in particular the inability to handle plotting the complete dataset, and as such this is more suited for shorter time windows. It did however provide useful results in initial work done (Irwin *et al.*, 2007) and in the development of additional requirements for the subsequent plotting tool.

Subsequent to the work in Irwin *et al.* (2007), a new geographical plotting scheme was developed. This simplified system made use of flexible chloropleth based

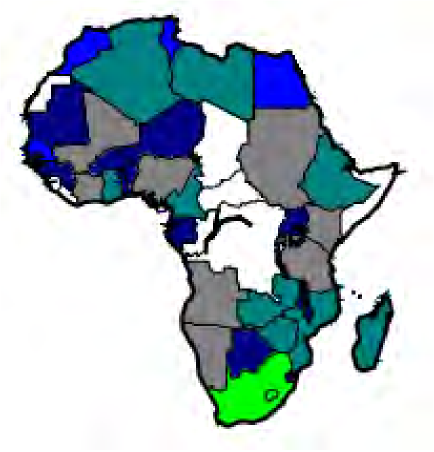


Figure 4.5: Initial Country based shading of Africa

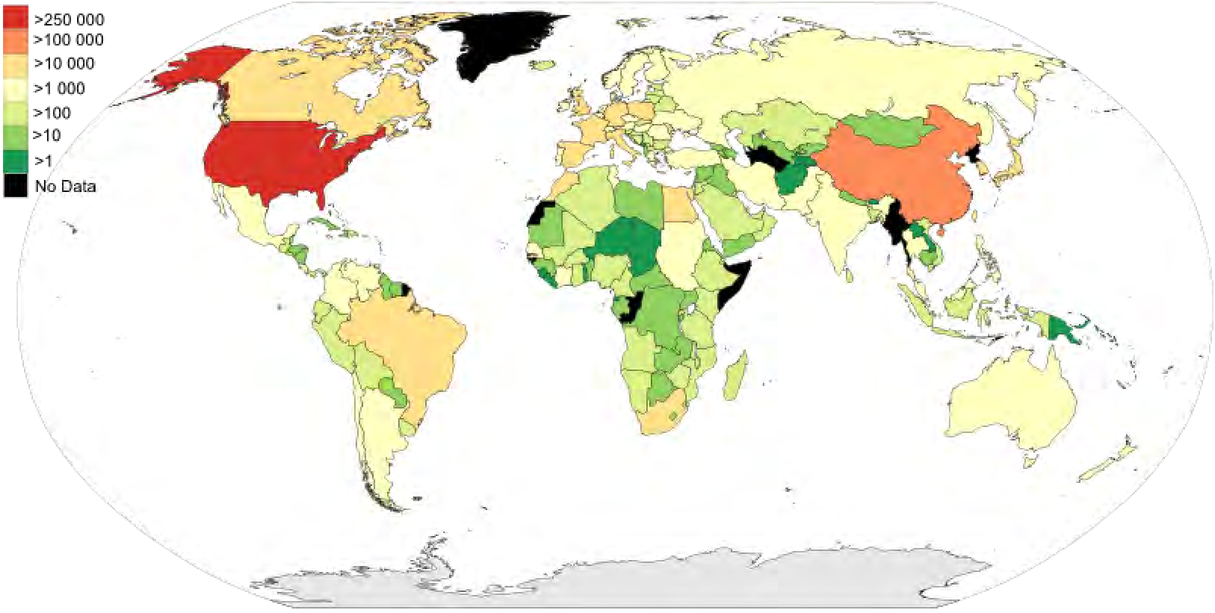


Figure 4.6: Sample Geopleth plot (Robinson projection)

colour schemes. The use of these graduated schemes allowed for both a simpler implementation and a better quality printed output, as opposed to the continuous graduations used in the Heatmaps in Section 4.7. The implementation was done by generating a CSS[11] file from database outputs, which was then applied to a SVG[12] map of the world[13] obtained from Wikipedia under a Creative Commons licence. The result was the production of a coloured plot that had a very high level of detail and which could be zoomed into and manipulated further. The vector nature of the image allowed for high-resolution scaling, and editing, far more so than bitmap outputs, the latter which could be produced at arbitrary resolutions. An example of this output is shown in Figure 4.6, in which countries that have no attributable packet data are coloured in black. Both Robinson[14] and Equirectangular[15] projection maps have been used, the latter being found to be more suitable for cropping.

Plots of this nature provide a clear high level view of the coverage of traffic as well as other attributes, such as the number of unique sources or datagrams received. It is also worth pointing out the inequality that exists globally with regards to the allocation of IPv4 Address space. Figure 4.7 is an image taken from the BGP

[11] Cascading Style Sheet

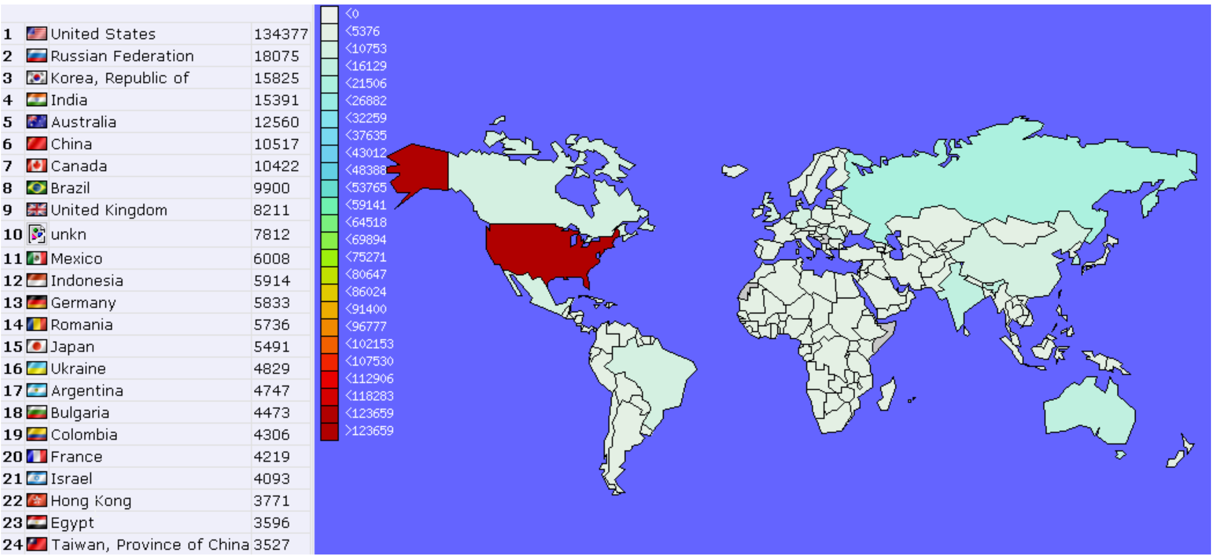
[12] Scalable Vector Graphic

[13] http://en.wikipedia.org/wiki/Wikipedia:Blank_maps

[14] http://en.wikipedia.org/wiki/Robinson_projection

[15] http://en.wikipedia.org/wiki/Equirectangular_projection

Weather Map[16] for IPv4 traffic. This image shows the Unites States of America with more than an order of magnitude more network blocks allocated for use within its borders than the next highest country, being the Russian Federation. Many of these network blocks are also quite large (/16 or larger) given the legacy of the original IP address allocation policies in the early days of the Internet. Africa has a similarly disparate allocation as shown in Figure 4.8. In this case it reflects the state of development of telecommunications infrastructure on the continent. The global disparity should be taken into account when evaluating plots of packet and host counts in later chapters.



1	United States	134377
2	Russian Federation	18075
3	Korea, Republic of	15825
4	India	15391
5	Australia	12560
6	China	10517
7	Canada	10422
8	Brazil	9900
9	United Kingdom	8211
10	unkn	7812
11	Mexico	6008
12	Indonesia	5914
13	Germany	5833
14	Romania	5736
15	Japan	5491
16	Ukraine	4829
17	Argentina	4747
18	Bulgaria	4473
19	Colombia	4306
20	France	4219
21	Israel	4093
22	Hong Kong	3771
23	Egypt	3596
24	Taiwan, Province of China	3527

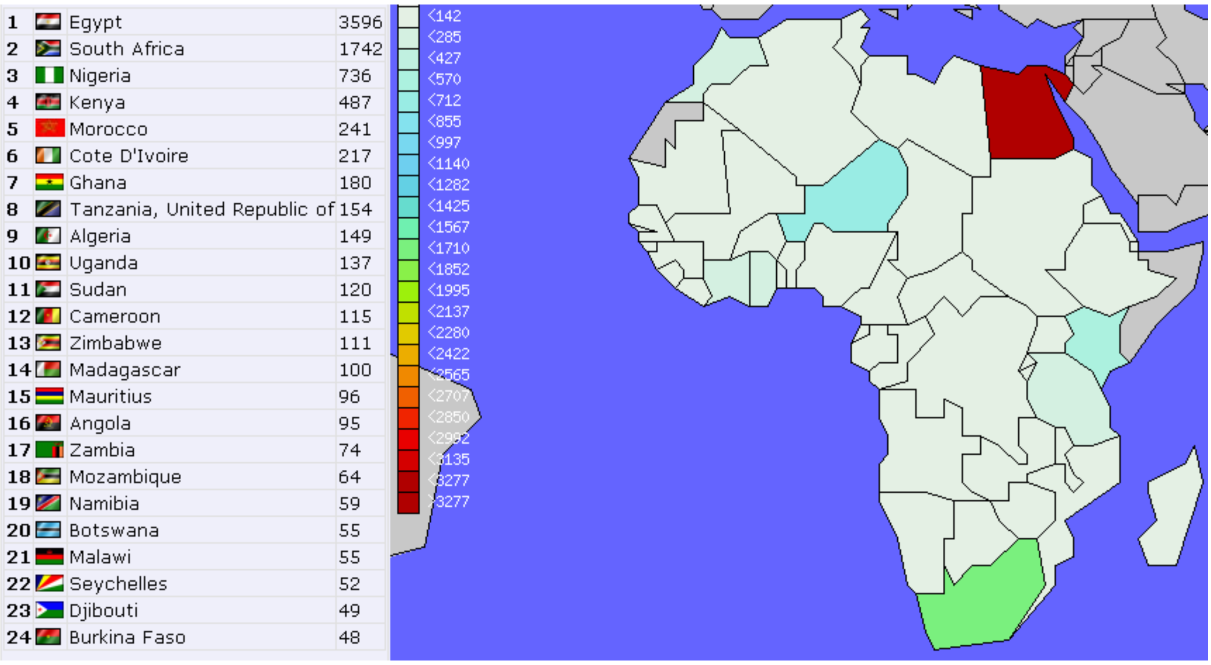
Figure 4.7: Global Netblock Allocations via BGP.

Source bgpmon.net 2010-12-28

4.5 Hilbert Curves

This section discusses the implementation of the Hilbert Curve visualisation tool and its application to the datasets being used in this study. This was developed with the intention of being a high level analysis tool for aiding in the analysis of large volumes of network telescope traffic, and in particular the comparison of data collected from multiple telescope datasets. While the layout was inspired by Randal Munroe, as he used this curve to the layout in his *Map of the Internet* panel on `xkcd.com` in December 2006 (Munroe, 2006a,b), no implementations of this were available at the time of the researcher's implementation in mid 2007. Munroe's

[16] `http://bgpmon.net/weathermap.php?inet=4`



1	Egypt	3596
2	South Africa	1742
3	Nigeria	736
4	Kenya	487
5	Morocco	241
6	Cote D'Ivoire	217
7	Ghana	180
8	Tanzania, United Republic of	154
9	Algeria	149
10	Uganda	137
11	Sudan	120
12	Cameroon	115
13	Zimbabwe	111
14	Madagascar	100
15	Mauritius	96
16	Angola	95
17	Zambia	74
18	Mozambique	64
19	Namibia	59
20	Botswana	55
21	Malawi	55
22	Seychelles	52
23	Djibouti	49
24	Burkina Faso	48

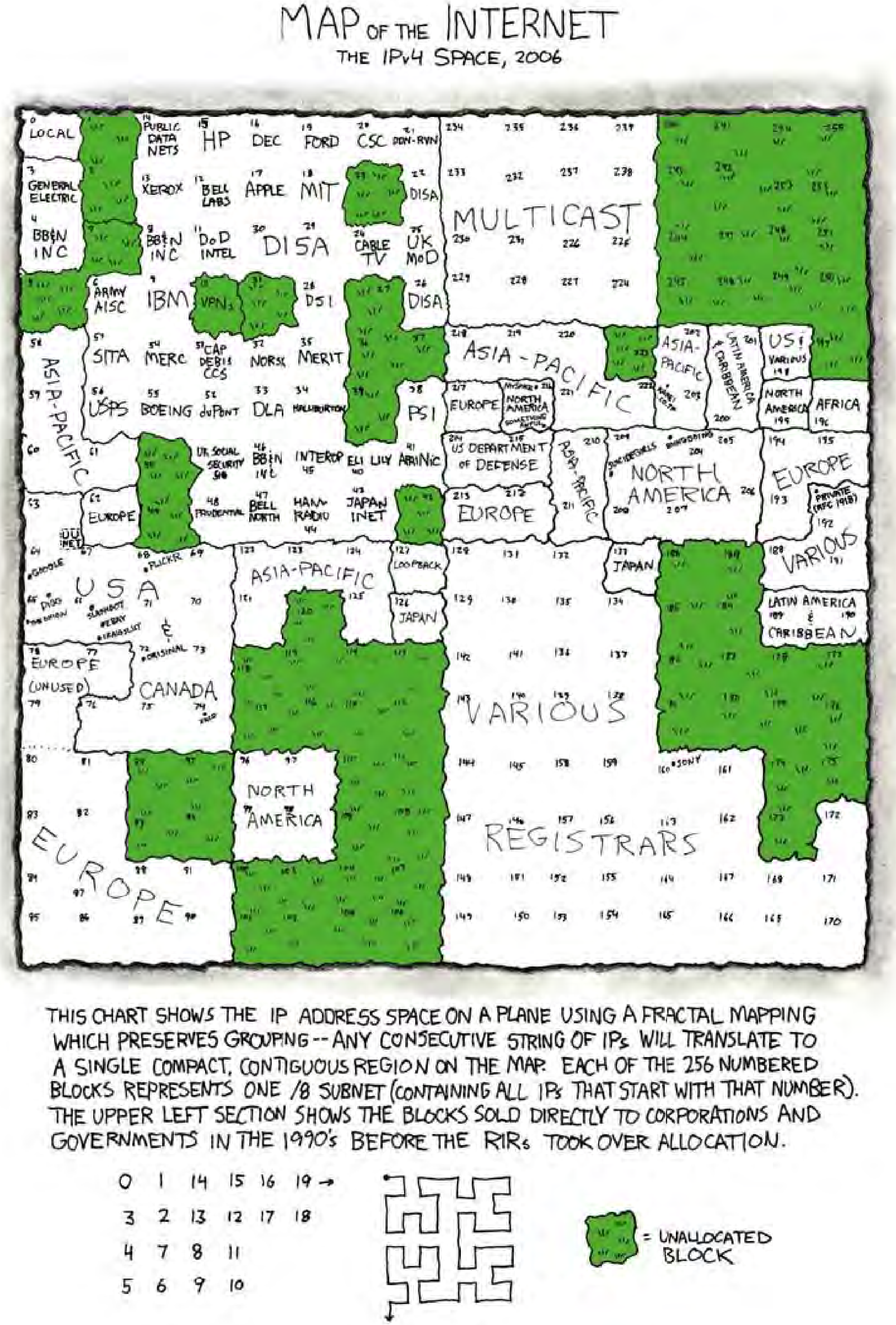
Figure 4.8: African Netblock Allocations via BGP.
Source bgpmon.net 2010-12-28

work as shown in Figure 4.9 only provided a mapping of the 256 top level address block allocations (/8 networks). The researcher's work was initially described in late 2007 (Irwin and Pilkington, 2008b) at the Workshop on Visualisation for Cyber Security (VizSec) Conference[17]. An expanded paper was published in the following year (Irwin and Pilkington, 2008a). Subsequently *The Measurement Factory* have made their code available for download[18]. The researcher's own implementation has subsequently undergone major revision in terms of scalability, functionality and performance. Further exemplar plots are provided in a Appendix B, together with further explanation on the interpretation of the resulting Hilbert images.

4.5.1 History

The Hilbert Curve[19] was first described by mathematician David Hilbert in 1891. It is a fractal curve that acts as a space filling curve as its order is increased. In this sense it is similar to the Peano family of space filling curves. Figure 4.10a-c shows the successive iterations of the curve from 1^{st} order to 4^{th}. The researcher's

[17] http://vizsec.org/
[18] ipv4-Heatmap - http://maps.measurement-factory.com/software/index.html
[19] http://en.wikipedia.org/wiki/Hilbert_curve



THIS CHART SHOWS THE IP ADDRESS SPACE ON A PLANE USING A FRACTAL MAPPING WHICH PRESERVES GROUPING -- ANY CONSECUTIVE STRING OF IPs WILL TRANSLATE TO A SINGLE COMPACT, CONTIGUOUS REGION ON THE MAP. EACH OF THE 256 NUMBERED BLOCKS REPRESENTS ONE /8 SUBNET (CONTAINING ALL IPs THAT START WITH THAT NUMBER). THE UPPER LEFT SECTION SHOWS THE BLOCKS SOLD DIRECTLY TO CORPORATIONS AND GOVERNMENTS IN THE 1990's BEFORE THE RIRs TOOK OVER ALLOCATION.

0 1 14 15 16 19 →
3 2 13 12 17 18
4 7 8 11
5 6 9 10

🟩 = UNALLOCATED BLOCK

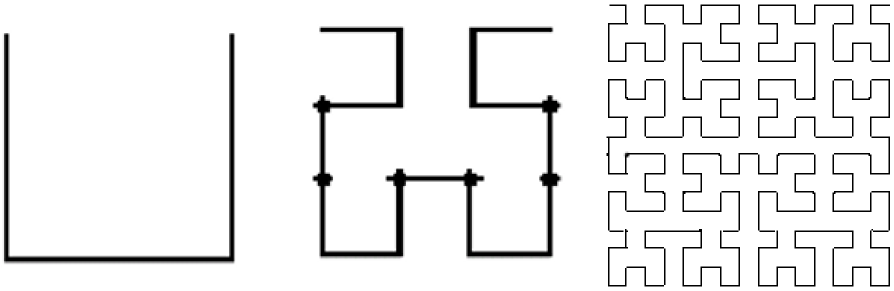
Figure 4.9: Map of The Internet (Munroe, 2006a)

interest in the curve is that it provides a means of traversing every square in a square grid by means of a continuous line. The number of points traversed in the grid relate to traditional means of grouping IPv4 address space. A Hilbert curve maintains locality of data on the curve. This means that data ordered a certain way in one dimension will still be ordered the same way along the curve in two dimensions.

Another interesting property of the curve is that it visits every lattice point in a square with side length a power of two. This is especially useful in extrapolating data which occurs in powers of two, onto a plane. With respect to the visualisation of IP address data it thus provides a convenient means of transforming the single dimensional sequence of IP addresses into a two-dimensional representation. The real value of the application of the Hilbert mapping to IP address data is that it preserves the locality of adjacent netblocks when the one-dimensional numerical ordering of octets is rendered to a two dimensional grid. This is seen to be a distinct advantage over other linear visualisation methods that make use of 'wrapping', although these can be useful in their own right as shown in the use of the 'Jupiter plots' by Pemberton (2007). It is worth noting that another similar ordered pixel-position plotting approach was proposed by Teoh *et al.* (2002) which uses a quadrant based mapping scheme based on the most significant bits of an IP address. The Hilbert plotting scheme has a more natural 'binning' effect, particularly useful to those habituated to dealing with traditional 'classful' subnets, than the aforementioned methods. As a tool the Hilbert Curve plots provided a accurate means of visualising data collected by the Network telescope at varying levels of resolution with respect to IP address space.

4.5.2 Implementation

The number of nodes (in essence the number of blocks traversed) for a given order (n) of the curve can be given by the equation $Nodes_n = (2^n)^2$. The Hilbert curves of order 4, 8, 12, and 16 are especially interesting as they have 256 (2^8), 65 536 (2^{16}), 16 777 216 (2^{24})and 4 294 967 296 (2^{32}) points respectively. These values correspond to the natural grouping of Internet networks blocks by Class A (/8), class B (/16), and class C (/24) with the range of 32-bit Internet address space. A 16^{th} order curve provides the same number of points as the total potential number of addressable nodes on the IP protocol version 4 Internet (2^{32}). This would however require



(a) Order 1 (b) Order 2 (c) Order 4

Figure 4.10: Hilbert Curve Orders

Listing 5 Lindemeyer System describing a Hilbert Curve

Alphabet: L, R

Constants:▲(forward), ◀(left), ▶(right)

Axiom: L

Production Rules:

$$L \Longrightarrow \blacktriangleright R \blacktriangle \blacktriangleleft L \blacktriangle L \blacktriangleleft \blacktriangle R \blacktriangleright$$

$$R \Longrightarrow \blacktriangleleft LF \blacktriangleright R \blacktriangle R \blacktriangleright FL \blacktriangleleft$$

approximately 16 gigabytes of memory if 32 bits of data were maintained per pixel. Plotting host level curves for IPv6 would require a curve of order 64 requiring a substantial amount of memory and system resources. Host level plots for IPv4 can be achieved though suitable data manipulation and are particularly well supported by the ipv4-heatmap tool. The only caveat with these is that host level resolution images can only be prepared for a single /8 network at a time showing 2^{24} nodes. In the future, the use of three-dimensional (or possibly even higher dimensional order) Hilbert structures may be of use in producing plots for IPV6. At the time of writing, the researcher is not aware of any tools providing support for IPv6.

The algorithm for the curve can be represented as a Lindemeyer system as shown in Listing 5, which provides the basic iterative drawing instructions required. Further specifics relating to the implementation and initial applications to analysis can be found in Irwin and Pilkington (2008a).

The base layout code to implement the Lindemeyer system was developed using C++ and OpenGL with the purpose of mapping a given sub-string of a dotted-quad IP address representation to a particular point on the grid being produced. In effect this associated a bin or bucket holding IP networks representing natural 8-bit divisions of the dotted-quad, and hence clustered by most significant octets is mapped to nodes on the curve of a given order. As previously discussed, curves of orders 4, 8 and 12 map easily to the natural netmasks of pre-CIDR class A, B and C network blocks occurring on the 8-bit boundaries used for presenting the IP address in dotted-quad format. Hilbert curves of order 16 were not implemented due to the complexities of mapping these to a reasonable screen size as well as memory constraints on the available hardware. Where host level views were needed, these were achieved though suitable data manipulation to allow for the viewing of a single /8 network at a time at the resolution of each point being a single host, rendered on a 12^{th} order curve.

While it is recognised that much Internet address space allocation today makes use of CIDR based variable length masks, for most purposes, the traditional masks based on 8-bit boundaries are still useful. The net result is a graphical output showing nodes which are coloured as containing elements aggregated dependant on the order of the curve. Initially a true/false flagging system was used where a node was drawn if it contained at least one member in the aggregation bucket. This was extended to use colouring based on criteria such as unique hosts within the bucket and total number of packets within the bucket. A number of discrete implementations were produced; each focusing on separate aspects of the visualisation, but all using the same underlying Hilbert Curve generation for layout. Specifically versions were produced that provided colour-indexed quantification of the number of unique hosts within a specific network block bucket and the number of packets received within a bucket.

In the program the vertices were generated recursively using the Lindenmeyer System representation of the Hilbert Curve— this was the simplest representation and allowed the curve to be generated quickly and accurately. The system used can be seen in Listing 5. Additionally, the use of this representation also generates the vertices of the curve in numerical order which allows points to be plotted at the same time the curve is generated thus providing much greater efficiency. The IP addresses to be plotted were stored in a hash table that allowed the required IP address count associated with the current vertex on the Hilbert curve to be quickly recovered from the table and plotted. Input to the system was by means of simple text generated as a result of database queries. This approach allowed for the internal database functions to be used for managing aggregation and enumeration.

The decision to use a text based input format was due to the simplicity to produce this kind of datafile from a number of different data sources such as libpcap format packet captures, databases, log files (such as from Intrusion Detection Systems, web servers and e-mail server logs), and simulation software. The ability to easily be able to edit and manipulate the input files was also deemed to be of importance. The code could also be potentially be extended in the future to work directly with formats such as libpcap, although this would require a significant increase in the data processing algorithm over the current implementation. Currently, pcap files can be processed directly using a helper script which produces output in a suitable format for consumption.

4.5.3 Application

The first application of the Hilbert curve maps was in assessing the coverage of IPv4 address space by various datasets. This was particularly useful in isolating and evaluating the level of activity for RFC1918 and other special purpose address blocks.

A sample plot of class A network blocks using a 4^{th} order curve is presented in Figure 4.11. The curve starts with 0.0.0.0/8 mapped to the top left (Origin), and 255.0.0.0/8 (End) to the top right corners respectively. The grid can be divided into quadrants showing the placement of the 256 network blocks, which can be used as a guide when interpreting output such as that shown in the next sections.

A detailed map of the labelling of the 256 grid points can be seen in Figure 4.12, with a larger annotated version in found in Appendix B. Raising the order of the curve for /16 and /24 networks used the same layout, but increases the tightness of the curve within the respective quadrants.

Generating an overview of data at /8 level provides a quick overview of input data, and particularly allows for rough checking that data is not being plotted in regions where it should not exist. During development, it was important to validate the output generated by the tool chain with the textual inputs extracted from the database system. Figures 4.13 to 4.15 show the successive plotting of 2.4 million unique data points sampled from CAIDA network Telescope data between 12h00 and 17h00 EST on February 28th 2007 (Shannon *et al.*, 2007) on curves of increasing order. These plots are to assess coverage only, and as such only record a single instance for each address block being considered. Figure 4.13 is the 4^{th} order curve with the actual Hilbert path plotted as a guideline. The same input data is used as shown in Figures 4.14 and 4.15, but plotted onto an 8^{th} order curve showing

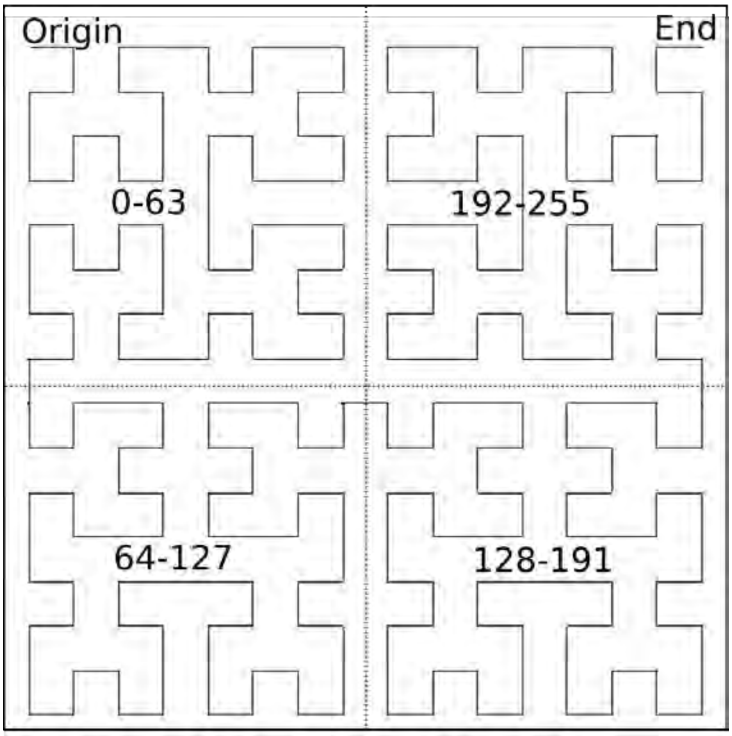
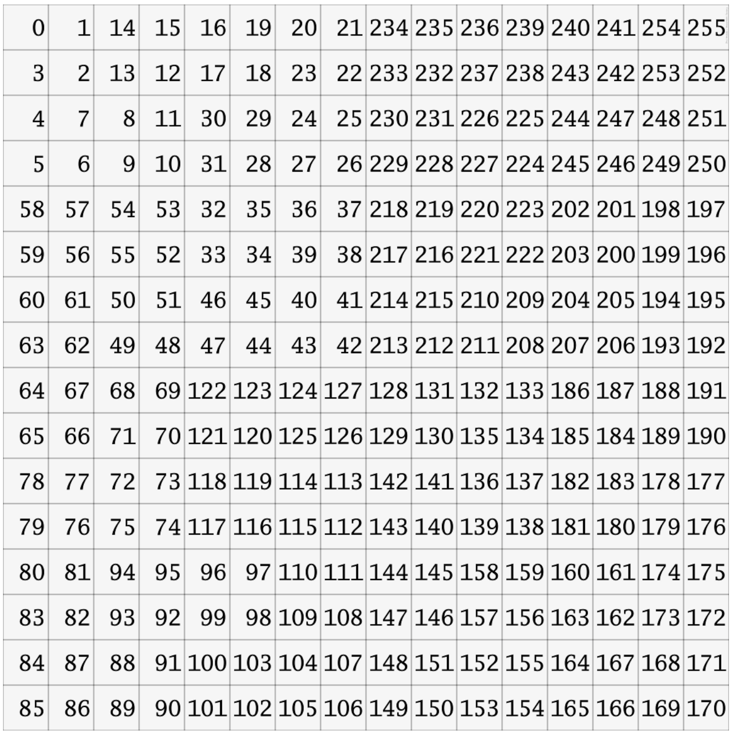


Figure 4.11: Sample 4^{th} order plot with overlay showing the four quadrants



0	1	14	15	16	19	20	21	234	235	236	239	240	241	254	255
3	2	13	12	17	18	23	22	233	232	237	238	243	242	253	252
4	7	8	11	30	29	24	25	230	231	226	225	244	247	248	251
5	6	9	10	31	28	27	26	229	228	227	224	245	246	249	250
58	57	54	53	32	35	36	37	218	219	220	223	202	201	198	197
59	56	55	52	33	34	39	38	217	216	221	222	203	200	199	196
60	61	50	51	46	45	40	41	214	215	210	209	204	205	194	195
63	62	49	48	47	44	43	42	213	212	211	208	207	206	193	192
64	67	68	69	122	123	124	127	128	131	132	133	186	187	188	191
65	66	71	70	121	120	125	126	129	130	135	134	185	184	189	190
78	77	72	73	118	119	114	113	142	141	136	137	182	183	178	177
79	76	75	74	117	116	115	112	143	140	139	138	181	180	179	176
80	81	94	95	96	97	110	111	144	145	158	159	160	161	174	175
83	82	93	92	99	98	109	108	147	146	157	156	163	162	173	172
84	87	88	91	100	103	104	107	148	151	152	155	164	167	168	171
85	86	89	90	101	102	105	106	149	150	153	154	165	166	169	170

Figure 4.12: Layout of the 256 cells on a 4^{th} order curve

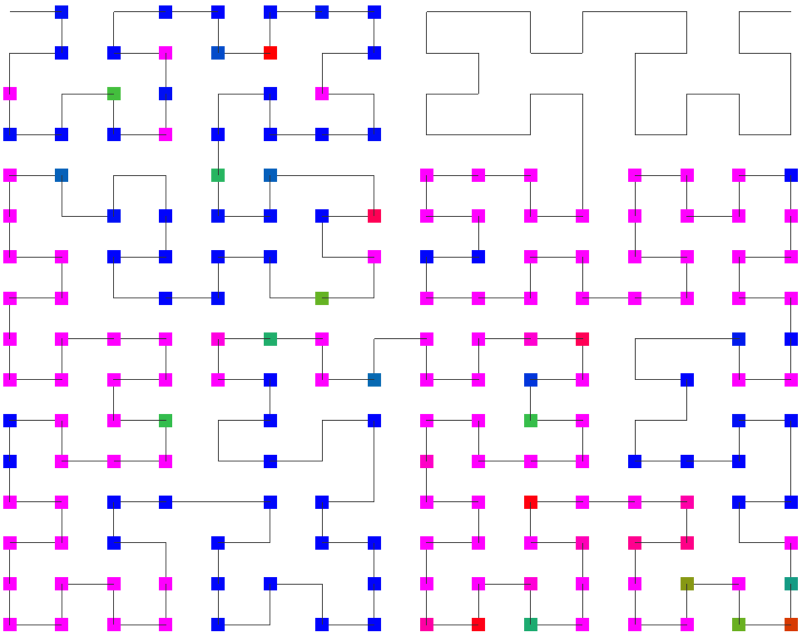


Figure 4.13: 4^{th} order Hilbert Curve showing plotting path for class A (/8) network blocks.

data bins with netmasks of /16 (thus holding 2^{16} IP addresses), and a 12^{th} (/24 bins) order curve respectively. The increase in granularity can be seen, particularly in the difference in between Figures 4.14 and 4.15.

Applying heat map colouring allowed for an extra dimension of information to be conveyed though the colouring associated with each point. Generally the packet counts in each bin are used, but interesting analysis was also done using the count of the distinct hosts observed in each bin. The latter application provides for a better understanding of how widespread the source activity is within a particular network, rather than just looking at the volume of traffic originating from particular netblocks. An example of this heat map colour scheme is shown in Figure 4.16, which is a re-plot using an 8^{th} order curve of the same data used in Figures 4.13 to 4.15, but with a colour index determined by the number of unique hosts in each /16 netblock. This plot shows a summary of the 63 million packets recorded by the CAIDA Telescope. Red dots show networks with the greatest number of unique hosts, with the colouring graduation running from green through blue, orange and finally red. The inset shown in the top right corner shows an enlarged view of the 210.0.0.0-221.0.0.0 network ranges. The area of the image occupied by this insert is traditional Class D (224.0.0.0/4) and E (240.0.0.0/4) address space, which are respectively used for Multicast and 'Reserved', and consequently are generally

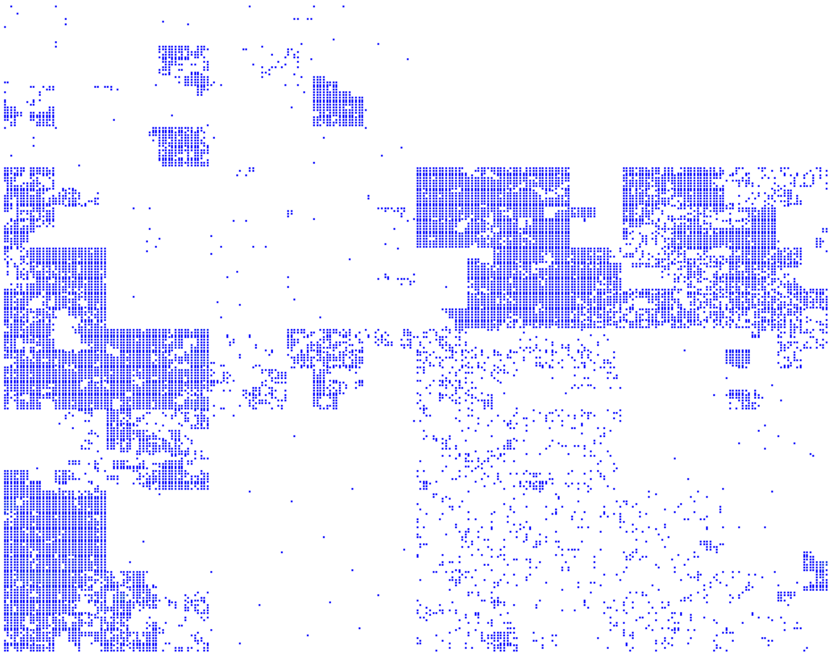


Figure 4.14: 8^{th} order Hilbert Curve representation with buckets corresponding to /16 networks (Class B) blocks

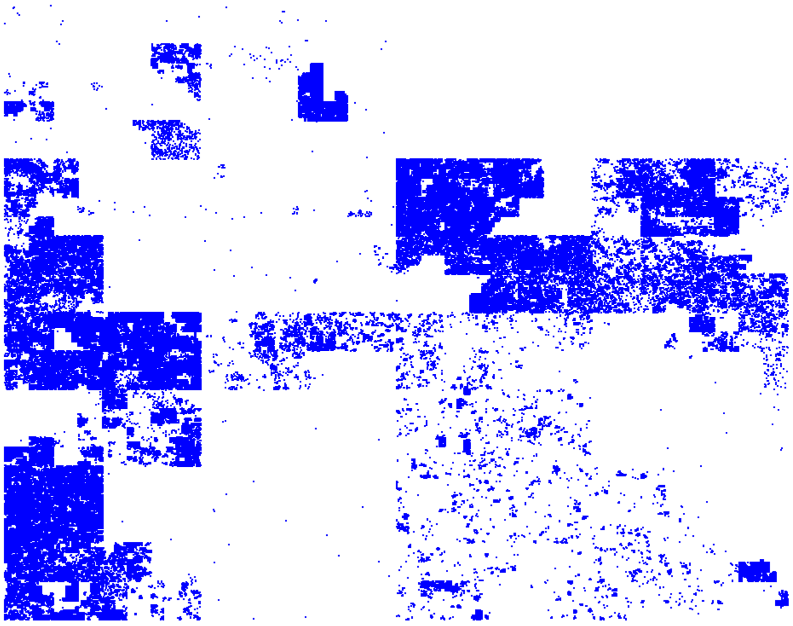


Figure 4.15: 12^{th} order Hilbert Curve representation with buckets corresponding to /24 (Class C) blocks

empty in the plots.

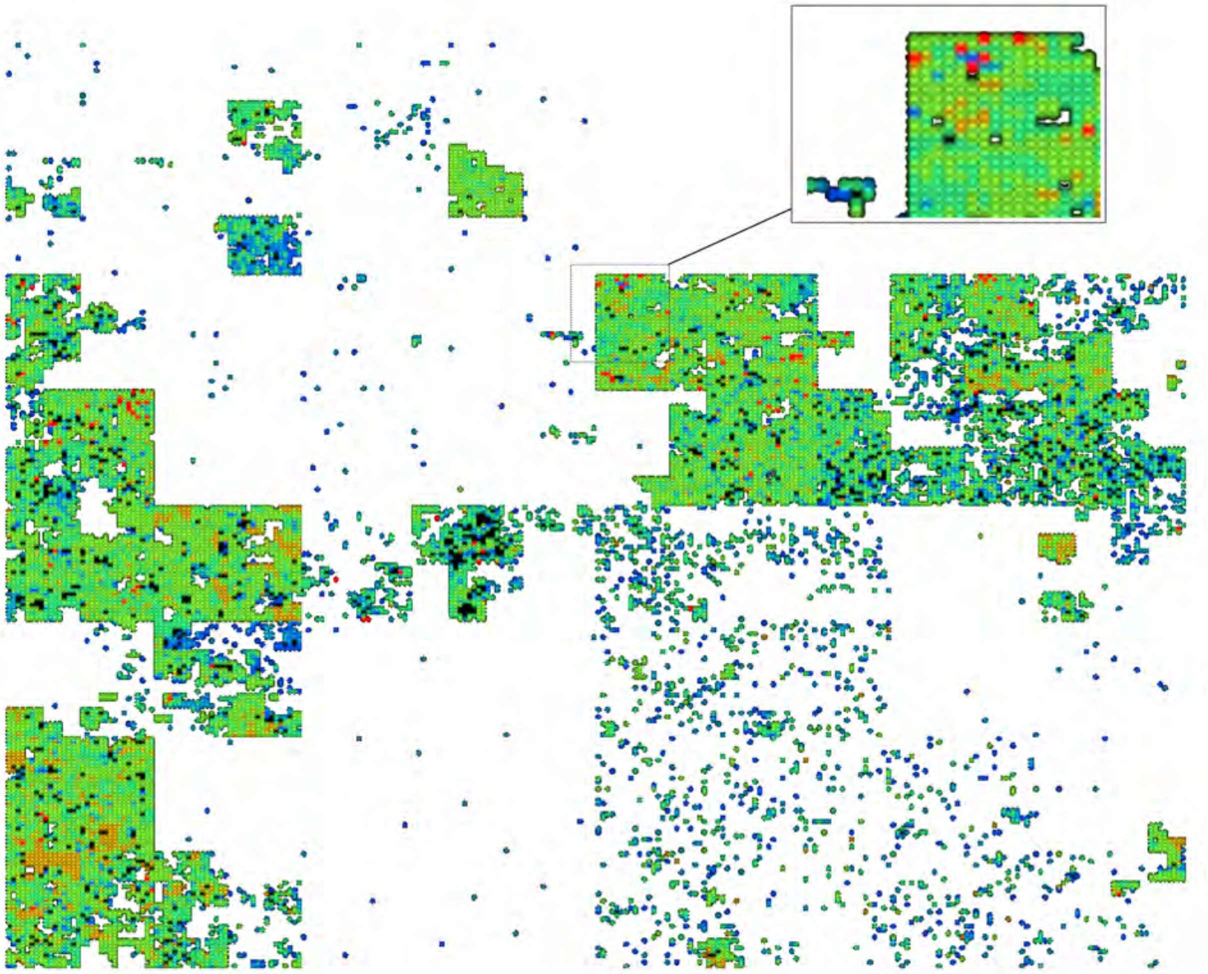


Figure 4.16: 63 million packets taken from the CAIDA Backscatter Dataset: February 28th 2007

4.5.4 Related Use

Another application of the tool, was in work done by Richter (2008) and Irwin and Pilkington (2008a) in analysing the coverage of various worm propagation algorithms. Other projects are making use of the same Hilbert algorithm for performing similar research, also inspired by Munroe's original work. The first is the ANT Censuses of the Internet Address Space (Heidemann and Pradkin, 2007), which uses the curve for plotting IPv4 address space usage as part of the Internet Census Project (Heidemann *et al.*, 2007). A second related project is that being run by Measurement Factory using this to visualise BGP route data (Wessels, 2007).

Similar work has also been done by CAIDA on visualising the use of IPv4 address space (2009). The CAIDA plots however, use a slightly different projection of the Hilbert Curve to that used by Measurement Factory/ISI and the researcher's own tool, resulting in a rotation of the images.

4.6 InetVis

InetVis is a novel three-dimensional tool developed for performing interactive inspection of both historical and 'live' network traffic. The initial design of the system was inspired by the *Spinning Cube of Potential Doom* (Lau, 2004) which interfaced with the BRO IDS system. The tool developed added a number of important features to Lau's original design. Details of the initial implementation can be found in van Riel and Irwin (2006b).

The basic design of the system is shown in Figure 4.17, which shows the three primary axes marked as Internet (red), home network (blue) and destination port for TCP/UDP (green). Underneath the cube, an additional plane is plotted to display ICMP traffic. An example plot showing a Nmap scan making use of parallel decoy scans is shown in Figure 4.18. Input to InetVis is via live capture using the libpcap library as an underlying framework, or by making use of the recorded pcap files produced by the telescope sensor. Used in conjunction with the Database system and WireShark, it proved to be a very useful means of analysing large data samples and isolating areas of of the captured data which warranted further examination.

Several extensions and refinements were made on Lau's work and are detailed in van Riel and Irwin (2006b) following a full discussion in van Riel (2005). Features found to be of particular use in analysing the RUSCOPE1 and CAIDA traffic were:

ICMP plane — In addition to the primary cube use in Lau's work, an additional plane was added under the cube to plot ICMP traffic. This was seen as logical due to there being no port information associated with ICMP datagrams as there is for those making use of TCP and UDP.

Time Manipulation — This feature proved to be one of the most valuable in terms of isolating areas of interesting traffic. The feature allowed for playback of the datagrams to be slowed down significantly and also sped up to a

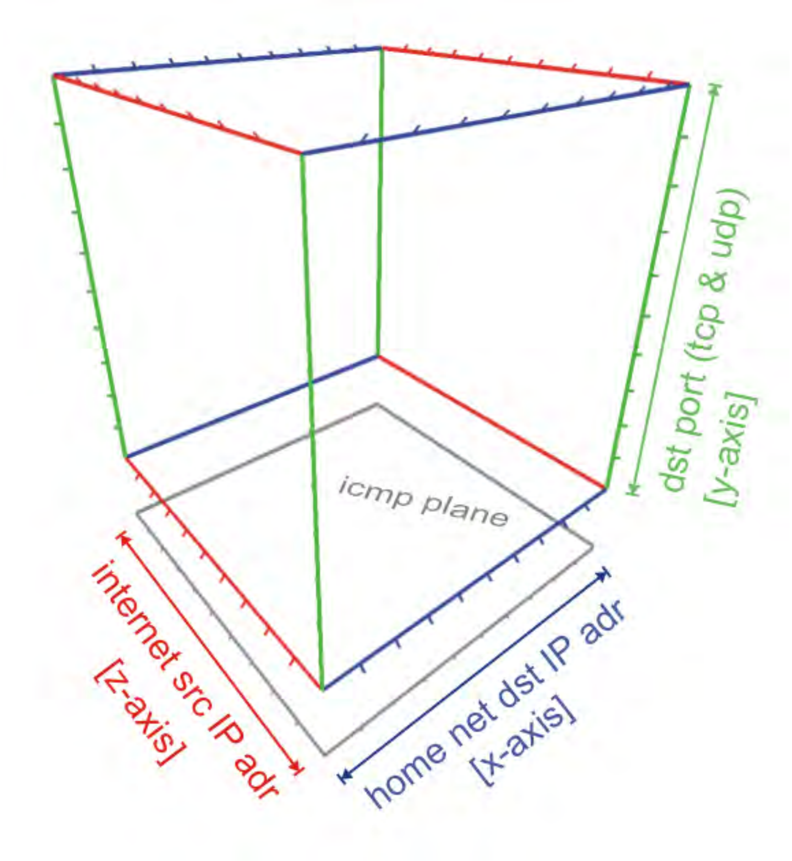


Figure 4.17: InetVis Plotting Scheme

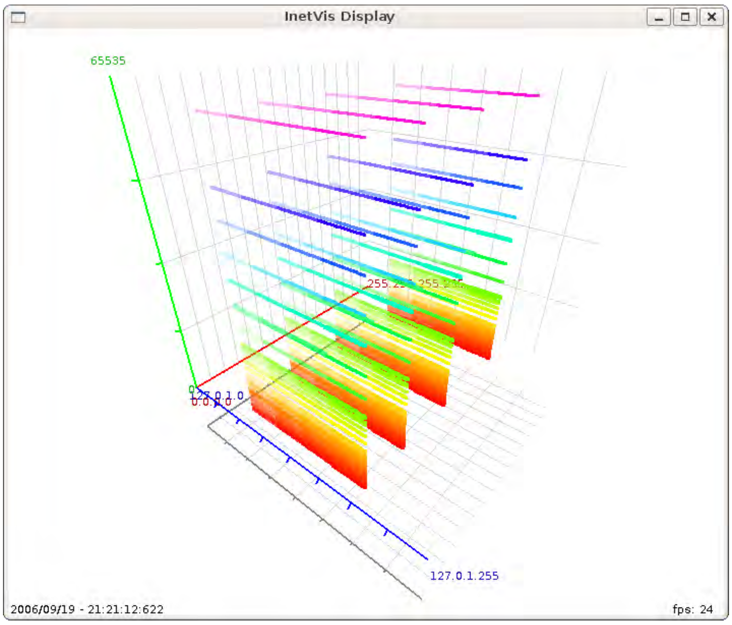


Figure 4.18: Sample InetVis plot

maximum rate of 86 400 times 'wall time', effectively displaying traffic at the rate of one day per second. This is only possible when processing pre-recorded traffic capture files.

Torpedo mode — So named due to its visual appearance, this plotting scheme drew attention to scanning activity by plotting new events with slightly larger points, giving the impression when watching scanning traffic of 'photon torpedoes' travelling across the cube. This effect is very noticeable, allowing for very easy visual identification of sequential scans. Random scanning was also detectable, although to a lesser extent due to the 'twinkling' effect that this created.

Filtering — A highly functional BPF based filter was built into the tool, allowing for narrowing down replayed traffic to specific hosts, network blocks, protocols or port ranges. The ability to perform the filtering internal to the application allowed for easier analysis than having to reprocess and reload the capture files subsequent to filtering by external tools.

One of the strengths of InetVis was in the identification of scanning activity. The application of this tool for this purpose has been discussed in van Riel and Irwin (2006a) and in Irwin and van Riel (2007). This tool has also been instrumental in isolating traffic used for training and testing of the tools and algorithms developed and used in the discussions contained in Barnett and Irwin (2008, 2009). A more generic application to visual intrusion detection is given in van Riel and Irwin (2006c). Conti and Abdullah (2004) provide some discussion of similar work in visual analysis of network attack traffic.

This tool was selected as one of the data visualisation tools included in DAVIX[20], a Linux based live CD for data analysis and visualisation (Monsch and Marty, 2008). It is also features as one of the visualisation tools discussed in *Applied Security Visualisation* (Marty, 2008), and was featured in the June 2008 *Toolsmith* column of the ISSA[21] journal (McRea, 2008). A re-implementation of the original C++ and OpenGL based tool was completed in 2010 with the resultant *dotNetVis* (Schwagele, 2010) making use of a client server architecture and the more modern Microsoft XNA framework.

[20] `http://www.secviz.org/node/89`

[21] Information System Security Association — `http://www.issa.org/`

4.7 Heatmaps

Two dimensional visualisations are produced using time series data, converting linear ‘time’ based data series into a two-dimensional array though the use of logical clustering. Such clustering is most commonly performed at the day level, or even at hour level. However at higher resolutions, the images tend to be a little overcrowded to be of any real use in providing any kind of detailed overview of the data concerned. They can provide overall trends, and allow for the identification of further areas of potential interest. A heat map colouring is used for the individual grid blocks, ranging from dark blue through to red (analogous to ‘cool’ to ‘hot’) over the HSV[22] colour space.

The use of the continuum was chosen in preference over chloropleth colour swatches (as used in the geopolitical analysis discussed in Section 4.4) due to the higher cell numbers to be filled in comparison to the world maps. The use of a continuous colour space also allows for subtle deviations to be shown, which is not possible with the banded chloropleth schemes. An example of the output produced by the heat map tool is shown in Figure 4.19; which shows data over the entire period of the RUSCOPE1 observation.

In common with many of the other tools developed by the researcher, the data was extracted from the database as CSV files, and then post-processed and reformatted into a suitable input to the tool. Data for this plot has been normalised to account for the capture period only starting in August 2005, and concluding in September 2009 (thus 2005 and 2009 being incomplete years) by means of the aggregate packet totals being divided by the number of days present, for each day of month combination. Black squares are used to indicate where there are no valid values Although 2008 was a leap year, data was removed for February 29th, hence the black square. In contrast, white squares are used to indicate missing data, as shown in Figure 4.20. Data is missing in most cases due to network or Internet connectivity failures, as mentioned in Section 3.2.

The real value of using such plots is to provide a quick overview of any anomalies in the data which warrant further exploration. Given the volume of data, tools such as this are invaluable in allowing one to narrow down the focus. For example, even at a highly aggregated level, the anomalous flood of ICMP Type 11 datagrams

[22] **H**ue, **S**aturation and **V**alue - a cylindrical colour co-ordinate system

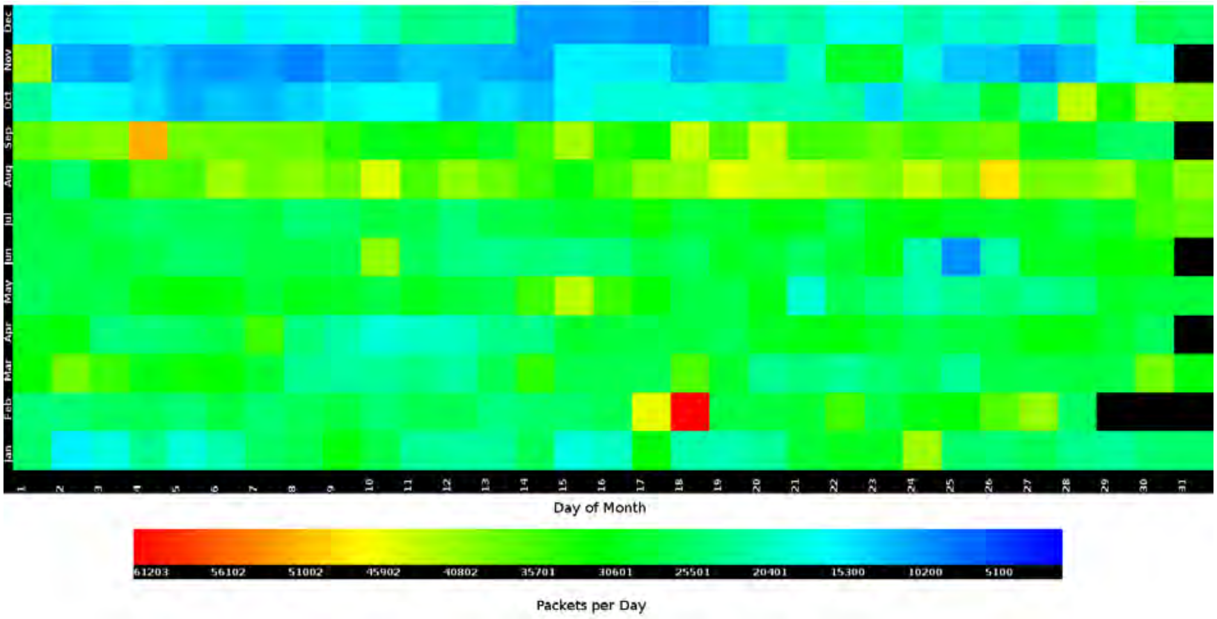


Figure 4.19: Sample Heat map Plot

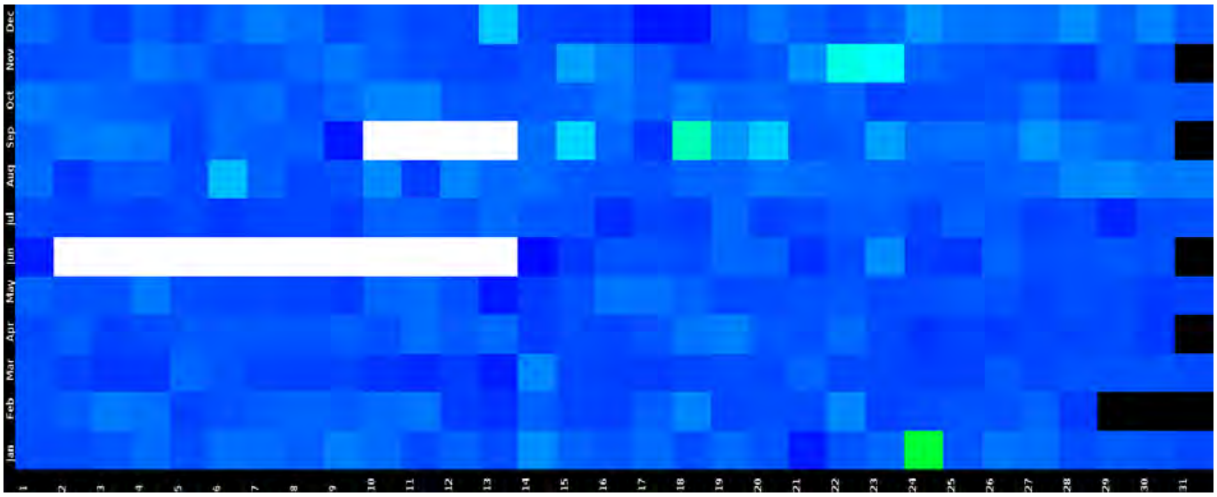


Figure 4.20: Heat map Plot by Month and Day - 2006

discussed in Section 5.3.4 can be seen as ‘hotspots’ of yellow followed by red on the 17^{th} and $18^{th}$of February in Figure 4.19. The actual cause of this anomaly was traffic observed in February 2009, the details of which are discussed in the following chapter.

The cooler colours in November and early December are most likely due to the lower observed traffic volumes linked to a decrease in traffic attributable to South African hosts during this period. The second half of December is somewhat warmer by comparison, most likely due to skewing of the result data by the massive influx of Conficker related traffic as discussed in Chapter 7. Further use of these plots is also made in the following chapters.

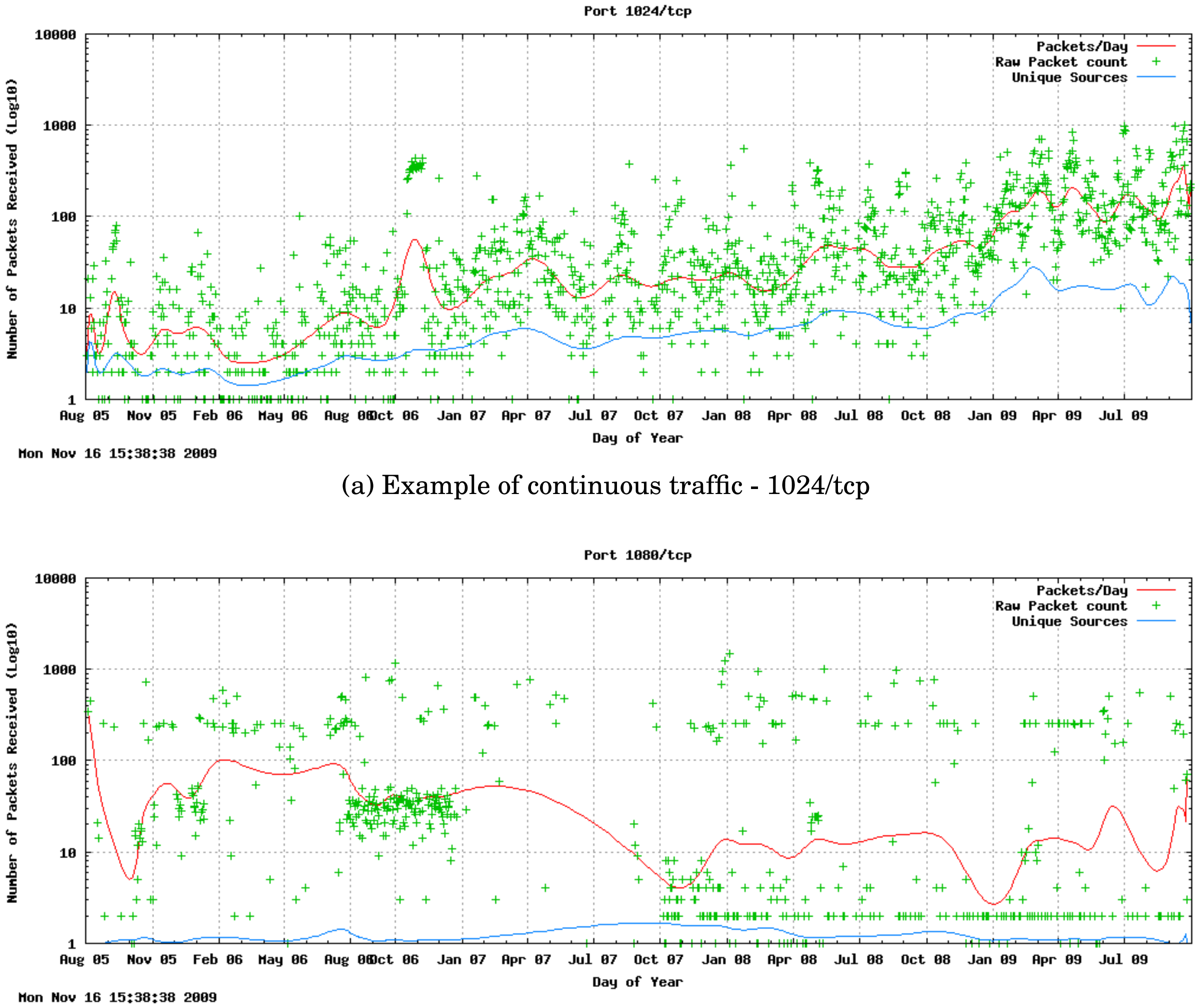
4.8 Time Series Plots

Time series plots were used extensively to evaluate trends in observed packet activity over time. These were produced by a scripted framework that extracted data from the RUSCOPE1 database and reprocessed it into a format suitable for plotting with gnuplot[23]. Readers are referred to *Gnuplot in Action* (Janert, 2009) for a good overview of the gnuplot tool. Two examples of differing types of observation are shown in Figure 4.21 for ports 1024/tcp and 1080/tcp. These were chosen for illustrative purposes due to the two plots exhibiting different behaviour.

In both Figures 4.21a and 4.21b, three values are plotted. The first of these is the raw packet count on a daily basis (green points). From this data, a fitted curve is calculated and plotted (red line) using a spline, with the intention of showing the dominant trend of the individual daily observations. The final value, plotted in blue, is the number of distinct source IP addresses seen on a daily basis. The relationship between the number of unique sources observed and the trend in packets received, can be used as an indicator of heavy scanning activity, either as the number of hosts spike, or as seen in Figure 4.21a, where in October 2006 there is an increase in packet counts, but no increase in host counts.

In the data used to produce these images there has been no discrimination between active and passive TCP traffic as discussed in Section 2.3. Figure 4.21a provides an example of traffic that was observed fairly continuously over the duration of

[23] `http://www.gnuplot.info/`



(a) Example of continuous traffic - 1024/tcp

(b) Example of bursty traffic - 1080/tcp

Figure 4.21: Sample Time Series Plots

the RUSCOPE1 dataset. In contrast, Figure 4.21b is an exemplar of bursty, yet persistent traffic, with a significant level of activity recorded during the period August-October 2006. Very little traffic was then recorded for the year through to October of 2007, at which point the recorded levels started fluctuating fairly erratically.

4.9 Summary

This chapter introduced the analytic methods and iterative approach that was used for evaluation of the collected datasets. An overview was presented of the graphical analysis tools developed in the course of conducting the research. The results of the data analysis and interpretation of the outputs of the application of these tools are explored in the following three chapters, as the RUSCOPE1 dataset is explored in detail.

The goal is to transform data into information,
and information into insight.

Carly Fiorina - former CEO and President of HP

5

Analysis: Protocol Level

BUILDING on the tools and methods previously discussed in Chapter 4, an exploration and analysis of the data from the RUSCOPE1 dataset is presented. While the primary focus is on the analysis and interpretation of the Rhodes Telescope data set, comparative values from other datasets are also presented where relevant.

The chapter opens with a high level analysis of the dataset — the so-called 'Top 10' by network address and protocol groupings found in the captured traffic. An analysis of the various protocols observed is discussed in Sections 5.1 and 5.2. Approaching the data analysis from the perspective of the observed source addresses, analysis is presented in Section 5.3. Section 5.4 presents an analysis of the traffic by size and TTL value. A discussion of the Active *vs.* Passive traffic breakdown for TCP and ICMP datagrams follows in Section 5.5. The chapter concludes with a summary of the analytic results. Appendix C contains an overview of much of the networking data, including protocol diagrams for IP, TCP, UDP and ICMP; details of network address block allocations; Bogon addresses and country code allocations used in DNS.

Throughout this chapter and the following, the monitored IP network used for the RUSCOPE1 sensor is anonymised as 196.$x.x.x$. In some cases, the last octet is presented (as in 196.$x.x.$1) where the actual host being targeted within the monitored network is of relevance. The purpose of this is to obscure the actual address space to prevent possible future pollution and potential poisoning of the sensor via deliberate targeted scanning. Similarly, individual hosts mentioned (such as in Table 5.14) have their last octet similarly masked with a letter. In cases where there are several hosts of interest, alternate letters are used. Many of these hosts may remain online and are possibly currently infected with malware, and as such are potentially vulnerable to further exploitation. The exception to this practise is when discussing Bogons in section Section 6.5, since these were either unallocated (as at the time of data capture) or are globally unroutable addresses in terms of the specifications contained in RFC 1918 (Rekhter *et al.*, 1996) and RFC 3330 (IANA, 2002).

Select examples and diagrams are presented in order to suitably illustrate the concepts being discussed. More detailed outputs and figures are provided in the accompanying electronic appendix, details of which can be found in Appendix F. Anomalies and other interesting observations are highlighted within the discussion. All times stated are GMT+2:00 unless otherwise noted.

In most cases the top ten items in each category are presented, along with the percentage contribution to the whole that these make up. While it is acknowledged that ten is a rather small sample size (out of 65 535 in the case of port numbers for TCP/UDP), it is shown to be a suitable measure representing a significant portion of the specific population as a whole, and provides a compromise in terms of achieving a balance towards readability rather than pages of detailed data[1]. Mention is made of other classifications outside of this range if relevant to the discussion at hand.

5.1 Top Talkers

This section deals with the so-called ‘top talkers’ for a number of different observed components. The analysis starts (Section 5.1.1) at the network (IP) layer of the

[1] Detailed data extracts are available on the accompanying electronic addendum detailed in Appendix F

Table 5.1: IP Protocol breakdown

Rank	Protocol	Number	%
1	TCP	6	81.57
2	UDP	17	12.62
3	ICMP	1	5.89
		Total	99.99

N=40 801 854

Breakdown of the three primary protocols in use on IPv4 networks.
Other protocols constituted 1 081 packets.

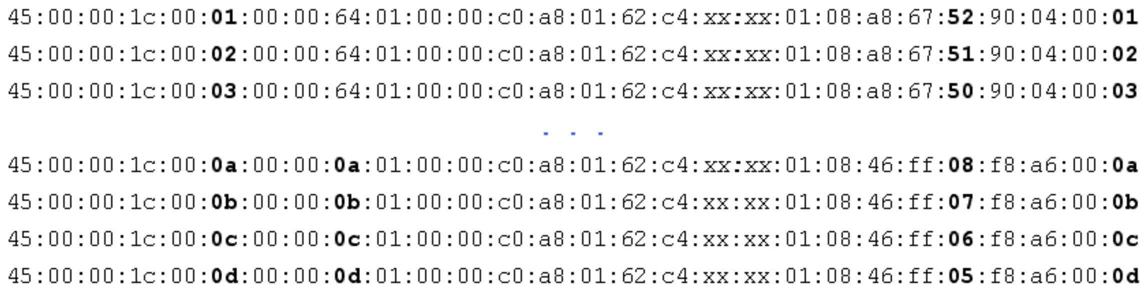
TCP/IP stack evaluating the protocols recorded as being carried by the IPv4 datagrams. Each protocol is addressed in detail in Section 5.2. Rankings are based on the number of datagrams received for each classification.

5.1.1 Protocol Analysis

The Internet Protocol version 4 (IPv4) allows an 8bit field for the higher level protocol being transported in the v4 datagram (Postel, 1981c). At the time of writing, the IANA registry[2] has allocated only 144 of the 256 possible values, although many of these allocations are defunct and currently deprecated. The most common traffic types seen online today are the three workhorse protocols that have been around since the early days of the Internet: ICMP, TCP and UDP. These three protocols account for 99.997% of the traffic collected. A breakdown of traffic received by protocol is shown in Table 5.1, based on the number of individual datagrams of each type received. This is similar to what one expects to see on a production network, with TCP constituting the bulk of traffic, both by byte count and volume, due to its use in protocols such as HTTP and FTP. UDP is primarily used for DNS resolution and some audiovisual streaming, with ICMP being used mostly for 'pings' (Types 0 and 8) as well as an out of band signalling mechanism (primarily through the use of Type 3 messages). Datagrams with protocols[3] not covered by the primary grouping amounted to only 1 081 (less than 0.003% of the whole), covering another 66 protocols, with the majority (884 packets) being of protocol 255 (classified as 'Reserved' by IANA), followed by types 46 (RSVP) and 96 (SCC-SP). From the unallocated range of protocol numbers (141-252), 37 datagrams were received, representing twenty-eight different protocols.

[2] `http://www.iana.org/assignments/protocol-numbers/`

[3] *ibid.*



```
45:00:00:1c:00:01:00:00:64:01:00:00:c0:a8:01:62:c4:xx:xx:01:08:a8:67:52:90:04:00:01
45:00:00:1c:00:02:00:00:64:01:00:00:c0:a8:01:62:c4:xx:xx:01:08:a8:67:51:90:04:00:02
45:00:00:1c:00:03:00:00:64:01:00:00:c0:a8:01:62:c4:xx:xx:01:08:a8:67:50:90:04:00:03
                                        . . .
45:00:00:1c:00:0a:00:00:0a:01:00:00:c0:a8:01:62:c4:xx:xx:01:08:46:ff:08:f8:a6:00:0a
45:00:00:1c:00:0b:00:00:0b:01:00:00:c0:a8:01:62:c4:xx:xx:01:08:46:ff:07:f8:a6:00:0b
45:00:00:1c:00:0c:00:00:0c:01:00:00:c0:a8:01:62:c4:xx:xx:01:08:46:ff:06:f8:a6:00:0c
45:00:00:1c:00:0d:00:00:0d:01:00:00:c0:a8:01:62:c4:xx:xx:01:08:46:ff:05:f8:a6:00:0d
```

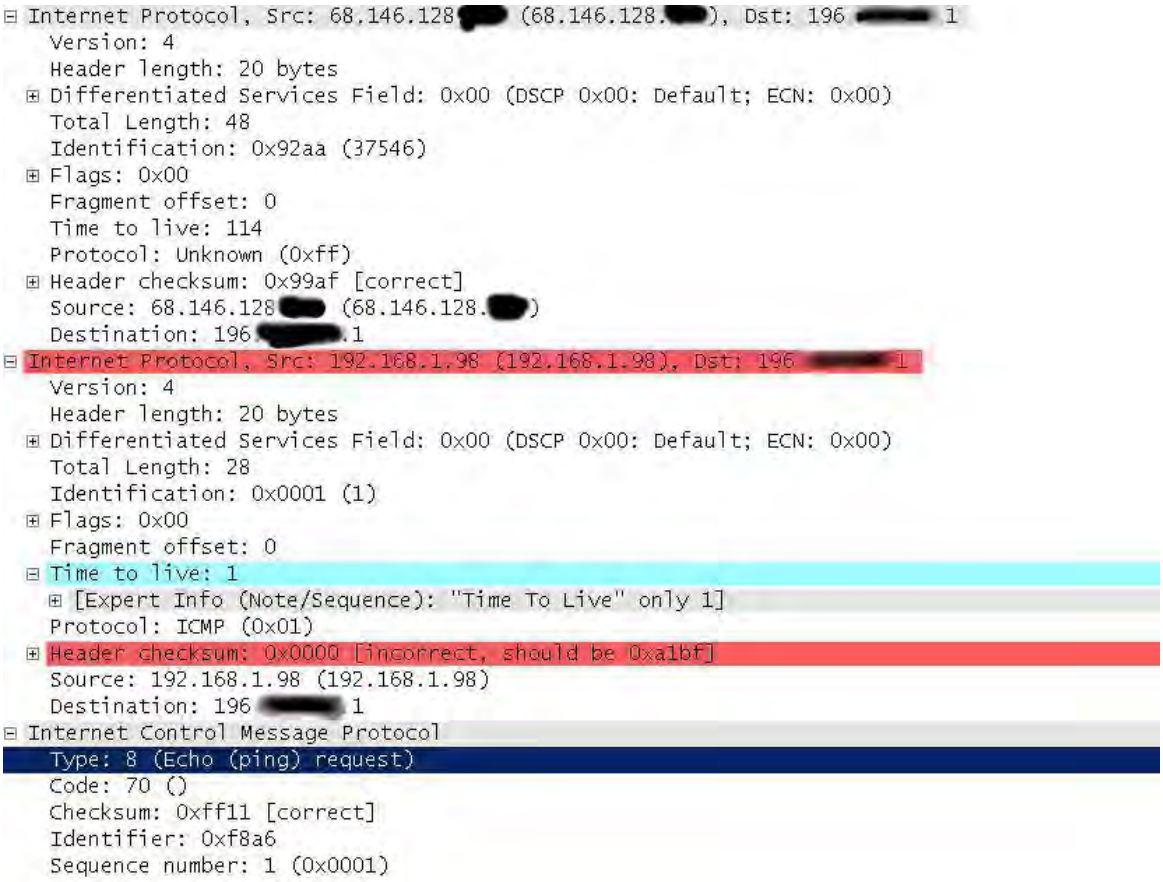
Note: xx:xx has been used to redact the destination addressing contained in the packet payloads. Varying values are bolded.

Figure 5.1: Strange packets observed using protocol 255

A closer look at the datagrams with a protocol type of 255, shows that the majority (800) of these arrived in a burst between 2009-02-17 09:20:36 and 2009-02-26 21:52:32 from host 68.146.128.x belonging to Shaw Communications and located near Calgary in Canada, and all destined for 196.x.x.1. The 28 byte payload is all binary, and contains several values which seem to vary monotonically with each packet. Hexadecimal dumps of a selection of the packet payloads are shown in Figure 5.1. The varying fields within these payloads are highlighted using a bold font.

Experimentation with various protocol decoders in WireShark showed that this was in fact an encapsulated IP datagram, which in turn held an ICMP datagram. A WireShark decode of a sample packet is shown in Figure 5.2 with some fields redacted. This indicates that there is a ICMP datagram encapsulated from an RFC1918 address (observed addresses include 192.168.1.{89,94,98}) which is being used to perform a traceroute to 196.x.x.1, hence the incrementing value shown in the last fields of the second block of payloads in Figure 5.1. The use of these datagrams as part of a traceroute, can be deduced by observing the TTL values to initially be one (as in Figure 5.2) and increasing.

The first highlighted field in Figure 5.2 is the incrementing IPID, the second a variable TTL, the third the changing checksum for the ICMP packet, and the final element the ICMP sequence number. One can conclude with a level of certainty that these datagrams are the result of packet mangling via a NAT device of some sort. The origin system is likely to be running an operating system in the Windows Family due to the use of ICMP for performing a traceroute. Similar packets received on 2006-09-19, also from a Canadian host also decode sensibly as ICMP datagrams containing echo replies (as shown in Figure 5.3), but were targeted at a



```
⊟ Internet Protocol, Src: 68.146.128[redacted] (68.146.128.[redacted]), Dst: 196 [redacted] 1
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 48
    Identification: 0x92aa (37546)
  ⊞ Flags: 0x00
    Fragment offset: 0
    Time to live: 114
    Protocol: Unknown (0xff)
  ⊞ Header checksum: 0x99af [correct]
    Source: 68.146.128[redacted] (68.146.128.[redacted])
    Destination: 196[redacted].1
⊟ Internet Protocol, Src: 192.168.1.98 (192.168.1.98), Dst: 196 [redacted]1
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 28
    Identification: 0x0001 (1)
  ⊞ Flags: 0x00
    Fragment offset: 0
  ⊟ Time to live: 1
    ⊞ [Expert Info (Note/Sequence): "Time To Live" only 1]
    Protocol: ICMP (0x01)
  ⊞ Header checksum: 0x0000 [incorrect, should be 0xa1bf]
    Source: 192.168.1.98 (192.168.1.98)
    Destination: 196 [redacted] 1
⊟ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 70 ()
    Checksum: 0xff11 [correct]
    Identifier: 0xf8a6
    Sequence number: 1 (0x0001)
```

Figure 5.2: WireShark decode of packet with Protocol 255

```
0000 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f ................
0010 10 11 12 13 14 15 16 17 18 19 1a 1b 1c 1d 1e 1f ................
0020 20 21 22 23 24 25 26 27 28 29 2a 2b 2c 2d 2e 2f !"#$%&'()*+,-./
0030 30 31 32 33 34 35 36 37 38 39 3a 3b 3c 3d 3e 3f 0123456789:;<=>?
0040 40 41 42 43 44 45 46 47 48 49 4a 4b 4c 4d 4e 4f @ABCDEFGHIJKLMNO
0050 50 51 52 53 54 55 56 57 58 59 5a 5b 5c 5d 5e 5f PQRSTUVWXYZ[\]^_
0060 60 61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f ‘abcdefghijklmno
0070 70 71 72 73 74 75 76 77 78 79 7a 7b 7c 7d 7e 7f pqrstuvwxyz{|}~.
0080 80 81 82 83 84 85 86 87 88 89 8a 8b 8c 8d 8e 8f ................
0090 90 91 92 93 94 95 96 97 98 99 9a 9b 9c 9d 9e 9f ................
00a0 a0 a1 a2 a3 a4 a5 a6 a7 a8 a9 aa ab ac ad ae af ................
00b0 b0 b1 b2 b3 b4 b5 b6 b7 b8 b9 ba bb bc bd be bf ................
00c0 c0 c1 c2 c3 c4 c5 c6 c7 c8 c9 ca cb cc cd ce cf ................
00d0 d0 d1 d2 d3 d4 d5 d6 d7 d8 d9 da db dc dd de df ................
00e0 e0 e1 e2 e3 e4 e5 e6 e7 e8 e9 ea eb ec ed ee ef ................
00f0 f0 f1 f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff ................
```

Figure 5.3: ICMP payload encapsulated in protocol 255 datagrams.

range of addresses within the monitored block. Similar datagrams were not found in other datasets examined, although there have been reports of similar traffic dating back to July 2002[4]. In this case the source was a host in Italy (IT).

The datagrams discussed above, are a prime example of the most likely non-malicious, yet inexplicable traffic observed as part of the general clutter of Internet Background Radiation (IBR). This observed anomaly provides three different classes of IBR. The first is anomalous use of a reserved protocol, the second incoming ‘active’ traffic as part of a traceroute, and the final one being reflected traffic as the result of spoofed packets using the address range monitored by the network telescope.

5.2 Protocols

This section presents an analytic overview of observed traffic grouped by the three primary protocols observed: TCP, UDP and ICMP. For TCP and UDP, analysis is performed on the basis of Source and Destination ports. While most people are concerned about the destination of traffic inbound to a network or host, the source ports used in the connection can shed light on some of the more subtle happenings

[4] http://seclists.org/incidents/2002/Jul/68

Table 5.2: Packet Fragmentation

	Packet Count	
Protocol	Fragmented	Unfragmented
TCP	1 780	33 281 463
UDP	117	5 147 850
ICMP	1 557	2 368 006
Other	0	1 081
Total	3 454	40 798 400

on the network, particularly when looking at backscatter and passive traffic analysis. ICMP traffic discussed in Section 5.2.3 used groupings by Type and Code, an overview of which is contained in Appendix C. For the purposes of the analysis performed in this section, all traffic was included with no discrimination between active and passive traffic as previously discussed in Section 2.3.

Fragmentation of datagrams is something one would not expect to see with traffic captured on a network telescope, primarily since one is not expecting any data to be received due to its passive nature. A summary of the fragmented datagrams received for each protocol is given in Table 5.2. While no packets from the other protocols observed had the fragmented flag set, several were found to be packet fragments on later inspection.

5.2.1 TCP

TCP traffic constitutes the lion's share of the traffic observed as previously indicated in Section 5.1.1. Of this, traffic destined to 445/tcp represented over half (50.75%) of the TCP traffic captured and 41.4% of traffic overall. Chapter 7 discusses traffic destined to port 445/tcp in further detail. The top ten ranked TCP ports as recorded by both source and destination ports are shown in Table 5.3. The values in this Table make no distinction between active and passive traffic, which is discussed in Section 5.5.

Performing an analysis of the source ports used in TCP datagrams is of interest for two reasons. The first is that ports may be selected by individuals running network scans in order to try bypass firewall rules, which may be incorrectly implemented, and allow bypass for common ports. The second reason is in characterising backscatter which is discussed in detail in Section 5.5.2. In the case of 80/tcp only

Table 5.3: Top TCP Ports

Source Ports				Destination Ports		
Port	P_{count}	%	Rank	Port	P_{count}	%
80	2 334 289	7.01	*1*	445	16 893 920	50.75
6000	845 593	2.54	*2*	135	2 749 950	8.26
7000	309 663	0.93	*3*	139	1 680 226	5.04
25511	179 829	0.54	*4*	22	1 224 074	3.67
3389	73 005	0.22	*5*	1433	1 156 668	3.47
5641	50 327	0.15	*6*	2967	888 935	2.67
6667	31 023	0.09	*7*	5900	491 818	1.47
22	28 262	0.08	*8*	23	449 281	1.35
3306	15 717	0.04	*9*	80	367 202	1.10
9201	15 679	0.04	*10*	50272	325 636	0.98
Total		11.67		*Total*		78.80

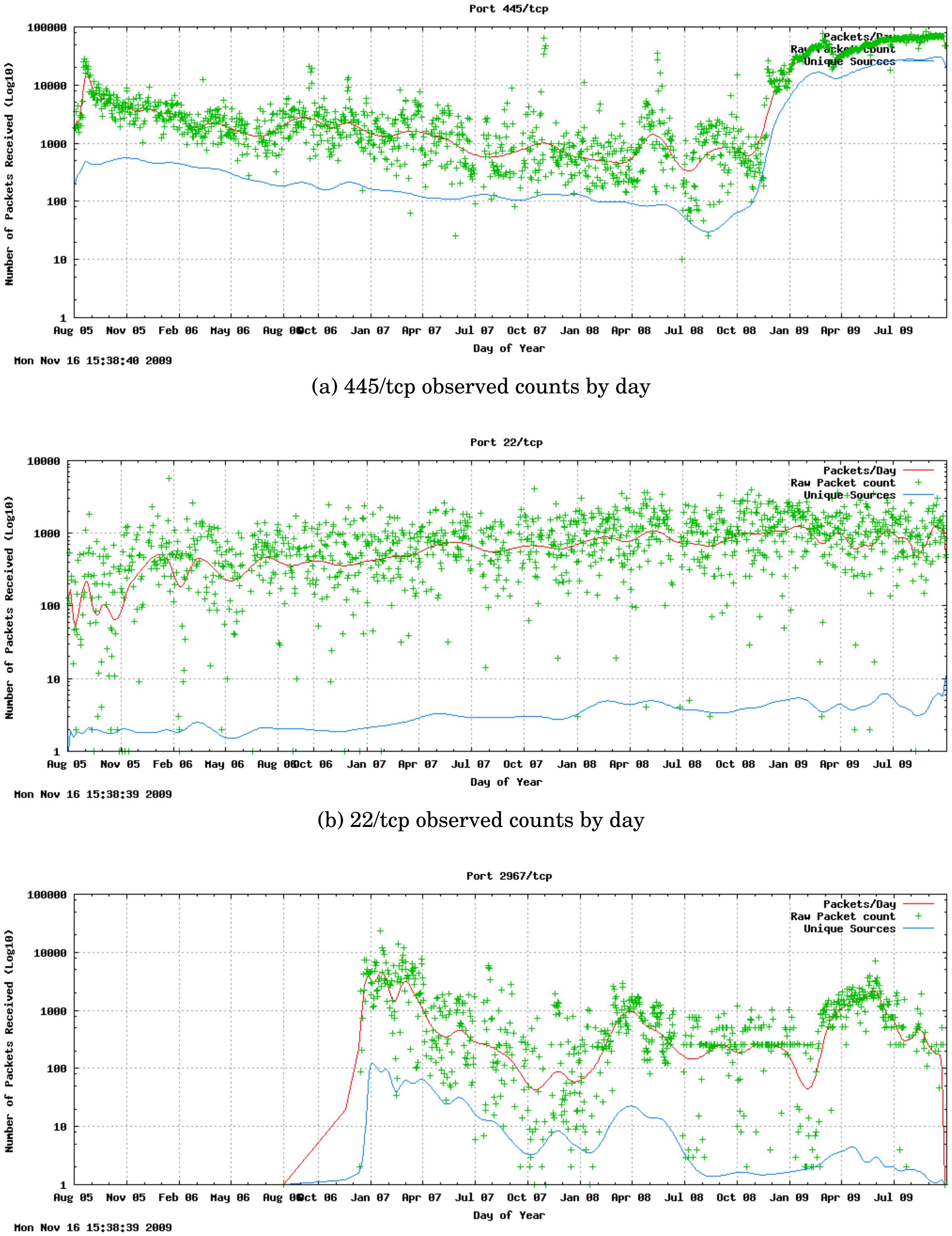
N= 33 283 243

13% of the traffic could be characterised as such. The distribution of traffic over source ports was more widespread with traffic originating from 65 492 distinct ports. The top twenty ports only accounted for 12.02% of TCP traffic.

The destination ports of observed datagrams are the usual focus of traffic analysis. Although traffic was observed as destined to 65168 ports, the majority was directed to a very small portion of possible ports. While the top ten ports accounted for 78% of traffic, the top twenty included nearly 84%, showing a a fairly concentrated focus of traffic. With the exception of 2967/tcp and 50272/tcp, the other ports are all well known services as allocated by IANA[5,6]. The former is used by the Corporate Edition Symantec AntiVirus, and the scanning for this port may be related to the traffic observed on 38293/udp discussed in Section 5.2.2. Timeline plots for selected destination ports from the top ten are shown in 5.4. The sharp rise in traffic observed in 445/tcp can be clearly seen in 5.4a, while a more consistent banded level of traffic for 22/tcp is evident in 5.4b. The graph in 5.4c shows an interesting trend in terms of the scanning having emerged fairly suddenly in late 2006. The number of distinct sources recorded on a daily basis decreased from an initial high of just over 100 to under 10. Even at these low levels, on average over 250 packets a day were received, as can be noted quite clearly during the period of July 2008 to January 2009.

[5] http://www.iana.org/assignments/port-numbers

[6] The augmented list at http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers provides information on non-official allocations



(a) 445/tcp observed counts by day

(b) 22/tcp observed counts by day

(c) 2967/tcp observed counts by day

Figure 5.4: Selected Top TCP destination ports

Table 5.4: TCP traffic by port range

Source Port		Port	Destination Port	
Count	%	Range	Count	%
2 473 196	7.430	0-1023	24 358 083	73.184
28 711 100	86.262	1024-49151	7 398 522	22.229
2 098 947	6.308	49152–65535	1 526 638	4.587

N= 33 283 243

Well-known ports: 0–1023

Registered ports: 1024–49151

Dynamic, private or ephemeral ports: 49152–65535

A final look at the distribution of traffic for both source and destination ports can be seen in Table 5.4. These results are not unexpected, since the majority of network services run in the well-known range of 0-1023, and in many cases require administrative privileges on the host to operate in this range. The bulk of the source ports are in what is known as the 'registered range', where other services can be registered with IANA. In practice many application developers simply choose an unassigned port in this range. The majority of TCP/IP stack implementations allocate a port from this range when an outgoing connection is made, hence the higher proportion of source traffic in this range.

Looking at the TCP flags received, 62 of the possible 64 combinations were received, although 54 of these combinations were recorded with less than 1 000 packets, the majority having less than 30. A summary of the major contributors is shown in Table 5.5. The three combinations present in more than 1% of the traffic together account for 99.2% of the total TCP packet count, once again showing the observed trend seen in the data that relatively few data items represent a significant portion of the whole. The prevalence of scanning traffic is apparent from the overwhelming presence of traffic with the SYN flag set. Prior to the spike in observed activity around the Conficker worm in November 2008, the percentage composition was roughly the same, with the ranking positions shown being consistent with the pre-Conficker and overall data. The top three ranked groupings however only accounted for 98.7% of traffic.

Considering the Backscatter datasets made available by CAIDA Shannon *et al.* (2005, 2006, 2007), and discussed in Section 3.3, given the large volumes of packets recorded in these, one could extrapolate the total data volumes arriving at the /8 network being used for monitoring, as there is probably in excess of 95% of traffic being rejected as 'active' and not falling into the definition of backscatter used in

Table 5.5: TCP Flags

	TCP Flags							
Rank	SYN	ACK	FIN	URG	PSH	RST	Count	%
1	✔						29 404 875	88.347
2	✔	✔					2 875 081	8.638
3		✔				✔	737 636	2.216
4						✔	189 304	0.568
5		✔			✔		34 173	0.102
6		✔					33 222	0.099
7		✔	✔				4 911	0.014
8	✔	✔		✔			2 082	0.006
							Total	99.99

N= 33 283 243

these datasets. Regarding fragmentation of TCP datagrams, only 1 780 datagrams were found to be fragmented. Some of these are discussed in Section 6.5.2.

5.2.2 UDP

The User Datagram Protocol (UDP) (Postel, 1980), provides a connectionless, lightweight means of transferring data between systems. The lack of a connection setup as required with TCP allows for fairly quick data transmission at the expense of reliability. What this means for a network telescope, is that UDP datagrams are likely to contain payloads, which can provide insight into the potential purpose of the packet, and in some cases, as discussed below, allow researchers to determine the exact purpose and application being targeted. This also accounts for UDP traffic having a much larger average packet size than TCP or ICMP data. This difference is discussed in Section 5.4.1.

An overview of the top ten ranked source and destination ports for observed traffic is presented in Table 5.6. As with TCP the traffic originated from a wide range of source ports, with 64 531 distinct ports observed. This range of sources is evident in that the top ten ports accounted for only 8.96% of traffic, and the top twenty 13.21%. The majority of the source ports are near the bottom of the registered port range (1024-49151). This is most likely due to the fact that new connection attempts are allocated ports greater than 1023 on the client side in the absence of a port number being explicitly specified. Particularly on Microsoft Windows family

Table 5.6: Top UDP Ports

Source Ports				Destination Ports		
Port	P_{count}	%	Rank	Port	P_{count}	%
1026	71 136	1.38	*1*	1434	2990 062	58.08
1231	68 071	1.32	*2*	137	633 620	12.30
1027	58 546	1.13	*3*	1026	320 355	6.22
1025	52 385	1.017	*4*	1027	249 309	4.84
1029	39 339	0.76	*5*	38293	32 235	0.62
1098	38 894	0.75	*6*	19932	30 976	0.60
1028	36 514	0.70	*7*	135	22 211	0.43
1031	33 227	0.64	*8*	1028	20 416	0.30
1030	32 858	0.63	*9*	1029	17 250	0.33
1036	30 526	0.59	*10*	5158	15 610	0.30
Total		8.96		*Total*		84.15

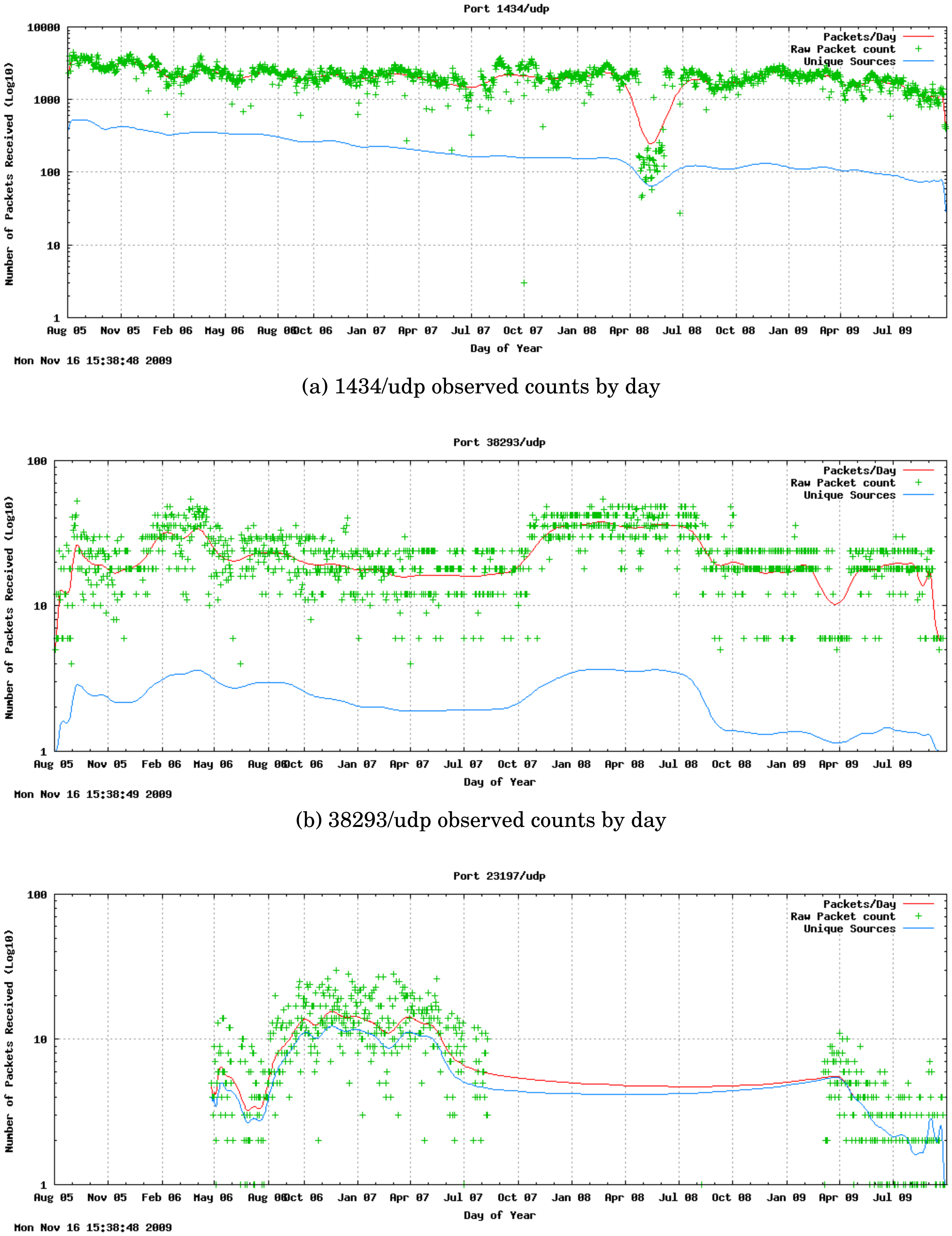
N= 5 147 967

systems, connections start from port 1025 and increment monotonically. As ports become available again they are re-used. A breakdown of the ranges used for both source and destination ports can be found in Table 5.7.

Looking at the destination ports, 65 283 ports were targeted during the observation period. As with TCP, the bulk of the traffic was directed to a relatively small number of ports. The top two ports accounted for over 70% of the traffic, with the top twenty accounting for 85% of the traffic. With the exception of 38293/udp, 19932/udp and 5158/udp the others are related to services commonly present on hosts running Microsoft Windows platforms. Port 135/udp is used by the DCE/RPC Locator service, 135/udp as part of netbios file sharing and networking, and 1026-1029/udp are commonly used by Windows DCOM services. Ports 1434/udp (Microsoft SQL Monitor) commonly exploited by the SQL Slammer worm and 38293/udp as used by the Symantic AntiVirus client are discussed in more detail below.

Port 1434/udp is used by the Microsoft SQL server Administration service[7]. It gained notice after the publication of MS02-039 (Microsoft, 2002) on July 24 2002, and ultimately the rise of the SQL Slammer/Sapphire worm on 25 January of 2003 (CERT/CC, 2003), some six months after patches were available. This worm targets systems running unpatched versions of SQL Server 2000 and Microsoft Desktop Engine (MSDE) 2000 software and results in a memory resident infection.

[7] `http://msdn.microsoft.com/en-us/library/ms175483.aspx`



(a) 1434/udp observed counts by day

(b) 38293/udp observed counts by day

(c) 23197/udp observed counts by day

Figure 5.5: Selected UDP destination ports

Table 5.7: UDP traffic by port range

Source Port		Port	Destination Port	
Count	%	Range	Count	%
22 989	0.446	0-1023	679 157	13.193
4 631 893	89.976	1024-49151	4 295 926	83.449
493 085	9.578	49152–65535	172 884	3.358

N= 5 147 967

Well-known ports: 0–1023

Registered ports: 1024–49151

Dynamic, private or ephemeral ports: 49152–65535

This continues to be present on the Internet today, seven years after its release. The packets observed for this accounted for 58% of the total UDP packet count and just over 7% of the overall dataset. Packets were all, with the exception of sixteen outliers, of a uniform size of 418 bytes, containing a 376 byte payload identical to that shown in Figure 5.6. The observed activity of this worm is shown in Figure 5.5a. It is interesting to note the gradual downward trend in the number of unique sources per day observed over time from a peak of over 600 in August 2005 to under 30 in late September 2009. Data for 1434/udp was recorded on all 1 429 days on which traffic was recorded. The dip in April-June 2008 was as a result of the multiple power outages mentioned in Section 3.2.

The other two ports chosen for producing illustrative plots in Figure 5.5(b) and (c) are 38293/udp (ranked 5^{th}) and 23197/udp (ranked 26^{th}). Traffic recorded as destined to 38293/udp is fairly uniformly distributed over the observation period. All packets are 60 bytes in size which allows for a 16 byte payload of the form: 0x020a00c04c4456504869434d00000000. Two payload variations have been observed with the printable portions of the string decoding to 'LDVPHiCM' and 'HiCMHiCM'. This has been reported as a means of probing for clients running Symantec AntiVirus[8]. When the crafted packets are processed by the client software, it will return a report back containing operational information about the client installation, including local computer name, NAV (Norton Antivirus) server group, current definitions, scanning engine version, and the last time of contact with its upstream server. So while not destructive in nature, traffic such as this is an example of active reconnaissance of target networks. This kind of traffic is discussed in more detail in Chapter 7 of *Hacking Exposed: Malware & Rootkits Secrets & Solutions* (Davis *et al.*, 2009), and in particular pages 236 and 237.

[8] http://seclists.org/incidents/2003/Apr/159 - Accessed 2010-03-27

```
 0000 04 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 ................
 0010 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 ................
 0020 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 ................
 0030 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 ................
 0040 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 ................
 0050 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 01 ................
 0060 01 dc c9 b0 42 eb 0e 01 01 01 01 01 01 01 70 ae ....B.........p.
 0070 42 01 70 ae 42 90 90 90 90 90 90 90 90 68 dc c9 B.p.B........h..
 0080 b0 42 b8 01 01 01 01 31 c9 b1 18 50 e2 fd 35 01 .B.....1...P..5.
 0090 01 01 05 50 89 e5 51 68 2e 64 6c 6c 68 65 6c 33 ...P..Qh.dllhel3
 00a0 32 68 6b 65 72 6e 51 68 6f 75 6e 74 68 69 63 6b 2hkernQhounthick
 00b0 43 68 47 65 74 54 66 b9 6c 6c 51 68 33 32 2e 64 ChGetTf.llQh32.d
 00c0 68 77 73 32 5f 66 b9 65 74 51 68 73 6f 63 6b 66 hws2_f.etQhsockf
 00d0 b9 74 6f 51 68 73 65 6e 64 be 18 10 ae 42 8d 45 .toQhsend....B.E
 00e0 d4 50 ff 16 50 8d 45 e0 50 8d 45 f0 50 ff 16 50 .P..P.E.P.E.P..P
 00f0 be 10 10 ae 42 8b 1e 8b 03 3d 55 8b ec 51 74 05 ....B....=U..Qt.
 0100 be 1c 10 ae 42 ff 16 ff d0 31 c9 51 51 50 81 f1 ....B....1.QQP..
 0110 03 01 04 9b 81 f1 01 01 01 01 51 8d 45 cc 50 8b ..........Q.E.P.
 0120 45 c0 50 ff 16 6a 11 6a 02 6a 02 ff d0 50 8d 45 E.P..j.j.j...P.E
 0130 c4 50 8b 45 c0 50 ff 16 89 c6 09 db 81 f3 3c 61 .P.E.P........<a
 0140 d9 ff 8b 45 b4 8d 0c 40 8d 14 88 c1 e2 04 01 c2 ...E...@........
 0150 c1 e2 08 29 c2 8d 04 90 01 d8 89 45 b4 6a 10 8d ...).......E.j..
 0160 45 b0 50 31 c9 51 66 81 f1 78 01 51 8d 45 03 50 E.P1.Qf..x.Q.E.P
0170 8b 45 ac 50 ff d6 eb ca .E.P....
```

Figure 5.6: Hex dump of SQL Slammer Worm packet payload.

```
0000 17 8f 87 23 47 1b f7 e8 57 cd 8b 0a a2 11 73 00 ...#G...W.....s.
0010 31 01 00 32 00 00 00 4c 49 4d 45 17 00 01 00 07 1..2...LIME.....
0020 75 72 6e 3a 73 68 61 31 3a 53 56 45 43 55 4c 54 urn:sha1:SVECULT
0030 46 41 51 4e 32 55 59 49 54 4e 36 37 4d 49 34 32 FAQN2UYITN67MI42
0040 37 48 32 50 58 58 44 4a 48                      7H2PXXDJH
```

Figure 5.7: Raw Payload

```
urn:sha1:SVECULTFAQN2UYITN67MI427H2PXXDJH
```

Figure 5.8: Sample Payload from packets destined to 23197/udp

Potentially based on the responses received, an attacker could plan further moves around avoiding the installed signature base on target clients. Very little definitive information surrounds this port and packet payloads, other than discussions on mailing lists[9].

Traffic observed as destined to 23197/udp represents another class of observation. This is traffic that is 'active' in nature in that it expects a response, but allows one to infer other activities. By examining the payload one can deduce that these are packets used by the LimeWire peer-to-peer file-sharing system which uses an implementation of the Gnutella file-sharing protocol[10]. A sample of the contents of the packet payload is shown in Figure 5.7 with the extracted textual component shown in Figure 5.8. These are requests for details of shared files where the files are identified by a sha1 hash. TCP connects to this port were also observed and it can be inferred that these also represent activity by p2p client software. This is however a non-standard port for the application. It is worth noting that during a three month period, from January to March 2007, packets were received from 983 hosts all destined for a single host on the network (196.x.x.57). This probably indicates misconfiguration of hosts running this client software, or alternately the inclusion of the target host as a file source within the p2p software. The payload contains a Uniform Resource Name URN[11]. In his case the sha1 hash value is encoded using base32[12]rather than the traditional base16[13] (hex) encoding as described in RFC 4648 (Josefsson, 2006). Online Searches for the base16 and base32 values failed to produce any results.

[9] http://library.pantek.com/MailingLists/snort.org/snort-users/03/07/5239.html - Accessed 2010-01-01

[10] http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf

[11] http://en.wikipedia.org/wiki/Uniform_Resource_Name

[12] http://en.wikipedia.org/wiki/Base32

[13] http://en.wikipedia.org/wiki/Base16

5.2.3 ICMP

The last of the major protocol groupings, ICMP (Postel, 1981a) provides much more functionality than the 'ping' utility (see RFC1739 (Kessler and Shepard, 1994)) which most people believe to be its sole purpose. A summary of the top observed types is shown in Table 5.8. ICMP Datagrams contain two components of primary interest: the Type of the packet, and dependant on this there may be a Code which provides further detail. A full listing of valid ICMP Types and Codes can be found in Appendix C.3. While echo traffic (Type 8 and Type 0) constitute the bulk of traffic (71.4%), much of the interesting analysis occurs when analysing the packets of Type 3 (Unreachable) and 11 (TTL Exceeded). Types 3, 8 and 11 combined constitute 99.61% of the ICMP traffic. Echo or ping requests are the most frequently observed packets and are commonly used as both as a means of reconnaissance and to detect the existence of a target host. In many cases these are the prelude to more detailed and targeted scans.

Many tools use a 'ping test' to determine if a host is reachable prior to commencing with the more intensive scans on TCP and UDP ports. Some tools such as NMAP[14] provide options to override this default behaviour and perform scans even in the event of hosts not responding (in this case using the -P0 command line option). The Type 0 datagrams received are echo responses (generated by the claimed source of the packets), which in turn are generated in response to Type 8 'echo request' packets with forged source addresses claiming to be from within the monitored netblock.

As previously observed in the protocols Section 5.1.1, there are a number of packets with nonsensical type values such as 46 and 255 (reserved). This again illustrates some of the mutations that one can see on the Internet, but are often missed on busy production networks.

A discussion of the example analysis of TTL exceeded messages is presented in Section 5.3.4 as part of the discussion around the flood of messages received from 219.158.4.*x*. As such this section will focus on Type 3 (Destination Unreachable) messages observed, a breakdown of which is shown in Table 5.9. In general, ICMP messages of this type are generated by routers or filtering systems in response to either missing routes, failed ARP requests or packet filters. Commonly observed

[14] `http://nmap.org/`

Table 5.8: Top ICMP Types

Rank	Type	P_{count}	%
1	8	1 471 537	62.10
2	3	585 904	24.72
3	11	302 891	12.78
4	0	4 586	0.19
5	5	2 176	0.09
6	255	1 881	0.08
7	17	254	0.01
8	12	165	0.007
9	14	46	0.002
10	4	34	0.001
		Total	99.99

N= 2 369 563

Table 5.9: ICMP Type 3 Datagrams

Type	Code	P_{count}	%
3	0	1 811	0.30
3	1	55 729	9.51
3	2	161	0.02
3	3	456 320	77.88
3	4	64	0.01
3	5	2	<0.01
3	6	18	<0.01
3	9	102	<0.01
3	10	4 827	0.82
3	13	66 869	11.41
3	46	1	<0.01

N=585 904

Note: Percentages are expressed as a portion of all ICMP Type 3 messages received.

Code values are defined in RFC 792, 1122 and 1812 (Postel, 1981a; Braden, 1989; Baker, 1995).

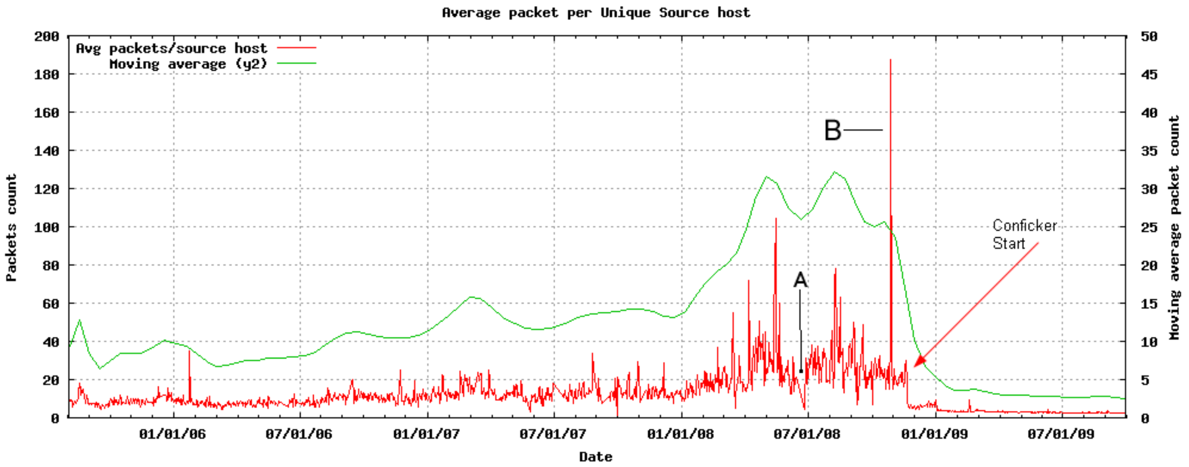
A further discussion of the details around the usefulness of these messages is contained in Section 5.5. All the Type 3 ICMP datagrams are generated in response to an event of some kind occurring as a result of receiving and processing an IP packet with a source address in the monitored range, thus providing a potential means of observing original packet payloads through the option to include a portion of the datagram in triggering the exception the ICMP response. Currently exploration of the payloads has been performed via manual browsing and manipulation using WireShark. In the future, a tool could be crafted with relative ease to carve the payloads out and convert to standardised 'libpcap' format frames, which could be further analysed by a variety of existing tools.

5.2.4 Summary

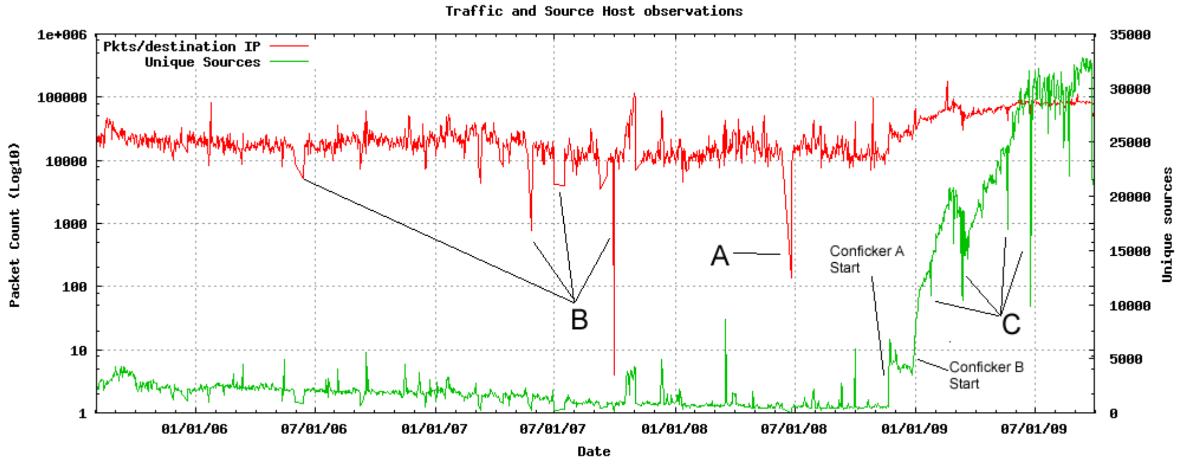
From the discussion above it can be seen that a relatively small proportion of the ports in the case of TCP/UDP, and response types for ICMP constitute a significant portion of the overall traffic for each protocol. The remainder of the traffic tends to be spread over a wide range of ports for TCP and UDP. Not all of the traffic observed can be attributed to 'well-known' protocols, particularly given the lack of payload information commonly experienced with TCP datagrams. The appearance of anomalous Type codes in the ICMP traffic points to potential data corruption or malfunction in data transmission by software. This is in itself an area for future investigation.

5.3 Hosts and Networks

The next analysis performed on the RUSCOPE1 dataset was the exploration of the source addresses of recorded datagrams. It is important to reiterate at this point that there is no way of ascertaining with 100% reliability that the addresses presented in the datagrams are actually those of the sending hosts, since spoofing of IP addresses is fairly trivial. This is a commonly used technique in both denial of service (DoS) attacks and as a decoy mechanism when scanning hosts. As an overview, Figure 5.9 provides two views of the traffic captured on the telescope system during the period of study. Clearly evident is the rapid growth in observed traffic from late November 2008 though until the end of the observation period in September 2009. The mostly likely cause of this explosion in observed hosts is due to the number of infections relating to the outbreak of the Conficker worm and the exploitation of the MS08-067 vulnerability (Microsoft, 2008a).



(a) Average Packet Counts per Unique Hosts by day



(b) Traffic per destination IP, and the count of unique source addresses

Figure 5.9: Traffic overview

An interesting aspect of the traffic shown in Figure 5.9a is that while there was a general increase in the volume of observed TCP scanning activity from early 2008, there was a sharp drop-off in the average number of packets received per source host once the Conficker outbreak occurred. This ties in with the observation that from this point most hosts observed were only sending packets destined to 445/tcp. Of these hosts only 127 045 (3.3%) sent ten or more datagrams. Taking the increase in observed hosts into account and the decreased contribution of these

Table 5.10: Breakdown of packet count present by netblock

		Number with Packet Count					
Netmask	N	> 10	$> 10^2$	$> 10^3$	$> 10^4$	$> 10^5$	$> 10^6$
/32	5 557 688	237 513	39 588	3 394	148	7	0
		4.273%	0.712%	0.061%	0.002%	-	-
/24	1 042 020	332 736	56 428	3 889	242	9	0
		31.931%	5.415%	0.373%	0.023%	-	-
/16	21 634	18 823	14 651	6 656	588	24	1
		87.006%	67.722%	30.766%	2.579%	0.110%	0.004%
/8	215	203	167	150	118	74	3
		94.418%	77.674%	69.767%	54.883%	34.418%	1.395%

Note: Percentages are calculated as for the proportion for each netblock grouping.

hosts to the total packet count, results in the sharp drop-off in the average number of datagrams observed per source host. This point is indicated in Figure 5.9b as the start of Conficker. The point marked A corresponds to the same major outage as indicated in Figure 5.9b (which also accounts for the dip in the moving average) resulting from a switch and cabling failure. Point B indicates a significant anomaly of backscatter traffic received, which is further discussed in Section 5.3.4 below.

The packet count received on average for each of the 256 monitored addresses on a per day basis is presented in Figure 5.9b, along with the number of unique source hosts observed during that day. A number of interesting features are marked in the image. Points marked A and B indicate periods of logging outages - either due to upstream connectivity failure (as discussed in Section 3.2), or equipment problems. Those marked C represent the same as A and B, but are more visible on the source count graph as the log scale increases for the packet count, tending to compress and flatten out the troughs. Also indicated are the beginning of the Conficker A and B worm variants, a more detailed discussion of which follows in Chapter 7.

Looking at the data overall, 5 557 688 individual IP addresses were observed. Of these only 237 513 (4.27%) sent more than ten packet into the monitored network address space, and 0.7% send more than 100. From this one can deduce that while there is a relatively large number of addresses observed, the majority of hosts send very little traffic. Aggregating the IP addresses in to network blocks falling on the traditional 8-bit boundaries (/8, /16 and 24) shows a similar decline as the packet count increases. A summary of the proportions of each netblock exceeding increasing packet count values is given in Table 5.10.

The analysis in this section is based on all traffic observed, and makes no distinction between active and passive traffic types. The remainder of this section consists of four parts, which present the analysis at increasing levels of granularity, starting with aggregation of data by /8 network and proceeding though to /32 which, in effect, is individual hosts. The rationale behind this is to provide an initial high level overview of the data which is likely to prove relatively stable over time, and becomes more volatile as the level of detail increases. Differing levels of granularity may also also suit different intended purposes for the data, such as those discussed in Chapter 9. Detailed breakdowns of the top twenty entries in each category, along with the corresponding graphs, are contained in the Electronic Appendix detailed in Appendix F.

5.3.1 CIDR /8 aggregation

A summary of the top ten networks when aggregated by /8 can be seen in Table 5.11. These comprise just over 40% of the total datagrams recorded. This percentage increases to 55% when the top twenty network blocks of /8 size are considered. In total 215 netblocks of size /8 were recorded, exceeding the number allocated for use at the time of writing. This is due to traffic being received from so-called Bogon address space, which is discussed in Section 6.5. A table showing the changes in allocation of the /8 networks to Regional Internet registries is contained in Appendix C. A Hilbert curve plot[15] of the observed address blocks is presented in Figure 5.10, which shows the overall coverage of IPv4 address space. Although no heat mapping has been done on this image, it shows that there is fairly wide coverage of IPv4 address space observed, with the large black areas in the top right being the 224.0.0.0/4 or old Class D (Multicast) and Class E (Reserved) ranges. Not all of the addresses observed are valid however, with a number being reserved or unallocated. Further discussion around these 'special class' addresses largely defined in RFC 3330 (IANA, 2002) is contained in Section 6.5.

The most notable of the networks observed is 196.0.0.0/8, which comprises over three times more traffic than the next highest rank, and almost as much as the next five places combined. This is the netblock in which the network sensor is located. The effect of this possible bias is discussed in Section 6.1. What is also interesting is that a fairly high number of distinct hosts and networks occur in this block in comparison to others, as noted in the following sections.

[15] Details on Hilbert curve interpretation can be seen in Section 4.5 and Appendix B

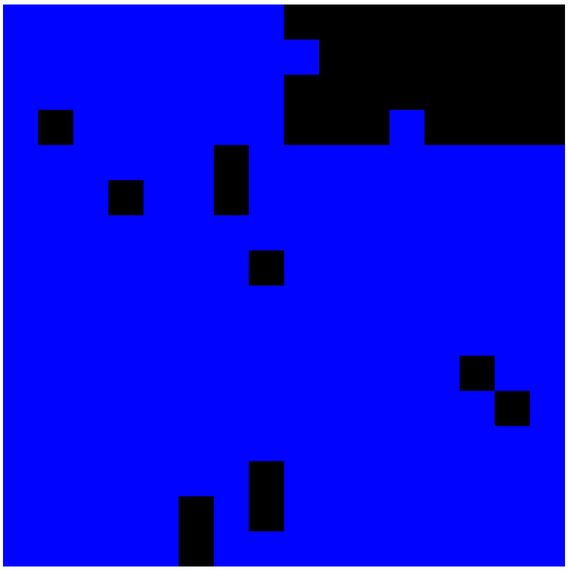


Figure 5.10: Hilbert Plot by /8

Table 5.11: Top Source networks by /8

Rank	Network(/8)	P_{count}	%
1	196.0.0.0	7428059	18.20
2	61.0.0.0	1734767	4.25
3	218.0.0.0	1491819	3.65
4	222.0.0.0	919612	2.25
5	60.0.0.0	897481	2.20
6	195.0.0.0	869136	2.13
7	202.0.0.0	828422	2.03
8	219.0.0.0	798974	1.96
9	190.0.0.0	716236	1.75
10	189.0.0.0	707022	1.73
		Total	40.17

N=40801854

A closer look at three of the top ten networks is presented in Figures 5.11a-c, respectively showing 196.0.0.0/8, 61.0.0.0/8 and 189.0.0.0/8. These networks were chosen as representative samples of the whole[16]. The 196.0.0.0/8 netblock displays some markedly different behaviour from the other two selected in that the packets per day (ppd) (displayed as a fitted curve to the raw daily samples) remains value bound in a band between $10^3$and 10^4, while the unique source count on a daily basis remains similarly banded between 10 and 350. In comparison one can see a marked spike in the number of distinct source hosts in both 61.0.0.0/8 and even more so with 189.0.0.0/8 after November 2008, and as such can most likely be attributed to Conficker related activity. 189/8 shows another interesting artifact in that it was allocated for use and further reallocation by LACNIC in June 2005, yet traffic was not observed until early July 2006. Other networks in the top twenty (89/8, 92/8, 95/8 and 190/8) are seen to have a similar 'cresting wave' pattern. The ramp up in traffic observed from mid November 2008 is also appears in the majority of the top twenty /8 networks.

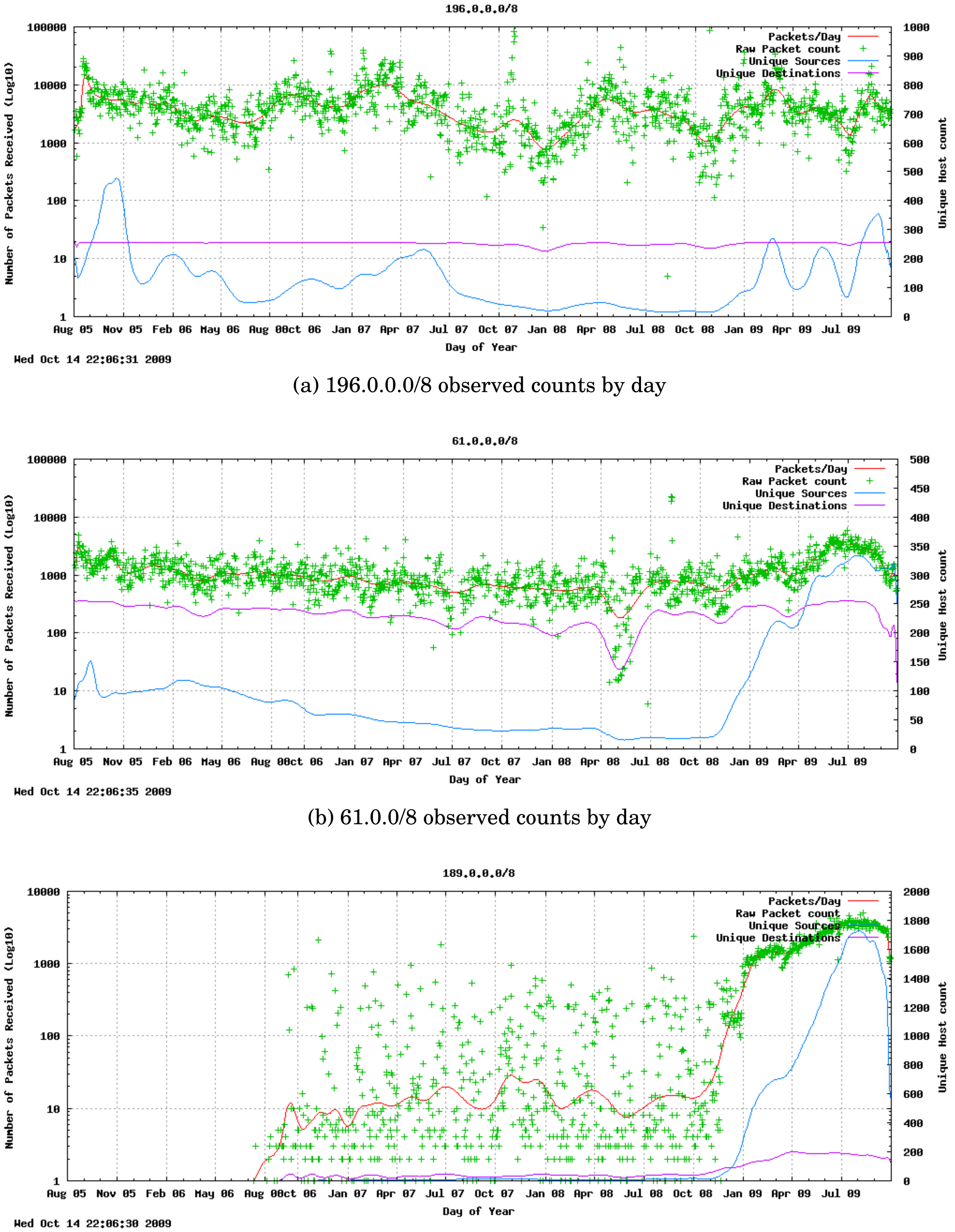
The following subsection takes a finer grained view of the traffic.

5.3.2 CIDR /16 aggregation

When aggregating traffic for analysis by /16 network, a much higher resolution overview of traffic is obtained, as shown in Figure 5.12. This shows the region of IP address space denoted by 128.0.0.0/2 which holds the highest number of 'hotspots', to be fairly sparsely populated. Even more sparsely populated is 0.0.0.0/2, with notable exception of the 24/8, 18/8 and 10/8 network blocks. Heat-map colouring has been applied in this plot with warmer colours indicating more traffic having been recorded. Each pixel in the image equates to a /16 network (65535 hosts) which is equivalent to the old pre-CIDR 'Class B' network blocks.

A summary of the top ten networks is shown in Table 5.12. Unsurprisingly, given the rank of 196/8 in Table 5.11, nine of these are within 196/8. Just over 12% of the total traffic is represented by these networks, with the top twenty covering 15.75% of the total data. In total, 21 634 netblocks of size /16 were identified, and it is significant that the top ten of these, representing only 0.046% of the population constitute such a large portion of the traffic. The 196.21.0.0/16 block was attributed

[16] Graphs for the full top 20 are available in the Electronic Appendix, see Appendix F for details



(a) 196.0.0.0/8 observed counts by day

(b) 61.0.0.0/8 observed counts by day

(c) 192.168.0.0/16 observed counts by day

Figure 5.11: Selected /8 netblocks

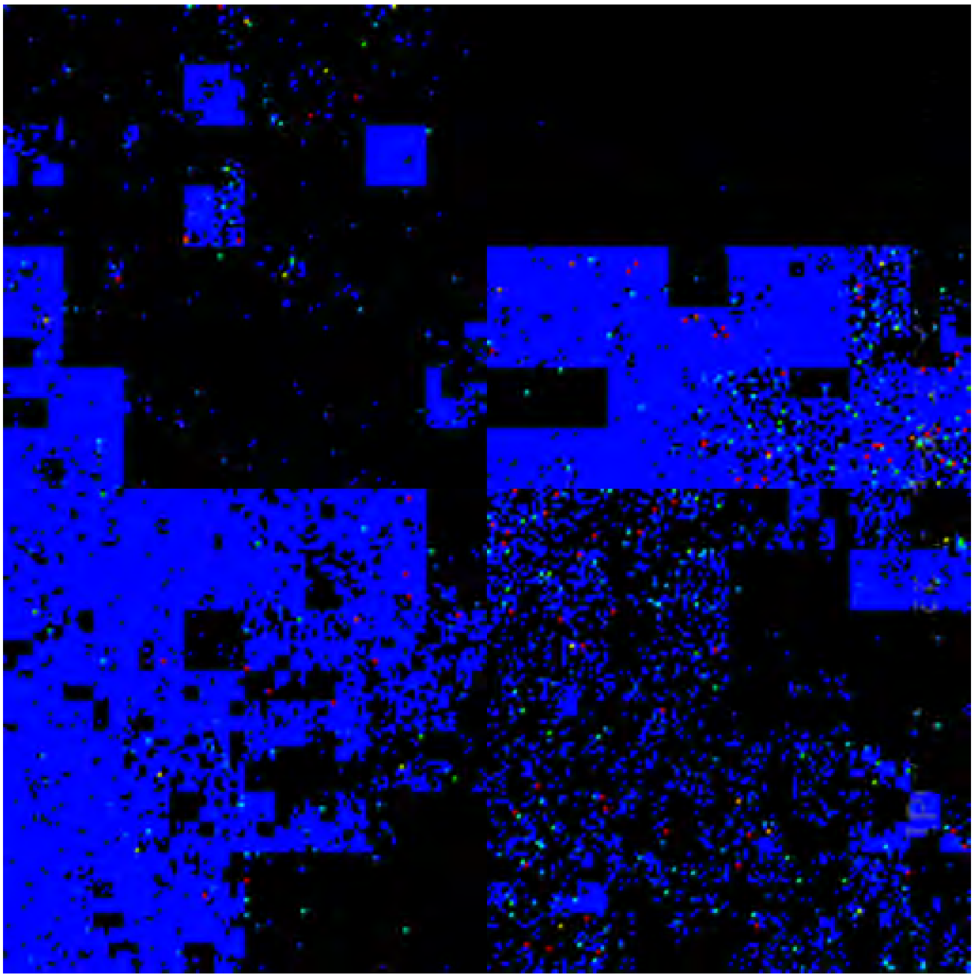


Figure 5.12: Hilbert Plot by /16

Table 5.12: Top Source networks by /16

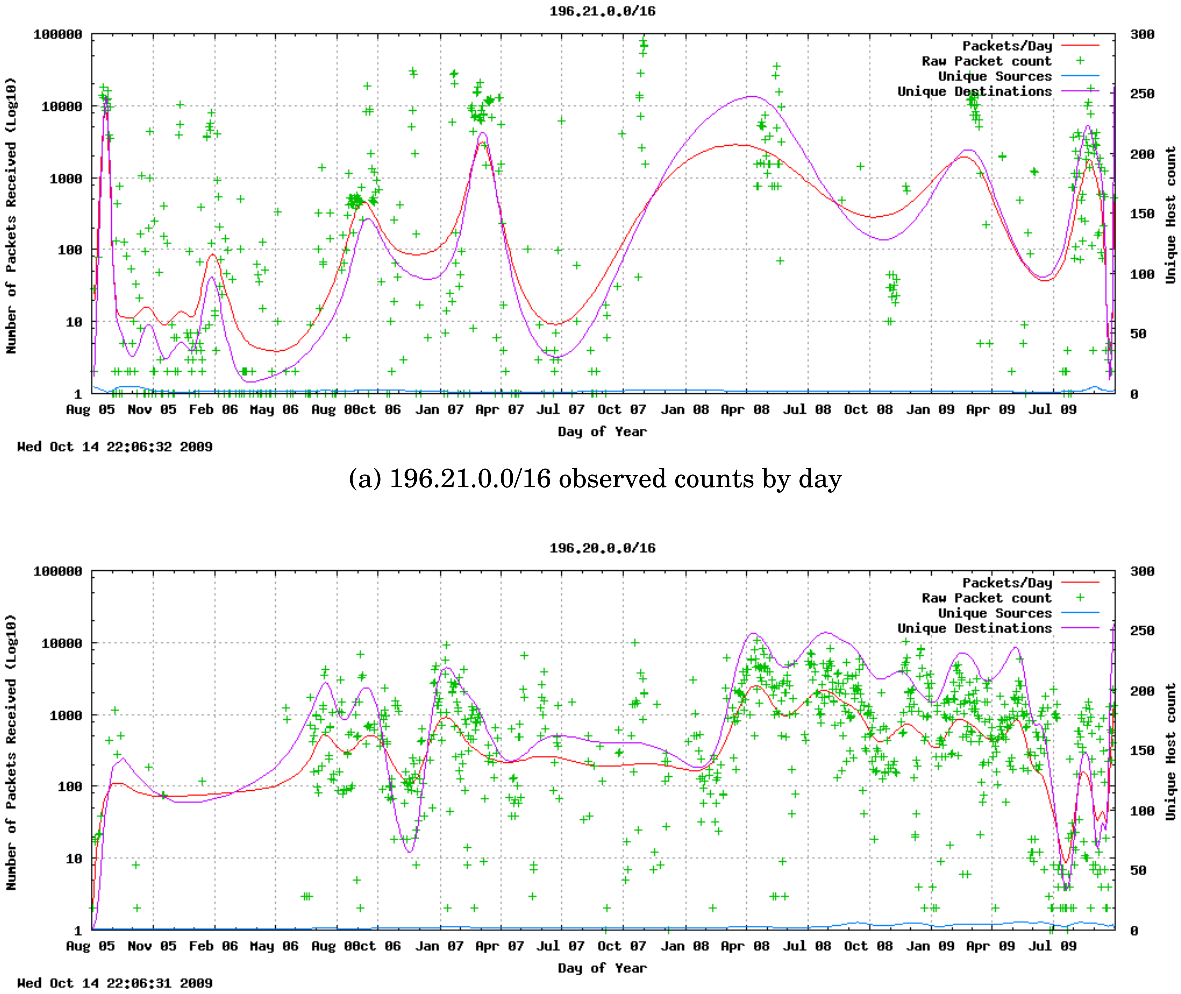
Rank	Network(/16)	P_{count}	%
1	196.21.0.0	1542882	3.78
2	196.20.0.0	995008	2.43
3	196.23.0.0	680489	1.66
4	196.14.0.0	346524	0.85
5	196.205.0.0	285794	0.70
6	196.12.0.0	260789	0.64
7	196.15.0.0	225141	0.55
8	204.16.0.0	202345	0.49
9	196.3.0.0	198510	0.48
10	196.40.0.0	183094	0.45
		Total	12.05

N=40 801 854

to be the source of just over 1.5 million packets over the observed period, but its portion of the whole has been diluted from 5.43% (as measured in December 2008) to only 3.78%. During 2009 204.16.0.0/16 moved from 6^{th} place to 8^{th}, also suffering a dilution from 0.91% to 0.49%. The top five positions remained consistent between 2008 and 2009, as did members of the top ten ranking group. This shows that there is some degree of stability despite the marked increase in traffic collected during Q1-Q3 2009, from a much wider range of sources (as shown in Section 8.3) in turn resulting in a dilution of the contribution to the overall packet count of a number of networks between December 2008 and September 2009.

Two selected plots for 196.21.0.0/16 and 196.20.0.0/16 are presented as Figure 5.13. It is interesting that neither of these show a marked increase in traffic post November 2008, and in the case of the latter there is actually a notable decrease in the mean traffic levels observed. While 196.21.0.0/16 is the top ranked /16 netblock, it is worth noting that actual traffic observed consists of a number of distinct spikes followed by periods of inactivity. Significantly, the number of unique hosts sources recorded remains very low throughout (varying between one and maximum of 16) with an average of 2.28 hosts observed on a given day within the traffic recorded. The fitted curves for packets/day and unique destinations have a close relationship, indicating that there were similar number of packets targeted at each IP address within the telescope sensor during the observation period.

A similar case is also seen in 196.20.0.0/16 which is ranked second. This indicates that while the packet counts are high, and hence the high ranking, there have been



(a) 196.21.0.0/16 observed counts by day

(b) 196.20.0.0/16 observed counts by day

Figure 5.13: Selected /16 netblocks

very few actual hosts involved. The bursty traffic in 196.21.0.0/16 (Figure 5.13a) is most likely due to single malware infected hosts performing scanning over a sustained period. The patchiness of the traffic observed in this case is likely due to either the system going offline (such as a laptop), or being identified as infected and being fairly rapidly detected and re-mediated. The majority of traffic from this network is destined for 445/tcp and 139/tcp, accounting for 74% of all observed TCP traffic for the netblock in question. TCP traffic was only observed originating from this network destined for ports (by volume) 445,139,135,1433,2967,2968,22,80 and 5900. These ports are all contained within the top 10 TCP destination ports by volume as discussed in Section 5.2.1.

Such activity is indicative of highly targeted scanning on common ports where potentially vulnerable services are likely to exist. This in turn highlights the potential danger of only considering a numerical rank analysis without performing a graphical analysis of the temporal behaviour as shown in Figure 5.14. Traffic from 196.21.0.0/16 is only observed on 333 days, representing a coverage of only 21% of the observed period, with an average of 4 583 ppd, although there was a moderately high standard deviation of 9 230 ppd. These values again point to highly focused, yet bursty scanning activity.

5.3.3 CIDR /24 aggregation

Analysis of the data by groupings using /24 sized bins (each holding 256 potential source hosts) provides a much clearer picture of where major traffic sources reside. At this level, the heat map colouring applied to a Hilbert curve plot in Figure 5.14, shows up as particularly warm in the upper portion of 195/8 and 196/8 areas (as indicated by the arrow). Half of the top ten netblocks in this aggregation class again come from the top /16 netblock: 196.21.0.0/16. In fifth position, 219.158.4.0/24 is discussed in detail in Section 5.3.4. The network blocks shown in Table 5.13 represent 4.81% of the total. The top 20 /24 netblocks represent 6.52%, again showing that relatively small blocks of addresses still constitute significant portions of the traffic total.

Four selected plots from the top ten are presented in Figure 5.15. The netblock 196.21.218.0/24 as presented in Figure 5.15a is an example of a situation where there is sporadic activity over the entire period, yet relatively high packet counts when there is traffic observed. This is in contrast to Figure 5.15c, which shows

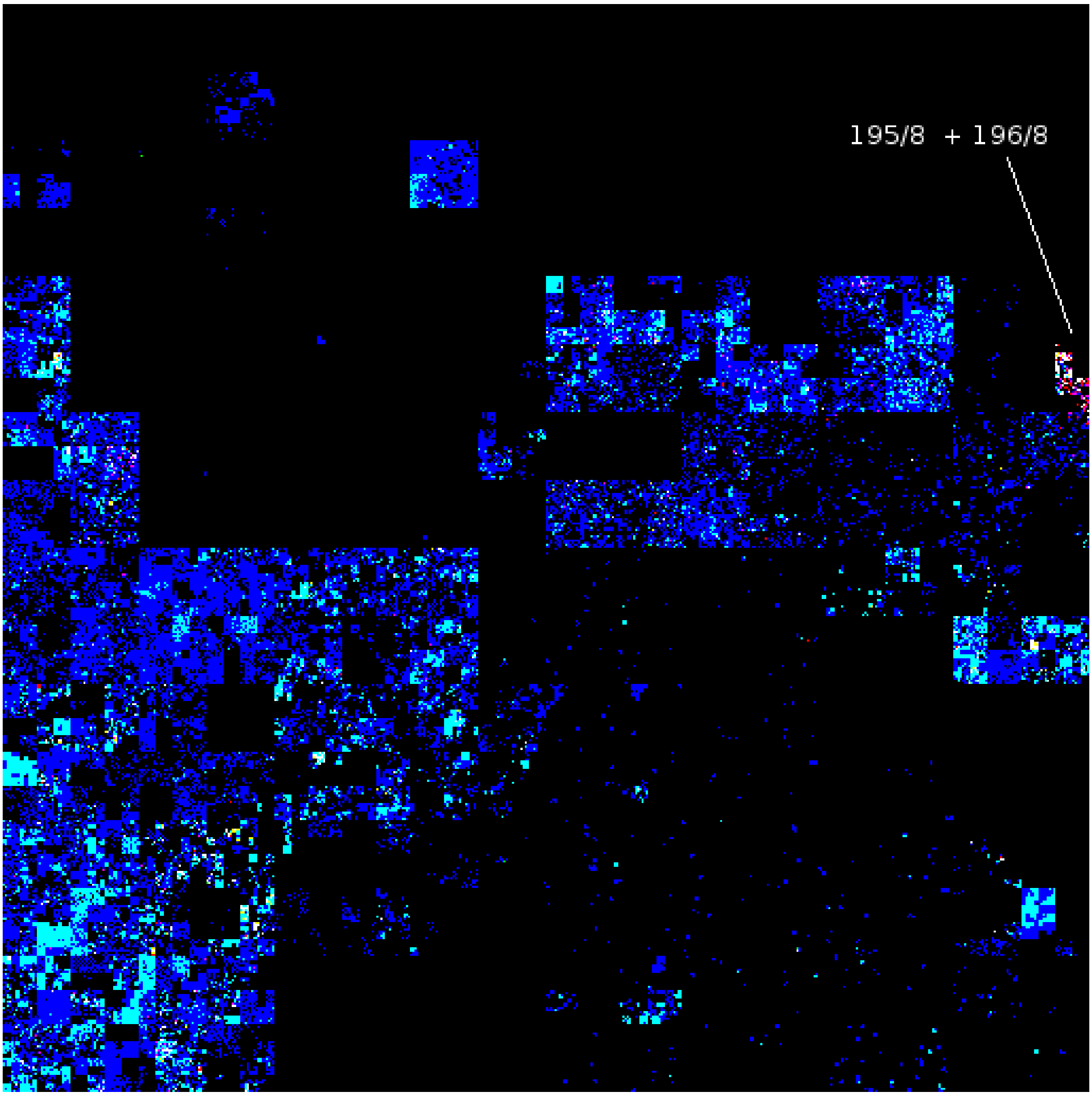


Figure 5.14: Hilbert Plot by /24

Table 5.13: Top Source networks by /24

Rank	Network(/24)	P_{count}	%
1	196.21.218.0	489032	1.20
2	196.21.220.0	340140	0.83
3	196.21.140.0	212629	0.52
4	196.21.139.0	164768	0.40
5	219.158.4.0	159700	0.39
6	196.21.142.0	151762	0.37
7	196.14.172.0	137319	0.34
8	196.14.141.0	110509	0.27
9	204.16.208.0	105868	0.26
10	60.173.10.0	96476	0.23
		Total	4.81

N=40 801 854

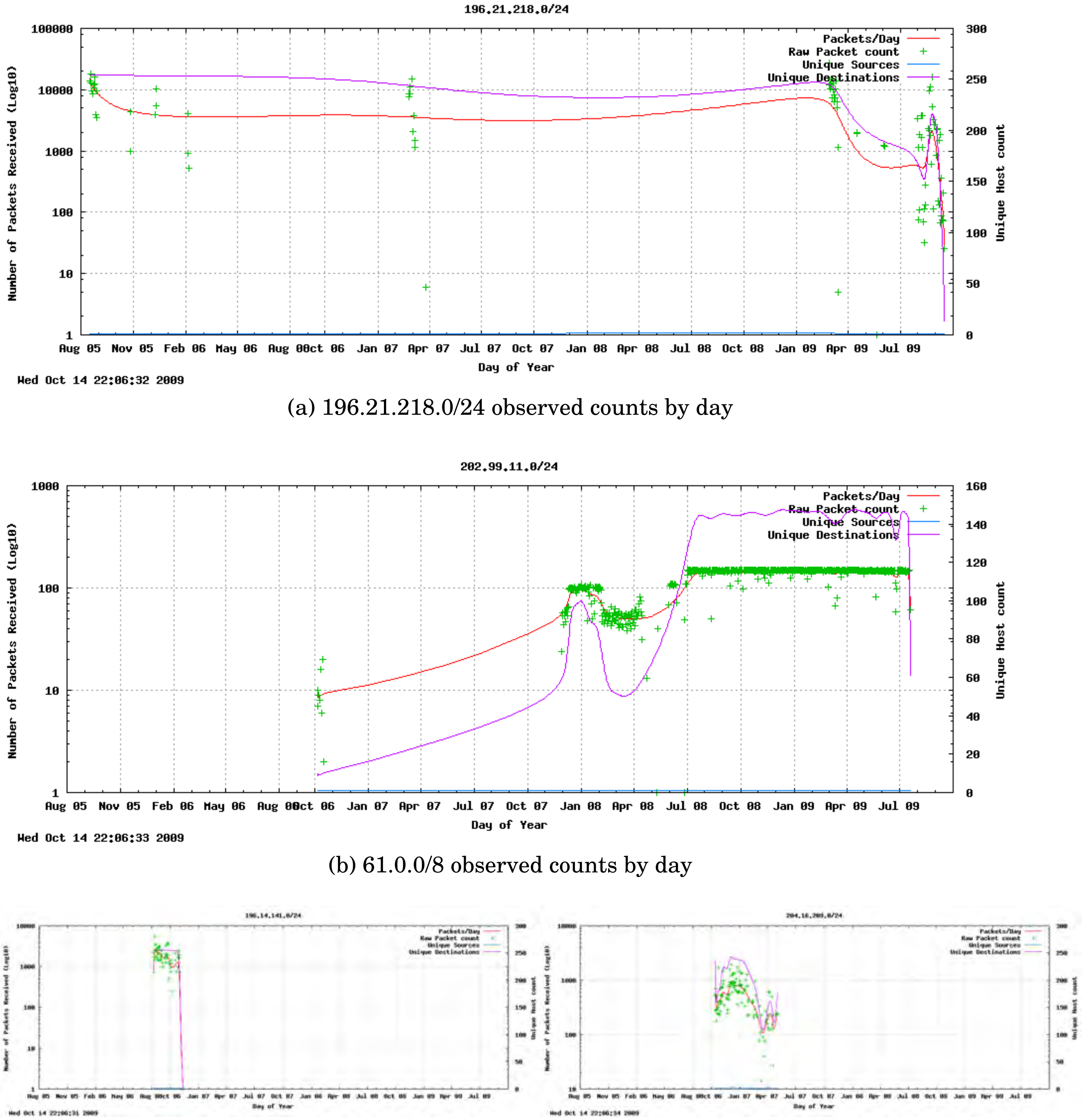
bursts of activity from the 8^{th} and 9^{th} ranked networks. In this case, high packet counts over very short activity periods were observed. In contrast, Figure 5.15c shows that a single host on the 204.16.208.0/24 network has ramped up its activity over a fairly long period of time and consistently sends a single packet to each of on average 148 target hosts until mid July 2009, at which point the traffic ceases. The only traffic was observed from this netblock was UDP packets to ports used by Windows Messaging (winpopup -1025/udp and 1026/udp). Various messages encouraging a user to follow a link to probable malware were received. Example payloads are shown in Listing 6. The URLs mentioned are not listed by Mcaffee Site Adviser or on current Google Safebrowsing blacklists[17]. It is worth noting that only xpregfix.com and regfixerpro.com are currently registered[18], both making use of so called anonymous registration, which is usually indicative of illicit activity. As such one can conclude that these packets were effectively unsolicited marketing using the Windows Messenger service, which would result in pop-ups appearing on vulnerable systems - in effect SPAM over UDP, rather than SMTP.

5.3.4 CIDR /32 aggregation

The final level of granularity at which the dataset is examined is that of the host level. Of the approximately 4.3 billion (2^{32}) possible hosts, 5 557 688 have been observed over the last 50 months, accounting for only 0.129% of potential IPv4 hosts

[17] Last checked 2008-12-31

[18] Whois registrations checked 2008-12-31



(a) 196.21.218.0/24 observed counts by day

(b) 61.0.0/8 observed counts by day

(c) observed counts by day

Figure 5.15: Selected /24 Netblocks

Listing 6 Sample payloads from UDP packets

```
SECURITYALERTSTOP!
SCAN CRITICAL SYSTEM ERRORS
To fix the errors please do the following:
1. Download Registry Repair from: http://www.regfixerpro.com
2. Install Registry Repair
3. Run Registry Repair
4. Reboot your computer
FAILURE TO ACT NOW MAY LEAD TO DATA LOSS AND CORRUPTION!
----
SYSTEMERROR! Warning!! The windows registry may have errors.
 Please visit http://www.xpregfix.com. to scan and repair the system registry.
---
WINDOWSERRORWarning!! The windows registry may have errors.
 Please visit http://www.xpregfix.com. to scan and repair the system registry.
---
WINDOWSERRORWarning!
The Windows registry may be corrupt or have CRITICAL errors.
 Please visit  http://www.fix-reg.com to scan and repair the system registry.
```

being observed on the Rhodes University Telescope system. In practical terms this is can be calculated to be a little higher as one could exclude the space at the upper end of IPv4 addressing (comprising some 536 million addresses), the percentage is at 0.148%. The top ten individual hosts by observed traffic volume are presented in Table 5.14. Of these, five of the hosts occur within the 196.21.0.0/16 netblock itself the top ranked network. Within this network, 320 distinct hosts were observed. The hosts detailed below account for 62.58% of the traffic recorded from the /16 block. The two hosts in 196.21.218.0/24 account for 85% of the traffic from the block. Despite there being only five hosts recorded from this network it is the top listed /24 block. This particular network block is evaluated further in Section 7.5.1. With the exception of the host in third position all these addresses were allocated for use by South African organisations. Of these the ones in 196.21.0.0/16 belong to three formerly disadvantaged tertiary institutions that are connected to the Internet via the TENET network, as is the researchers telescope system. The others have been used by commercial Internet Service Providers.

That nine of the top ten hosts come from the same /8 network as the researcher's sensor, may well indicate some kind of bias either topological or address based. This is considered further in Chapter 7, given localised scanning mechanisms of the Conficker worm.

The most interesting of those listed in the top ten in Table 5.14 is that of 219.158.4.x which is ranked third. The traffic observed from this host was observed in a 7h12m

Table 5.14: Top Source hosts

Rank	Host	P_{count}	%
1	196.21.220.x	340 140	0.83
2	196.21.218.x	258 615	0.63
3	219.158.4.x	159 614	0.39
4	196.21.218.x	157 180	0.38
5	196.14.172.x	135 804	0.33
6	196.21.142.x	121 427	0.29
7	196.14.141.x	110 509	0.27
8	196.21.139.x	88 282	0.21
9	196.27.0.x	86 309	0.21
10	196.40.71.x	74 515	0.18
		Total	3.75

N=40 801 854

burst starting Feb 17, 2009 21:30:49 through to Feb 18, 2009 04:50:38 during which time all of the 159 614 packets were received - a mean rate of 369 packets/minute (6.15pps). Further analysis of the packets shows that they are all ICMP packets reporting a TTL exceeded in transit (Type 11, Code 0). As a part of the ICMP payloads, a portion of the original IP datagrams is returned. The host producing the datagrams belongs to ChinaUnicom and resides on their network backbone. That it is producing TTL expired messages indicates it is most likely serving as a network gateway, as the traffic must have been routed though the node in question when the TTL dropped to 0, causing the emission of the TTL expired messages. All of the ICMP payloads contain TCP datagrams claiming to originate from 196.x.x.88 and destined to port 80 on 67.159.55.x.

Given that the source TCP port (26310) and TCP sequence id (76657869) as recorded in the data portion of the ICMP packets do not vary, this suggests that one of two events are likely to have occurred. The first, and most probable, is that there was some automated generation of packets as part of a Denial of Service (DoS) attack against a target host. This is supported by the fact that the source IP range was spoofed, and as such no data could be received. The payload is truncated after the sequence number so no flag information is present, but it is likely that the observed packets are part of a SYN-flood against the target host. The second possibility, although unlikely given the number of packets generated, is that there was some form of malfunction which caused massive packet duplication — yet the fact that source IP addresses of the packets are spoofed significantly detracts from this version.

Approached from the perspective of this being traffic resulting from a Denial of Service attempt, one is able to ascertain that some host on the ChinaUnicom network (AS4837) performed the attack, spoofing the source address to be from a host in the range monitored by the Rhodes Network Telescope. The packets were observed due to the gateway expiring them and sending notification to the supposed source. A number of possibilities exist as to why the packets expired and resulted in the emission of the observed datagrams. The first is that the automated code set a very low TTL — on today's Internet most hosts are reachable within 20 hops or less (Lu and Lin, 2009). A more detailed discussion on observed TTL values is in Section 5.4.2.

A second is that the host operating system sending the packets may have enforced a lower TTL. A third and most probable explanation is that there was some form of routing loop caused by a de-announcement of the target address from global BGP tables, which is a common response to incoming DoS attacks. Further discussion on the analysis of IP TTL values is contained in Section 5.4.2.

The target itself is located near Chicago in the United States belonging to FDC-servers.net, quite far removed from Beijing where the ICMP datagrams originated. The target IP is listed by Site Dossier as hosting 30 different domains[19], which are shown in Listing 8. These are all apparently registered to the same individual resident in China (Listing 7) and reside on a cluster of adjacent servers (given the consecutive IP addresses they are recorded as having). Further analysis of these domains shows that many of them have in the past been hosted across multiple IP addresses. This could either be for redundancy or as part of Fast-flux botnets command and control channel (Cooke *et al.*, 2005; Larkin, 2007). Searching on Google.com for "SICHUAN BW" provides a substantial number of links to other similarly named domains all registered with the same details as in Listing 7. The researcher was unable to find out much more about these domains. One domain that was responding was `http://9ttss.com/`, a screen-shot of which is shown in Figure 5.16. This image is similar to a number of other 'domain for sale' or generic landing pages. This however does not preclude possibility of malware or other potentially malicious resources being hosted on non-public URLs on the site.

From the perspective of a network telescope, this anecdotal case study of a particular host provides an example of the type of information that can be extracted from data collected using a network telescope. In this case the datagrams observed

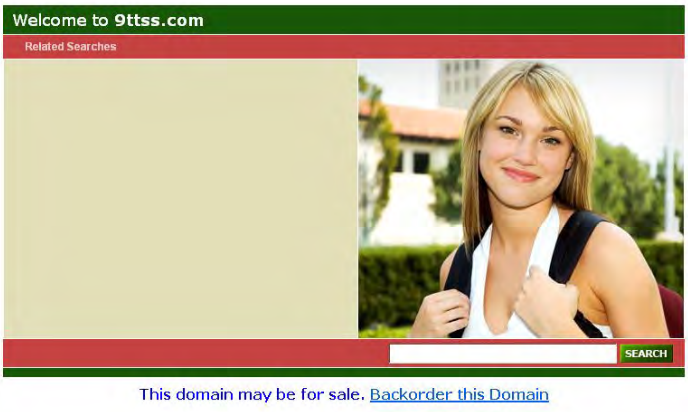
[19] `http://www.sitedossier.com/ip/67.159.55.129` Accessed 2010-01-10

Listing 7 Sample Whois Output

```
Domain name:  4aaxx.com
Administrative Contact:  - wei xiaoming (XXXXX@gmail.com) +1.2866288290
Fax:  267 sichuan in china chengdu, SICHUAN BW
Technical Contact:  - wei xiaoming (XXXXX@gmail.com) +1.2866288290
Fax:  267 sichuan in china chengdu, SICHUAN BW
Registrant Contact:  - wei xiaoming () Fax:  sichuan in china chengdu, SICHUAN BW
Status:  Locked
Name Servers:  dns1.name-services.com
dns2.name-services.com
dns3.name-services.com
dns4.name-services.com
dns5.name-services.com
Creation date:  19 Jan 2009 17:18:32
Expiration date:  19 Jan 2010 17:18:00
Information Updated:  Thu, 7 Jan 2010 01:54:48 UTC
```

Listing 8 Domain names hosted on probable DoS target

```
http://1ggss.com/ http://1ttss.com/ http://2ggss.com/ http://2ttss.com/
http://3ttss.com/ http://4ttss.com/ http://5ttss.com/ http://9ttss.com/
http://se.4aaxx.com/ http://se.6ttss.com/ http://se.8qqxx.com/ http://www.1ggss.com/
http://www.1qqxx.com/ http://www.1ttss.com/ http://www.2aaxx.com/ http://www.2ggss.com/
http://www.2qqxx.com/ http://www.2ttss.com/ http://www.3aaxx.com/ http://www.3ttss.com/
http://www.4aaxx.com/ http://www.4ggss.com/ http://www.4ttss.com/ http://www.5ttss.com/
http://www.6ggss.com/ http://www.7qqxx.com/ http://www.9ggss.com/ http://www.9qqxx.com/
http://www.9ttss.com/ http://www.se.1yyxx.com/
```



This domain may be for sale. Backorder this Domain

Snapshot taken 2010-01-10 23:24

Figure 5.16: Screen-shot of 9ttss.com

were true backscatter, as the result of spoofed address usage. In the case of ICMP, packets of this type constitute 14.65% of all ICMP traffic, and ranked in third place. Further discussions around the constituency of ICMP traffic and Active and Passive traffic can be found in Sections 5.2.3 and 5.5 respectively.

5.4 Size and Lifetime

Two other aspects of IP datagrams that are worth investigating in the context of a network telescope are the size of received packets and the differing Time To Live (TTL) values observed. Given that the network telescope was operating in a completely passive mode, the size of recorded packets is expected to be relatively small since only a small portion of packets — UDP and ICMP along with a few vagrant TCP datagrams were expected to contain payloads for reasons discussed in Chapter 2. An analysis of the observed TTL values in particular can provide insight into the topological distances that hosts are from the sensor network.

5.4.1 Size

The fact that the sensor network has operated in a completely passive manner, and that no 'connection' as such has been established with remote hosts, would lead one to hypothesise that the majority of traffic observed would have relatively low packet sizes. An overview of the average sizes for each of ICMP, TCP and UDP on a daily basis over the period is shown in Figure 5.17, along with the median line for each protocol. The large ICMP spike is due to the anomalous burst of traffic described in Section 5.3.4. What is noticeable is that that both UDP and ICMP show a general decline in the average packet sizes from early August 2008. The marked dip in UDP traffic in April of 2008 is due primarily to there being hardly any activity on 1434/udp, and power and network outages mentioned in Section 3.2. TCP traffic remains constant other than a slight increase experienced in December 2007.

Overall packets captured in the RUSCOPE1 dataset amounted to 3 962.73 Megabytes of data. The overall composition of the dataset from a size perspective can be seen in Table 5.15. Of significance in this table is that despite only accounting for 12.6% of traffic by Packet count, UDP accounts for 43.2% by size.

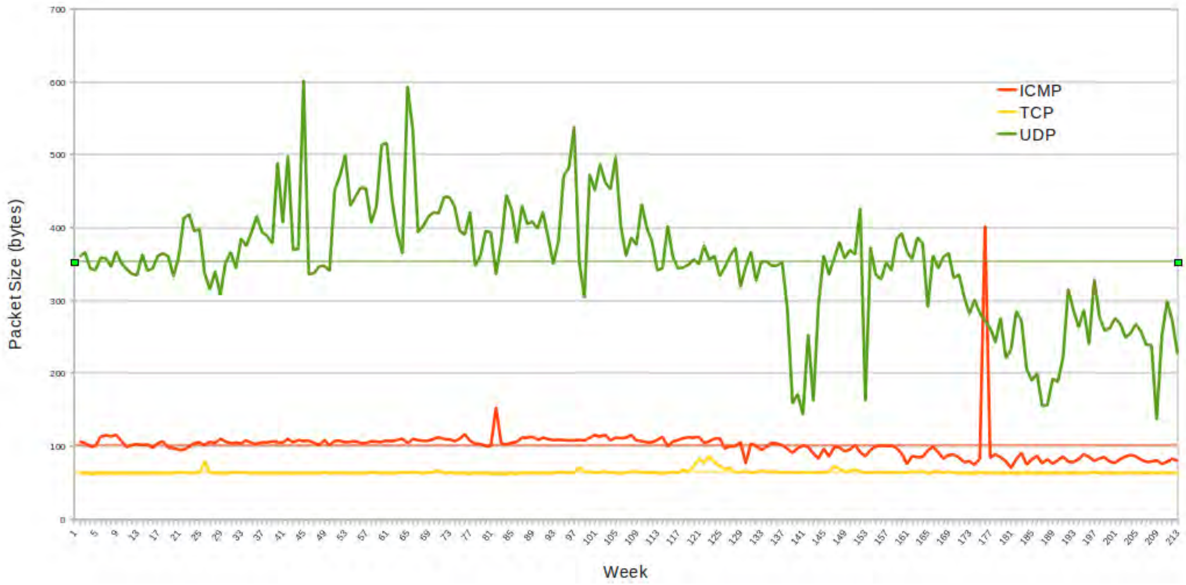


Figure 5.17: Average packet sizes per protocol by day

Table 5.15: Traffic composition by Protocol

Rank	Protocol	Number	$\%_{Pcount}$	Size(MB)	$\%_{Size}$
1	TCP	6	81.57	2 019	50.96
2	UDP	17	12.62	1 711	43.20
3	ICMP	1	5.89	231	5.83
		Total	99.99		99.99

N=40 801 854

Breakdown of the three primary protocols in use on IPv4 networks.
Other protocols constituted 1 081 packets, amounting to ≃164 Kilobytes.

This is mostly due to the prevalence of the SQL Slammer worm, and the fact that the UDP datagrams carried a payload. Each of these protocols is addressed in more detail below.

TCP datagrams exhibited a range of sizes, with 110 different values being recorded. Those accounting for more than 1% of TCP traffic accounted for 99.78% of the total. These top five values were 62,60,74,66 and 78 bytes. These were TCP SYN packets with differing TCP options set, resulting in differing packet sizes. The majority (85.27%) of traffic is of either 60 (12.19%) or 62 (73.07%) bytes. Packets of size 60 bytes is the minimum size required to encapsulate a TCP header (20 bytes) inside an IP datagram (20 bytes) transmitted over an Ethernet network (20 bytes from the header and trailer), without the use of any options. The average size over the 33 million TCP datagrams was 63.6, which re-enforces the notion that the majority

Table 5.16: Selected TCP Flag combinations and sizes

TCP Flags						Size (bytes)		
SYN	ACK	FIN	URG	PSH	RST	Avg	Max	Min
✓						63.48	1454	60
✓	✓					64.57	1514	60
	✓				✓	60.04	1514	60
					✓	60.00	78	60
	✓			✓		195.99	1514	60
	✓					64.74	1514	60
	✓	✓				65.80	78	60
✓	✓		✓			60.03	78	60
		✓				62.62	1514	60
	✓	✓		✓		113.75	491	70
✓		✓				60.12	78	60
						480.12	1514	60

of the packets were connection attempts rather than vagrant fragments, although some larger packets of up to 1 514 bytes were observed. A breakdown of the most significant TCP flag combinations and the associated packet sizes is found in 5.16. The traffic with the ACK+PSH flags set, was all reflected traffic directed largely at 22/tcp in a burst on 24^{th} January 2006. The large 1 514 byte datagrams are discussed in Section 6.5.2. The largest average was attained by the 251 NULL packets (those with no flags set), of which the bulk of the 1 514 byte datagrams were received destined to port 0/tcp from April to September 2009.

UDP, however, requires no connection state to be established and as such packets were considerably larger due to the fact they contained payloads. The largest received datagrams were 1 514 bytes in size, with an average over the entire sample of UDP datagrams being 348 bytes. The 418 byte packets containing SQL Slammer payload (Cai *et al.*, 2007; Goebel *et al.*, 2007; Mattsson, 2007) were recognisable and constituted 58% of the total UDP traffic. This is evident in Figure 5.17 where the traffic can be seen to be above 400 bytes on average on a number of days. UDP traffic saw the widest range of sizes, with 633 different values observed. A breakdown of the sizes of the top ten UDP destination ports (as discussed in 5.2.2) is shown in Table 5.17. The large average sizes for ports 1026, 1027, 1028 and 1029 were largely as a result of the DCE/RPC messages being broadcast on these ports as discussed in Sections 5.3.3 and 6.5.4.

ICMP traffic had an average size of 102 bytes, given that a standard ‘ping packet’

Table 5.17: Top 10 UDP destination ports and sizes

	Size (bytes)		
Port	Avg	Max	Min
1434	418.00	460	60
137	92.00	97	92
1026	592.91	1266	60
1027	617.14	1266	60
38293	60.00	60	60
19932	80.00	80	60
135	197.76	1049	60
1028	582.61	1266	60
1029	553.10	1266	60
5158	79.98	80	60

Table 5.18: ICMP Types and sizes

	Size (bytes)			
Type	Avg	Max	Min	%
8	99.0	1514	60	62.101
3	120.6	1514	60	24.726
11	72.3	759	60	12.782
0	574.0	1514	60	0.193
5	187.3	590	70	0.091

N= 2 369 563

Percentage is expressed as a percentage of the total ICMP traffic

is usually of size 102 (20 byte Ethernet header 20 bytes IP header, 8 bytes ICMP header and a 64 byte payload. Across the ICMP packets, 315 different sizings were observed. A breakdown of the top five ICMP Types along with the average sizes for each, are presented in Table 5.18. The largest average was that of Type 0 (Echo Response) packets, although these represented only 0.19% of the traffic. The top three Types (8, 3 and 11), accounted for 99.6% of the traffic. Of these, the largest average value was found to be had by the Type 3 (Unreachable) datagrams with an average of 120 bytes. This was likely skewed by the large spike of traffic discussed in Section 5.3.4. What was observed is that ICMP had a wide range of datagrams with larger packet sizes, mostly due to the fact its is connectionless, and in the case of Types 3 and 11, payloads were often included containing the packets that generated the notification.

5.4.2 TTL

The Time to Live (TTL) value within the IP header is defined in RFC 791 (Postel, 1981c, p30) where it is defined as:

> The time to live is set by the sender to the maximum time the datagram is allowed to be in the Internet system. If the datagram is in the Internet system longer than the time to live, then the datagram must be destroyed.

Different TCP/IP stack implementations provide different base or default values for the TTL of packets that are emitted by the host. While there is no comprehensive definitive published list of these, a classification was determined though research and experimental testing. Given that the TTL is decremented by one for every hop that a packet transits, this metric can be used to evaluate the possible distance of the sender from the sensor network. Even in the case of spoofed source IP addresses, this value will still generally decrement at the same rate for all packets originating from a given host. In the case where live monitoring is being performed, this can be a useful tool in the discrimination of spoofed *vs.* actual originating addresses, by comparing the traceroute paths to the address to the received TTL values. This method may not be perfect due to the potential for asymmetric routing paths, but can be a discriminator.

A plot of the observed TTL values is shown in Figure 5.18. Two peaks as indicated by A and B are of interest. Peak A is a number of values clustered just below the TTL value of 64. A similar peak is observed at point B, where the values cluster just below 128. This clustering is the result of the common use of 64 and 128 as default TTL values by different operating systems. Generally Microsoft Windows Family operating systems have a default TTL of 128, which results in the major peak as depicted in Figure 5.18. Unix type systems, particularly Linux and the BSD family make use of a default TTL of 64. The final cluster just below 255 is dominated by ICMP traffic, and is probably due to a significant portion of this being Type 3 and Type 11 datagrams generated by Cisco (and possibly other) routers, which default to using a TTL value of 255 on generated packets.

Over 82% of all TCP traffic observed lies in the TTL range of 96-128. Figure 5.19 shows the TTL plots for major operating systems. Operating System identification

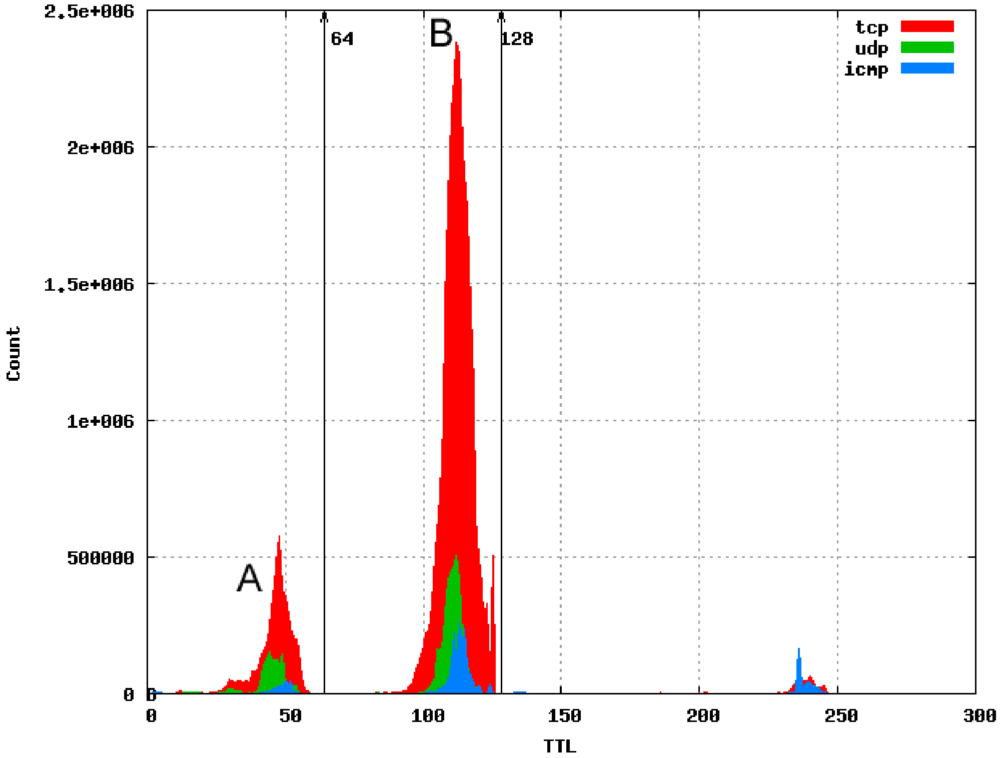


Figure 5.18: Observed TTL values

was performed using the p0f[20] - a passive operating system fingerprinting tool developed by Michal Zalewski. More details on operating system analysis can be found in Section 7.3.1. This image is plotted using a log_{10} scale for the packet count. Microsoft windows can be seen to dominate, with FreeBSD and Linux systems peaking at the sub 64 point, with relatively little traffic, in other areas. The relationship between TTL value and the numerical distance of sources is discussed in Section 6.1.

Table 5.19 shows a breakdown of the top five recorded TTL values for each of ICMP, TCP and UDP. Of these three protocols, ICMP not only had the highest representation in the top five of 45% of the packet count, but a fairly wide spread of values (as seen in Figure 5.18). The bulk (65.78%) of ICMP datagrams had values occurring in the region between 64 and 128 as with the other two protocols: UDP 73.33% and TCP 82.56%.

Datagrams with TTL values greater than 128 accounted for only 2.64% of the total number of packets across all three protocols. By protocol, ICMP had nearly 20% of its packets in this region, reflecting the influence of the router (most probably

[20] `http://lcamtuf.coredump.cx/p0f.shtml`

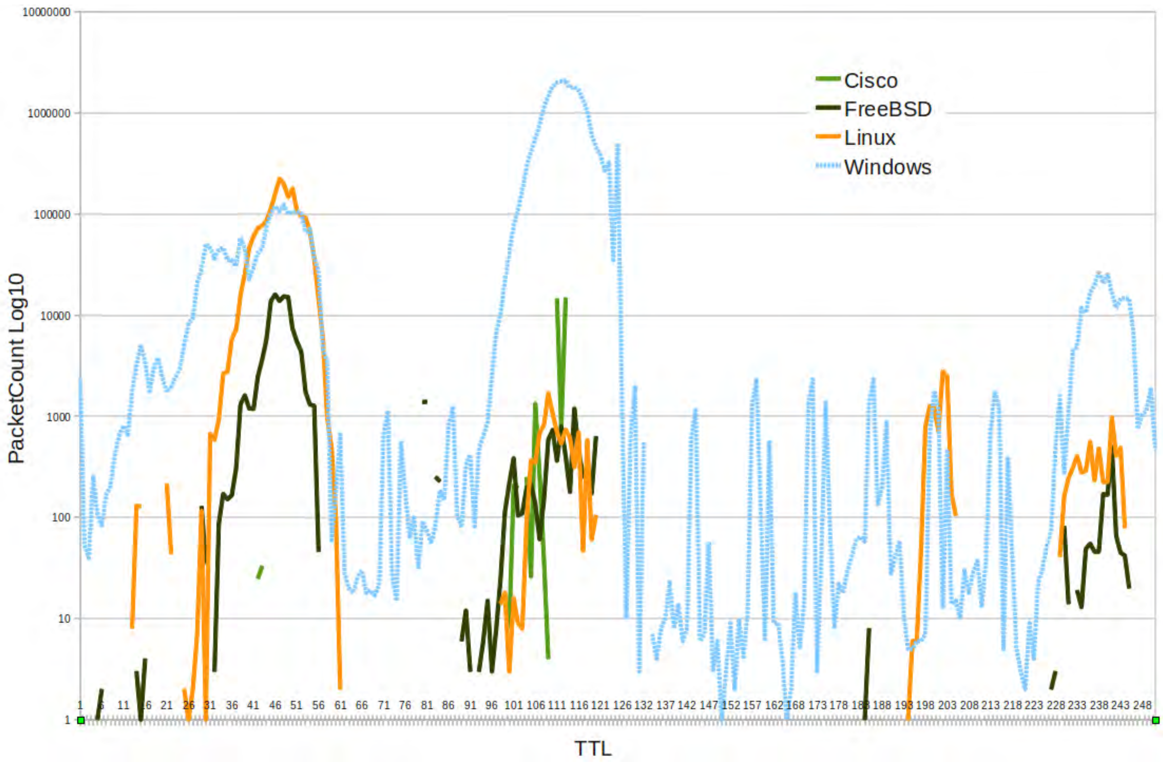


Figure 5.19: Selected Operating systems and TTL values

Cisco) generated Type 11 datagrams previously discussed. TCP had only 1.79% with UDP having less than 0.203%. These observations are consistent with the type of observed traffic which is mostly attributable to systems running one of the Microsoft Windows family of operating systems.

The top values for each protocol are shown in Table 5.19, all of which can be seen to cluster around the centre point of Point B in Figure 5.18, with the exception of the 5th ranked ICMP value of 236 representing 164 618 packets, 159614 (96.96%) of which are related to the incident discussed in Section 5.3.4.

The TTL data can also be used to compute average distances at a topological level, by subtracting the observed TTL values from the common maximal cluster points of 64, 128 and 255. The results of these calculations are shown in Table 5.20. The average path length that can be arrived at using a weighting based on packet count is 16.67, which can be rounded up to 17. This is similar to the average path length of 15.13 arrived at in the work conducted by Lu and Lin (2009) using active measurements. The discrepancy can probably be accounted for by the way the routing of the netblocks on the telescope was set up within the Rhodes University network. What can be determined, is that the majority of traffic observed originates less

Table 5.19: Top TTL values by protocol

	ICMP		**TCP**		**UDP**	
Rank	**Val**	%	**Val**	%	**Val**	%
1	113	11.01	112	7.15	112	9.77
2	115	9.84	113	7.02	111	9.14
3	111	9.49	111	6.79	110	8.83
4	114	7.76	110	6.40	109	8.15
5	236	6.94	114	5.97	113	8.01
$Total_{\%}$		*45.05*		*33.34*		*43.91*
δ_{TTL}		59.04		30.20		30.66
$N_{Distinct}$		203		251		193

Percentages are expresses as a percentage of all packets for that protocol
$N_{ICMP} = 2\,369\,563$ $N_{TCP} = 33\,283\,243$ $N_{UDP} = 5\,147\,967$

Table 5.20: Computed path length from TTL data

TTL	$Avg_{Distance}$	$Count_{Packets}$	$\%_{Packets}$
≤ 64	18.78	6 906 733	16.93
64<128	15.972	32 814 707	80.42
≥128	24.42	1 080 414	2.65

N=40 801 854

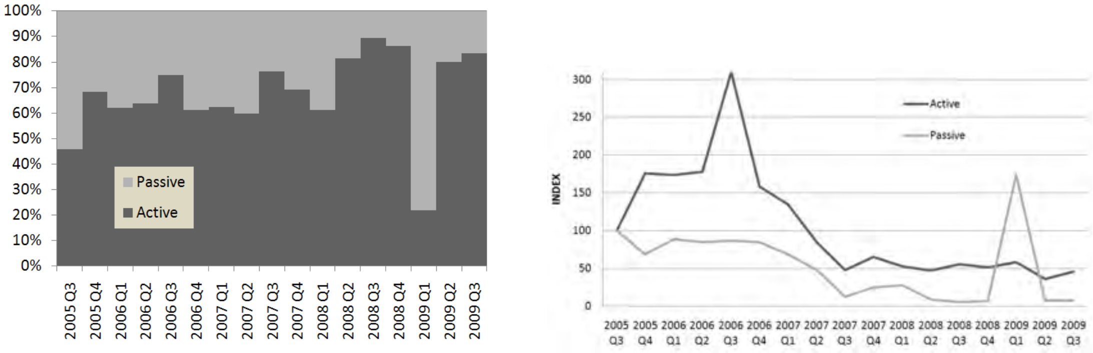
than twenty 'hops' or router nodes away from the Network Telescope. Only 17% of the traffic observed has longer probable path lengths.

5.5 Active *vs.* Passive Traffic Analysis

As discussed in Chapter 2, a network telescope can expect to see a number of different types of traffic which can broadly be split into two categories — that traffic which is active in that it is intended to generate a response, and that which is reflected back to the sensor as the result of 'active' traffic generated elsewhere on the Internet. In this section, a discussion of the active and passive traffic considered for ICMP and TCP is presented.

5.5.1 ICMP

Much of the detail relating to the composition of the ICMP traffic recorded has been dealt with in Section 5.2.3, and particularly Tables 5.9 and 5.8. Passive datagrams



(a) Percentage Composition (b) Index

Figure 5.20: ICMP Traffic: Active *vs.* Passive analysis

Table 5.21: ICMP classifications

Traffic	Types	%
Active	8,13,16,18,30,33,35	62.102
Passive	0,3,4,11,12,14,16,18,31,34	37.712
	Total	99.815

N=2 369 563

4389 packets lay outside of the above ranges, predominantly of Types 5 (Redirect) and 255 (Reserved), with 2176 and 1881 packets respectively

accounted for 37.7% of the ICMP traffic recorded with Types 3 and 11 accounting for 37.5%. A significant portion ($\approx 18\%$) of this originated in the incident discussed in Section 5.3.4. The details of the comparison between active and passive traffic can be seen in Table 5.21, while Figure 5.20 illustrates the change in ICMP composition over the monitored period.

The burst in passive traffic in the first quarter of 2009 can be attributed to the flood of traffic discussed in Section 5.2.3. The overall level of ICMP traffic has dropped significantly in both real and relative terms (as evidenced by Figure 5.20b), with only 50% of the volume of active packets being recorded by September 2009, and around 10% in the case of passive packets, as were observed at the start of the study. More discussion around the use of proportional and index plots is contained in Section 8.3.

5.5.2 TCP

For the purposes of this analysis, active traffic was defined to be those TCP datagrams received that had the SYN flag set comprising 96.99%. Passive traffic was deemed to be those packets which had the RST flag set (2.79%). Only 73 687 datagrams (0.22%) matched neither of these criteria. While this is a fairly crude delineation, it avoids the potential problem of traffic with both the SYN and ACK flags set as described in Section 2.3, by grouping all incidences of these datagrams as active, by virtue of the fact they are capable of generating a response. SYN-ACK traffic make up 8.6% of the TCP total. Further decomposition of the make-up of these datagrams could potentially be performed by detailed analysis of window sizing, options and sequence number — a research area that has not been addressed in this work. Such packets made up 2 877 275 (8.64%) of the TCP datagrams recorded.

An index plot providing a comparison between the active and passive TCP packet groupings over the monitored period is shown in Figure 5.21. The use of this plot style allows for the values to be represented on the same set of axes, despite having largely differing numeric scales. More on this plot format is discussed in Section 8.3. The values observed for the two traffic classes tended to be relatively stable until the beginning of 2008 when they start slowly increasing, with a rapid increase in the third quarter of that year. This rapid increase coincides with the advent of the Conficker worm in November 2008 and is discussed in Chapter 7. The passive traffic, however, started declining from early 2009, in contrast to the active traffic which continued to climb, attaining nearly a 700% increase since the start of the monitoring period. The passive TCP traffic recorded from 2009-01-01 is primarily directed towards 445/tcp (97.4%), possibly due to scanning with spoofed addressing.

A summary of the top ten source ports RST traffic was received from is shown in Table 5.22, accounting for 58.9% of the total. The bulk of these (35%) are from 80/tcp — indicating that they are mostly likely to be the product of spoofed connection attempts to web servers, where there is either no service listening or the service is being firewalled, resulting in the generation of the TCP RST response packets.

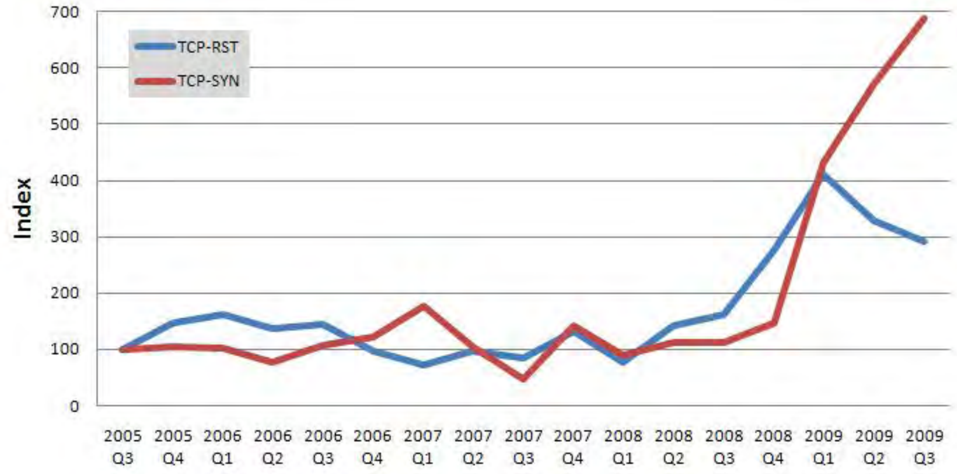


Figure 5.21: TCP Active vs Passive relative indexes

$Index_{100}$ = 998 674 and 32 383 respectively for Active (SYN) and Passive (RST) traffic

Table 5.22: Top 10 source ports for passive traffic

Source Port	Count	%
80	327 194	35.29
25511	107 968	11.64
7000	52 345	5.64
9201	15 602	1.68
29999	10 818	1.16
22	8 066	0.87
3389	7 216	0.78
2999	7 199	0.77
10416	5 198	0.56
119	4 443	0.48
	Total	58.90

N=927 062

5.6 Summary

This Chapter has provided an overview of the RUSCOPE1 dataset, looking initially at the constitution of the observed traffic on a protocol basis. This was further explored for each of TCP, UDP and ICMP. A breakdown of traffic by network source address was evaluated at differing levels of aggregation. The size of datagrams was analysed and correlated with certain type of probative traffic received on the network sensor, the observed differences between the primary protocols was discussed. The Time to Live (TTL) values of traffic were investigated and compared to identified probably originating Operating System. The Chapter concluded with a comparison of active and passive traffic. In the course of discussion, a number of anomalous events were discussed along with general traffic trends observed in the dataset.

The following Chapter continues with the evaluation of the RUSCOPE1 dataset, but rather than looking directly at attributes of the datagrams themselves, analyses the meta-information which can be derived from the timing and addressing information.

Statistical thinking will one day be as necessary for efficient citizenship as the ability to read or write.

Attributed to H. G. Wells – Author

6
Analysis: Semantic

FOLLOWING on from the previous chapter, the analysis presented in the following sections makes use of higher level aggregate and meta-data relating to the packets observed in the RUSCOPE1 dataset. The results of performing a logical distance analysis is contained in Section 6.1. The results of looking at the data from a temporal point of view are to be found in Section 6.2. A brief geopolitical analysis of the collected data is presented in Section 6.3. Sections 6.4 and 6.5 cover the analysis specifically focusing on so-called 'Bogon traffic' and from the perspective of network topology and groupings of data by Autonomous System Number (ASN). The Chapter concludes with a summary.

6.1 Distance Score Analysis

This section discusses the interpretation analysis performed on the RUSCOPE1 Dataset by calculating the Distance Score as introduced in Section 4.3. Initial work on this was presented in Irwin and Barnett (2009), where a relationship was found to exist between packet counts and the calculated $IP\Delta$ score. This study was on on the RUSCOPE1 dataset prior to the advent of the Conficker worm in late 2008. The effects of this worm on the network telescope traffic are discussed in Chapter 7

6.1.1 Early work

The rationale behind the calculation has already been addressed in Section 4.3. What was found is that the calculation showed an increasingly weak relationship to packet counts as the dataset grew. The initial hypothesis proposed by the researcher was that one would see a a number of hosts with high packet count and low $IP\Delta$ score, given the naive scanning algorithms employed by much of the network aware malware prevalent on the Internet. The initial analysis done in mid 2008 showed that while there was a low statistical correlation, when plotted graphically, some relationship could be seen.

Figure 6.1 contains a radar plot of the Packet count and absolute $IP\Delta$ values for the 250 closest recorded addresses to the network telescope as determined by the distance score, having a range of 1.2 to 3.0. What can be seen is that at a very small scale the hypothesis seems to hold true, in that there is a relationship between the two values. This can be seen at particularly at the lower end of the $IP\Delta$ range, where higher packet counts are observed. Worth noting is that log_{10} value of of the packet count has been taken in order have the two series on comparable axes. Only 38 nodes were found to have a distance score of <2.00, and as such can be deemed to be very close representing a minimal distance of 1 753 addresses away, to a maximum of 62 692. Effectively this means that at a minimum the closest nodes were within 6.8 natural /24 (class C) netblocks of the telescope midpoint, with a maximum of 244 /24 netblocks distance (or in effect nearly one /16 netblock away).

Similar results are shown in standard linear plots of the data. such as shown in Figure 6.2 where the inverse relation between the DVS and the packet count can be seen. This image shows the same dataset as figure 6.1. Further to this Figure 6.3 shows the percentage contribution to the total packet count of the closest 250 nodes, as ordered by the $IP\Delta$Score. This again shows that the original hypothesis had some merit, when applied to the constrained RUSCOPE1 dataset.

Examining a Hilbert curve plot of the entire dataset, Figure 6.4 shows a trimmed portion of the resultant image, showing the section of 196.0.0.0/8 (administered by

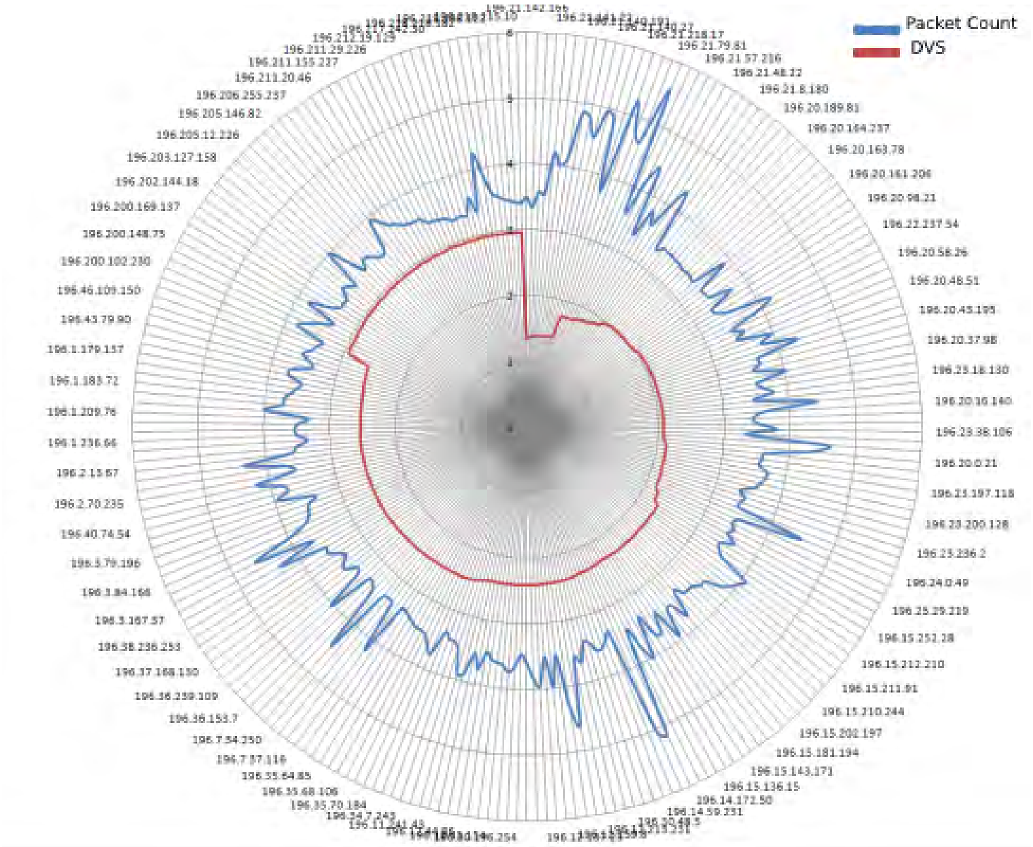


Figure 6.1: Radar plot of distance vector score ($IP\Delta$) vs. log_{10} Packet Count

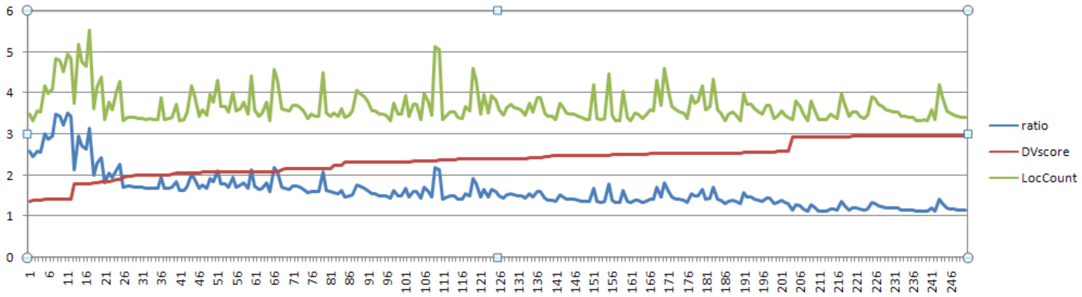


Figure 6.2: Linear plot of DVS vs Packet count and computed ratio

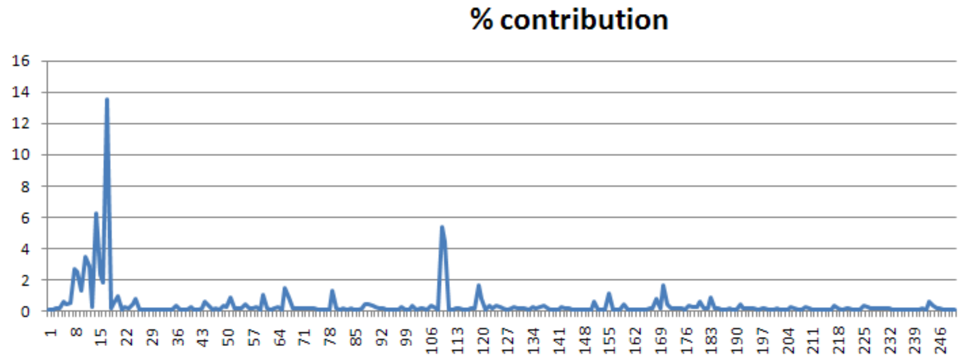
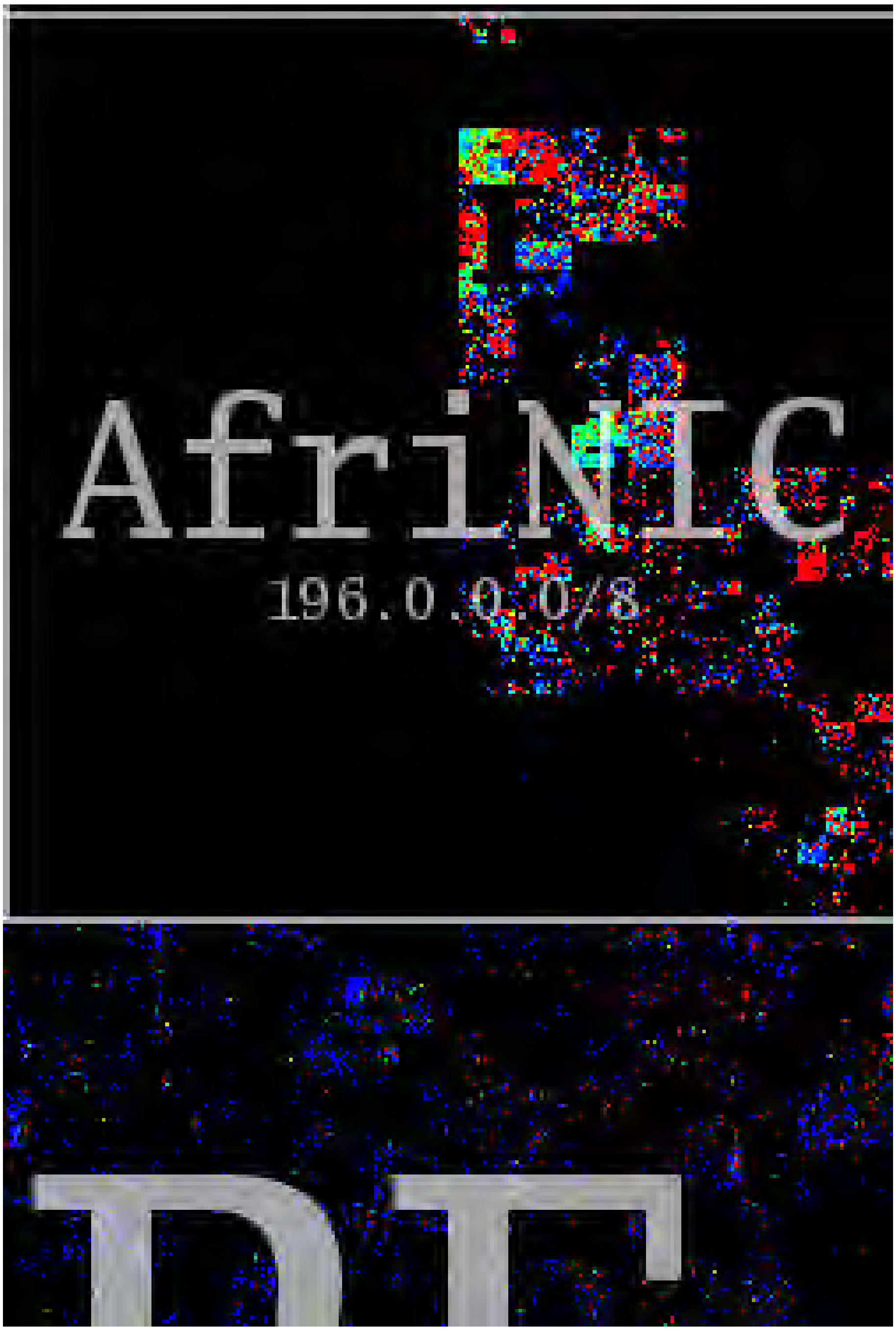


Figure 6.3: Percentage Contribution to packet count of top250 nodes

AfriNIC) and the upper portion of 195.0.0/8, administered by RIPE. 197.0.0.0/8 has been omitted as it was unallocated at the time of the data analysis performed for Irwin and Barnett (2009). This areas was chosen as since the network telescope resides within the 196.0.0.0/8 netblock, these represent the hosts and networks closes numerically, although not topologically. The widespread use of the 196.0.0.0/8 address block is discussed further in Section 6.3.3. The heat map colouring shows the number of packets received, with red showing significantly high levels in comparison to the other data. In this image each dot represents a discrete /24 netblock. Most of 196.128.0.0/9 can be seen to empty. This was determined to be unallocated though the analysis of BGP routing tables. The upper quartile of the 195.0.0.0/8 block (195.192.0.0.0/10) can be seen to show as significantly warmer than the other shown elements within this range. It is proposed that this is due to the closeness of this range to that of the telescope sensor network, with a maximal distance of $\Delta 2.8$, or approximately 7 million IP addresses of the telescope.

6.1.2 Long Term View

Reflecting back on the entire RUSCOPE1 Dataset, and reapplying the $IP\Delta$ analysis, the results were found to not have as close a relationship has had been observed previously. One possibly reason for this is the presence of a number of anomalous spikes in traffic attributable to single hosts. A second reason is that the nature of the dataset changed substantially after the emergence of the Conficker work, which is itself discussed as a case study in Chapter 7. A plot of the packet count *vs.* *IP*ΔScores for the entire dataset is shown in Figure 6.5. This plot used a signed value for *IP*Δ and as such the packet counts appear in two distinct groupings of >2.0 and < -2.0, with the latter making a relatively small contribution to the whole.



The Sensor Network is a class C network that lies within this range, and thus this represents the closest IP addresses. Data used spanned the period August 2005–June 2008

Figure 6.4: Hibert Curve plot showing 196.0.0.0/8 which contains RUCSOPE1

IT can be seen in this plot that the highest packet counts were in fact received from hosts with relatively high $IP\Delta$ scores, somewhat disproving the hypothesis.

A second interesting analysis performed on the RUSCOPE1 dataset was to consider the relationship between the observed TTL of datagrams, and the $IP\Delta$ score calculated against their source address. The result of this is given in Figure 6.6. The three distinct areas of activity around TTL values of 64, 128 and 255 can be observed, as discussed in Section 5.4.2. The interesting feature noticeable in this plot is that hosts with TTL values between 96 and 128 (generally attributable to Microsoft Windows systems) tended to be closer to the network telescope. Two horizontal bands are also of interest, appearing at approximately 3.6 and 3.8. These signify hosts at these distances presenting traffic with a very wide range of TTL values.

Reflecting on the use of this score, it is of interest and may warrant further exploration in the future and in particular comparison with dataset attributes other than packet count, for which the initial hypothesis would appear to have been disproved.

6.2 Temporal Analysis

Looking at the dataset as a whole, over the entire period of capture, one can start explore the dataset for trends, or changes that occur over time. One benefit of the RUSCOPE1 dataset is its long temporal baseline in comparison to other network telescope datasets that have been reported on. While the CAIDA backscatter datasets span from 2004 to the present, these are only snapshots into prevalent activity at a particular time. Other datasets that the researcher is aware of that have had research conducted on them over a period of time and have longer temporal baselines than most of the published works, are those discussed in Cooke (2007) and Pemberton (2007). These both however have much shorter time periods covered than the dataset collected by the researcher in this study.

This section highlights some of the significant observations made. Along with the basic numerical analysis done in the database, use was made of both the Heatmaps, and temporal line plots as discussed in Sections 4.7 and 4.8 respectively. The line plots as described, and other variations thereof were used extensively for deciding

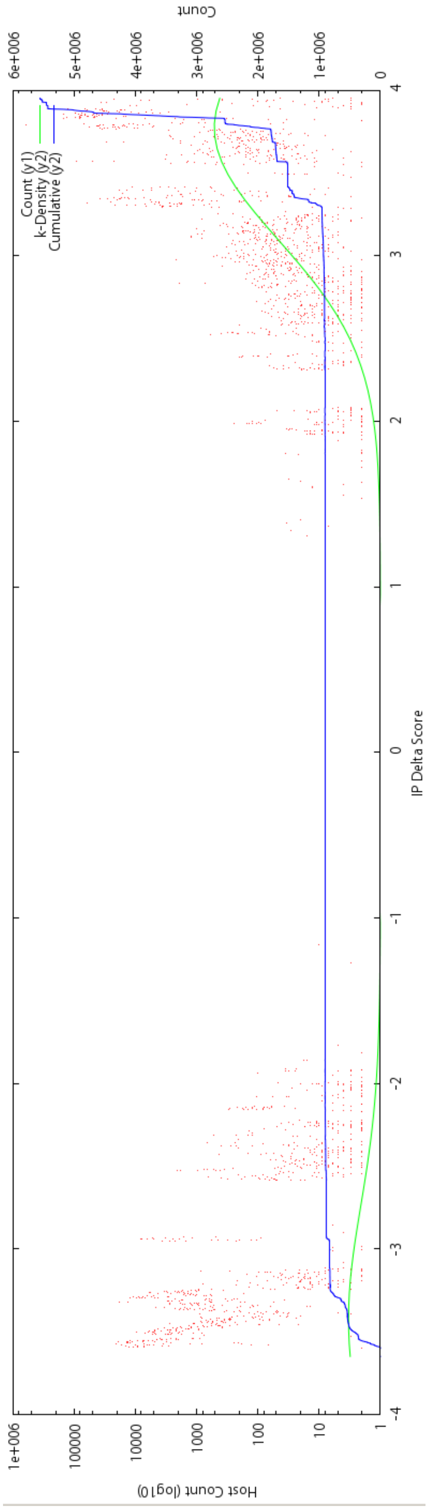


Figure 6.5: DVS range plot

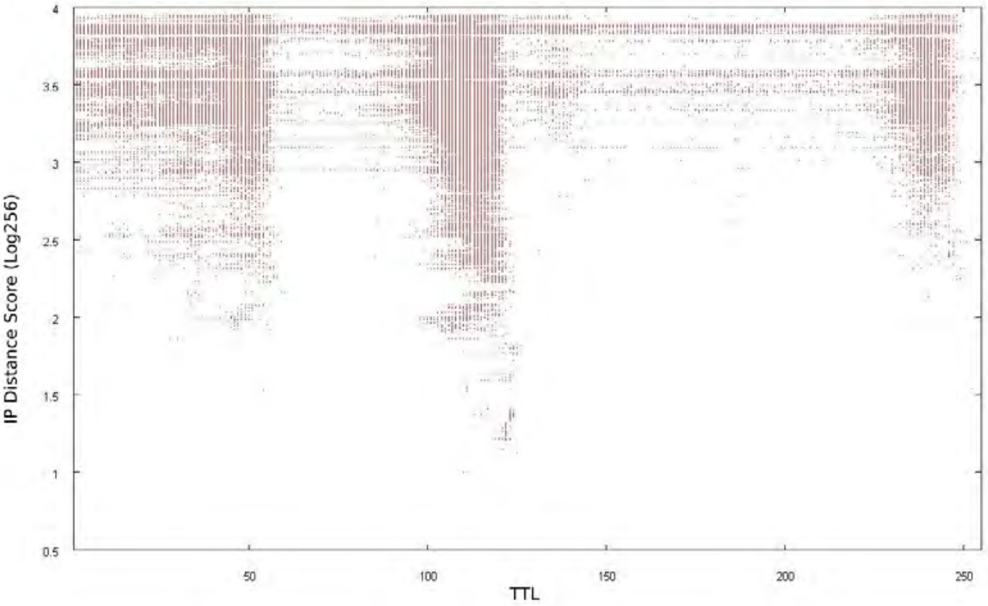
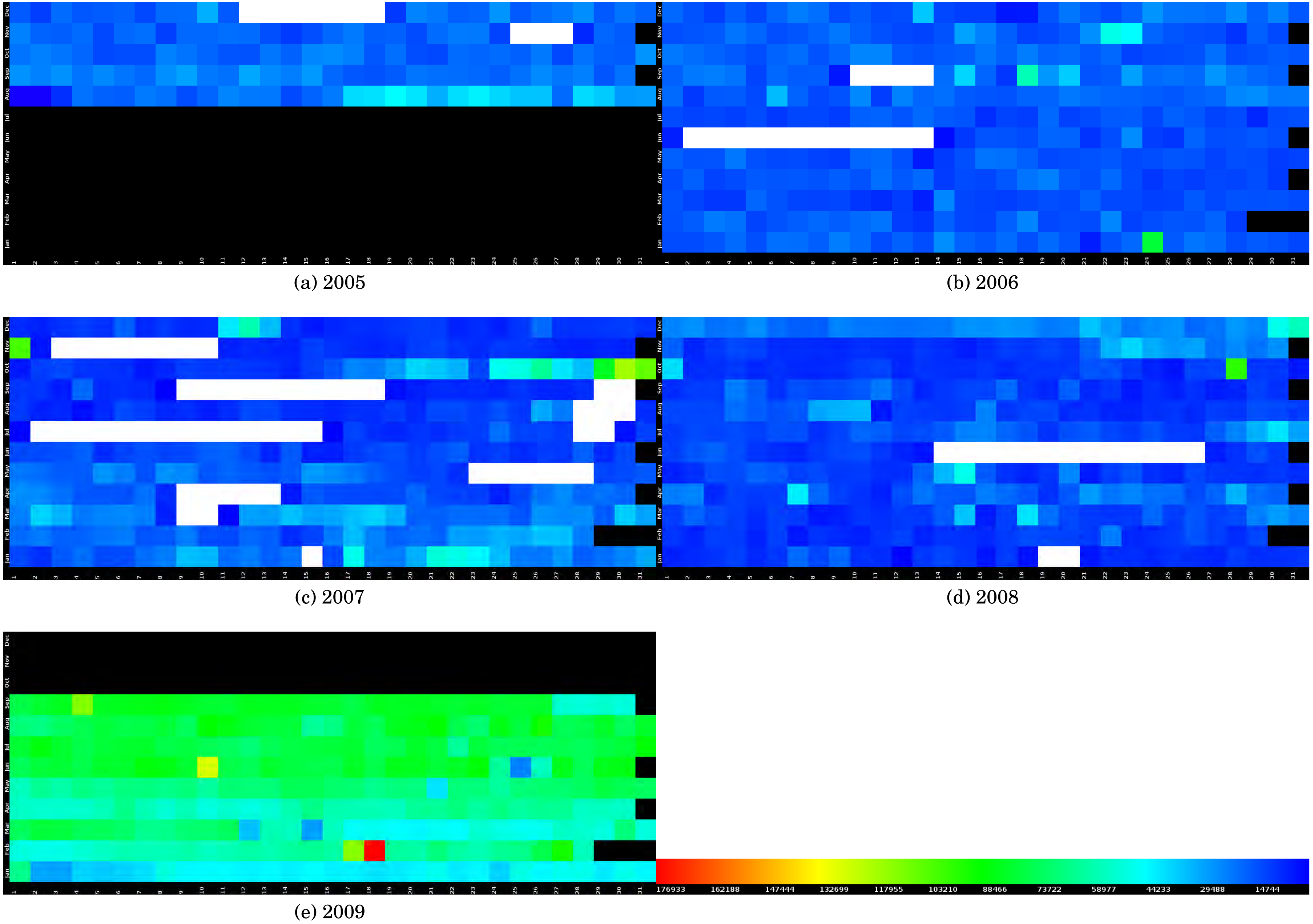


Figure 6.6: DVS vs TTL

what groupings of traffic warranted further examination, or exhibited interesting trends and possibly anomalous behaviour. A scripted framework that allowed for the automated periodic generation of plots was used. The ability to gain an overview of the data visually was found to be important as data volumes increased. Current work by the researcher on the implementation of automated numerical methods for performing similar identification and possibly isolation of incidents is discussed in Cowie and Irwin (2010a,b).

An overview of the traffic received on the telescope is presented in Figure 6.7. The black areas in these sub-figures and in subsequent Heatmaps, indicate periods for which there is no valid data — such as the 30^{th} of February or where the plot includes time periods outside of the dataset. In contrast white areas indicate blocks where data was missing. This was chosen so as to optimise the images for on-screen viewing where white areas are readily apparent. Figures 6.7a-e are plotted as a 12 by 31 matrix with each block representing one day's worth of traffic. The colouring of the block indicated how many packets were received on that day. Overall one can see that the level of activity on a daily basis has increased over the period, with a significant difference in the shading between the plots for 2005 and 2009. During the course of 2009 (Figure 6.7e) traffic levels continue to increase. The red marker in this image on the 18^{th} of February relates to the massive spike of ICMP traffic previously discussed in Section 5.3.4.



(a) 2005

(b) 2006

(c) 2007

(d) 2008

(e) 2009

Figure 6.7: Heatmap plots of Rhodes Data August 2005-September 2009

6.2.1 Port activity trends

The primary means of evaluating these plots was to manipulate the numeric range as mapped against the colour space in order to see what other features may be occluded by high spikes such as in Figure 6.7, where very little is coloured in the warmer range. An example of this can be seen in Figure 6.8. In this case, the maximum range of the colour-scale has been capped at 20 000 packets. Showing the packet counts at a weekly level for each of the top 10 TCP destination ports, as discussed in Section 5.2.1, it provides an overview as to the occurrence of traffic over the period under consideration. Port 50272/tcp shows an interesting pattern in that it was only observed on five weekly periods, yet accounted for nearly 1% of TCP traffic. This is in contrast to traffic destined to TCP ports such as 22,80,135,139 and 445 which were present in all periods. This kind of analysis, evaluating the volume of traffic received over time is important in understanding both the composition of the dataset, but also the emergence of new activity trends.

Such emergent trends can be be observed in Figure 6.8 with traffic destined to 2697/tcp being almost non existent, until a large spike in late 2006 after which, activity levels decreased until a brief resurgence in mid 2009. The gradual increase in traffic targeting 23/tcp is also evident.

Temporal overviews such as those shown was also found to be useful when looking at the changing composition of traffic at a protocol level, as well as with the discrimination between active and passive traffic for ICMP and TCP data as discussed preciously in Section 5.5.

6.2.2 Time based activity

Further investigation was done on potential biases of traffic to particular times of day, and days of week. Basic numerical analysis was performed on the percentage of the total traffic attributable to each hour in a day was found to vary from a low of 3.42% in the 6^{th} hour (05h00–06h00), to a peaks of 4.80% twelve hours later (17h00–18h00) with the highest value of 4.84% obtained at during the hour from 13h00–14h00. Looking at the day of the month, the 18^{th} was found to have the highest percentage (3.58%), although this is likely skewed by the 156 000 packet spike observed on late on the 17^{th} and into the early morning of the 18^{th} of February

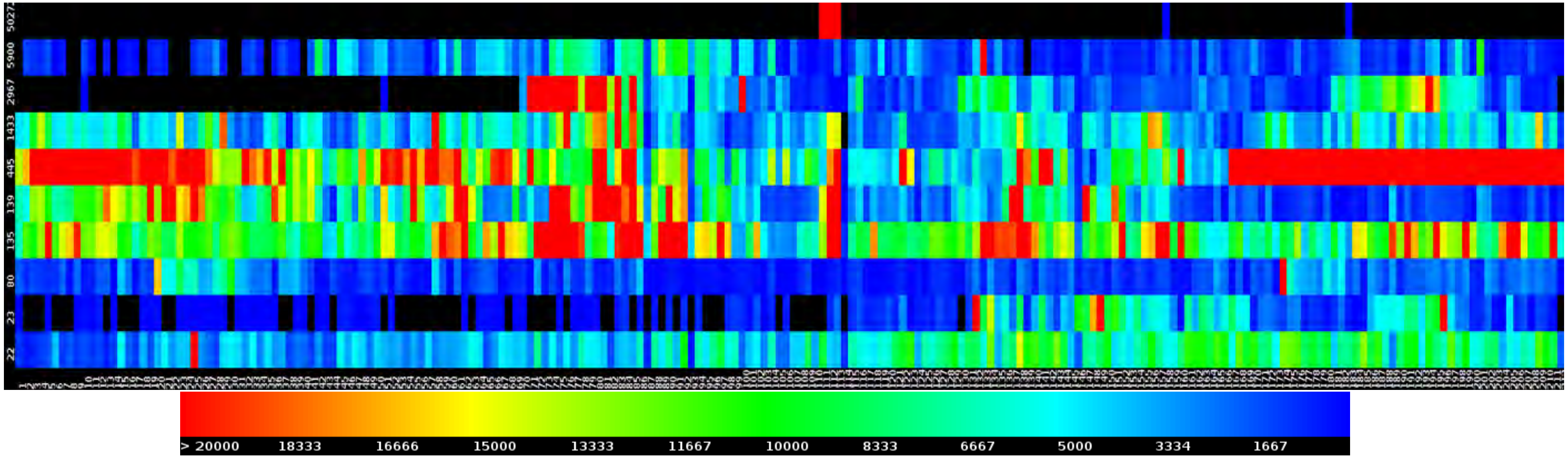


Figure 6.8: Heatmap of the top 10 TCP destination ports

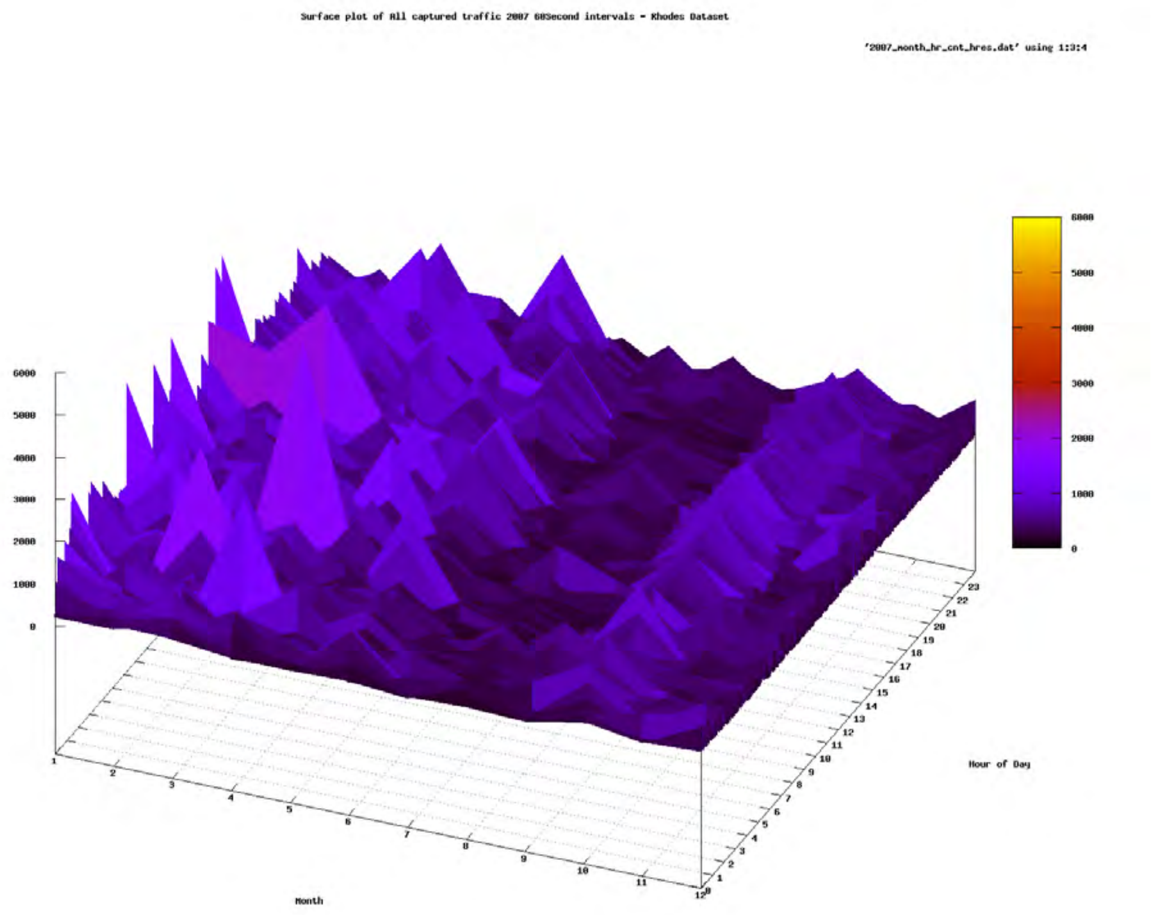


Figure 6.9: Rhodes Dataset Packet count (60 minute intervals)

2009 (Section 5.3.4). The two lowest periods of the month on average were the 11^{th} to 13^{th} and the 2^{nd} and 3^{rd}.

With a relatively small range in the actual numeric data, visualisation methods were applied to the data. The initial attempt at visualising the output was done using gnuplot, as shown in Figure 6.9. This output was found to be unsatisfactory for general printed output, although was a useful means of exploring the data in gnuplot's interactive mode. Subsequent application of the Heatmap tool produced much more usable static output.

An overview of the traffic from the entire period is presented in Figure 6.10. The 7 by 24 matrix shows the relative packet counts observed for each hour of the week. The times used are based on those recorded in the packet captures which are South African Standard Time (SAST) which is GMT+2. What can be seen is that traffic levels rise from the start of the week towards midweek with Monday

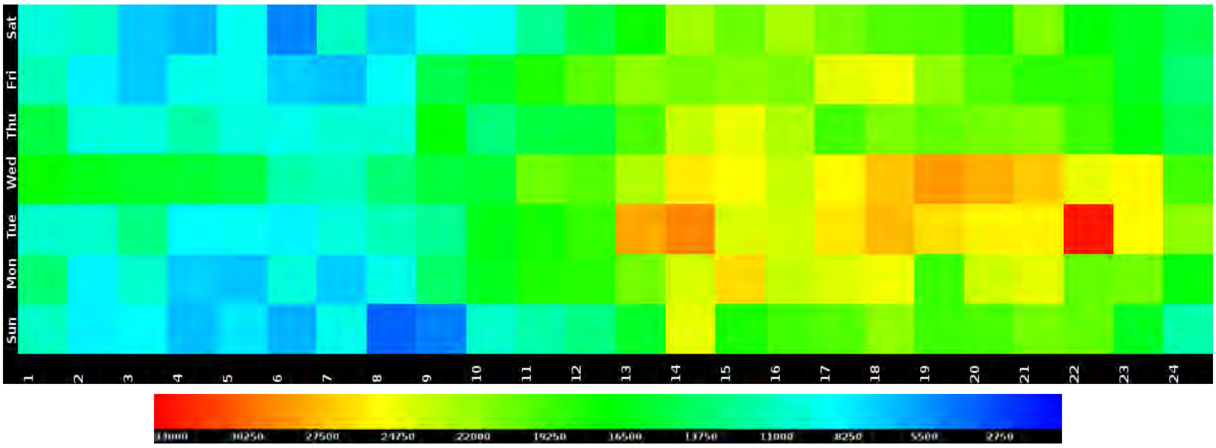


Figure 6.10: Overall traffic

having 14.3%, Tuesday 15.0% and Wednesday having a high of 15.4% of the total traffic. Levels then decline towards the weekend, with traffic levels dropping each day. The lowest levels overall were observed on a Sunday (13.3%), closely followed by a Saturday (13.5%). Considering the time of day, traffic levels rise from around 08h00, with the majority of traffic being observed between 14h00 and 22h00. The early morning period between 02h00 and 07h00 is the when the lowest levels of traffic are observed.

This analysis was repeated, but using data only for hosts that could be geolocated as being in South Africa. The result of this is presented in Figure 6.11. This image differs from the overview in Figure 6.10. Key differences are in the majority of traffic being concentrated between 10h00 and 15h00, with peaks being observed between 11h00 and 13h00. The low periods of Sunday and early morning hours are also lower than those seen overall. Lastly, while in Figure 6.10 Wednesday has a significant overall increase in traffic compared to other days, in contrast, with the South African hosts, Wednesday is actually lower in the morning, with Thursday being the day with the highest sustained levels of traffic.

What can be determined from this analysis is that generally relatively lower levels of traffic are observed outside of traditional business hours. This in turn points to the likelihood of much of the activity observed originating from systems which are only connected during business hours. This is somewhat counter-intuitive to what would be expected with automated scanning by malware, which should operate on a near continual basis, especially given the prevalence of always-on broadband connections. One plausible explanation is that for machines located at businesses, they are powered down over weekends (or the link is disconnected or down), and

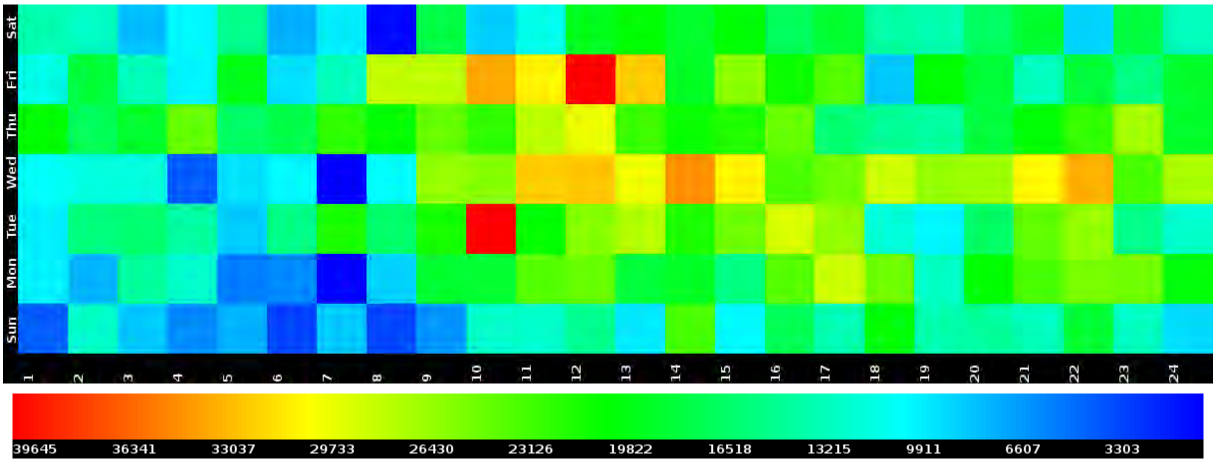


Figure 6.11: South African Hosts

for home users, systems are similarly generally either powered off or disconnected during the early hours of the day. The increasing use of laptops by many home users in preference to desktop systems may account for this behaviour as well.

Further temporal analysis work relating to the Conficker worm and traffic targeting 445/tcp is discussed in Section 7.2.3.

6.3 Geopolitical Analysis

Initial work done on the geopolitical analysis of observed network telescope traffic has been previously published in Irwin *et al.* (2007). This paper concluded that while significant traffic was observed from traditional 'hotspots' such as China (CN) at the time of publication, the United States of America (US) closely followed by Canada (CA), was actually the primary source of traffic. A somewhat unexpected second tier consisting of developing countries such as South Africa (ZA), Egypt (EG) and Brazil (BR), all emerging economies, was also identified. For the purposes of this Section, countries are referred to by their ISO 3166[1] two-letter short-codes. A selected list of these is presented for reference in Appendix C Section C.6. The geopolitical aspects of the analysis of traffic observed are also discussed in Sections 6.4 and 7.6.

[1] http://www.iso.org/iso/country_codes.htm

6.3.1 Total Traffic

An analysis was performed on the RUSCOPE1 dataset in order to ascertain the change in originating sources of traffic for the period under investigation. A listing of the top ten originating countries for each year of study is shown in Table 6.1. During the period 2005-2008, the top ten countries represented at least 70% of the total packets observed for each year, and over 83% when the top 20 were considered. The observed change in composition when considering the 2009 data is most likely due to both the wider spread of countries observed and the marked increase in recorded packet counts — 2009 accounting for over 45% of the total data.

The cause of this increase and observed wider spread of origin states is most likely due to the spread of the Conficker worm as discussed in Chapter 7. China (CN) remains a top source of traffic overall, with the exception of 2007, when an anomalous spike was observed in traffic from South African (ZA) hosts, relegating it to second place. Traffic originating from the United States (US) remained in the top three positions, most of this being attributable to the various commercial Internet Service Providers serving the SME and residential markets. A significant portion of the traffic observed originates from the 24.0.0.0/8 block, which is utilised by providers of 'Internet over Cable'. South African traffic was also present in relatively large volumes, placing it among the top three in three of the five annual periods, and third overall. It is strongly believed that this is due to an observation bias experienced by the telescope, as discussed in Section 6.1.

Other than South Africa, Egypt (EG) and Mauritius (MU) both feature in the top ten rankings. Both of these countries have relatively good broadband access available in comparison with the rest of Africa, where dial-up or satellite based connectivity remains common. The availability of fast and affordable connectivity can probably account for the higher showing of these countries in the rankings. The other countries present in the top ten are all known to have relatively well-established broadband telecommunications offerings in reasonably liberalised markets, leading to much greater penetration. Western Europe is well represented with Germany (DE), France (FR), Italy (IT), the United Kingdom (GB) and Spain (ES) all appearing.

The general lack of caps or quotas on traffic in particular leads to infected or compromised systems staying online much longer than they would in Africa, where these restrictions are commonly enforced. 'Bad traffic sources' have traditionally

Table 6.1: Top 10 Countries by Packet Count

Year	2005		2006		2007		2008		2009		Overall	
Rank	cc	%	cc	%	cc	%	cc	%	cc	%	cc	%
1	CN	24.87	CN	23.37	ZA	20.77	CN	27.60	CN	15.36	CN	19.73
2	ZA	18.11	US	18.02	CN	18.73	US	10.54	RU	10.47	US	10.94
3	US	15.17	ZA	15.55	US	12.44	MU	8.54	US	7.31	ZA	9.51
4	JP	3.75	DE	4.27	ES	6.49	ZA	6.33	BR	6.13	RU	5.55
5	KR	3.31	TW	3.16	EG	3.53	TW	4.38	TW	4.06	TW	3.65
6	TW	3.18	JP	3.00	TW	2.52	EG	4.27	IT	3.86	BR	3.53
7	EG	3.01	EG	2.76	DE	2.51	RU	2.58	DE	3.43	DE	3.11
8	DE	2.32	KR	2.40	KR	2.15	IT	2.38	KR	3.17	KR	2.66
9	CA	1.49	GB	1.97	JP	1.92	ES	2.10	ZA	3.16	IT	2.58
10	FR	1.44	FR	1.78	FR	1.52	DE	1.87	RO	3.07	EG	2.42
Total		76.66		76.30		72.58		70.59		60.01		63.69

Percentage is expressed as a portion of the annual packet count

$N_{2005} = 271882$ $N_{2006} = 6638185$ $N_{2007} = 6193984$ $N_{2008} = 5990987$

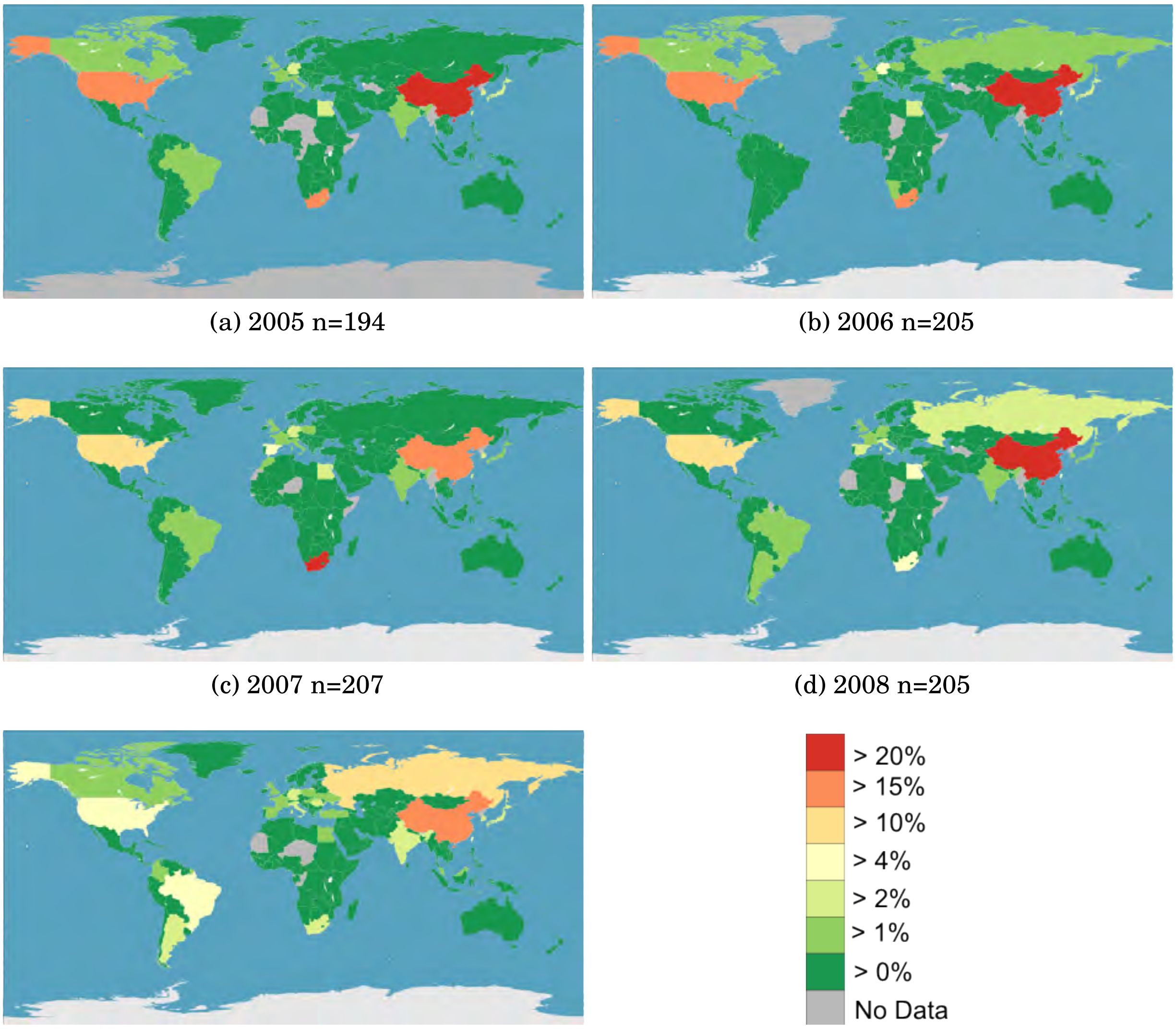
$N_{2009} = 18706816$ $N_{Total} = 40801854$

been deemed to be China (CN), Korea (KR) and Taiwan (TW) mostly due to general apathy from service providers in dealing with abuse complaints, and almost non-existent legislation relating to cybercrime. All of these countries feature prominently in the top ten rankings.

The rankings shown in Table 6.1 should be compared to the markedly different results shown in Table 6.2, discussed in the following section, which ranks the countries by the host count.

The full packet count dataset is shown in Figure 6.12, which plots the traffic observed and attributable by country, based on the percentage composition of the year's traffic. The chloropleth colouring schema is based on the total percentage contribution of traffic determined to be originating from a particular country, in relation to the annual traffic total. China can clearly be seen as a hot spot, along with South Africa and the United States. Of interest in these plots is the absence of observed traffic from a number of African countries, particularly Congo (CG), Niger (NE), Mauritania (MR), Somalia (SO) and Western Sahara (EH). The lack of traffic attributable to the Western Sahara is explainable given its disputed status. Traffic that may originate from here is most likely included in the traffic from Morocco (MA), who currently controls the territory. Two other areas notable for their total lack of traffic are the Democratic Peoples Republic of (North) Korea (KP) and Myanmar (MM), formally known as Burma. The traffic plot from 2009

(Figure 6.12e) presents the widest coverage observed with packets attributable to 218 of the 246 countries recognised by the International Standard Organisation in their ISO 3661 standard. Taking the entire dataset into consideration, traffic was observed from 228 countries, although of these 71 could be attributed to fewer than 100 packets.



(a) 2005 n=194

(b) 2006 n=205

(c) 2007 n=207

(d) 2008 n=205

(e) 2009 n=218

Figure 6.12: Geopolitical breakdown of monitored traffic by year
Colouring is based on the percentage of traffic received on an annual basis. 2005 and 2009 do not represent full years, being 5 and 9 months respectively.

6.3.2 Unique Hosts

Looking at the geolocation data from the perspective of the number of unique IP addresses attributed to each country shows a somewhat different ranking as displayed in Table 6.3, where it can be seen that the US dominates from 2005 to 2008, yet ranks only 12^{th} in 2009 despite there being a 3.24 times growth in the number of attributable hosts observed. One should recognise that due to the use of technologies such as Network Address Translation (NAT) and Dynamic IP address allocation (via DHCP or other means) there is not necessarily a direct mapping of IP address to a single host. However, assessing the data from this point of view provides a best effort assessment. As seen in Table 6.2, the top 10 represent a significant portion of the total traffic. While some marked dilution of the proportion represented is present over the observed period, from a high of 81% in 2006 to 63% in 2009 the values represented remain significant. When taking the top 20 countries (in effect the top 8% of the countries in ISO3661) into account the lowest recorded coverage is 81% (2009), from a high of 90% in 2006.

The rise of traffic attributable to Russia (RU) and Romania (RO), among several other countries, is most likely attributable to the vast numbers of compromised hosts performing scanning. While packets counts were low, the scanning activity was quite widespread. These two countries ranked 2^{nd} and 18^{th} respectively in 2009 on the basis of packet counts. This issue is revisited in Chapter 7.

A comparison of the rankings of countries over the five year period is shown in Table 6.3, which lists the rank positions of the eighteen countries that appeared in the top 10 in Table 6.2. The table is ordered by an average ranking score determined from the 2005-2009 values. The most consistent appearances in the top 10 are from China, United States of America, Germany (DE), Taiwan and Japan (JP) which appear in every annual grouping. Two countries worth taking particular note of due to their meteoric rise are Russia and Brazil (BR). Russia has risen from 22^{nd} position in 2005 (accounting for 0.5% of both traffic and hosts) to first place in 2009 (14.9% of hosts and 10.5% of traffic). The corresponding growth in the volume of hosts observed and total traffic volume is 41 581% and 11 351% respectively. At the same time the actual level of activity normalised on a per host basis has dropped from 11.5 pkts/host in 2006 to 3.15 pkts/host in 2009, showing a 72% decline. Considering the same analysis for Brazil one can see a rise in the number of hosts from 1% to 9.44% of the annual host totals observed, while traffic volumes increased from 1.2% to 6.3%. Similar to the Russian case, the

Table 6.2: Top 10 Countries by Host Count

Year	2005		2006		2007		2008		2009		Overall	
Rank	cc	%	cc	%	cc	%	cc	%	cc	%	cc	%
1	US	41.91	US	38.81	US	27.01	US	13.20	RU	14.60	RU	11.16
2	CN	9.87	CN	9.98	DE	11.67	CN	12.39	BR	9.44	US	10.18
3	DE	5.39	DE	9.84	ES	10.05	TW	9.20	CN	7.98	CN	8.43
4	CA	4.61	CA	4.92	CN	7.38	AR	8.63	IT	6.71	BR	7.44
5	GB	3.56	TW	4.86	TW	3.41	ES	5.40	TW	5.50	DE	6.09
6	TW	3.52	GB	4.12	CA	3.28	IT	3.92	DE	5.28	IT	5.41
7	JP	3.07	FR	3.71	JP	2.99	DE	3.82	AR	4.18	TW	5.33
8	FR	2.64	JP	2.78	KR	2.81	KR	2.81	IN	3.98	AR	3.57
9	ES	2.50	KR	1.70	GB	2.68	BR	2.69	JP	2.98	IN	3.19
10	ZA	1.88	MA	1.19	FR	2.60	JP	2.59	RO	2.82	JP	2.97
Total		78.95		81.91		73.89		64.64		63.48		63.79

Percentage is expressed as a portion of the annual distinct host count

$N_{2005} = 296020$ $N_{2006} = 576391$ $N_{2007} = 336416$ $N_{2008} = 253934$

$N_{2009} = 4169024$ $N_{Total} = 5557688$

number of packets observed per host address dropped from 12.69 to 2.91 (a 77% decline). What can be inferred from this is that more systems are either scanning less frequently or are utilising more focused scanning.

Countries such as the United Kingdom (UK) and France (FR) are different from the others shown in terms of the marked decline in the rankings over the observation period. Although as with the other countries, the volumes of traffic increased 753% as did the number of hosts observed, 523% in the case of the United Kingdom. A more important observation is the decrease in the relative proportions of hosts from 3.56% (UK) and 2.64% (FR) in 2006 to 1.32% and 1.08% in 2009 respectively. The most likely cause for this general decrease (other than the massive inflation in the volumes of traffic received) is the general prevalence of licensed operating system software and therefore access to preventative technologies such as Microsoft Windows Update, along with public awareness. One would expect a similar trend in Germany (and other western European countries). However, while there has been a decrease in rank, the change has not been as significant, possibly due to activities in former Eastern Germany, where there is still likely a large proportion of unlicensed Operating Systems, resulting in the inability to apply patches against threats such as that detailed in MS08-067 (Microsoft, 2008a).

Other than South Africa (ZA), Morocco (MA) is the only African country to feature in the top ten, although Egypt (EG), and Tunisia (TN), both countries with

Table 6.3: Rankings of Countries appearing in the top 10 by host count

	2005	**2006**	**2007**	**2008**	**2009**	**Avg†**	**Overall**
CN	2	2	4	2	3	2.6	3
US	1	1	1	1	12	3.2	2
DE	3	3	2	7	6	4.2	5
TW	6	5	5	3	5	4.8	7
JP	7	8	7	10	9	8.2	10
ES	9	11	3	5	17	9	12
BR	14	13	12	9	2	10	4
KR	13	9	8	8	14	10.4	14
CA	4	4	6	11	29	10.8	18
IT	15	16	13	6	4	10.8	6
GB	5	6	9	17	20	11.4	15
FR	8	7	10	14	22	12.2	17
RU	22	22	18	12	1	15	1
IN	21	18	20	16	8	16.6	9
AR	30	25	23	4	7	17.8	8
ZA	10	14	11	24	60	23.8	33
RO	44	30	25	18	10	25.4	11
MA	19	10	16	40	69	30.8	45

† Average is computed over rankings from 2005-2009

$N_{2005} = 194$ $N_{2006} = 205$ $N_{2007} = 207 N_{2008} = 205 N_{2009} = 218$ $N_{Total} = 228$

Table 6.4: African Countries by host count

Year	2005		2006		2007		2008		2009		Overall	
Rank	cc	R_G	cc	R_G	cc	R_G	cc	R_G	cc	R_G	cc	R_G
1	ZA	10	MA	10	ZA	11	EG	23	EG	41	EG	29
2	EG	11	ZA	14	SN	15	ZA	24	ZA	60	ZA	33
3	TN	18	EG	17	MA	16	MU	30	MA	69	MA	45
4	MA	19	TN	23	EG	17	MA	40	MU	71	DZ	52
5	CI	25	CI	26	TN	21	BF	63	KE	84	TN	62
6	MU	26	MU	35	MU	35	SD	71	SD	86	SD	66

R_G is the global ranking on a per year basis

$N_{2005} = 194$ $N_{2006} = 205$ $N_{2007} = 207 N_{2008} = 205 N_{2009} = 218$ $N_{Total} = 228$

relatively modern telecommunications infrastructures and Internet penetration, appear in the top 20 on a number of occasions. Surprisingly Senegal (SN), and Sudan (SD), also appear relatively highly rated, with the latter achieving a rank of 86 in 2009, however, this still places the Sudan in the top 40% of countries observed. A breakdown of top African states is presented in Table 6.4 which provides a continental ranking, along with the global ranking (R_G) in that year.

While South Africa can be seen to dominate, it should be borne in mind that this country has the most developed telecommunications infrastructure, and arguably the highest density of IP address allocations in Africa. As shown in Section 4.4 and Figure 4.8, despite having a lower number of total netblock routes, many of the South African netblocks are of size /16, particularly those blocks on the TENET network. This is largely due to the early allocations to Universities in the early 1990's. The diluting effect of the massive increase in traffic experienced during 2009 can be seen with the entry of countries such as Algeria (DZ) and Kenya (KE) into the upper portions of the ranking tables. While the ranking of African states dropped significantly, the number of hosts observed grew dramatically in comparison to previous years. Egypt, for example, rose from a host count of 1 857 in 2008 (with a global ranking of 23^{rd}) to 12 375 in 2009 — a startling 666% increase — yet dropped to 41^{st} in the global rankings. Similarly the count of observed hosts from South Africa rose from 1 815 to 4 980 (growth of 269%) from 2008 to 2009 yet the ranking slipped 26 placings.

6.3.3 AfriNIC

The African Regional Internet Registry (AfriNIC) maintains authoritative control over the 196.0.0.0/8, 197.0.0.0/8 and 41.0.0.0/8 network address blocks, the latter two of which were assigned in October 2008 and May 2005 respectively. The use of the 196.0.0.0/8 netblock pre dates the creation of the Regional Internet Registries, and allocations have been made out of this block since May 1993 (Gerich, 1993; Internet Assigned Numbers Authority (IANA), 2008b) and was consequently used for address assignments on a global basis, particularly for organisations needing direct assignment of smaller address allocations, than were available out of the historical 'Class B' (/16) blocks. Consequently allocations were typically of /24 or /23 in size. Geographical allocations from this address space seem to have been particularly prevalent in the developing world, including Africa, the Caribbean, the Indian Sub-Continent and Latin America. This diversity is displayed in Figure 6.13 reflecting those IP addresses within 196.0.0.0/8 from which traffic was observed and their geolocated countries of origin. In reality there are address blocks within this IP address space which are in countries not illustrated in the Figure, from which traffic was not observed. Subsequent to the allocation of this legacy address space to fall under the administrative control of AfriNIC, allocations are awarded only to countries under its sphere of control. A plot of the geographical zones of operation of the Registries can be seen in Section C.4 of Appendix C.

The RUSCOPE1 network telescopes operates both within 'African' IP address space, and on the African continent. Consequently it is of particular interest to look at the traffic received from African states. This is shown in Figure 6.14, which displays plots of traffic received from African states from the three aforementioned network blocks under African control, *viz* 41/8, 196/8 and 197/8. It is important to be cognisant of the fact that these are by no means the only address blocks in use in Africa, with many organisations in South Africa maintaining use of address space allocated in 128.0.0.0/2 (pre CIDR 'Class B' space), that were obtained prior to the establishment of the Regional Internet Registries. Similarly other organisations on the continent make use of allocations from both this space, and addresses coming from the former 'Class C' ranges of 192.0.0.0/3. In both sub-figures, South Africa and Egypt are dominant, with Morocco ranking in third place.

A more detailed example of the use of geolocation can be seen in Chapter 7, where the outbreak and spread of the Conficker worm is analysed.

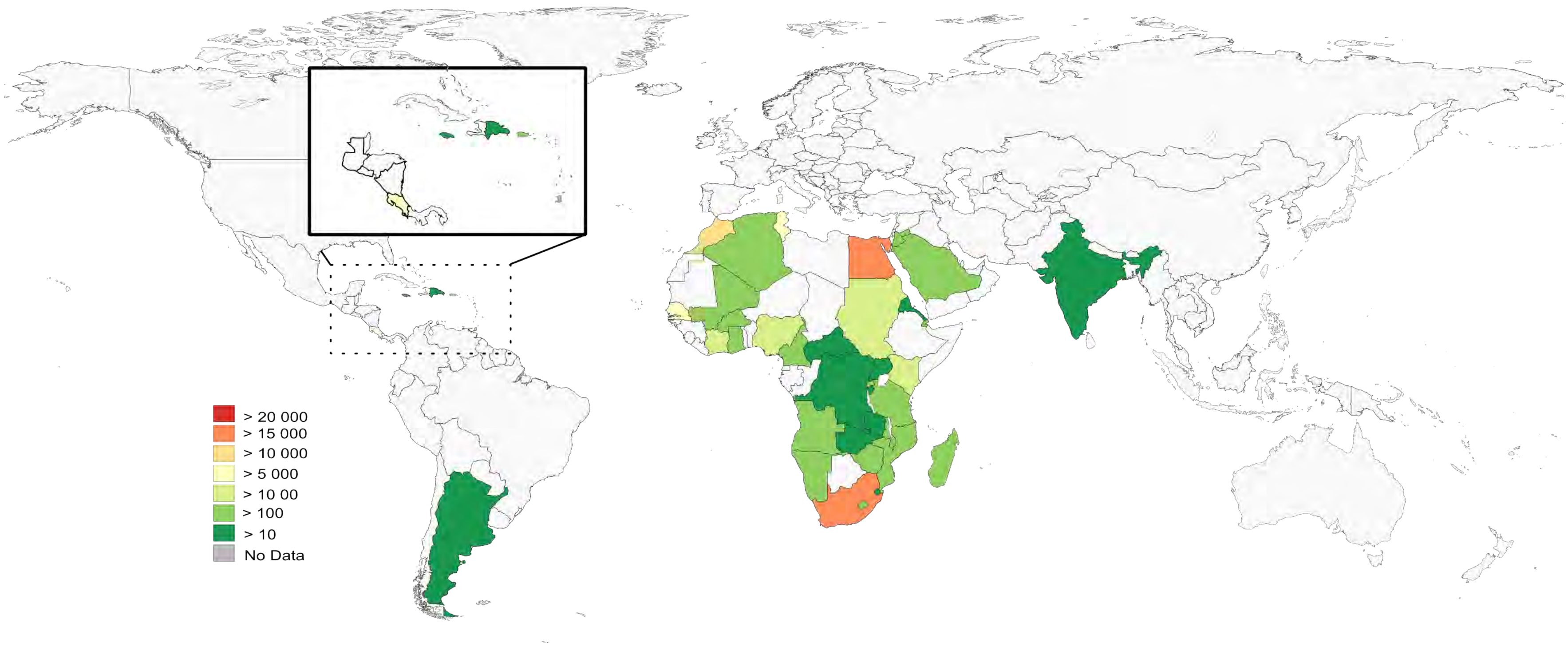
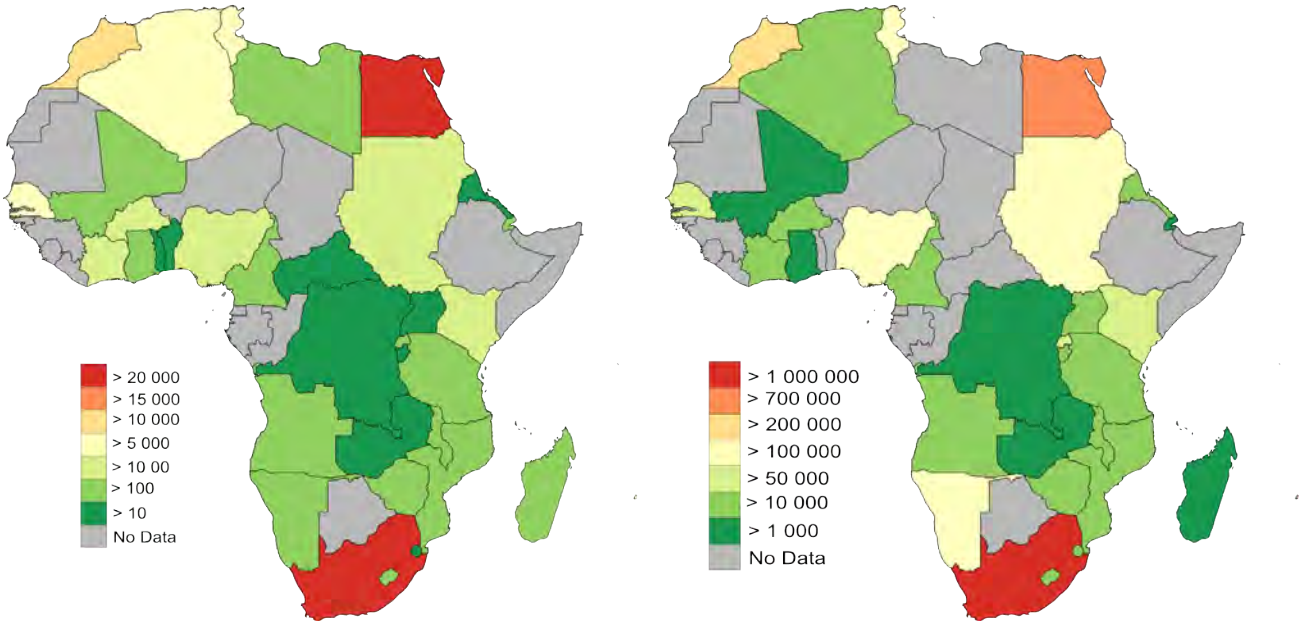


Figure 6.13: Geographical dispersion of traffic from 196.0.0/8 by unique hosts



(a) Host Count

(b) Packet Count

Figure 6.14: Geopolitical plots of African Traffic from 196/8

6.4 Topological View

Taking a topological view of the observed traffic provides a slightly different picture to that seen thus far. This analysis is based on the actual network routing topology of the Internet. As a part of the operation of the Internet, global routing tables are based on the concept of Autonomous Systems (AS). In effect an AS represents a group of network address blocks, possibly aggregated together, as a routing prefix which are under the control of a single organisation. In the case of TENET (AS 2018), all the routes to netblocks at higher education and research centres on the network are aggregated and published globally via BGP under a single AS number (ASN). This occurs similarly with Internet Service Providers (ISP), telecommunications operators and other large organisations responsible for providing IP connectivity and upstream service to endusers. These ASNs are registered via IANA, but are assigned and maintained by the Regional Internet Registries such as AfriNIC, per RFC 1930 (Hawkinson and Bates, 1996). The current list of ASN assignments may be obtained at `http://www.iana.org/assignments/as-numbers/`. Traditionally these have been 16bit values, but the introduction of 32bit values per RFC 4893 and RFC 5396 (Vohra and Chen, 2007; Huston and Michaelson, 2008) has allowed for expansion. Due to the registration process required, the data can be treated with a larger measure of trust than DNS registrations.

While it is acknowledged that BGP spoofing can occur either though malicious intent or as a consequence of misconfiguration such as in the case of the network operators in Pakistan inadvertently hijacking the IP address space for YouTube[2], and China[3,4] receiving routed traffic for US address allocations, it is reasonably rare. Thus, resolving observed addresses to their respective ASN entries in global BGP routing tables provides a means of ascertaining at a fairly coarse level the location and organisations responsible for them. Table 6.5 provides a summary of the top ten observed autonomous systems from the telescope data. Ranking of ASNs was based on the count of the number of distinct /24 sized blocks of addresses that were observed in the dataset.

6.4.1 Methodology

In order to perform this analysis, the 40 million observed packets were reduced to 5 557 688 unique IP addresses. These were in turn reduced to a list of 1 017 429 addresses which needed to be queried against the look-up service. This was done by selecting the unique /24 netblocks present in the dataset, by masking off the last octet of the address. The look-up list was further trimmed by removing the netblocks described in RFC 3330, and the current Cymru Bogon list (see Listing 10 in Appendix C). The Cymru.com IP to ASN lookup tool[5] was used to resolve listing.

A sample of output is shown in Listing 9. This was then further post-processed and loaded into a PostgreSQL database. The accuracy of this approach of performing look-ups after the fact, rather than at the time of capture, is relatively high as organisations generally tend to accumulate address space over time — the endpoint allocations and assignments may well change, but the assignments from RIRs stays with the larger organisation performing the aggregation and providing the Internet connectivity. As such this means of evaluating the source of traffic has merit. The results of this analysis are discussed in the following section.

Table 6.5: Top active AS

Rank	**AS**	**Counts**[1]	**Nets**[2]	**Name**	**Detail**	**Country**
1	4134	77066	5949	CHINANET-BACKBONE	No.31,Jin-rong Street	CN
2	4713	25184	87	OCN	NTT Communications Corporation	JP
3	3320	24237	25	TAG	Deutsche Telekom AG	DE
4	4837	22716	418	CHINA169-BACKBONE	CNCGROUP China169 Backbone	CN
5	3352	22592	192	TELEFONICA-DATA-ESPANA	TELEFONICA DE ESPANA	ES
6	7132	19849	102	SBIS-AS	AT&T Internet Services	US
7	3269	19382	235	ASN-IBSNAZ	TELECOM ITALIA	IT
8	3462	18768	241	HINET	Data Communication Business Group	TW
9	19262	18365	808	VZGNI-TRANSIT	Verizon Internet Services Inc.	US
10	5089	12444	81	NTL	NTL Group Limited	GB

N= 1 023 193

[1] Count is defined as the number of distinct /24 blocks containing traffic observed on the telescope

[2] Nets is determines as the number of distinct IPv4 address blocks (or prefixes) being advertised via BGP that appeared in this list

Data was extracted using the Cymru.com IP to ASN lookup tool available at `http://www.team-cymru.org/Services/ip-to-asn.html`. Look-up data last confirmed 24-12-2009

Listing 9 Sample IP to ASN output

```
ASN     Query IP      ParentNetblock   CC   RIR       DATE_Alloc   BLOCKNAME  DESCR
8447  | 195.3.88.0  | 195.3.64.0/18  | AT | ripencc | 1997-06-27 | TELEKOM-AT Telekom Austria
34251 | 195.3.128.0 | 195.3.128.0/22 | UA | ripencc | 2006-07-31 | IMC-AS ISP IMC,Kiev,Ukraine
12998 | 195.3.159.0 | 195.3.156.0/22 | UA | ripencc | 2006-08-09 | BGNET-AS ISP BGNet
NA    | 195.3.181.0 | NA             |    |         |            | NA
```

Note: "NA" indicates that the address is not in the historical or current routing tables, and is most likely to indicate spoofed addresses

6.4.2 Results

Considering the top ten organisations as determined by ASN, as shown in Table 6.5, China (CN) shows up as the highest ranked source by number of distinct /24 networks, with two providers accounting for 9.8% of observes blocks ranked first and third. The next highest nation is the United states of America, with two providers accounting for 3.75%. In total the top ten organisations accounts for 25% of observed /24 blocks. A researcher can take two views when considering this kind of result. The first is that these are large providers, and hence one would expect a larger presence, even if infection rates were in line with an observed average. The second is that these providers may have a weak or apathetic approach to handling abuse complaints, and clients may be running wild. Further investigation would be needed to support either theory, with one approach being to calculate the coverage of the address blocks observed as a proportion of those actually allocated to the organisation.

Looking at the aggregated coverage on a per country level, the situation changes slightly. Table 6.6 contains the top ten ranking of country by the total number of /24 networks observed. The United States of America (US) and China top the rankings with a combined representation of 31% of the total number of /24 netblocks observed. The countries shown account for 63.66%. Expanding this to the top twenty rank countries out of the 208 observed, accounts for 81% of the networks. All of the countries have relatively cheap available broadband, and fairly high levels of Internet penetration within the population. Looking at representation by African states, South Africa (ZA) comes in position 35 (5 328 networks) followed by Egypt at 42 (3 561). These values should be considered in light of the inequitable distribution of IPv4 Address space as shown in Section 4.4. Current Data on Address

[2] http://www.ripe.net/news/study-youtube-hijacking.html

[3] http://www.bbc.co.uk/news/technology-11773146

[4] http://www.infoworld.com/t/routers-and-switches/chinas-internet-hijack-attack-or-accident-461

[5] http://www.team-cymru.org/Services/ip-to-asn.html

Table 6.6: Coverage by /24 network

Rank	CC	/24 nets	%
1	US	197 649	19.31
2	CN	129 648	12.67
3	JP	56 991	5.56
4	DE	53 045	5.18
5	BR	43 586	4.26
6	ES	37 543	3.67
7	GB	35 374	3.45
8	RU	34 170	3.34
9	KR	32 579	3.18
10	TW	30 878	3.01
		Total	63.66

N= 1 023 193

allocations can be obtained from the BGP Weather map[6] website.

6.5 Bogon Analysis

The last portion of traffic to be considered in this analysis of the RUSCOPE1 dataset is that which should not actually be there by virue of it originating from addresses that are reserved for use, and should not be observed on the public internet. These address blocks are known within the network security community as Bogons, referring to any bogus or incorrectly formed packet sent on a network[7]. This traffic accounts for a very small portion of the whole, but is of particular interest since in theory it should not exist. As seen however in Chapter 5, and particularly in the Section 5.2, datagrams were observed with various attributes that were 'out of spec' or contained undefined values. Ideally this kind of traffic should be filtered at ingress points onto the global Internet as per RFC 1918 (Rekhter *et al.*, 1996) and RFC 2827/BCP 38 (Ferguson and Senie, 2000). The term 'Bogon' is defined by Team Cymru[8] as:

> *A bogon prefix is a route that should never appear in the Internet routing table. A packet routed over the public Internet (not including over VPNs*

[6] `http://bgpmon.net/`

[7] `http://www.retrologic.com/jargon/B/bogon.html`

[8] `http://www.team-cymru.org/Services/Bogons/` Accessed 2009-11-13

> *or other tunnels) should never have a source address in a bogon range. These are commonly found as the source addresses of DDoS attacks*

Bogons, also referred to as 'Martian'[9] packets, include both the Special-Use IPv4 Addresses as currently defined in RFC 3330 (IANA, 2002) and unallocated address space from the IANA perspective[10] – that which is currently reserved and pending allocation to a RIR for further onward assignment. In terms of this designation, packets with address sources originating from such network blocks should not be present on the global Internet. The main reason for this traffic being observed is most likely due to misconfiguration of networking devices such as broadband modems or organisational gateways. A small portion of this traffic may be due to address spoofing, as part of decoy scans or flood attacks.

This section looks at Bogon traffic in two groups, the first in Section 6.5.1 being those addresses as initially defined in RFC1918 (Rekhter *et al.*, 1996) and subsequently in RFC3330 (IANA, 2002) (along with other special-purpose addresses) as being for use on private IP networks. These address blocks are commonly used by organisational intranets and for providing connectivity to multiple systems on broadband connections. In both of these cases network address translation (NAT) is commonly used to enable hosts with these addresses to have full 'layer 3' (IP) connectivity.

The second group deals with address space as it transitions from unallocated to allocated, and is discussed in Sections 6.5.2 to 6.5.4. For the purposes of this evaluation, this has only been done at the /8 network level as managed by IANA (Internet Assigned Numbers Authority (IANA), 2008b), and use was made of the 'Bogon List' (Team Cymru, 2008), a copy of which can be found in Appendix C. Again misconfiguration is the most likely cause of traffic to be observed as originating from these network blocks.

6.5.1 RFC1918 Blocks

Of all the Bogon address space, the most packets were received from the 'private use' IP address ranges as defined in RFC1918. By definition (Rekhter *et al.*, 1996) these are private address ranges and as such:

[9] `http://en.wikipedia.org/wiki/Martian_packet`

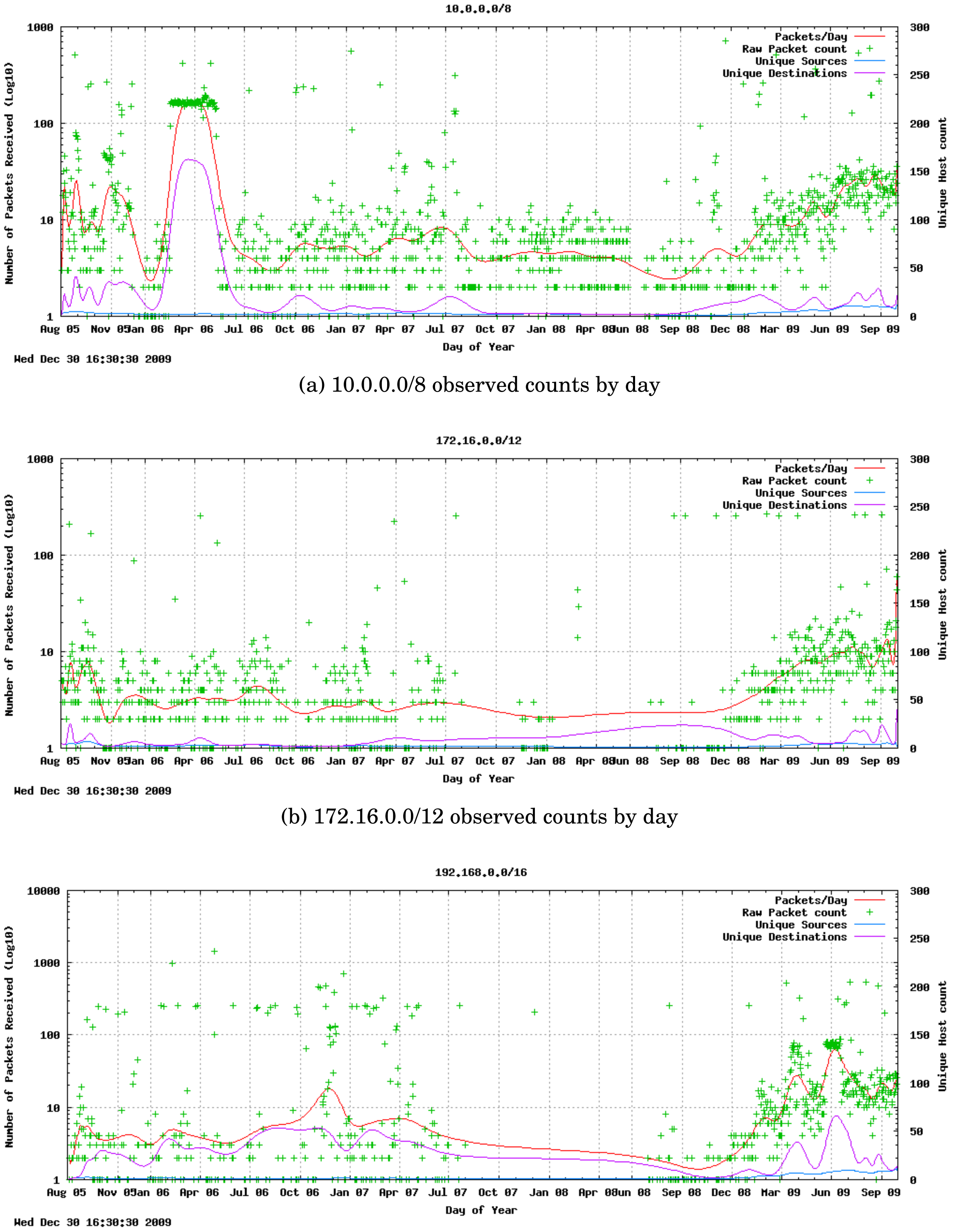
[10] Address blocks allocated to regional registries but not assigned to end-users are not included

> *It is strongly recommended that routers which connect enterprises to external networks are set up with appropriate packet and routing filters at both ends of the link in order to prevent packet and routing information leakage. An enterprise should also filter any private networks from inbound routing information in order to protect itself from ambiguous routing situations which can occur if routes to the private address space point outside the enterprise.* — RFC1918 Section 5.

The most likely source of these packets are misconfigured or malfunctioning Network Address Translation (NAT) gateways. While RFC 2827 (Ferguson and Senie, 2000) discusses ingress filtering, egress filtering has become more prevalent in recent years as organisations try to deal with internal malware infestations from threats such as such as SQL Slammer, Blaster and most recently Conficker. The breakdown of traffic received by protocol is shown in Table 6.7. Despite this traffic accounting for only 0.16% of the total packets received, it is still of interest to the researcher since these packets in particular should not be normally seen outside of private networks.

An overview of the traffic received from these three network blocks over the duration of study, is given in Figure 6.15. Looking at this in more detail, an anomalous spike in traffic is observed in Figure 6.15a originating from the 10.0.0.0/8 network from February to May 2006. During this period 14 377 datagrams were received, the majority (92.8%) originated from 10.12.1.1 targeted at a range of hosts on the monitored netblock on 1434/tcp. Size and payload inspection confirmed these to be SQL Slammer packets, as discussed in Section 5.2.2. Other than this spike, volumes remained fairly consistent, with the same up-sweep in traffic from November 2008, almost exclusively consisting of traffic destined to 445/tcp.

Traffic from 192.168.0.0/24 (Figure 6.15c) and 172.16.0.0/12 (Figure 6.15b), although to a lesser extent, virtually dissipated from late July 2007 through August 2008. The reason for this is unknown. Possible causes are systems being offline. The brief resurgence of traffic seen in January 2008 in Figure 6.15b, could possibly relate to the start of the new academic year at institutions connected to the TENET network, and the introduction of new infected hosts.



(a) 10.0.0.0/8 observed counts by day

(b) 172.16.0.0/12 observed counts by day

(c) 192.168.0.0/16 observed counts by day

Figure 6.15: RFC1918 blocks

Table 6.7: Packet counts by RFC1918 block

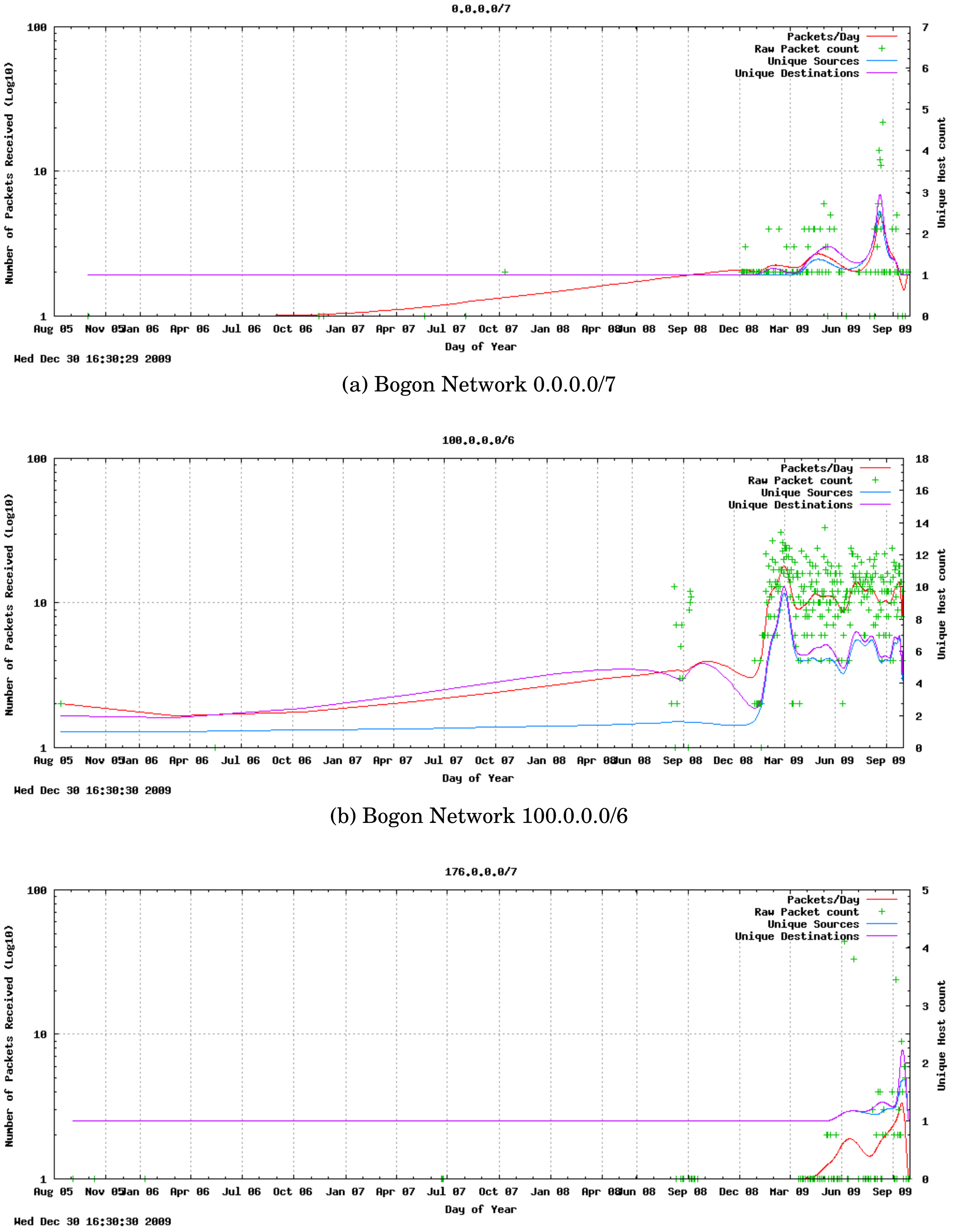
Netblock	Packet Count	TCP	UDP	ICMP
10.0.0.0/8	33337	14984	17450	903
172.16.0.0/12	8365	3354	3565	1446
192.168.0.0/16	25445	19286	5743	415

6.5.2 Bogon blocks

A full list of Bogon blocks current as at the time of writing is contained in Listing 10 in Appendix C, and is based on version 5.0 of the Cymru Bogon List (Team Cymru, 2009), current as of September 2009. The analysis below is based on these blocks. Of the twenty-three netblocks listed (excluding the RFC1918 IP ranges) packets were recorded on all but 3.0.0.0/8, 42.0.0.0/8, 179.0.0.0/8 and the special networks of 192.0.2.0/24 and 198.18.0.0/15 (RFC3330 IANA (2002)). Figure 6.16 shows traffic plots from the three most populous of the Bogon blocks in their aggregated forms: 0.0.0.0/7, 100.0.0.0/6 and 176.0.0.0/6 in sub-figures a,b and c respectively. These are aggregated and hence one would expect more traffic from these, consisting of two contagious /8 networks in the case of 0.0.0.0/7 and four in each for 100.0.0.0/6 and 176.0.0.0/6. It is interesting to note the marked increase in traffic in all three of these since early December 2008.

A closer look at the traffic shows it to be almost exclusively 'active' traffic destined to 445/tcp with the SYN flag set, and hence attempting to establish a connection. This is most likely automated scanning and propagation from misconfigured systems by the Conficker malware. The other interesting anomaly observed was a flood of fragmented datagrams such as those shown in Figure 6.17. These were received from hosts in 176.0.0.0/7 in June 2009. No redaction has been carried out on the packet other than the three bytes (0x1f-0x21) indicated in bold. One probable source of these packets is datagrams directed to port 80, and containing the payload of the Code Red II worm[11], which used a long flood of 'X' characters passed as a parameter in order to overflow the buffer (eEye Digital Security, 2001a), although it could also be a case of the 'X' just being used as padding on some other attack. The original Code Red worm used a sequence of 'N' characters (eEye Digital Security, 2001b; CERT, 2001). However lacking the first fragment of the data-stream, it is impossible to tell where these packets were originally targeted.

[11] http://en.wikipedia.org/wiki/Code_Red_II



(a) Bogon Network 0.0.0.0/7

(b) Bogon Network 100.0.0.0/6

(c) Bogon Network 176.0.0.0/7

Figure 6.16: Observed counts by day on Bogon netblocks

The fact that TCP datagrams containing payloads were observed is anomalous in itself.

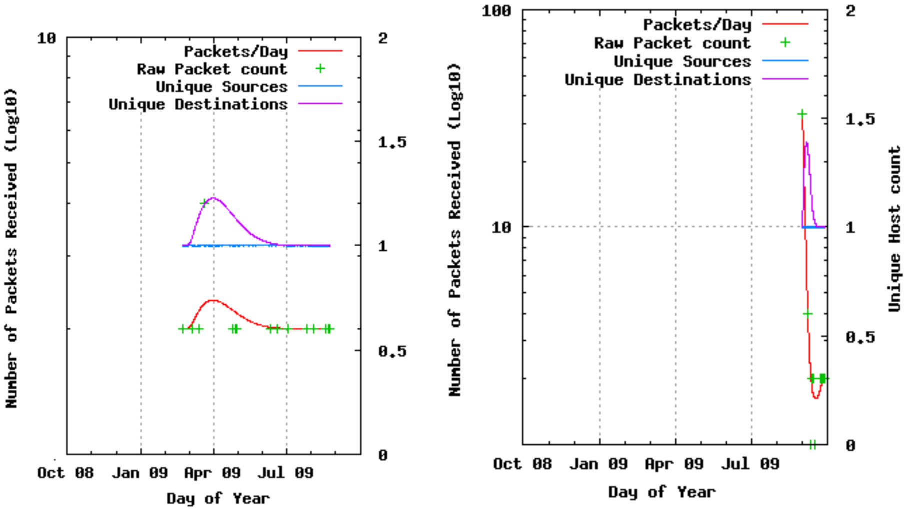
```
0000 00 03 ba 5c ae 0b 00 1d a2 9a 95 f5 08 00 45 00 ...\....  ......E.
0010 05 dc 00 3b 20 00 33 06 69 43 b0 4d f1 58 c4 XX ...; .3.  iC.M.X..
 0020 XX XX 29 b2 00 00 16 30 7b 0d 02 e3 10 72 50 00 ..)....0 {....rP.
 0030 02 00 a9 ff 00 00 58 58 58 58 58 58 58 58 58 58 ......XX XXXXXXXX
 0040 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX
 0050 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX
 0060 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX
 0070 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX
 0080 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX
 0090 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX
                               ....
 05c0 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX
 05d0 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XXXXXXXX
            05e0 58 58 58 58 58 58 58 58 58 58 XXXXXXXX XX
```

Figure 6.17: Hex dump of strange fragmented packets received

6.5.3 Allocated blocks

Over time IP blocks come into and out of use, though their allocation or reclamation by IANA. Of particular interest in this study are 14.0.0.0/8 and 46.0.0.0/8 which were retired from use in January 2008 and April 2007 respectively. In September 2009 46.0.0.0/8 was returned to use and allocated to the European Regional Registry (RIPE NCC). A summary of the IP block allocations during the period of research is shown in Section C.5. The analysis performed in this section tracks the appearance of traffic from these and other address blocks both before and post official allocation by IANA to a RIR. Metrics of time to sight post allocation allow for the measurement of the rate of allocation of usable address space to endusers by the RIR responsible for administering the particular block.

Observation of packets prior to allocation can be due to organisations pre-emptively configuring devices, or even test equipment for routing being preconfigured, prior to the official allocation. An example of this is the 2.0.0.0/8 block allocated in September 2009, at the end of the RUSCOPE1 capture period. Figure 6.18 provides an interesting example of this, with activity observed from February of 2009, some months prior to the official allocation of the block to RIPE by IANA. Packets observed were all destined to 445/tcp with the exception of a burst of fragmented



(a) Pre allocation

(b) Post allocation 09/2009

Figure 6.18: Traffic observed with an origin of 2.0.0.0/8

packets of 1514 bytes in size in a one second burst at 20:19:37 on the first of September 2009. The contents of these packets was identical to those discussed in the previous section, and illustrated in Figure 6.17.

A number of examples of networks being allocated to RIRs and appearing soon after are also apparent. Some spoofing may be present — again the lack of a three-way handshake or active back-path probe limit the ability to determine this definitively, although in the future, live probes of global BGP databases could be used to differentiate between spoofed ranges and those actually published.

6.5.4 Back chatter

An analysis of Bogon traffic would not be complete with taking into consideration traffic with source addresses claiming to be from within the monitored network telescope IP range. Ten packets matching this criteria were observed, arriving in two batches in March and May of 2009. Each of these consisted of a single burst of five datagrams directed at the broadcast address of the monitored netblock. A summary of these is shown in Table 6.8. An examination of the traffic in WireShark, reports the datagrams to be making use of the Messenger Service Remote Protocol (Microsoft, 2007) used for sending messages between Windows Family systems. These packets are of the same type as discussed in section as seen in Section 5.3.3.

Table 6.8: Traffic from the RUSCOPE1 address space

Date	Source	Destination	Protocol	SPort	DPort	Size	TTL
2009-03-29 17:07:44	196.x.x.254	196.x.x.255	UDP	0	1025	684	110
2009-03-29 17:07:44	196.x.x.254	196.x.x.255	UDP	0	1026	684	110
2009-03-29 17:07:44	196.x.x.254	196.x.x.255	UDP	0	1027	684	110
2009-03-29 17:07:44	196.x.x.254	196.x.x.255	UDP	0	1028	684	110
2009-03-29 17:07:44	196.x.x.254	196.x.x.255	UDP	0	1029	684	110
2009-05-18 16:40:51	196.x.x.254	196.x.x.255	UDP	0	1025	684	111
2009-05-18 16:40:51	196.x.x.254	196.x.x.255	UDP	0	1026	684	111
2009-05-18 16:40:51	196.x.x.254	196.x.x.255	UDP	0	1027	684	111
2009-05-18 16:40:51	196.x.x.254	196.x.x.255	UDP	0	1028	684	111
2009-05-18 16:40:51	196.x.x.254	196.x.x.255	UDP	0	1029	684	111

The actual content of the datagrams is fairly indecipherable, despite the DCE/RPC[12] protocol indicating ASCII type encoding. Two strings that can be extracted are: “http://www.subian.net” and “$$$ Powered by www.NetGuarder.com.”. Current domain registration records for the subian.net domain[13] indicate it was only registered on 2009-12-20 by ZhuKongXian in ShanDong China. Netguarder.com[14] on the other hand has been registered since 2002-07-15 by HiChina Web Solutions (Hong Kong) Limited. The URL referred to appears to be a legitimate vendor of mass advertising and messaging software. What can be surmised from these datagrams is that Netguarder.com’s software was used on a misconfigured system. Another possibility, given that the traffic us UDP based and hence requires no connection, is that forged source addressing was used in order to try and circumvent firewall rules on target networks. The targeting of the broadcast address 196.x.x.255 would likely result in the incoming datagrams reaching all hosts on the subnet, thus maximising any impact that they may have. Based on the TTL values the system is most likely 17 or 18 hops away from the telescope sensor, given that there is a strong likelihood that the originating system was running Microsoft Windows and thus had a default TTL of 128.

[12] `http://en.wikipedia.org/wiki/DCE/RPC`

[13] Checked 2010-03-12

[14] Checked 2010-03-12

6.6 Summary

This Chapter dealt with the analysis of the collected data, focussing on the meta-information that can be determined about components of data contained in the captured datagrams. Various components of the collected datagrams were analysed, and several short illustrative case studies presented. The Distance Score analysis was introduced as a potential means for assessing the bias that a telescope may have due to its placement both geographically and topologically. A high level temporal overview of the data was also presented, along with an analysis of the geopolitical origins of traffic. The origins of traffic were again assessed from an organisational and topological view. Finally a discussion of Bogon traffic was presented. The following chapter presents a case study of the Conficker/DownAdUp worm as observed by the Rhodes University network telescope and recorded within the RUSCOPE1 dataset. Techniques as described in Chapter 4 and applied in Chapter 5 and this chapter are applied to a subset of traffic collected. In addition, a number of metrics as described in Chapter 8 are applied to the filtered data to aid in the explanation and illustration of the case study.

In 2006, the attackers want to pay the rent. They don't want to write a worm that destroys your hardware. They want to assimilate your computers and use them to make money.

Mike Danseglio - program manager Microsoft Security
4 April 2006

7
Case Study: 445/tcp

THIS chapter explores the application and value of the use of a network telescope in the tracking and monitoring of a global malware outbreak. As shown in Chapters 5 and 6, the volume of traffic observed arriving at the research telescope destined for port 445/tcp grew dramatically over the last 14 months of the RUSCOPE1 dataset, reaching a peak nearly two orders of magnitude higher than the previously observed baseline traffic. Much of this can be like attributed to the prevalence of the Conficker worm (Microsoft, 2008b), also known as Kido and DownAdUp (Microsoft, 2009).

This malware exploits a vulnerability in the Microsoft RPC stack detailed in the Microsoft MS08-067 (Microsoft, 2008a) security bulletin released on 23^{rd} October 2008. The vulnerability exploited is similar to those discovered in July 2003 detailed in MS03-026 (Microsoft, 2003a) and later in MS03-039 (Microsoft, 2003b) — and subsequently exploited by the Blaster[1] and Welchia[2] worms in August of that year (Microsoft, 2003c). A further vulnerability in the RPC stack was exploited by

[1] http://en.wikipedia.org/wiki/Blaster_worm

[2] http://en.wikipedia.org/wiki/Welchia

Sasser in April 2004, taking advantage of the vulnerability disclosed in MS04-011 some seventeen days previously (Microsoft, 2004). The problems with the RPC/DCOM stack in Microsoft Windows Family operating systems continued and MS06-40 released in September 2006 Microsoft (2006), patched a further vulnerability that was exploited by various malware such as Mocbot. Given this history of vulnerability, and the widespread adoption of the Windows operating platform and the rapid devlopment of code expoiting these vulnerabilities, researchers were justifiably concerned when the MS08-067 vulnerability was announced.

A detailed analysis of the Conficker malware is beyond the scope of this research. For details on the actual origins, and analysis from a payload and reverse engineering perspective, readers are encouraged to consult in particular the work done by SRI (Porras *et al.*, 2009) and Symantec (Nahorney, 2009) on reverse engineering and documenting the spread. The website maintained by the Conficker Working Group[3], a body constituted by members of the information security industry, is also a valuable reference, particularly for its ongoing monitoring of the outbreak as is the Conficker watch site run by the Shadowserver Foundation (ShadowServer, 2010).

What is important to bear in mind when analysing the data collected using the Rhodes University system, is that one of the shortcomings of the current network telescope setup is that only the first packet of the potential TCP 3-way handshake is actually captured. Since the handshake cannot complete, no data payload can be captured. Due to this limitation it can only be inferred, albeit with a high level of certainty, that the increase in observed traffic is directly related to the Conficker malware. It is believed that the majority of the recorded connection attempts are automated connections from Conficker, but there is certainly a component which is scanning activity from other sources looking for operational targets which may also be vulnerable to the MS08-067 issue. Evidence for the strong likelihood of the majority of this traffic being Conficker related is presented in Sections 7.3 and 7.4, where the actual scanning observed is analysed. Some of the initial research relating to Conficker observations in the RUSCOPE dataset was presented in Irwin (2010).

This chapter presents a discussion of how the spread of this malware was observed from the perspective of the RUSCOPE system. An overview of the evolution of the worm is presented along with a time-line of the major points in the evolution of this

[3] `http://www.confickerworkinggroup.org/`

software in Section 7.1. This is shown to match fairly accurately with the observed changes in traffic presented in 7.2. An analysis of the traffic is presented in Section 7.3, with a focus of on the traffic distribution across the target addresses in Section 7.4. Section 7.6 presents an analysis of the change in geopolitical origins of the traffic over time. The chapter concludes with a reflection on the application of a network telescope to the monitoring of this kind of event, and the views of traffic as observed by other sensors.

7.1 Conficker Evolutionary Time-line

The evolution of the threat posed by the Conficker malware can be traced back to the release of the MS08-067 advisory on October 23^{rd}, 2008. Described as "*Vulnerability in Server Service Could Allow Remote Code Execution*" (Microsoft, 2008a), this critical bulletin was released as an emergency out of sequence patch by Microsoft after exploitation of the vulnerability was observed in the wild. An abridged summary of the evolutionary time-line for Conficker is shown in Table 7.1. This has been taken largely from the time-line[4] maintained by the Conficker Working Group (CWG).

One of the issues to be aware of when analysing Conficker and research around the threats, relates to the two different naming conventions used by Microsoft, and the Conficker Working Group. The former appears to be in more widespread use. These differences are shown Table 7.2. In this document the Microsoft naming conventions are used. When analysing the traffic, inflexion points can be seen relating to the version changes in the Conficker malware, as seen in Section 7.2.2.

7.2 Telescope traffic observations

Traffic destined to 445/tcp makes for an interesting case study on a number of fronts. Firstly, it is the single most significant contributor to the total, both in terms of the number of packets and source addresses observed. Secondly, it is used by the Microsoft Windows family of operating systems for RPC/DCOM communications, including file sharing, and is usually enabled on such systems. The

[4] `http://www.confickerworkinggroup.org/wiki/pmwiki.php/ANY/Timeline`

Table 7.1: Major Conficker Evolutionary Events

Date	**Event**	**Comment**
	2008	
20 August	Gimmiv Trojan first spotted in South Korea	
mid September	Chinese malware brokers are spotted selling a $37 tool	
29 September	Gimmiv seen in the wild (Vietnam)	Mistakes limit its ability spread
23 October	MS08-067 advisory released	Out of order patch in release
26 October	$37 malware kit starts being given away for free	Related malware exploiting MS08-067
mid October - early November	Gimmiv attacks unfold against unpatched PCs in Asia	Researchers worry about a Worm
20/21 November	Conficker.A launches	
22 November	Microsoft security alert	
late November	Conficker A census: 500,000 infected machines	
1 December	Conficker moniker coined due to trafficconverter.biz domain	Previously known as DownAdUp
24-27 December	Conficker A census: 1.5 million infected machines	
28 December	Conficker.B launches	38 days post A variant
	2009	
1 January	Conficker B payload activation	
mid January- Early February	Estimates of A & B variants 3-12 million infected	
12 February	Microsoft offers USD 250 000 Reward	Still unclaimed
20 February	Conficker.C Variant emerges	53 days post B variant
4 March	B & C variants upgrades to Conficker.D	12 days post C variant
31 March	IBM reverses the p2p client	
1 April	Conficker.C activates	Begins checking domains
7 April	Conficker.E seeded to p2p net	
3 April	'E' variant set to deactivate	
8 April	An update begins spreading via P2P to Conficker C machines	Waledav AntiAV/ and Spam engine

Sourced largely from `http://www.confickerworkinggroup.org/wiki/pmwiki.php/ANY/Timeline`

Table 7.2: Conficker Naming

Date	Microsoft	CWG[a]
20 Nov 2008	Conficker.A	Conficker.A
28 Dec 2008	Conficker.B	Conficker.B
20 Feb 2009	Conficker.C	Conficker.B++
4 Mar 2009	Conficker.D	Conficker.C
8 Apr 2009	Conficker.E	

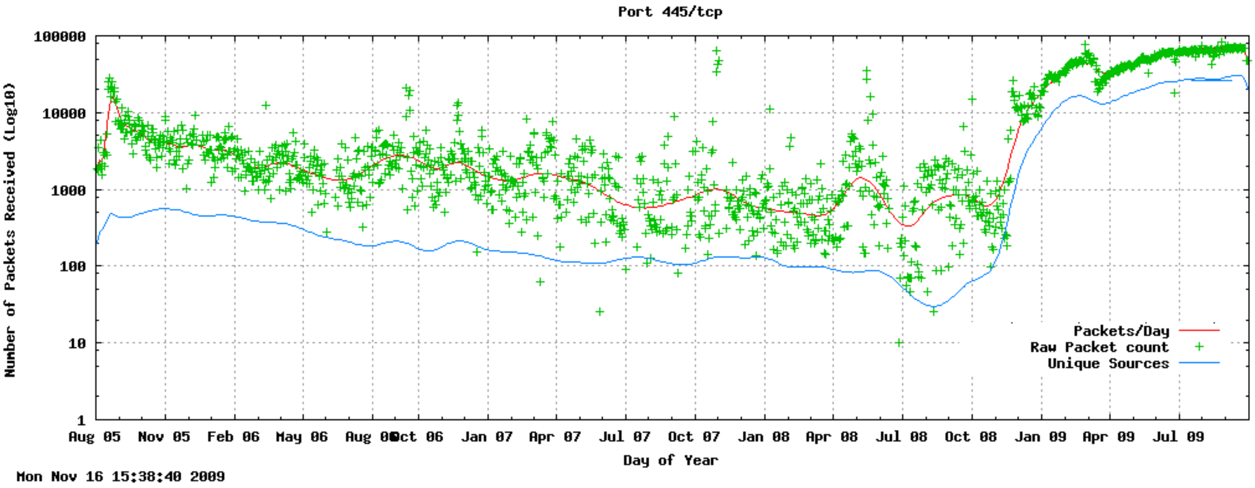
[a] Conficker Working Group

popularity of the deployment of these systems make this an inviting target when vulnerabilities are found. Several significant vulnerabilites have been found in the RPC/DCOM stack, as discussed in the introduction to this Chapter. There has also been wirespread exploitation of these vulnerabilities. Furthermore, this port is generally firewalled by most organisations, and often by home users as well, although ususally for inbound traffic only. The evidence points to the fact that this is a popular port for which scanning attempts can be recorded on a network telescope.

7.2.1 Overview

Traffic destined to port 445/tcp as a whole, can be seen to be fairly persistent over the entire duration of the network telescope observation, being observed on all but one of the 1 429 days having data (and 98.1% of hourly observations) within the dataset. Over the period it was consistently ranked in the top ten ports observed, by both month and year. During the observation period, packet counts for traffic destined to port 445/tcp was the top ranked in 10 of the 17 quarters under study, with its lowest positions being 4th in Q1 2007 and Q4 2008. As previously discussed in Section 5.2.1 it also accounted for 41.4% of the traffic overall. Figure 7.1 shows the prevalence of this traffic over the observation period. Data shown in this figure reflects only that TCP traffic destined to 445/tcp that has the SYN flag set, and hence can be considered ‘active’. The rapid increase in traffic from approximately 1 000 packets/day (ppd) in October 2008 through to nearly 100 000 ppd by the end of September 2009 can be clearly seen.

Looking at the data from a different point of view, Figure 7.2 provides a heatmap plot of traffic destined to 445/tcp on a weekly basis over the observation period of the RUSCOPE1 dataset. Values range from under 50 000 packets per week over



All packets can be considered as *active* as the SYN flag was set.

Figure 7.1: TCP packet received on port 445 by day.

most of the period to nearly half a million a week in August 2009. As mentioned in Sections 4.7 and 6.2, the use of heatmaps can provide a visual overview that is easy to interpret.

The spike in observed activity the in early portion of the graphs (most notable in Figure 7.1) is most probably attributable to the Zotob worm (Schneier, 2005) exploiting a vulnerability disclosed on August 9th 2005 in MS05-039 (Microsoft, 2005; White, 2005), or related scanning in response to this event by individuals looking for vulnerable hosts. Traffic levels had, however, decreased and largely normalised by November 2005 and continued to drop though to mid October 2008. This gradual decrease is likely due to the increased uptake of automated patching of systems though the Windows update mechanisim, the release of Service Pack 3 for Windows XP (April 2008) and Service Packs 1 and 2 for Windows Vista (March 2008 and April 2009) resulting in the remediation of the vulnerability. More significantly the lack of any significant vulnerabilities affecting this protocol durring this period, would have reduced the incidence of scanning. The rapid increase in traffic observed from this point onwards can be attributed to activities surrounding the exploitation of the MS08-067 Vulnerability in Microsoft Windows operating systems, most notably by the Conficker worm. The remainder of this chapter focusses on this activity.

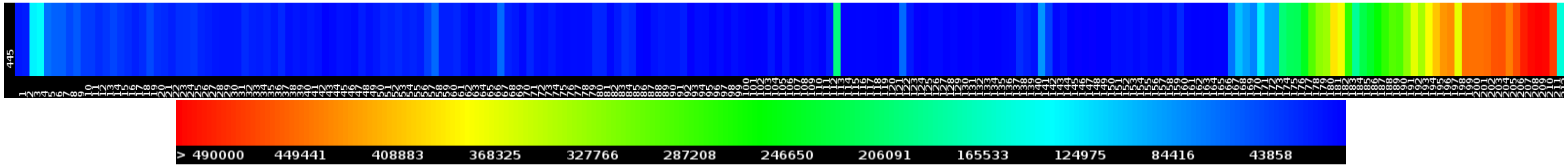


Figure 7.2: Heatmap of 445/tcp

7.2.2 Conficker Related

The Conficker worm was first observed on the $20^{th}/21^{st}$ of November 2008 (depending on time zone), and after almost a year, over 7 million infected nodes were observed as still infected[5]. After the 20^{th} of November, traffic destined to 445/tcp constituted 70% of traffic recorded. Over the entire observation period, 4 002 119 unique hosts (86% of the total) were observed sending packets to a destination port of 445/tcp. Of these addresses seen originating traffic to 445/tcp, 3 809 160 (95%) were observed after the 20^{th} of November 2008. Of those IP addresses observed after the 20^{th} of November targeting 445/tcp, only 5 544 (0.14%) had been seen prior to the 1^{st} of November sending traffic to 445/tcp, and 11 630 (0.3%) having sent any traffic at all.

This is not the first work to have been done on looking at Conficker from the perspective of a network telescope, with some detailed analysis having been performed by CAIDA researchers in Aben (2009), and the subsequent release of a portion of their *3 Days of Conficker* dataset (Hick *et al.*, 2009). What is novel in this analysis is the fine level of detail at which it has been performed along with the use of data from a single /24 network telescope, rather than the aggregated level previously reported utilising the CAIDA /8 telescope, and a further dataset gathered on a /16 netblock. The discussion relates to the Conficker related traffic, considered traffic from Mid October 2008 through to the end of the dataset at the end of September 2009, in effect covering nearly a year of activity related to the MS08-067 vulnerability.

An overview of the total traffic observed by the telescope system is shown in Figure 7.3 as the calculated average number of packets received by each of the IP addresses in the monitored range. A number of distinct spikes in the traffic are noticeable, along with the general increase in traffic over time. The increase is, however, not nearly as rapid as that observable in the latter part of Figure 7.1. Particularly notable events are the large spike on 28^{th} October 2008, followed by a rapid climb on the 21^{st} November 2008. A second rapid increase can be seen on 1^{st} January 2009, with a consistent increase in traffic rates observed though to mid February, and a large increase in activity on the 218^{th} . This is followed by a sharp drop-off mid March and a small spike prior to 1^{st} April. From this point the traffic continues to increase, other than two dips which were caused by network outages.

[5] http://www.shadowserver.org/wiki/pmwiki.php/Stats/Conficker

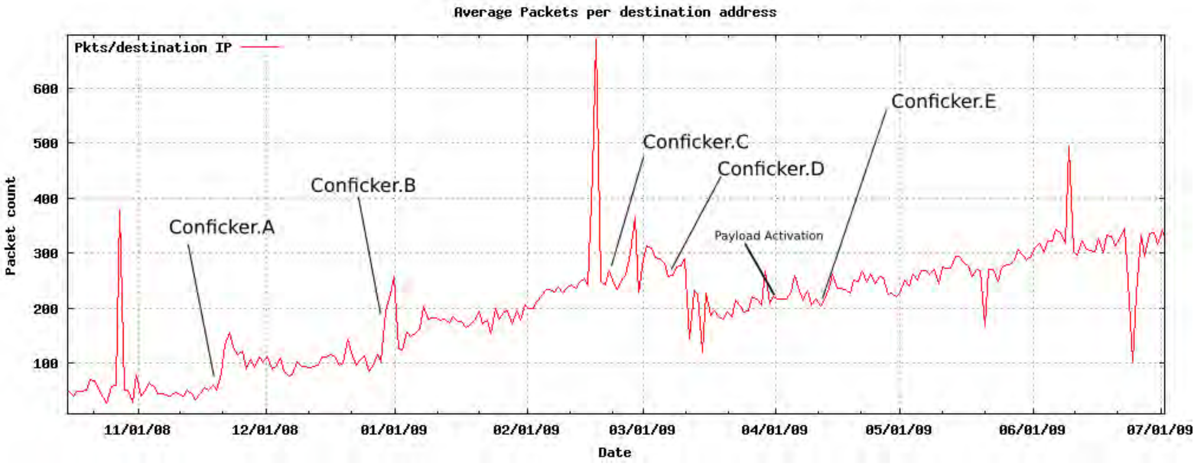


Figure 7.3: Traffic November 2008 — September 2009

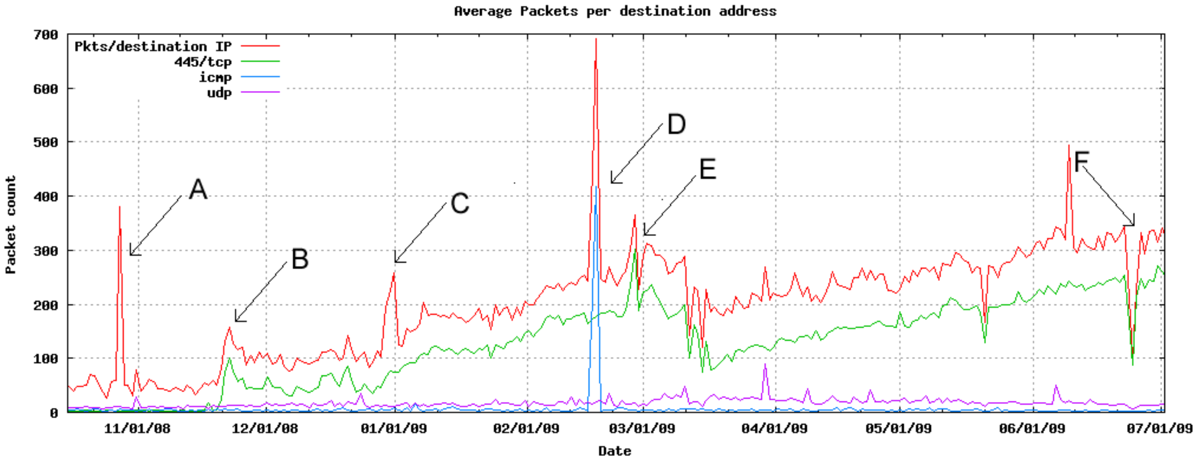


Figure 7.4: Protocol breakdown (November 2008 — September 2009)

At first glance, these dates seem to relate remarkably well to those outlined in the timelines in Tables 7.1 and 7.2.

Looking a little deeper, and analysing the composition of the traffic down by protocol, one can see that the spikes observed cannot be correlated directly to activity on port 445/tcp. This detailed breakdown of the same dataset and time period as previously shown in Figure 7.3 can be seen in Figure 7.4. In the detailed plot, ICMP and UDP traffic have been shown along with the contribution made by traffic destined to 445/tcp on the sensor network. The total traffic shown is the same as in Figure 7.3. Several points in Figure 7.4 are worth highlighting:

- Although the spike shown at point A ties in with the release of the MS08-067 security bulletin, it is not related to it, but rather is the result of a burst of 87

818 SYN-ACK TCP packets originating from 80/tcp and destined to a range of 46 monitored addresses, with varying ephemeral destination TCP ports. These classic backscatter packets originated from a unix system located in Jordan.

- The ‘birth’ of Conficker on the 21^{st} of November 2008 (point B) can be seen by the sharp rise in 445/tcp traffic as a portion of the whole.
- Conficker.B was released on the 28^{th} of December 2008 and although there is a spike in total traffic (point C) this was not due to Conficker, but again a reflected flood of 74 074 SYN-ACK packets from a web server located in Costa Rica. While p0f was unable to identify the host’s operating system, it was most likely a unix-type platform, given the low TTL.
- Point D indicates a further anomaly in the traffic pattern, with the spike caused by a flood of 159 000 ICMP TTL expired messages, received from a host in China on the 17^{th} and 18^{th} of February 2009 (this incident is discussed in detail in Section 5.3.4).
- The spike in late February 2009 (point E) can be attributed to the release of Conficker.C on the 20th.
- Point F is worth noting in that the drop in recorded traffic was due to a series of extended network outages adversely affecting Rhodes University’s upstream Internet connection over the period 24^{th}-26^{th} June 2009.

7.2.3 Conficker Outbreak

Analysis of the data in the first few days of the Conficker outbreak revealed some interesting trends. The first of these is illustrated in Figure 7.5, which plots data relating to traffic received on 445/tcp during the period $21^{th} - 24^{th}$ November 2008. Times noted in the Figure are SAST (GMT+2:00). What is immediately noticeable is that while the packets follow a rough circadian rhythm, this trend is even more noticeable when the number of distinct sources for each hour interval are plotted. Similar results were found with the data processed in the CAIDA Conficker report (Aben, 2009). Considering the 24 hour period from midnight on the 21^{st} November, the number of observed sources per hour can be seen to climb rapidly, from fewer than ten at 05h00 to over 250 by midnight the following day. What is interesting

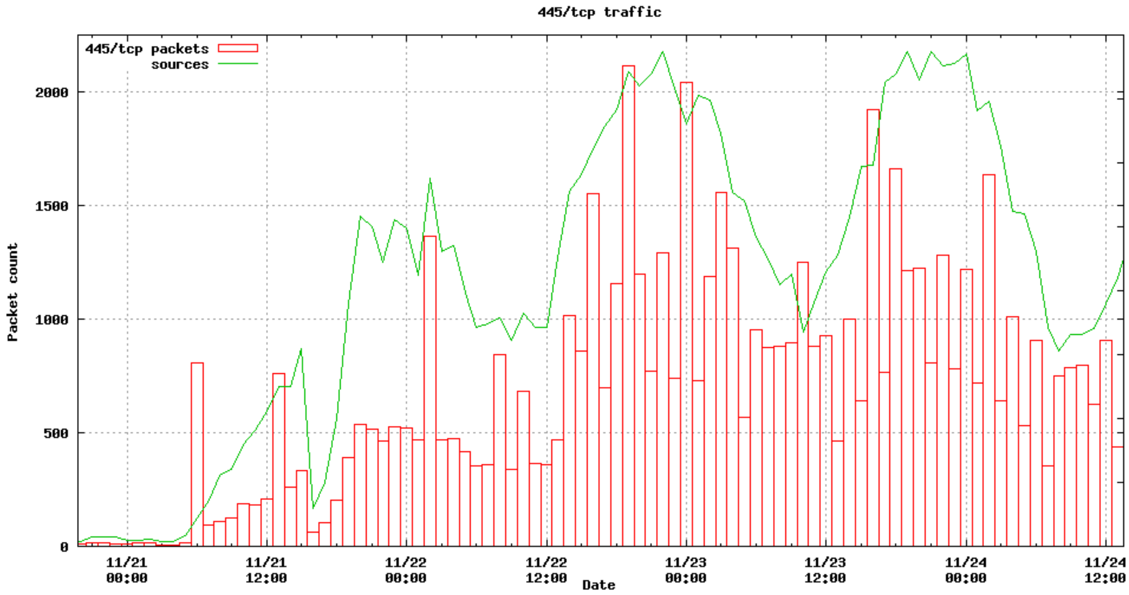


Figure 7.5: Early days of Conficker: 21—24 November 2008

is that there is a large increase in packets observed around 06h00, yet only twenty source hosts had been observed at this stage. From this point the packet rate per hour dropped dramatically and the host count started to climb — in essence, more hosts were sending relatively fewer datagrams. The origins of these hosts is discussed in more detail in Section 7.6. By 16h00 traffic had reached a low point and subsequently started to increase again, with a rapid growth in the number of sources observed; a high sustained rate being maintained for nearly ten hours before dropping back. This is pattern can be seen to be repeated over the next two days.

Delving somewhat further back in the data, an interesting anomaly was found in the data for 445/tcp. A spike in scanning activity was detected over a period from late on the 30^{th} September, through to the evening of the 1^{st} of October 2009. This increase in scanning was particularly noticeable, due to there being almost no traffic targeting 445/tcp in the days leading up to this, and very little afterwards. A plot of the relevant traffic is presented in Figure 7.6. The traffic is also seen to originate from a relatively high number of sources, with 3 476 IP addresses being logged between 03h00 and 18h00 on the 1^{st} of October, having sent some 14 808 packets targeting 445/tcp. Top geopolitical sources as determined by geolocation tools were Brazil (BR), France (FR), the USA (US), Japan (JP) and the Philippines (PH), together accounting for nearly half (46%) of the sources, a summary of which is shown in Table 7.3. Sources were observed from 100 countries, although only 30

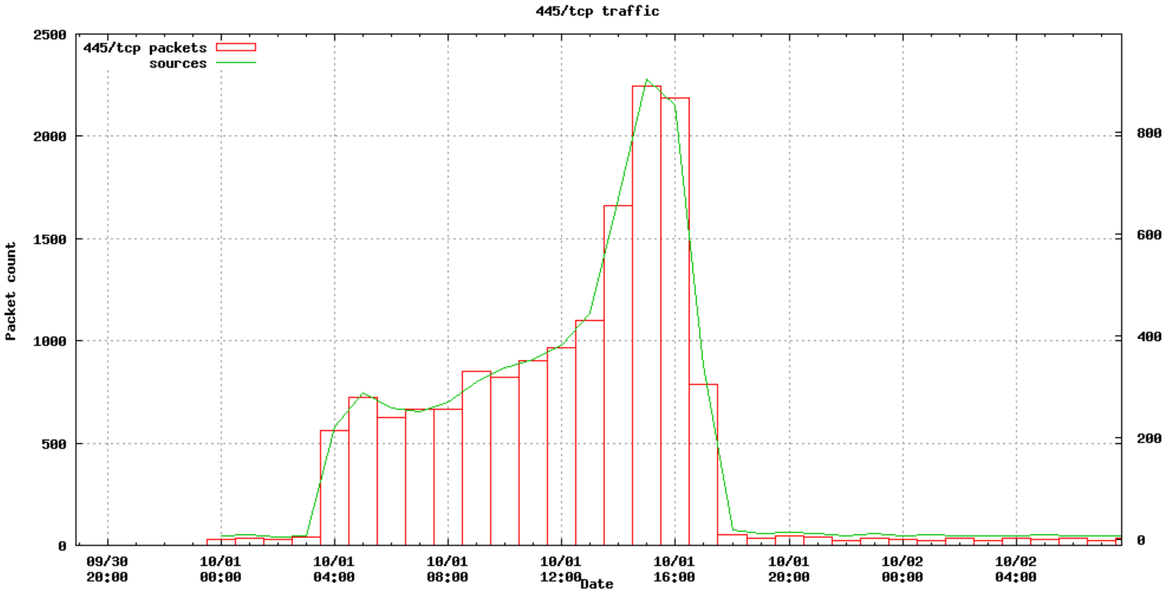


Figure 7.6: Early reconnaissance: 30 September — 2 October 2008

Table 7.3: Top 5 countries - 1 October 2009

Rank	CC	$Count_{Packets}$	$Count_{Source}$	$Count_{Dest}$
1	BR	2 203	625	249
2	FR	1 551	388	231
3	US	1 877	279	249
4	JP	654	167	177
5	PH	573	155	157
Total		6 858	1 614	

of these had more than twenty sources.

Without payloads, it is impossible to determine the exact nature of this traffic, or the means of its generation. It could be plausibly be activity from the Gimmiv trojan, although it was noted that there were problems with its replication mechanisms. Another possibility is that it could be some custom malware utilising the Chinese exploit kit. It is nontheless interesting to note that there is a fairly even distribution across the entire monitored range, in contrast to what is seen in Section 7.4. The majority of the sources (99%) scanned less than ten target addresses, with 1832 (52%) only probing one host. Only three hosts scanned more than 100 addresses, while the majority of sources sent two packets were in relatively quick succession, before disappearing.

It is the researcher's hypothesis that there is a strong likelyhood of this having been a distributed scanning attempt, with multiple sources scanning to look for vulnerable hosts possibly for further targeted exploitation, or as a means of seeding initial distribution points for later malware release. The fact that such a high number of hosts only probed a single target points to a well co-ordinated, distributed scan, or these addresses possibly being used as a decoy scan. The three main hosts located can be determined to be in Taiwan (TW) and the USA, and are most likely the real hosts. This is supported by the fact that hosts that are geolocated as originating from the Phillipines (PH), Croatia (HR), Turkey (TR) and Austria (AT) all have TTL values above 240. This is highly unlikely given Rhodes University's Internet connectivity which had, at the time of collection, at least ten hops to get to international peering points in London. Further examination of the TTL values shows that a significant number of the hosts have the same TTL values despite being geolocated to vastly different parts of the globe, further strengthening the likelihood of packet forgery. The fact that only two packets are sent is also interesting as generally most TCP/IP stacks send three SYN connection attempts before timing out. This could point to the fact that custom code was being used with a short timeout, or that packets were being constructed using 'raw' sockets.

7.3 Packet Data

This section evaluates aspects of the packets received on 445/tcp by the network telescope, considering the observed TTL, packet structure, packet retransmission, and source operating system.

7.3.1 Time to Live

An analysis of the TTL values recorded for all incoming traffic destined to 445/tcp showed a very narrow banding where it was observed that the values were, with few exceptions, between 50 and 100. This range covers default TTL settings for both Windows and unix platforms as discussed in Section 5.4.2. This banding is further evident when plotted against packet counts for TTL values received overall as presented in Figure 7.7. In this Figure the TTL values for 445/tcp packets, can be seen to be largely grouped in the 96-128 range, with very few packets recorded in the 32-64 and 224-255 ranges. This again provides strong evidence towards

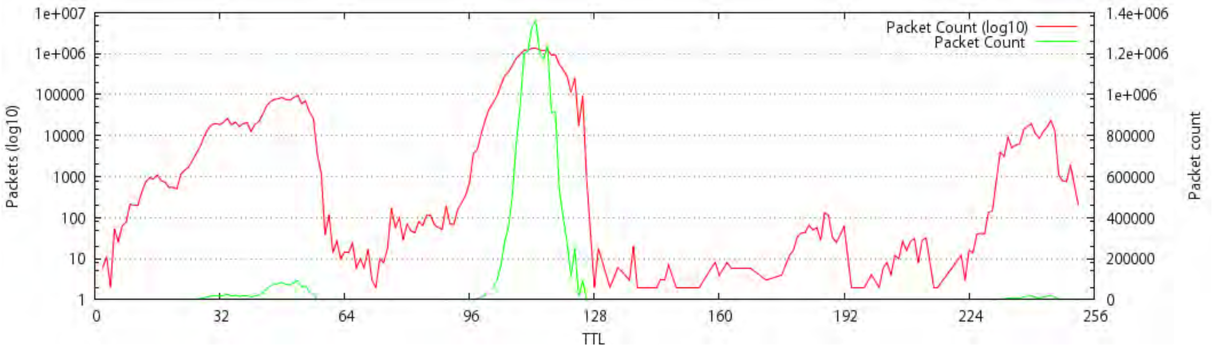


Figure 7.7: Packet counts by TTL for 445/tcp

Microsoft Windows platforms being the origin of the packets, as confirmed by the passive operating system fingerprinting that was performed, and is discussed in Section 5.4.2. This would in turn lend weight to the supposition that when considering the number of distinct sources, the packets observed were actually generated by the automated scanning modes of the Conficker worm.

7.3.2 Packet Structure

Based on an analysis of of the datagrams received, the majority were found to be of 62 bytes in size. A summary of packet sizes is given in Table 7.4, with a plot of all observed sizes and their relative frequency in Figure 7.8. In this Table, it can be seen that more than 85% of both the hosts and packets match this sizing. This is reached by having a TCP packet with no data and encapsulated inside an IP and finally an Ethernet datagram. In order to reach the value of 62 bytes, rather than the default of 60, TCP options are set. The combination found set most often was “MSS:1460, NOP NOP TCP SACK:TRUE”, which accounted for 8 bytes. These options enabled the Selective Acknowledgement (SACK) on the connection being established, along with a maximum sent size of 1 460 bytes. A sample WireShark decode of such a packet is presented in Figure 7.9. These settings were found to match captured Conficker propagation traffic. There is a fairly high probability that the TCP SYN packets being sent to addresses on the telescope actually originated from the Conficker Malware.

This provides an example of how, despite being somewhat handicapped by the lack of payloads in a network telescope dataset, comparative data from honey-net or other systems with higher levels of interaction can be used to augment the anal-

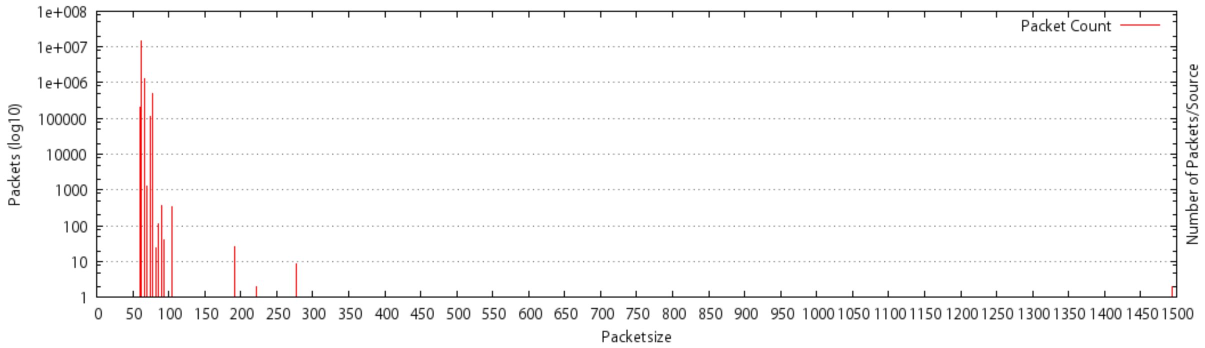
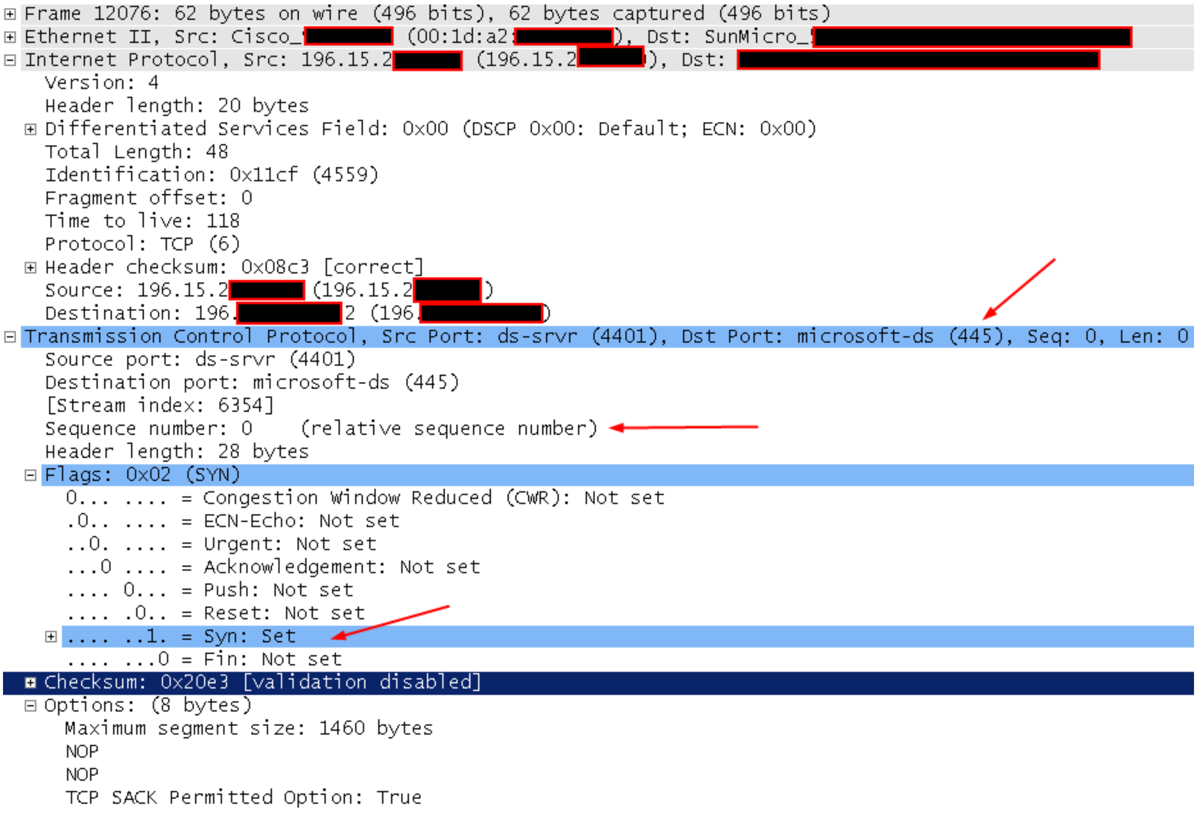


Figure 7.8: Recorded Packet size distribution for 445/tcp



```
⊞ Frame 12076: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
⊞ Ethernet II, Src: Cisco_[redacted] (00:1d:a2:[redacted]), Dst: SunMicro_[redacted]
⊟ Internet Protocol, Src: 196.15.2[redacted] (196.15.2[redacted]), Dst: [redacted]
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 48
    Identification: 0x11cf (4559)
    Fragment offset: 0
    Time to live: 118
    Protocol: TCP (6)
  ⊞ Header checksum: 0x08c3 [correct]
    Source: 196.15.2[redacted] (196.15.2[redacted])
    Destination: 196.[redacted]2 (196.[redacted])
⊟ Transmission Control Protocol, Src Port: ds-srvr (4401), Dst Port: microsoft-ds (445), Seq: 0, Len: 0
    Source port: ds-srvr (4401)
    Destination port: microsoft-ds (445)
    [Stream index: 6354]
    Sequence number: 0    (relative sequence number)
    Header length: 28 bytes
  ⊟ Flags: 0x02 (SYN)
      0... .... = Congestion Window Reduced (CWR): Not set
      .0.. .... = ECN-Echo: Not set
      ..0. .... = Urgent: Not set
      ...0 .... = Acknowledgement: Not set
      .... 0... = Push: Not set
      .... .0.. = Reset: Not set
    ⊞ .... ..1. = Syn: Set
      .... ...0 = Fin: Not set
  ⊞ Checksum: 0x20e3 [validation disabled]
  ⊟ Options: (8 bytes)
      Maximum segment size: 1460 bytes
      NOP
      NOP
      TCP SACK Permitted Option: True
```

Figure 7.9: Sample 445/tcp datagram

Table 7.4: 445/tcp sizes

Size	$Count_{Packet}$	Packet %	$Count_{Source}$	Source %
62	12 207 669	86.18	3 248 492	85.16
66	1 281 603	9.04	443 325	11.62
78	423 043	2.98	153 076	4.01
60	150 549	1.06	25 211	0.66
74	100 043	0.70	31 278	0.81
	Total	99.96		102.26[a]

$N_{Packets} = 14\,165\,097$ $N_{Sources} = 3\,814\,447$

[a] This adds up to > 100% since hosts may have sent packets of different sizes

```
1 01:16:21.685360 IP 190.50.x.80.2725 > 196.x.y.3.445:  S 2062323770:2062323770(0)
2 01:16:23.509228 IP  77.28.x.55.4853 > 196.x.y.34.445:  S 1323192692:1323192692(0)
3 01:16:24.677814 IP 190.50.x.80.2725 > 196.x.y.3.445:  S 2062323770:2062323770(0
4 01:16:26.514630 IP  77.28.x.55.4853 > 196.x.y.34.445:  S 1323192692:1323192692(0)
5 01:16:27.693010 IP  79.0.x.248.1731 > 196.x.y.18.445:  S 1786561877:1786561877(0)
6 01:16:29.808481 IP 189.101.x.133.2499 > 196.x.y.3.445:  S 3114908412:3114908412(0)
7 01:16:30.696890 IP  79.0.x.248.1731 > 196.x.y.18.445:  S 1786561877:1786561877(0)
8 01:16:32.751635 IP 189.101.x.133.2499 > 196.x.y.3.445:  S 3114908412:3114908412(0)
```

Figure 7.10: Examples of the two packet repetitions

ysis process. While absolute certainty is not possible without payload analysis, researchers can attain a high level of confidence in their analyses.

7.3.3 Transmission

A further interesting characteristic observed in the 445/tcp traffic after the advent of Conficker, is that it has a very noticeable signature 'on the wire' in terms of the way connection attempts are made. Most operating system TCP/IP stacks will send at least three TCP SYN packets in an attempt to establish a connection. By contrast, in the majority of the 445/tcp traffic received after the 20^{th} of November 2008, one sees only two connection attempts. An example of this is shown in Figure 7.10, where source addresses (in the 4^{th} column) make two connection attempts approximately three seconds apart. Similar behaviour has been observed by Aben (2009) and . This was further validated by the researcher with captures of live propagation traffic obtained from hosts with confirmed Conficker infections.

Considering the total number of sources observed and the total number of packets targeting 445/tcp after November 20^{th}, these are in a ratio of approximately 1:4,

Table 7.5: Traffic to 445/tcp by attributed Operating System

Rank	Protocol	Number	%
1	Windows	16336052	99.671
2	Proxyblocker	19401	0.118
3	MacOS	10066	0.062
4	FreeBSD	7114	0.043
5	Linux	4361	0.026
6	NetBSD	3981	0.024
7	Cisco	3230	0.019
8	Solaris	1910	0.011
9	Checkpoint	1343	0.008
10	NMAP	1258	0.007
		Total	99.992

N=16 389 887

Percentage of Packets attributable to an IP address with an identified Operating System

N is calculated as the packets received after 2008-11-20 00:00 GMT+2

indicating that most sources scanned two hosts at most. This is discussed further in Section 7.4, where an analysis of the target addressing is provided.

7.3.4 Operating System Fingerprinting

Operating System attribution was performed using the methods discussed in Section 5.4.2. While it is recognised that this method is not flawless, and may be skewed by the use of NAT and dynamic addressing, it nervertheless provides a useful measure. The results are presented in Table 7.5. Microsoft Windows family platforms accounted for 99.7% of the sources that could be attributed, which is is unsurprisingly given the fact that the Conficker malware targets these platforms, and the TTL data as seen in Section 7.3.1.

What is surprising is that machines are still being observed as infected, despite the patch having been out since 23^{rd} October 2008, and the Microsoft malware removal tool having had functionality to clean the Conficker malware off a system since January 2009. The removal tool is automatically run by the patch update procedure that occurs as part of Microsoft's monthly 'Patch Tuesday' patch cycle. One of the actions taken by the Conficker malware as a means of self preservation is to disable the automatic update and patching mechanism provided by Microsoft operating systems. What can be concluded from this is that these machines remain

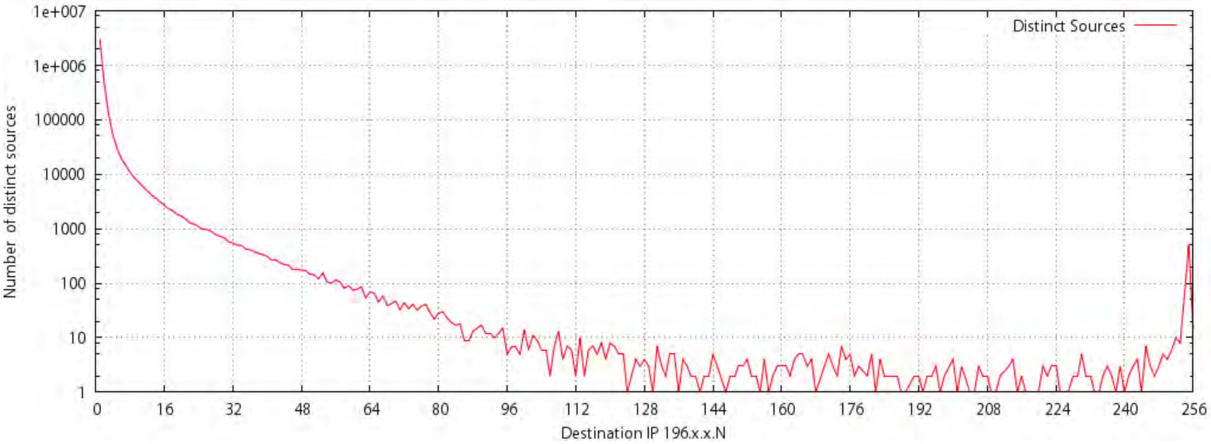


Figure 7.11: 445/TCP traffic: Distinct Sources per sensor IP

infected due to one of the following reasons:

User unaware — The user is unaware of the infection.

User unable to update — The user is unable to update because of the defence mechanisms put in place by the malware

System unable to update — The User is unable to apply updates due to the system platform likely being pirated and therefore unable to update[6].

7.4 Target Analysis

Changing focus away from the sources of the traffic to the addresses being targeted in the network telescope address space, a very uneven distribution pattern is observed. The lower half of the monitored space, i.e. 196.x.x.0/25, is targeted substantially more than the upper half (196.x.x.128/25). Particularly heavily targeted is 196.x.x.1, closely followed by other addresses in the lower 16. The first eight addresses in the address block all received more than 100 000 distinct sources. This bias is shown in Figure 7.11, which considers the number of distinct sources rather than packets observed for each IP address in the monitored range.

The strong bias towards the lower portion of the address space can be seen clearly. Notably, the last address in the monitored range (196.x.x.255) received a much

[6] Interestingly this is a policy that Microsoft changed with the advent of the Windows 7 platform. Even pirated copies of this OS will be able to receive security updates.

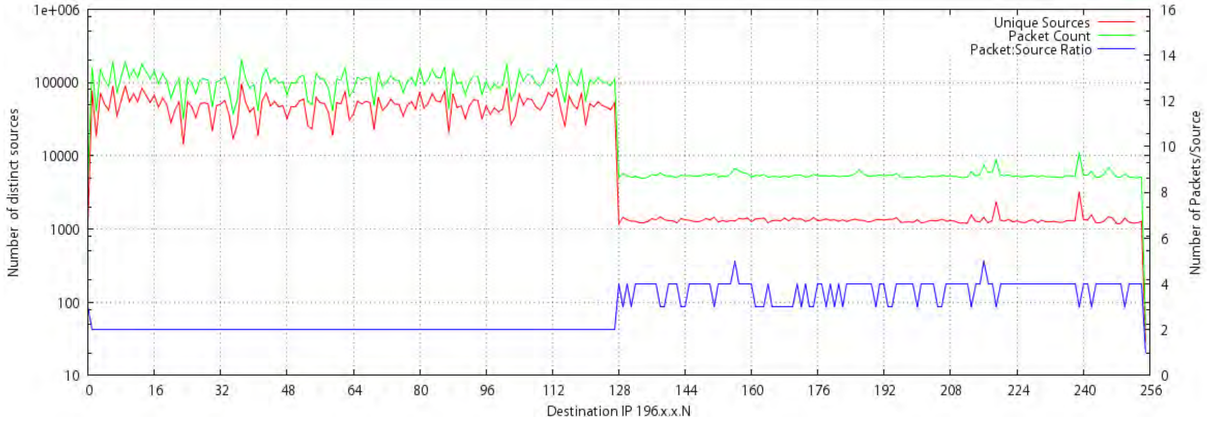


Figure 7.12: 445/TCP total traffic received per IP in the sensor network

higher coverage than other IPs in the upper /25 portion. The reason for this bias is most likely a naive scanning optimisation, which attempts to probe one or more addresses on a range, and if no response is received, moves onto another range. The probes to the last address on the range may serve a similar purpose. By convention default gateways on most IP networks either make use of the first or last address in a given subnet.

Considering the packet counts received for each address, presents a markedly different picture, as shown in Figure 7.12. Here a sharp change in the levels can be seen occurring at 196.x.x.128, with a drop in recorded traffic of nearly two orders of magnitude. The Figure contains the number of sources, and the packet count observed for each address. A third value is plotted on the second *y-axis*, on the right, being a ratio of the number of packet to distinct sources. This value also makes a change as the target address moves into the upper half of the range, doubling from just over 2 to 4. The dip at the beginning of the graph is due to some traffic having been recorded to 196.x.x.0, which is not normally targeted as it it would be regarded as the 'network address', rather than a host address.

This observed bias towards the lower 128 hosts in the network is due to a bug in the scanning algorithm implemented by Conficker (Richard and Ligh, 2009; Carnivore.IT, 2009). Due to the way the pseudo random number generator works, a 15-bit value is generated and then used in a 16-bit field, resulting in the most significant byte of the 2^{nd} and 4^{th} octets in an IPv4 address to being zero; in effect limiting these values to the range 0-127. The side effect of this is that it significantly reduces the portion of the Internet that can possibly be scanned (Aben, 2009; Wustrow *et al.*, 2010), limiting it to only 25% of total address space —

although it will attempt to scan all the /8 blocks and half of all /16 blocks. Taking this into account, one can use the traffic to the upper 128 address to quantify the other scanning activity present on the port that is not attributable to the Conficker Worm.

Looking at the sources of the traffic directed to the lower half of the monitored range, 1 049 562 had a 2^{nd} octet greater than 127, and 1 842 784 with the last octet having a value greater than 127. Common to both these groupings were 515 834 hosts. What is interesting is that these these could not have been infected by direct scanning, although given dynamic addressing they could have had other addresses previously, or have changed networks (as is common with mobile systems). Infection is still possible though other measures such as via the Windows 'autorun' mechanism on removable media, which was used by the earlier Conficker variants. These sources are analysed further in Section 7.5.

It is worth noting that the differences in traffic observed between the upper and lower ranges is substantially more than the three times differential described by Wustrow *et al.* (2010) and attributed to this activity. This bug most likely accounts for the fact that the RUSCOPE2 dataset has so little traffic destined to 445/tcp, since both its 2^{nd} and 4^{th} octets are greater than 127. In addition to this bug in the way random IP's are generated, IP ranges known to belonging to security research firms, and Reserved IPv4 Address space are also avoided by the malware. This is an interesting case which shows the value in having distributed IP address space for monitoring global trends. It is also an argument for chunks of contiguous space rather than smaller fragmented blocks as advocated with the use of greynets (Baker *et al.*, 2010b).

7.5 Source analysis

Analysing the source address data for the Conficker provides an interesting insight, especially when comparing the top sources ranked by packet count and host count. These are presented in Tables 7.6 and 7.7 respectively, and discussed in the sections below. The value in discriminating between these two means of ranking data is that the first quantified the volume, and to some extent the persistence of the infection. In contrast, ranking and subsequent analysis by the number of distinct sources observed provides a means for assessing how widespread the infection and related scanning activity was.

7.5.1 Packet Count

Considering the rankings by /8 netblock, 196.0.0.0/8, managed by AfriNIC, can be seen to have the highest packet count by a significant margin. It is worth taking note that a single host within this block accounted for 23% of the packet count in this netblock. Looking at the remainder of the top ten, over 38% of the total packet count can be observed. The numerical sequencing, of the top ten /8 netblocks have very close numerical groupings. The allocations within 189.0.0.0/8, 190.0.0.0/8 and 201.0.0.0./8 are controlled by LACNIC, with the remainder of the top ten being under the control of RIPE. Three adjacent groups of netblocks are observed. This is most likely due to the scanning an propagation mechanisms used by the Conficker malware.

This numerical closeness is also evident when considering the /16 rankings, with two contiguous address blocks in positions one and two. This can also be seen with the two blocks in 93.80.0.0/15. In the /24 rankings four contiguous blocks can be observed in 196.20.164.0/24, contributing to the high rankings of both the 196.20.0.0/16 and subsequently 196.0.0.0/8 netblocks. However, even combined these four blocks comprising 1024 addresses account for less than a third of the volume the top observed host. The traffic attributable to individual hosts shows a rather dramatic decrease, with the top ranked host of 196.21.218.x accounting for more than two and a half-times the sum of the remainder of the top ten hosts. A second host from the same /24 netblock appears in 6^{th} position. All but one of the top ranked hosts are within the 196.0.0.0/8 netblock, to some extent explaining the reason that this netblock was also the highest ranked overall as discussed in Section 5.3.1.

These hosts are of interest considering the pattern of the traffic observed over the duration of the study. Traffic for the 196.21.218.0/24 netblock is presented in Figure 7.13. What can be clearly seen is that the activity is bursty, occurring in four major groupings. While this Figure represents all traffic observed, after the advent of Conficker only traffic targeting port 445/tcp was detected, across the entire range monitored by the network telescope. What is interesting is that there is not a significant drop-off in the volumes of traffic from this host targeting the upper /25 of the monitored block. This in turn indicated it is most likely not Conficker, but most likely scanning looking for the vulnerability. The activity relating to 445/tcp was first observed on the 26^{th} of February 2009, and from this point to the end of the observation period was observed on 62 days, with the majority of the scanning being observed in the period ending on the 11^{th} of March, with a high of 26 859 packets received on February 27^{th} — twice the average daily rate during this period.

One interesting anomaly is that while in excess of 85% of the packets from this network were of size 62, a small portion were much larger at 66 and 78 bytes, indicating a strongly likelihood of a different source. TTL Values were also seen to vary between 109 and 118, while the majority of packets had a TTL value of 116. Packets received from the same source at nearly the same time, yet having differing TTL values, is indicative of the use of Network Address Translation (NAT), with the observed address being that used by the NAT gateway. Differences in the internal topology of the network behind the gateway would account for the differing TTL values. The use of NAT would explain the extremely high packet counts in relation to other hosts observed, if a larger block of systems were having their traffic coalesced into that of a single IP address as far as the outside world is concerned. Given the institution at which this netblock is utilised, and the absence of other addresses from this block (with the exception of that noted above), the researcher believes that the use of NAT is highly probable.

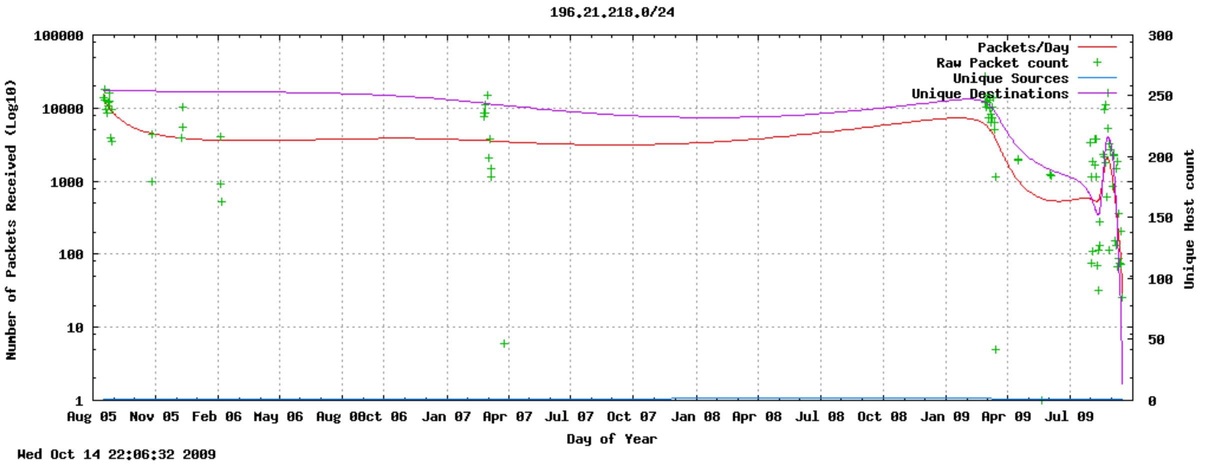


Figure 7.13: 196.21.218.0/24 - Traffic observed

A comparison of the percentage contribution for each of the netblock aggregation levels decreases sharply from the level covered by the /8 netblocks. However there is relatively little difference between the /24 and /32 levels, largely due to the contribution by the hosts in 196.21.218.0/24. It is still significant that the top host rankings accounted for 2.5% of the total, despite there being over 3.8 million hosts observed, even more so the percentage of traffic attributable to the top ranked hosts. The nearly 80% decrease in composition moving from /8 to /16 is an indicator of how widely spread the scanning activity was, although the proportion covered (7.9%) is less than the 12% observed in the overall RUSCOPE1 dataset, with the values for /24 and host level proportions being much closer.

7.5.2 Source Count

Rankings of the number of distinct sources observed as origins for traffic destined to 445/tcp, aggregated of by network block is presented in Table 7.7. This presents a somewhat different picture to that discusses in the previous section. This ranking provides an indication of how widespread the activity was within a particular netblock, and can be used as a means of determining infection rates. These are not absolute values, and may well be influenced by the use of NAT and dynamic addressing of endpoints, or even a combination of these — as is particularly common in modern broadband networks.

At the /8 aggregation level, the first eight ranked netblocks also occur in the top ten by packet count, with 196.0.0.0/8 and 93.0.0.0/8 being omitted. These placed 54^{th} (with 14 914 sources) and 72^{nd} (4 268 sources) respectively. The top four aggregated blocks had a fairly equitable portion of the total hosts served, at just over 5% each. The top ten accounted for over 40% of the total hosts. Again sequential netblock are seen to occur indicating a strongly likelihood of propagation activities favouring 'near' or 'local' addresses, although this could have been influenced by some of the other propagation mechanisms used.

This sequential locality is repeated again with the /16 aggregation. A strong geopolitical bias can be seen at this level, with five of the top 6 ranks netblocks being under the control of Corbina Telecom[7] (AS8402), head-quartered in Moscow, Russia. These blocks include blocks used in Moscow, St Petersberg, and other centres within the Russian Federation[8]. The number of sources observed as a proportion of each of the /16 netblocks is given in the percentage cover column. The highest of these is within 78.106.0.0/16 and 93.80.0.0/16 where over 45% of hosts in each netblock have been observed. This indicates a fairly widespread presence of scanning activity, either directly attributable to the Conficker Malware, or to

[7] http://www.corbina.net/

[8] Readers can get a view into their current routing via the BGP Looking Glass provided at http://noc.corbina.net/usr-cgi/lg.pl

Table 7.6: Top 445/tcp origins by Packet count

Rank	**/8**	**count**	%	**/16**	**count**	%	**/24**	**count**	%	**/32**	**count**	%
1	196.0.0.0	1 118 121	7.921	196.21.0.0	287 270	2.035	196.21.218.0	266 790	1.890	196.21.218.x	258 615	1.832
2	189.0.0.0	600 718	4.255	196.20.0.0	234 092	1.658	196.20.164.0	22 734	0.161	196.20.17.x	21 611	0.153
3	190.0.0.0	586 827	4.157	78.106.0.0	91 845	0.650	196.20.17.0	21 655	0.153	196.14.169.x	21 576	0.152
4	92.0.0.0	565 721	4.007	93.80.0.0	91 248	0.646	196.14.169.0	21 576	0.152	196.20.13.x	15 992	0.113
5	95.0.0.0	539 401	3.821	93.81.0.0	83 169	0.589	196.20.165.0	20 762	0.147	196.21.125.x	8 848	0.062
6	89.0.0.0	494 255	3.501	95.28.0.0	76 423	0.541	196.38.187.0	20 077	0.142	196.21.218.x	8 175	0.057
7	78.0.0.0	463 784	3.285	89.178.0.0	75 996	0.538	196.20.167.0	18 283	0.129	59.162.166.x	6 549	0.046
8	79.0.0.0	387 708	2.746	95.24.0.0	68 342	0.484	196.20.166.0	17 758	0.125	196.32.152.x	6 302	0.044
9	93.0.0.0	367 343	2.602	190.51.0.0	54 041	0.382	196.20.13.0	15 992	0.113	196.15.239.x	5 958	0.042
10	201.0.0.0	353 829	2.506	196.205.0.0	53 587	0.379	196.20.140.0	14 668	0.103	196.34.217.x	5 841	0.041
Total			38.801			7.902			3.115			2.542

$N_{Packets} = 14\,115\,791$

related scanning for the vulnerability exploited by this worm. Overall the top ten /16 netblocks accounted for nearly 6% of all hosts observed.

Finally when considering the top ten /24 netblocks, the top source network (196.1.232.0/24) originates from the Sudan (SD), and most likely Khartoum, belonging to the 196.1.232.0/22 allocation operated by SudaTel (AS15706) . For this network, 233 individual IP addresses have been observed out of a possible 254 operable host addresses, providing a coverage factor of over 90%. Sequential address blocks can be seen with 79.114.134.0/23, and the aggregate block of 89.179.104.0/22. Other /24 blocks in the top also exhibit both numerical and topological (being part of the same ASN in global routing tables) closeness. The face that blocks such as those shown appear in the top ten out of 495 602 observed /24 blocks indicates highly concentrated level of activity on a very localised level. Of interest is that despite 89.179.0.0/16 containing five of the top ten /24 netblocks, is is not in the top ten when aggregation is performed by /16, appearing only in 39^{th} place in these rankings.

7.5.3 Sources and Conficker Scanning

The final aspect to be considered regarding sources related to the construction of the observed addresses, given the flawed random number generation discussed in Section 7.4. Evaluating the data presented in Tables 7.6 and 7.7, one can observed that the majority of the /16 networks had a second octet with a value of less than 128. Regarding /24 networks, only the 89.179.104.0/22 as this octet greater than 127. Considering the individual hosts in the top ten, six have a fourth octet value > 127. Further Analysis of the top 100 000 IP addresses, showed 28% had a second octet greater than 127, with 40% having a fourth octet meeting this criteria.

One explanation for the observed behaviour is that the flaw only affects the random IP selection and scanning phase of Conficker's propagation. It also implements localised scanning on infected hosts, scanning those addresses nearby numerically. Thus given that IP addresses may be sparsely allocated when one considers the last octet, the influence of the second octet may be seen to be more significant. Considering the total sources identified sending traffic to 445/tcp after the advent of the Conficker worm, just over a quarter (27.5%) of the hosts originated from networks with the second octet greater than 127. These could be considered to be under represented, given an even statistical distribution. These hosts accounted for 3

Table 7.7: Top origin netblocks for 445/tcp by source count

Rank	/8	Sources	%	/16	Sources	%	%	/24	Sources	%
1	189.0.0.0	209 921	5.511	78.106.0.0	30 407	0.798	46.397	196.1.232.0	233	91.015
2	92.0.0.0	204 970	5.381	93.80.0.0	29 643	0.778	45.231	79.114.134.0	229	89.453
3	190.0.0.0	191 752	5.034	93.81.0.0	25 766	0.676	39.315	79.114.135.0	226	88.281
4	95.0.0.0	190 630	5.004	95.28.0.0	25 703	0.674	39.219	89.179.78.0	224	87.500
5	79.0.0.0	151 942	3.988	89.178.0.0	25 677	0.674	39.179	89.179.104.0	222	86.718
6	78.0.0.0	143 154	3.758	95.24.0.0	23 763	0.623	36.259	89.179.105.0	215	83.984
7	201.0.0.0	113 530	2.980	190.51.0.0	19 982	0.524	30.490	89.179.106.0	213	83.203
8	89.0.0.0	113 106	2.969	77.28.0.0	17 173	0.450	26.203	89.179.107.0	211	82.421
9	59.0.0.0	111 005	2.914	59.93.0.0	15 446	0.405	23.568	93.81.128.0	209	81.640
10	87.0.0.0	110 858	2.910	77.29.0.0	14 024	0.368	21.398	93.81.132.0	209	81.640
		Total	40.449		Total	5.970				

$N_{SourceIP} = 3\,809\,104 \quad N_{/8} = 194 \quad N_{/16} = 14\,896 \quad N_{/24} = 495\,602$

The host percentage is calculated as the proportion of all hosts sending traffic to 445/tcp

The coverage is the proportion of of observed host to the relevant netblock size.

Data columns omitted are due to the values being too small to be meaningful.

920 612 packets representing 27.77% of the total traffic directed at 445/tcp, again a somewhat lower contribution than would be expected. Repeating this analysis, for those having both the second and fourth octets greater than 127, resulted in a coverage of 27.99% of the traffic, again highlighting the significance of the second octet.

7.6 Geopolitical Analysis

The last aspect of the case study of Conficker related traffic, is that of the geopolitical origins of the traffic, and particularly how these changed over time. The origins have to some extent been touched on in Section 7.5, but rather than focussing on specific network address blocks and IP addresses, this section takes an aggregate view at a country level[9]. This approach is taken in order to provide a bigger picture of the spread and evolution of the worm on a global scale. Country codes as assigned in ISO 3166 are used in tabled for brevity. A list of selected codes is contained in Appendix C.

7.6.1 Pre-Conficker

Prior to the advent of Conficker in November 2008, some 2.7 million datagrams were observed targeting 445/tcp, from 198 thousand source addresses. The geopolitical origins of this traffic are shown in Table 7.8, which provides a top ten ranking of countries by the number of sources observed, and the number of packets received. Notable in both rankings is the high proportion of the whole with in excess of 70% of both IP addresses and packets being covered. This is was observed to change dramatically following the publication of the MS08-067 vulnerability, and subsequent widespread exploitation.

Comparing the two rankings, seven countries are common to both of the top ten rankings. The United Kingdom (GB), Poland (PL) and Italy (IT), while having fairly significant host counts ranked in positions 14, 17 and 19 when packet counts were considered, significantly still within the top twenty. Conversely, South Africa (ZA) while the top ranked source by packet count was only placed 16th by the

[9] Readers can explore the individual attributions of the netblocks described previously using several tools such as whois and online geolocation services.

Table 7.8: Top Sources for 445/TCP — pre Conficker

cc	$SrcIP_{count}$	$SrcIP$	**Rank**	cc	$Packet_{count}$	$Packet_{\%}$
TW	40 072	20.187	**1**	ZA	981 349	35.326
US	39 728	20.014	**2**	US	275 276	9.909
JP	12 772	6.434	**3**	EG	251 975	9.070
FR	12 436	6.264	**4**	TW	185 946	6.694
DE	11 574	5.830	**5**	MU	82 372	2.965
CN	7 307	3.681	**6**	JP	78 462	2.824
GB	5 902	2.973	**7**	CN	68 403	2.462
PL	5 449	2.745	**8**	DE	60 138	2.165
EG	5 217	2.628	**9**	FR	52 297	1.882
IT	3 492	1.759	**10**	NG	39 578	1.425
Total		72.517		Total		74.724

$N_{SourceIP} = 198503$ $N_{Packets} = 2777950$ $N_{Countries} = 190$ $Avg_{Packets/Src} = 13.994$

source ranking, with 2 592 hosts observed. This serves as an example, were a relatively small number of hosts were responsible for a fairly significant volume of traffic – on average these host sent 378 packets each, or 1.479 packets per address monitored. This rate can be seen to drop off significantly, and should be compared to similar rates after the advent of Conficker.

7.6.2 Post Conficker

The situation observed after the widespread exploitation of the MS08-067 vulnerability in Microsoft Windows family systems, changed dramatically. The top ten ranked attributable countries of origin by source and packet count are presented in Table 7.9. While the relative percentage coverage for top ten rankings both by sources and count decreased, the impact on the latter was most notable. Although only a 55% representation of total packet count was achieved by the top ten source countries, this was far more evenly distributed than that given in Table 7.8.

Comparing the top sources as ranked, shows that although India and Japan had a significant number of sourced identified, they ranked much further down when looking at their contribution to the total traffic, ranking respectively in positions: 13 and 14. The remainder of the top ten by number of sources are all present in the top ten by packet count, joined by US, ZA . These two countries ranked 14^{th} and 60^{th}, by the number of identified sources. Hosts from the Russian Federation (RU) and Brazil (BR) maintained their first and second placings in both rankings. Looking

Table 7.9: Top Sources for 445/tcp — post Conficker

cc	$SrcIP_{count}$	$SrcIP$	**Rank**	cc	$Packet_{count}$	$Packet_{\%}$
RU	577 261	15.154	**1**	RU	1 791 475	12.691
BR	357 982	9.398	**2**	BR	1 062 521	7.527
IT	267 660	7.027	**3**	US	802 232	5.683
CN	265 809	6.978	**4**	TW	706 241	5.003
TW	233 693	6.135	**5**	IT	669 286	4.741
DE	215 123	5.647	**6**	CN	633 169	4.485
AR	181 141	4.755	**7**	DE	592 773	4.199
IN	149 420	3.922	**8**	ZA	552 573	3.914
JP	119 299	3.131	**9**	RO	548 204	3.883
RO	109 987	2.887	**10**	AR	519 605	3.681
Total		65.038		Total		55.810

$N_{SourceIP} = 3809104$ $N_{Packets} = 14115791$ $N_{Countries} = 214$ $Avg_{Packets/Src} = 3.706$

at the average traffic contribution by host for these countries, Russia and Brazil achieved values of 3.1 and 2.96 respectively. This is significantly lower than the average packet contributions observed before the Conficker worm. This measure, while crude, can be used to determine the level of activity. The values themselves would be expected to tend towards 2.0 as discussed in Section 7.3 where only two packets were observed before a host stopped. The observed values being higher than this can be attributed to a relatively small number of hosts which scanned the entire monitored range, often on a number of occasions.

These rankings can be broken down and evaluated smaller time frames. This is explored in the next section which provides an overview of the dynamics of the spread of the Conficker malware by geographic region.

7.6.3 Conficker Evolution

The final geopolitical analysis of the Conficker Traffic, considers the changing composition of the traffic as the malware evolved. Table 7.10 shows the lifetime of the Conficker worm segmented into five phases designated A to E. These phases correlate to the five worm variants that were identified (using the Microsoft naming scheme). While it is acknowledged that the actual traffic observed may have originated from multiple variants of the malware, the dates on which new variants were identified serve as a useful means by which to segment the traffic. The dated bounding the periods, as well as the percentage contribution (by packets count and host) to the total traffic directed to 445/tcp after the advent of Conficker is given at the bottom of the table.

The first consideration is that of the ranking by packet count as shown in the upper portion of Table 7.10. The increasing prevalence of the malware on a global scale can be see by the dilution shown in the overall percentage of traffic covered by the top ten countries in each of the periods, from a high of 70% in A to 57% in D and E. The low of 53.79% in period B would appear to be anomalous.

Looking at the changing composition of top geopolitical sources, countries consistently appearing in all periods are Taiwan (TW), the United States of America (US), China (CN) and Brazil (BR), which are highlighted in bold font. The absence of Russia (RU) from the top rankings in period A may be due to a condition in the original variant of the malware that checked for Cyrillic keyboard types, resulting in Russia appearing in 14^{th} and the Ukraine in 42^{nd} place, leading many researchers to initially believe that these countries were the possible origin of the worm. South Africa ranked highly in all but the last period, where the dilution due to the in excess of 10 million packets received, resulted in a rank of 16. For each of the periods, with the exception of B, the top three ranks represent a significant portion of the traffic.

Re-evaluating the same dataset, but ranking countries by the number of hosts observed provides a slightly different picture (as seen in Sections 7.6.1 and 7.6.2). Five countries were found to constantly maintain a top ten ranking across the five periods. These were Argentina (AR), Taiwan (TW), China (CN), Brazil (BR) and Russia (RU), and have been boldfaced in the lower half of Table 7.10. One possibility for these countries having a persistent ranking is that they are regarded has having fairly high incidence of software piracy[10]. A general dilution of the percentage of hosts covered by the top ten can be seen, and is similar to that observed with packet counts. The proportion of hosts covered drops from 74% in the starting period to only 61% in period B, this then increased up to 66% by period E. Interestingly the Ukraine (UA) makes an appearance in period B, possibly as a flood of hosts were infected with Conficker.B which removed the restriction on not infecting host with Cyrillic, and particularly Ukrainian keyboard settings.

In closing, Figure 7.14 shows the dominance of the contribution by Russian hosts to the overall 445/tcp traffic related to the Conficker worm, for six of the top countries — Russia, Taiwan, Brazil, China, Germany and the United States. This plot

[10] `http://portal.bsa.org/idcglobalstudy2007/`

Table 7.10: Changing Geopolitical sources by Evolutionary Phase

	Period									
	A		B		C		D		E	
Packet Count										
	cc	*%*	**cc**	*%*	**cc**	*%*	**cc**	*%*	**cc**	*%*
1	MU	18.53	RU	9.50	ZA	16.84	RU	13.03	RU	13.84
2	**TW**	11.67	**US**	6.00	RU	10.31	ZA	9.77	**BR**	8.51
3	AR	10.40	**BR**	5.53	KR	6.07	**BR**	5.74	**US**	5.69
4	ZA	6.94	**TW**	5.51	**US**	4.62	**US**	5.39	IT	4.92
5	**US**	6.32	ZA	5.08	**BR**	4.51	KR	4.64	**TW**	4.86
6	**CN**	5.64	IT	5.02	**CN**	4.27	IT	4.50	DE	4.44
7	ES	3.16	**CN**	4.97	IT	4.02	**CN**	4.03	**CN**	4.41
8	CL	2.85	KR	4.56	**TW**	3.60	**TW**	3.46	RO	4.28
9	CO	2.71	DE	4.29	EG	3.53	RO	3.44	AR	3.59
10	**BR**	2.50	AR	3.33	DE	3.52	DE	3.27	IN	3.34
Total		*70.72*		*53.79*		*61.29*		*57.27*		*57.88*
Source Count										
	cc	*%*	**cc**	*%*	**cc**	*%*	**cc**	*%*	**cc**	*%*
1	**AR**	19.57	**RU**	13.74	**RU**	15.89	**RU**	18.27	**RU**	15.97
2	**TW**	18.98	IT	7.66	**CN**	6.54	**BR**	7.25	**BR**	10.25
3	**CN**	9.26	**CN**	7.34	IT	6.23	IT	6.59	IT	6.87
4	CL	5.19	**BR**	6.66	**BR**	6.21	**CN**	5.99	**CN**	6.61
5	ES	4.67	**TW**	6.11	KR	6.02	DE	4.33	TW	5.92
6	US	4.61	DE	5.78	DE	5.03	**TW**	4.32	DE	5.67
7	CO	3.93	AR	4.56	**TW**	4.75	IN	4.29	**AR**	4.46
8	**BR**	2.97	KR	3.29	RO	3.69	**AR**	3.95	IN	4.21
9	**RU**	2.96	UA	3.05	**AR**	3.54	KR	3.84	JP	3.32
10	DE	2.43	IN	3.03	IN	3.14	RO	3.43	RO	2.96
Total		*74.57*		*61.22*		*61.04*		*62.26*		*66.24*

Range	Start	End	Packets	%	Hosts	%
A	20 Nov 2008	28 Dec 2008	482 311	3.41	93 103	2.44
B	28 Dec 2008	20 Feb 2009	1 751 334	12.40	484 931	12.73
C	20 Feb 2009	4 Mar 2009	640 640	4.53	180 631	4.74
D	4 Mar 2009	8 Apr 2009	1 194 125	8.45	348 844	9.15
E	8 Apr 2009	30 Sep 2009	10 047 561	71.17	2 912 630	76.46

presents the average packets received per unique source host on an hourly basis for the period of October 2008 to September 2009. The rise in the average rate can be clearly seen.

7.7 Global traffic observations

Looking at the spread of the infection at a global level presents an interesting contrast. Dshield.org is a project run by the Internet Storm Centre (ISC), which collects submissions from people around the globe. These submissions are generally produced from firewall logs recording filtered traffic. A plot of the data extracted from the Dshield archive is shown in Figure 7.15. Absent from this, however, is the distinct increase in traffic seen in the RUSCOPE1 data from November 2008. Reasons for this could be due to the address space issue discussed previously, or due to pre-filtering of 445/tcp at border gateways, and thus a reduced incidence of reporting.

As of the time of writing, the Conficker worm continues to persist more than two years after its initial release. Estimates of the total population size vary, but is is generally acknowledged that there are between 5 and 7 million systems still infected, although this is slowly reducing. The data collected[11] by the Shadowserver Foundation for the period January 2010-January 2011 can be seen in Figure 7.16. Top sources as observed by their sensors continue to be in China and Brazil.

7.8 Analysis

This focussed analysis of traffic destined to 445/tcp has covered two distinct global malware threats — that of Zotob in August 2005, and Conficker in November 2008. In the intervening period traffic levels remained consistent, and can be attributed to remnants of the Zotob malware, and similar other software, and scanning by individuals for hosts having services on 445/tcp exposed to the Internet at large, and being potentially vulnerable to compromise. Over the period of the RUSCOPE1 Dataset, and particularly in the last 14 months, traffic destined to 445/tcp made a

[11] http://www.shadowserver.org/wiki/pmwiki.php/Stats/Conficker

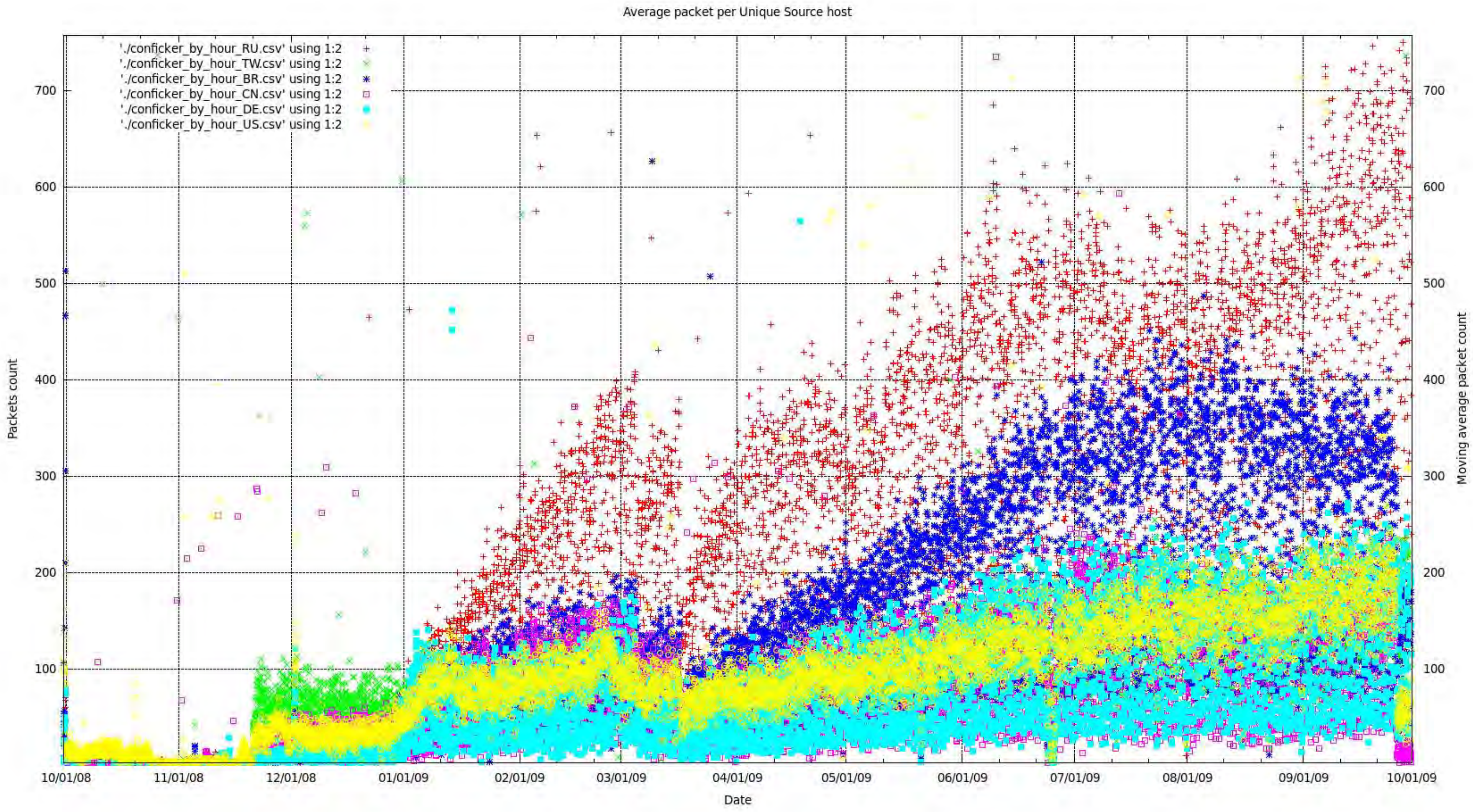


Figure 7.14: 445/tcp : Average Packets by Host per country

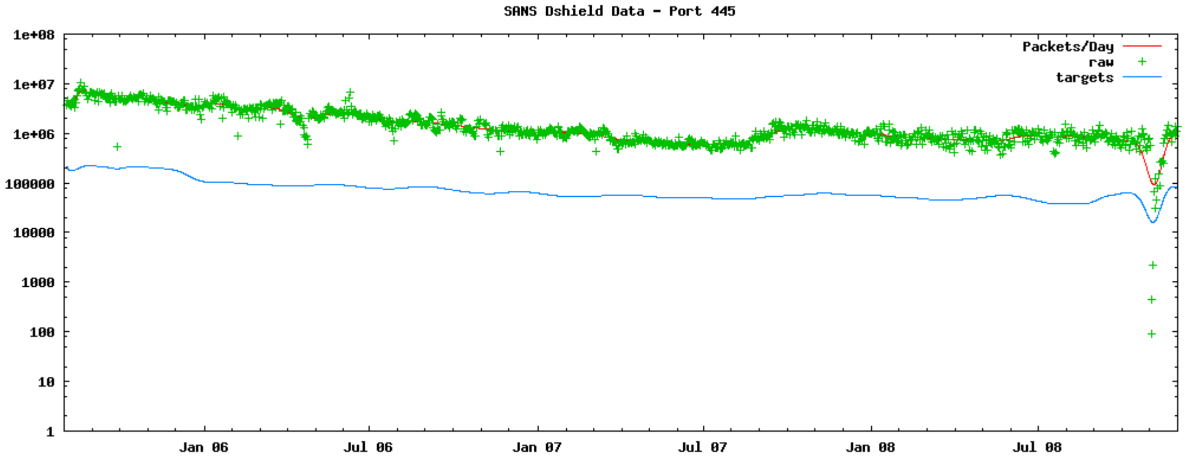
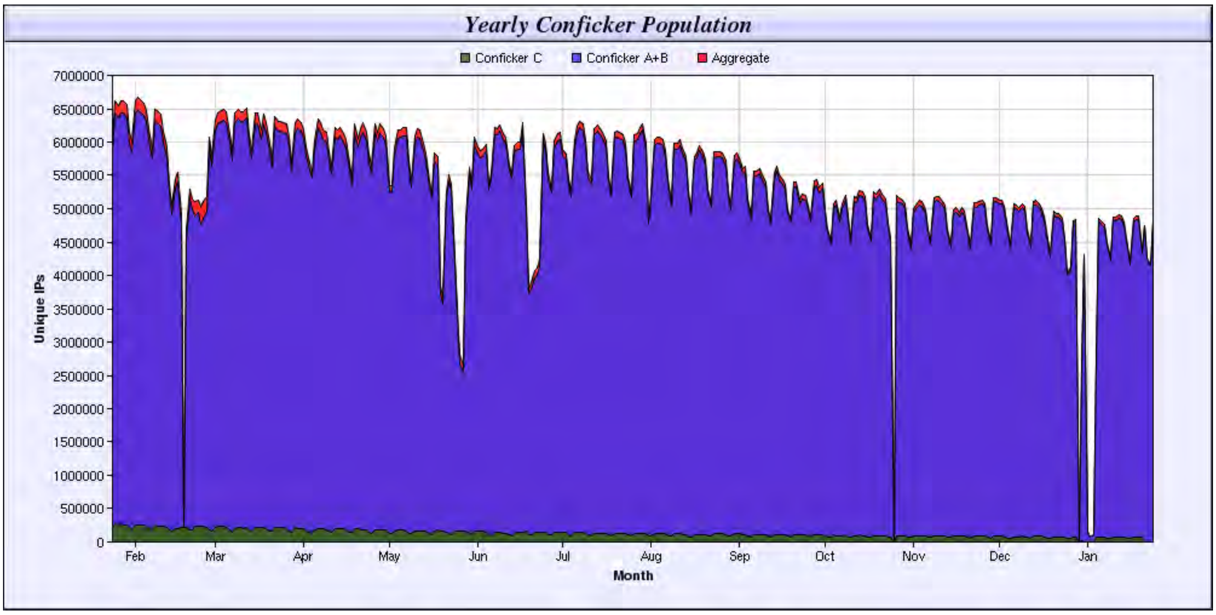


Figure 7.15: Traffic Observations from Dshield.org



Source: ShadowServer (2010)

Figure 7.16: Global Conficker Statistics January 2010 - January 2011

significant contribution to the whole. Given this, it is important to investigate the nature and origins of the datagrams.

While the analysis carried out in this chapter is by no means complete, it provides an example of the kind of focussed analysis that can be done with a network telescope. The evolution of the Conficker worm is plotted, with a particular focus on the changing geopolitical composition of the traffic over time. The problem with the random scanning and propagation algorithm can be clearly observed, and this is seen to be a plausible explanation for the significant difference in traffic observed by the researcher between the RUSCOPE1 and RUSCOPE2 datasets, even given the short duration of the overlap.

7.9 Summary

This chapter concludes Part II of the research and the analysis of the observed traffic. The final Part reflects on the findings presented and looks at the application of network telescopes both as a research tool and as part of a holistic information security architecture. The following chapter explores more succinct, high level mechanisms for conveying network telescope data summaries between researchers, in order to be able to perform comparative work, without the need to exchange large datasets. The use of such succinct data exchange could also solve some of the issues around privacy, and non-disclosure of monitored address space.

Part III

Application

Any idiot can tell you what something is. It is much harder to say what that thing means.

George Forrester Colony - Forrester Research Founder

8 Network Telescope Metrics

One of the problems faced when dealing with large numbers of data points is deciding which means of data summarisation best conveys the overall picture of the data being analysed in such a way that meaningful conclusions can be drawn from it. The approach taken in this chapter is to present a number of metrics that can be used to provide aggregate and summarised data for a network telescope dataset. These discussions are illustrated by using the data collected in the RUSCOPE1 dataset.

A series of common metrics specific to network telescopes are proposed which can be used to quantitatively compare datasets from differing sources/organisations without the need to transfer increasingly large raw capture sets, which in themselves have issues relating to privacy of the payloads, and possibly the anonymity of the actual sensor. The proposed metrics can be divided into two categories — those describing the telescope/sensor itself (Sensor Metrics in Section 8.1) and those describing the characteristics of a particularly sampled dataset (Dataset Metrics in Section 8.2). A single network telescope may produce several types of datasets, for example one may be processed at monthly intervals while other systems may produce weekly or even annual bundling of data. Provided the operating conditions remain the same, the sensor metrics should remain consistent and only the dataset metrics change.

The graphical representation of these metrics is presented in Section 8.3, which provides useful trending information without too much detail. The chapter concludes with a discussion relating to proposed common metrics that could be used for comparative analysis of network telescope data sets, both for intra-institutional and inter sensor use.

This chapter is not intended to provide a detailed overview of the field of metrics and information security related metrics in particular. Andrew Jaquith's *Security Metrics* book (Jaquith, 2007) provides an excellent overview of the topic. More detail on the visualisation of security metrics can be found in *Security Data Visualization* (Conti, 2007) and *Applied Security Visualization* (Marty, 2008). Further resources can also be found on the `http://securitymetrics.org` website and accompanying mailing list.

8.1 Sensor Metrics

Sensor metrics are those metrics that describe the configuration of a particular network telescope (sensor). These are applicable to a system for a given period of time and may change as the operation and configuration of the sensor system evolves. They should, however, remain relatively constant across multiple datasets at smaller time scales, such as months or quarters, but may be stable over much longer periods. The purpose of reporting these values is to describe the operation and configuration of the network sensor in such a manner that easily allows comparison of results with data from other organisations or even with other sensors in the same organisation.

One of the issues that arose in this research was the lack of descriptive meta-data about the configuration of the network sensor, particularly when trying to perform comparative analysis against the other datasets encountered. The following subsections propose a number of these metrics, as well as suitable meta-data which should be included in the presentation of such metrics. The section concludes with the presentation of these metrics and meta-data based on the data from the RUSCOPE1 dataset.

8.1.1 Lens Size and Shape

The size and type of the IP address space being used for monitoring can have a large bearing on the volume and quality of data collected. Knowledge of this is important in order to be able to perform comparative analysis among different datasets from differing collection sources. It is proposed that the size of the address space (as an aggregate total if smaller components are used) — being analogous to the lens size in a traditional astronomical telescope — be expressed in CIDR (also referred to as 'slash') notation (Fuller and Li, 2006). What is important is that the makeup of the address block be made clear – in essence the shape or form of the collecting mechanism used. Details as to whether the address space is a single contiguous block or an aggregate block comprised of smaller sized portions of address space should be disclosed.

Information that is useful, but not critical, would be how the sensor space was dispersed in the event of it being comprised of smaller constituent parts, and if the monitored block is part of a larger allocation used by an organisation. For example a block of size /29 (eight addresses) in each subnet in an organisation with a /16 allocation may be monitored. While there may be a relatively wide coverage (dependant on overall subnet size), there is still little address space in use, ranging from the equivalent of a /21 if subnets of size /24 are to used to the equivalent of a single contiguous /23 block if subnet assignments of size /22 are utilised. The fact that the blocks are not contiguous can have an impact on certain kinds of analyses such as that investigating the scanning algorithms behind observed scan traffic as described in van Riel and Irwin (2006a); Barnett and Irwin (2008, 2009).

Should the sensor be a portion of a larger block this could be expressed in a suitable format, preferably as a fraction rather than a percentage. As an example, two /29 networks within a larger /23 netblock may be used for network telescope monitoring. This amounts to a /28 in total, which in effect represents $\frac{16}{512}$ addresses or $\frac{1}{32}$ of the address space, which is a more understandable measure than when expressed as a percentage: 3.125%.

While ideally metrics by their nature represent a numeric distillation and quantification of data, it may well be beneficial to augment these with explanatory notes should there be anything other than a simple contiguous assignment of address space for use by the network telescope.

8.1.2 Mode of operation

The mode of operation of the network sensor(s) is essential when processing either historical data or data obtained from another organisation. Four common classes of operations for network telescopes are described below, and it should be stated as to which of these best fits the mode of operation of the sensor being described. These have been discussed in more detail in Section 2.5.

Passive: The sensor operates as a traditional network telescope, being completely passive, only logging traffic and affording no response on the monitored address space.

Low Interaction: Some form of low interaction system is used in order to enable the observation of data payloads in the subsequent first data packet. This is particularly important if TCP payloads are of interest to the researcher as these will only be present for active traffic once the initial TCP handshake is completed. This is usually performed by the software system responding to incoming TCP requests, and completing the 3-way handshake. The connection is then dropped after the first data packet has been received. Payload capture can be of value in determining the nature of the inbound traffic and may be of use when combined with some form of NIDS.

Full – Medium Interaction: A honeypot system such as honeyd[1], dionaea[2] or nepenthes[3], or a protocol specific tool such as kippo[4] (a SSH honeypot) is used to simulate live systems on given addresses/ports or a combination of the aforementioned. The advantage of this is that TCP payloads can be obtained beyond the first packet, as in the low interaction systems. Honeypot systems are discussed in more detail in Section 2.7.

Live system: Live physical or virtualised systems are used as endpoints for the monitored address space.

The mode of operation has a major bearing on the type of traffic and consequently data that is likely to be observed. Thus it is important to disclose to other researchers the mode of operation used by the sensor in order to capture traffic (and to record for historical use).

[1] http://www.honeyd.org/

[2] http://dionaea.carnivore.it/

[3] http://nepenthes.carnivore.it/

[4] http://code.google.com/p/kippo/

8.1.3 Sampling

Details of the means and frequency of sampling for the data collection process should be documented. This would include the following aspects:

Snap length: The value of how many bytes are recorded for each packet should be reported. Larger telescopes, or those recording large volumes of traffic, may elect to only capture the first n bytes of a datagram, rather than all bytes as seen ‘on the wire’, as done with the RUSCOPE datasets. It is worth noting that in its default mode of operation, tcpdump uses a *snaplength* (capture size per packet) of 64 bytes which is sufficient to capture the Ethernet header, IP header and the appropriate transport layer headers for TCP/UDP/ICMP, but not any actual payload, although portions of ICMP and UDP data may be present. The volume of data captured has direct bearing on the storage requirements and the potential flexibility of the data in future research. An alternate approach is that used by the University of Michigan, which calculates the MD5 checksum of payloads, and only stores the payload if it is not already present in the data store as described in Bailey *et al.* (2005a).

Capture Interval: This metric is a statement of how frequently data is sampled and stored from the sensor system. The CAIDA backscatter project (Moore *et al.*, 2004; Shannon *et al.*, 2005), for example, produces a set of captures for one week of every month. Although the interval may be driven largely by the storage requirements, interval sampling is different from a network telescope in its simplest form, which captures on a continuous basis. Another important item of information is detail of how the actual packets are sampled. In the case of data collected by the University of Wisconsin, only every 10^{th} packet was recorded for later analysis (Kumar *et al.*, 2005).

Disclosing this type of information may allow for more accurate selection of datasets for comparison, or even possible extrapolation of missing data.

8.1.4 Noise Suppression and Filtering

It may be to the benefit of other researchers for publishers of network telescope datasets to disclose if there is any known noise suppression or other filtering that is performed by the sensor. Traffic originating due to service discovery, infrastructure management or monitoring can potentially generate huge volumes of traffic which may well skew results obtained in later analysis. In addition there may well be a number of known misconfigured devices, or even active poisoning of the dataset traffic which can be regarded as noise and thus removed. This could be performed either at capture time through the use of suitable filters or as part of post-processing prior to archiving or further analytical processing. There may also be the case where certain ports, or even protocols, are missing from captures due to filtering by upstream providers or through choice. An example of this could be a sensor system dedicated to investigating and monitoring SSH scanning activity. In this case all that may be relevant is traffic destined for 22/tcp and ICMP ping (Type 8) packets.

The case of the CAIDA Backscatter and Telescope datasets as discussed in Section 3.1 can be considered here. The Backscatter dataset excludes all '*active*' traffic from the published captures, whereas the Telescope datasets contain all traffic other than that attributable to known production systems.

The suggested preferred means of expressing filters used is in the commonly used BPF syntax used by libpcap and many other common tools for working with packet traces and captures. This should be augmented by suitable notes where clarification may be needed. When publishing filters, care should be taken not to disclose potentially sensitive information relating to address space, which could result in future pollution of the sensor.

8.1.5 Meta-data

There are various other pieces of relevant meta-data relating to the network telescope which are worth recording and including in published descriptions of the network telescope. The first two of these deal with the location of the network from a physical (geographic) and topological perspective. While it is recognised that there is some need for keeping the exact specifics of the network location hidden, it is useful for researchers publishing datasets to be able to provide these details below.

Geographic Region: The geographic region that a network resides in can have a bearing on a number of factors, such as the available bandwidth and the amount of network address space available. In the case of the latter, North American and European organisations traditionally have relatively large IP address allocations while those in developing countries have been much more restricted. The regions could be described in general, or preferably at the level of the country or possibly even countries that a sensor may operate in. This can also be of use when measuring local geographic bias of traffic that may occur.

Topological Location: Similarly, it may be of use to other researchers to disclose which major networks provide upstream Internet connectivity for the network sensor. Sites with better peering to Tier-1 ISPs may receive better coverage of traffic. Networks could be described through the use of the name of the upstream organisation (e.g. Sprint, AT&T, and JANET) or via the registered Autonomous System (AS) Number(s) allocated to the provider by IANA through the Regional Internet Registries (RIRs).

Contact information for the group or person(s) operating the network telescope and/or publishing the data is important to enable queries regarding the datasets produced by the sensor in question. Sensible information would also include the organisation (and group if appropriate).

The final two items of meta-data would be if the telescope/sensor is still operational or not, and the date that the information was last updated. In the case of a sensor that is no longer in operation, the dates of operation could be stated.

8.1.6 Summary

In summary, Table 8.1 presents an example of the above metrics using data from the RUSCOPE1 dataset telescope. It is hoped that by producing summaries of sensor configuration as described above, researchers will be able to exchange information in a more effective manner. The use of standardised means of describing these sensors will allow for more meaningful comparisons of the data collected. When combined with the actual dataset metrics as described in the following Section 8.2, these could be published and allow for high level comparative analysis to be performed *without* the need for large dataset transfers and their associated problems.

Table 8.1: Sample Sensor metrics for RUSCOPE1 sensor

RUSCOPE1		
Metric	**Value**	**Note**
Lens size	/24	
Operational Mode	passive	
Sampling	24/7 all IP datagrams	★
Noise suppression	none	★
Geographic Region	South Africa (ZA)	
RIR	AfriNIC	
Upstream Network(s)	TENET (AS 2018)	
Operational Status	Active - Since August 2005	
Organisation	Department of Computer Science Rhodes University Grahamstown	
Contact person	Barry Irwin <b.irwin@ru.ac.za>	

★: BFP capture expression: ip and net 196.x.x.0/24

8.2 Dataset metrics

Dataset metrics would be used to describe a particular dataset as produced through the logging of packets on a network telescope or sensor network. A single dataset could be regarded as an aggregated collection of captures, or a single contiguous series of captures over a defined temporal period. The purpose of these proposed metrics is to be able to communicate the more useful and salient features of the dataset in a format that allows for easy comparison with other similar datasets, both from the same sensor, and ideally from other sources.

8.2.1 Top Items

The concept of reporting the 'top *N*' items has been covered in Section 5.1 and to some extent in the geopolitical analysis presented in Section 4.4. The purpose of presenting these metrics is to provide an overview of the dataset. In terms of ease of comparison, it is suggested that a relative scoring scheme be used in preference to raw numeric data (although both can be provided).

Two possibilities are available, the first being a simple percentage. This would indicate, for example, that traffic destined to 445/tcp amounted to 41.4% of the observed packets in the period and 50.75% of TCP traffic. The advantage of this approach is it provides a basis for relative comparison without having to worry about scaling issues when dealing with sensors of differing lens sizes and configurations.

A second method is to use an index based scoring scheme with some value set as a starting value, and all others normalised against this. Common choices for such values could include the values when monitoring was first started (which then provides some idea of the growth in observed traffic volumes over a period of time), or probably less useful, as it is problematic for inter-dataset comparisons, is an average score of sort. The use of such index based metrics is discussed in the following Section 8.3 where they are used for producing plots.

Whichever of the above means of displaying the quantitative information is chosen, the final choice is what number of values should be presented — the value of N. While a full detailed analysis is useful, in practise the majority of useful information is contained in the top 10-20 items for port based traffic and that analysed at a network block level, as already evidenced in Chapter 5. When presenting lists of 'top N' items, the total percentage of traffic represented by the values should ideally be disclosed. Items that are likely to be useful as summaries and to other researchers are:

Top Hosts/Networks: The top network sources observed as aggregated at differing levels of granularity (/8, /16, /24 and /32) can provide some indication as to the distribution of traffic. For the RUSCOPE1 dataset, these have been previously discussed in Section 5.3.

Top Geopolitical: The origins of the observed traffic. Bearing in mind that traffic can be trivially spoofed, these can provide some indication of hot-spots of malicious activity, or prevalence of malware. For example common wisdom in the network security community is that generally there is very little legitimate traffic originating out of countries such as South Korea (KR) for organisations not dealing directly with clients there. There are published block lists[5,6,7] to allow for easy blocking of traffic from South Korea and of other countries. As shown in the analysis of the RUSCOPE1 traffic, while countries such as

[5] http://www.countryipblocks.net/

[6] http://www.blockacountry.com/

[7] http://www.ipdeny.com/ipblocks/

Russia (RU) and Brazil (BR) may be very active sources for certain types of activity such as MS-RPC scans to 445/tcp commonly associated with Conficker (see Section 7.6), they are lower for others. Most importantly in this section it should be disclosed how the geolocation was performed, generally describing the method and libraries/sources used to perform the correlation. A selected list of relevant country codes can be found in Appendix C.

Top Topological: While network level aggregation may have some use, when looking at the traffic from a topological perspective, it is possible to further aggregate networks into the groupings by Autonomous System (AS) number, whereby they are routed on the Internet by the BGP protocol. One may find, for example, that while there may be a small number of hosts coming from individual netblocks, when these are aggregated by AS number, certain organisations or network providers may show up as hot-spots. This was shown in Section 6.4 for the RUSCOPE1 dataset.

Destination Ports: These represent the targeted destinations of traffic and allow for the monitoring of emerging threats and trends in scanning and other malicious activity. Summary data should be produced for both TCP and UDP as relevant for the datasets. Any specific filtering (such as backscatter or active only) should be noted.

Source Ports: While destination port traffic generally receives much attention, an analysis of the source ports can be interesting. For while the spread of these is much wider than with destination ports, certain ports do stand out. Source ports can generally be trivially controlled on the sending side (particularly on Microsoft Windows family systems and when using popular scanning tools such as Nmap) and may be used for evading IDS systems or attempting to bypass firewall rules. An example of this is the very common use of 80/tcp as a source port. Source ports also reveal other interesting information when viewed from the perspective of backscatter, as they allow a sensor operator to ascertain to some extent possible DDoS activity which commonly involves spoofed addresses, which in turn are reflected back to the sensor. As with Destination ports, processing and reporting should be done for both TCP and UDP.

Protocols: A breakdown of the composition of traffic by protocol is useful to other researchers when evaluating the dataset fitness for use in their own research. For example, the CAIDA backscatter datasets contain no UDP traffic, as it has been filtered out as part of the backscatter isolation process. While one would expect the common three protocols of ICMP, TCP and UDP to dominate, there may well be anomalies worth highlighting. ICMP data can be further summarised, as per Section 5.2.3, in terms of the characterisation of the traffic. It is suggested that traffic composition is reported as percentages.

For all of these basic, statistical information such as minimum, maximum, mean, median and mode can be reported. Due care should be taken when working with these values as many of the distributions of the traffic are not normal, often tending towards Poisson or multi-modal distribution. More detailed statistical metrics than these are in most cases better included in separate reports rather than in a metric summary report.

8.2.2 Temporal Aspects

The disclosure of period of time which the dataset covers is critical when making data available for comparative analysis. Date and timestamps should either be adjusted to UTC/GMT or if local time is used, suitable offsets from UTC/GMT are indicated (taking into account any local daylight saving). Along with the period captured, the duration should be noted along with the percentage of this time that the address space was actually monitored. Outages from both network and equipment perspectives do occur and it can be useful if these are recorded. It is suggested that a coverage score be disclosed as a percentage of when the sensor was actually active during the overall capture period. The granularity at which this was calculated should also be disclosed (days, hours, minutes, seconds). Of these units, days and hours are probably the more useful units of measure on which to base the calculation. Examples of this can be seen in Section 3.2.4.

8.2.3 Traffic Rates and Coverage

Other common metrics relating to a dataset that are of potential benefit to other researchers are some measure of the intensity and volume of the observed traffic. To this effect, the following are suggested:

Inter packet arrival times (IPAT): This metric can provide an indication of how dense the arrival of traffic at the sensor is. For example, a site monitoring aggressive scanning traffic or DDoS reflection is likely to have a much lower IPAT than a network sensor not receiving such traffic. This is in effect a score measuring how quickly the packets were arriving. The calculation is taken by determining the time between packets as recorded by the telescope. Care needs to be taken to remove any outage periods from such calculations.

Packet Arrival Rate (PAR): This rate, usually expressed in packets per second (pps), is again a measure of the overall average rate at which incoming traffic was received. This gives some indication of how busy the telescope was during the data collection period. While pps may be a viable reporting unit, on smaller or quieter telescopes, other temporal units such as hours or days may be more appropriate. As a guideline, this value could be scaled to appropriate time units to avoid having only fractional packets arriving in the particular time period chosen.

When interpreting these values, due care should be taken to factor in the lens size of the telescope when performing comparisons. As such it is suggested that normalised metrics also be presented which present the values scaled to a single IP, and to that of a nominal /24 for very low rates. These rates also need to be qualified in terms of directionality (inbound/outbound) should network sensors other than completely passive types (see Sections 2.5 and 8.1.2) be used for data collection.

The level of coverage observed should also be disclosed. This would include the number of unique IP addresses observed during the temporal period in question. Related to the Top networks, the total number of discrete observed networks at differing levels of granularity could also be shown.

8.2.4 Active/Backscatter Ratio

Although methods to discriminate between active and passive or so called backscatter traffic are imperfect, particularly for UDP traffic (which is why collections such as CAIDA backscatter remove it completely), it can provide an interesting analysis; in particular what the ratio of active to backscatter traffic was. Ratios for TCP are relatively easy to calculate, with ICMP a clear cut issue. Traffic composition should be expressed as a percentage-wise ratio of the total traffic for the protocol under consideration.

8.2.5 Summary

A partial summary of the metrics described above is shown in Table 8.2 which is a sample taken from the RUSCOPE1 dataset for traffic captured during July 2009. This table provides a sample of the reporting of the metrics not already represented elsewhere in this section.

8.3 Graphical Metrics and Temporal Sequences

This section builds on the metrics previously described in Section 8.2. In many cases being able to represent data in a graphical format provides a far more succinct means of conveying a large volume of information. The most common format for doing this is as a line plot with time being represented on the x-axis and volume or count on the y-axis. While Chapter 5 has made extensive use of these, the plots used were highly detailed and dealt with actual volumes of traffic recorded.

This section introduces a variation of the type of plot seen so far. While maintaining the temporal progression on the x-axis, the granularity is markedly decreased, as the purpose is to convey trends rather than detailed information. To this effect raw numeric data is converted into index based representations, which allow for the plotting of comparative data on the same set of axes. This is discussed further below.

The value of including the types of non-textual information described in this section is that they provide context in which the data can be interpreted. In the case of large bundled datasets, they can show the inherent trends within the data, which in many cases may be the information of greatest value to other researchers.

The issue of the granularity at which data is plotted needs to be considered. In most network datasets such as RUSCOPE1, the level of granularity goes down to sub-second accuracy. For most practical evaluation, particularly over longer time spans, a temporal bin size of days or weeks may be preferable, particularly if

Table 8.2: Example Dataset Metrics for Rhodes University Telescope: July 2009

Netblock Grouping	/32	703 855
	/24	250 937
	/16	1 4135
	/8	181
Total Packets		2 426 940

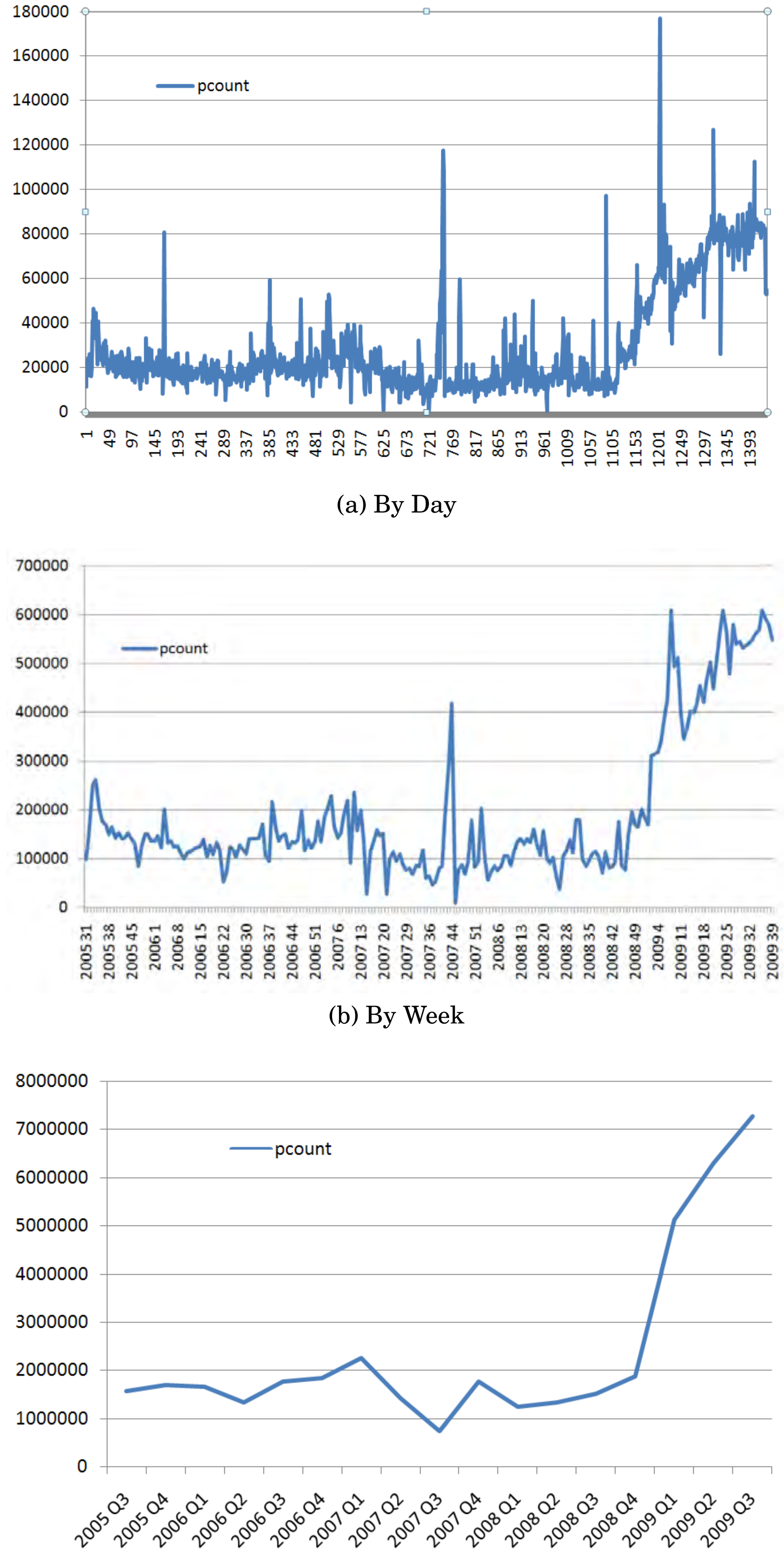
Protocols	**Ports**	**N**	**Top 10**
TCP	Source	56 406	6000 80 25511 5641 6667 3306 25521 6005 10000 4496
	Destination	9 844	445 135 22 1433 2967 5900 139 25 80 4899
UDP	Source	13 808	1859 1231 3106 1029 3302 2373 1102 53 4659 4150
	Destination	2 683	1434 137 38293 5060 33435 33436 33437 33438 22105 7548
ICMP	Type:Code		8:0 11:0 3:3 0:0 8:204 8:74 3:10 8:196 8:7 8:225
Traffic Composition			TCP 95.453% UDP 3.985% ICMP 0.561%
Geopolitical	Regions	191	"CN" "RU" "BR" "US" "TW" "IT" "DE" "RO" "KR" "IN"

trends are being looked at rather than the minutiae of the data. Figure 8.1 shows the difference between graphs when using increasingly lower levels of granularity when plotting packet count data for the RUSCOPE1 dataset. As the granularity decreases, the jitter factor on the lines becomes less, the overall shape of the graph however remains the same. The plot by week (Figure 8.1b) reveals two interesting spikes in 2007 Week 43 and 2009 Week 8, the latter has already been addressed in Section 5.3.4. These are also visible in Figure 8.1a, but are absent in Figure 8.1c. Appropriate levels of granularity should be chosen to convey data that is deemed to be important or significant, particularly when a reader may not have access to the raw dataset for further exploration. Summary numeric data should ideally be provided along with such plots, to allow for more detailed analysis and comparison.

8.3.1 Index plots

An index plot provides a useful means of plotting multiple data series onto the same set of axes, particularly where the raw volumes represented may differ by orders of magnitude. While not intended to provide detailed information, it serves to convey trends and facilitates comparison of these trends of different data series relative to each other. An example is shown in Figure 8.2. In this plot, the raw values for each series are mapped to be relative to a starting index of 100, although depending on the level of variance, other values such as 1 000 or more could be used. This is the same as the system commonly used to denote stock market performance. In this case the base values used in the calculation were those for 2005 Q3 were 114, 8 626 and 106 012 for networks aggregated respectively by /8, /16 and /24 masks. As can be seen these values differ quite substantially, and would be difficult to plot with any type of discernible meaning on a standard line graph even when using logarithmic scaling. The remainder of the data is calculated as a score relative to the starting index as in:
$Index_{T=N} = (Count_{QuarterN}/Count_{Q1}) \times 100.$

What is immediately apparent from Figure 8.2 is that by the end of 2009 Q3 more than four times as many distinct /24 networks were being observed per quarter than four years previously, while twice as many /16 blocks were being observed. The effects of the Conficker Worm in late 2008 can be clearly seen in both this Figure, and in Figure 8.3, which provides a similar index based plot broken down by protocol. Figure 8.3 also shows a clear drop in the real volume of ICMP traffic,



(a) By Day

(b) By Week

(c) By Quarter

Figure 8.1: RUSCOPE1 packet count data with varying granularity over the duration of the observation

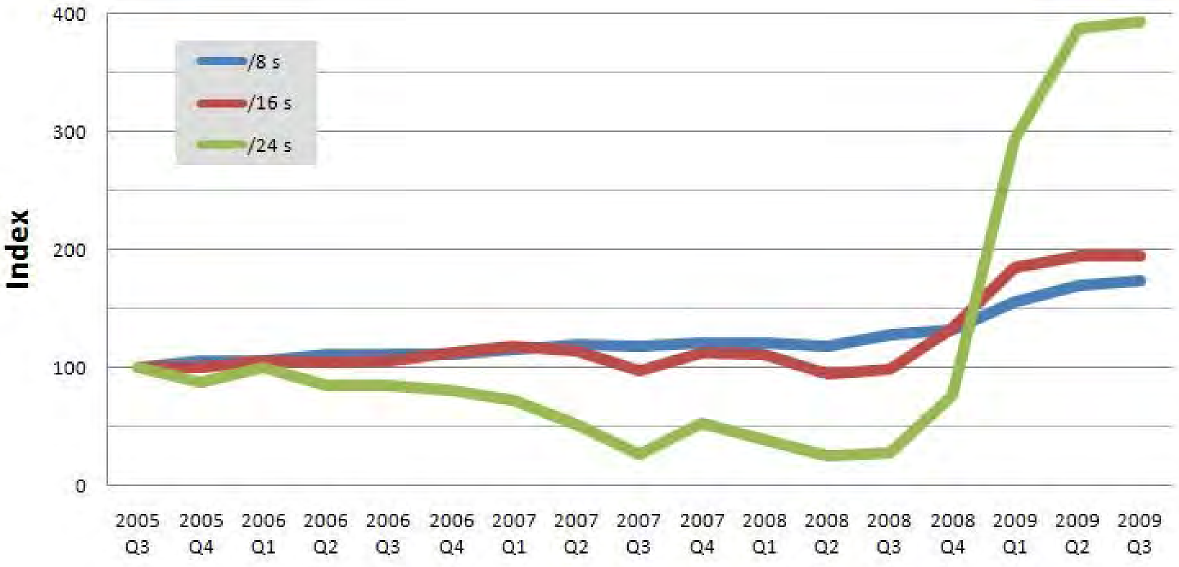


Figure 8.2: Traffic by Network Size

Note: Indexes at $T_0 = 100$ were $I_{/8} = 114$, $I_{/16} = 8626$ and $I_{/24} = 106012$

ending on an index of 24.92. Viewing these two Figures in conjunction, one can clearly observe a widespread increase in TCP activity: the number of /24 netblocks rises dramatically, as does the volume of TCP traffic. This activity is most likely attributable to the Conficker Worm which is discussed in more detail in Chapter 7.

8.3.2 Proportional plots

As discussed in Section 8.2, there is merit in sometimes representing the data in terms of percentage contribution of the parts to the whole. The use of percentage representation is similar to the index plots above but is a far coarser, yet often serves as a better understood means of communication. The intention of the proportional plot format discussed in this Section is to provide an overview of relative constitution of a dataset sample. Examples of this are shown in Figure 8.4 which illustrates two common methods of representing the percentage composition of the observed traffic from the perspective of the three primary constituent protocols of ICMP, TCP and UDP. In both cases, the sub figures illustrate the contribution of the various series to the whole. Depending on the data being displayed, either the bar or line formats may be more appropriate. These should be seen to be complementary to the index based plots. Although they also allow for a comparison of the relative contribution of each protocol to the traffic composition, it is at a fairly low level of granularity. Comparing Figure 8.4 with Figure 8.3, one can observe that

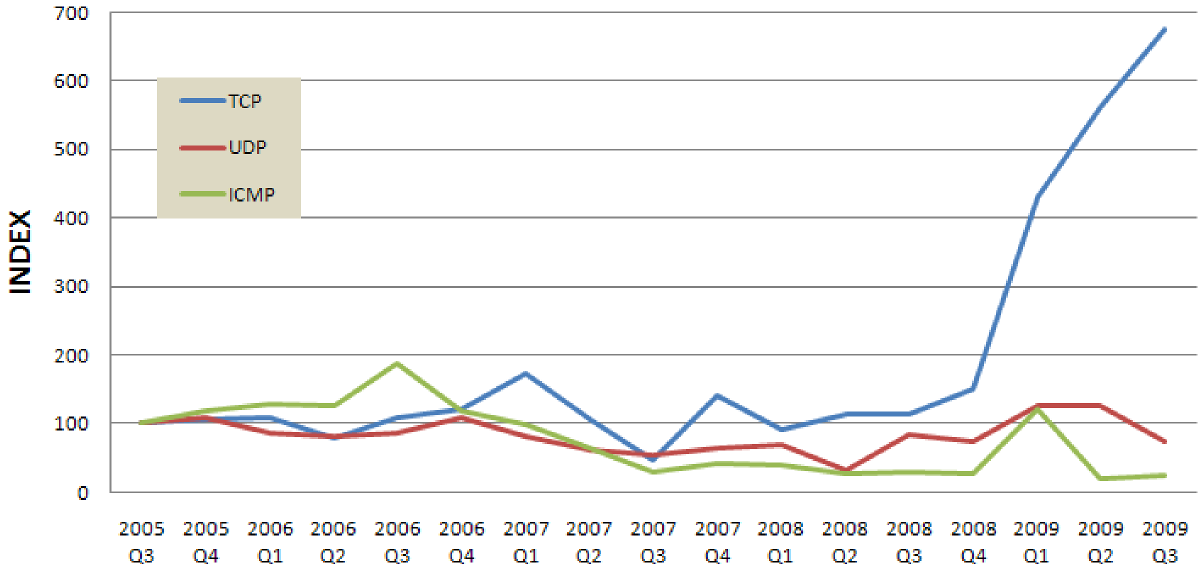
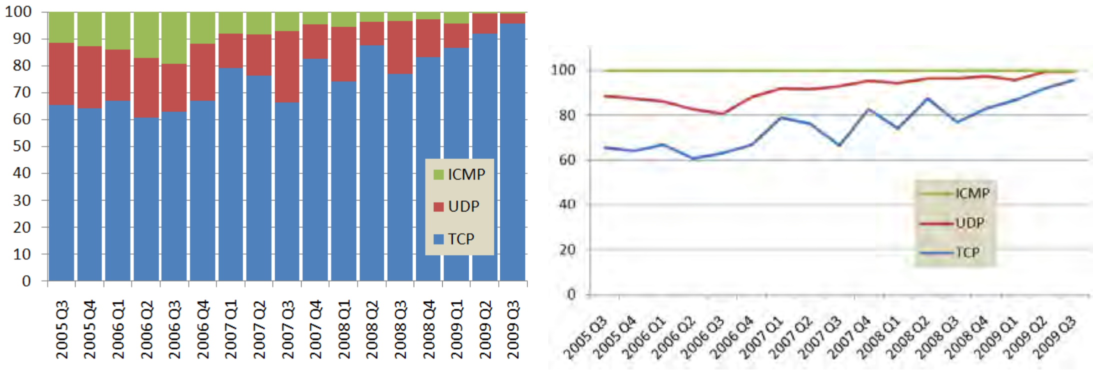


Figure 8.3: Traffic by Protocol



(a) Bar Plot

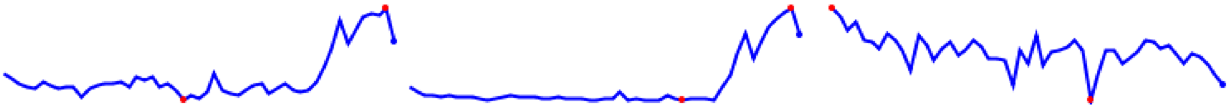
(b) Line plot

Figure 8.4: Percentage composition of traffic

the proportion of ICMP traffic observed is decreasing, most likely due to the large increase in TCP traffic in the last two plotted quarters. In real terms the volume of ICMP traffic is dropping too as shown by its lower index value in Figure 8.3.

8.3.3 Sparkline plots

The final form of plot to be considered is that of the Sparkline, popularised by Edward Tufte (2004). This is a a simple 'word sized' line graph. Sparklines are most often used to show trends over time and are generally dimensionless in that there are no values attributed to the axes. One variation to this plotting scheme is that the current or most recent point may have a value attached along with



(a) Total traffic (b) 445/tcp (c) 1434/udp

Figure 8.5: Sparkline plots
Plots are over the period 3^{rd} August 2005 to 30^{th} September 2009

high and low markers. The intention of these plots is to provide succinct trend information in a dense format. As a general rule this plotting technique is used only to show a single series per plot. Common uses of the Sparkline plot are inline in text and in dashboard type information displays and reports, where trend information is of more value than specific point data. Examples of Sparkline plots produced from the RUSCOPE1 dataset are shown in Figure 8.5. This figure shows the high and low points for the period as red dots. Considering the similar shapes of Figures 8.5a and 8.5b, the significance of the contribution to traffic destined to 445/tcp to the whole can be seen.

8.4 Summary

This chapter has presented an set of guidelines and motivation for the use of a number of different metrics which can be used to suitably describe both the network telescope and data collected. The use of graphical summaries as discussed in Section 8.3 allows for trend information to be conveyed in a very succinct format. The metrics proposed have been illustrated with examples from the RUSCOPE1 dataset. It is hoped that the adoption of metrics such as those described, will allow for easier comparison among multiple datasets. The publication of metrics may in some cases also obviate the need to have access to full packet captures — which have other issues such as privacy and anonymity to consider, quite aside from the potential problems of moving multiple gigabytes of data.

I sense an insatiable demand for connectivity. Maybe all these people have discovered important uses for the Internet. [...] At least a few must wonder what the big deal is.

Clifford Stoll - *Silicon Snake Oil*, 1995

9

Implementation & Recommendations

THE Internet is a constantly evolving system as are the risks that are faced by individuals and organisations connecting to this new global communications medium. The concept of a 'network telescope' has been introduced in this research as a tool for aiding researchers and information security professionals gain insight and a better understanding into the happenings on the modern Internet which has evolved from its fairly safe humble beginnings as ARPAnet to the relatively lawless and unrestricted space which it is today. This Chapter initially reflects on the information obtained though the analysis of the Network Telescope data presented so far. This is then followed by proposals and discussion around the operational and research applications of network telescope systems as discussed in earlier chapters.

9.1 An Analysis Framework

At the outset of this research, very little information was available on how to best approach the establishment, operation and subsequent analysis of data collected by a network telescope. Mention of the concept was found in the work by Nazario (2003) on strategies for dealing with Internet worms. This along with the work published by David Moore in (Moore, 2002; Moore *et al.*, 2002, 2003, 2004) provided nearly the sum total of the material available.

Subsequent to this a substantial body of more work has been published, notable works of which have been discussed in Chapter 2. The majority of these have focussed on looking at data sets of a few months duration at most. Notable exceptions to this norm are the work done by Pemberton (2007), Cooke (2007) and Wustrow *et al.* (2010), in turn building on the seminal work of Pang *et al.* (2004).

While the operational and technical aspects relating to network telescopes have become increasingly well understood, there is very little in the way of guidance for a systematic approach to the analysis of the data collected by the sensor, regardless of its mode of operation. This section looks to provide a high level framework which can be applied to the analysis of such data. Operational issues are discussed in Section 9.3. The analysis proposed below is approached with the premise that the data collection process already been established and that packet capture files are available.

9.1.1 Summary Data

Regardless of the volume of data being processed, the first step in performing analysis is to obtain a summary of the high-level constitution of the data in question. Key features that should be determined are:

Timing — Start and stop times for the capture file should be determined, and a duration calculated.

Sizing — Total packet count, byte count and packet arrival rate.

Addressing — The number of unique source and destination addresses should be reported.

Protocol Overview — An overview of the number of protocols detected, and details of the three primary protocols, this could also include a calculation of the relative percentages of the protocols constituting the sample.

TCP — Distinct source and destination port counts.

UDP — Distinct source and destination port counts.

ICMP — Distinct Type/Code pairings observed.

This kind of data can be generated by tools such as `tracerpt` which is part of the `libtrace` toolkit. Ideally this summary information can be generated at the time of capture or at the point of data rotation and migration from the collector system. In some cases, much or all of the above may have been calculated either at the time of capture, or by other data providers, such as CAIDA, from which capture files have been obtained.

The value of having this kind of overview, is that it can easily be assessed without the need for further tools, and can be used to locate temporal regions of interest (assuming as with most telescope data, the rotations are on a temporal basis). This is particularly relevant where dataset files are large, or numerous, as the summary data should be small in comparison. The summary data also serves to ensure that researcher's are working with the right datasets. When looking at the comparative analysis from two or more telescope sensors, care needs to be taken to adjust timestamps as appropriate to account for the offset of local time from GMT.

9.1.2 Data Processing

The second phase involves the processing of the raw packet captures. Two different approaches can be taken at this time, the first, as used in this study, is to process all the data and then perform an analysis on this output. The second, which can be used in conjunction with the first, applies pre-filtering on the data. This could be used in cases where only certain, protocols, ports or address blocks are of interest. An example of this is where researchers are only interested in the backscatter traffic being collected by a telescope. Given that backscatter has been observed to constitute a very small proportion of the whole, pre-filtering of the data at an input stage will speed up all subsequent processing. Doing so will obviate the need to perform this filtering with each tool applied. New projects will need to perform some initial runs over sample data in order to assess the best approach.

Approaches to the actual processing can vary, depending on the research outcomes at hand and volume of data. How long the processed data should be maintained for is another factor which needs to be borne in mind. The analysis presented in this work was largely achieved working on packet captures that had been loaded into a relational database system. While this approach has been shown to function well, and provide great flexibility when querying packets, it may not prove to be scalable on very large telescope sensors.

The database can be further supplemented by tools working directly with the packet captures. Custom tools can also be developed utilising libraries such as libtrace and libpcap which can produce suitable analysis and report, and may be more applicable to larger sets. An alternate approach is to leverage the performance gains detailed in Nottingham and Irwin (2009a, 2010b) making use of GPU based classifiers to perform the processing over larger datasets, being able to achieve high speed classification and processing given the parallel nature of the GPU architecture.

Data processing can also incorporated the segmentation of larger datasets into smaller component pieces. The architecture presented and used in this research has discounted the payloads of recorded packets during initial processing. Once anomalous or interesting activity was identified, the packet captures were reprocessed and payloads of matching datagrams extracted, and processed primarily in WireShark.

9.1.3 Graphical Overviews

The generation of graphical overviews of the data using methods such as Hilbert Curves, Heatmaps, or Time series plots is the next phase in understanding the data and trying to identify trends, and any particular areas of interest. While interactive exploration with tools such as Inetvis and Hilbert plots can be useful, this is often time consuming and requires suitably skilled operators in order to recognise abhorrent behaviour.

What is proposed is that static analysis be performed on the data as a first step, and then the interactive methods are used to validate data. The tools used to produce the static images for analysis should be should be template, in order to produce the same kinds of images on a repeatable basis, with differing inputs. This uniformity allows researchers to make direct visual comparisons. This process can then be automated such that a daily report could be generated as the previous days traffic is processed.

The graphical overviews serve to allow rapid high-level inspection of the data. Subtle changes in traffic or other anomalies not being catered for in the visualisation tool-chain are likely to be missed. The interactive analysis described,along with the numerical processing can help identify these.

9.1.4 Basic Numerical Analysis

Some kind of basic numeric processing is required as a pre-requisite to several of the visualisation techniques discussed. Further analysis and the production of tabular rankings can also prove useful. one of the primary advantages experienced by the researcher in loading the data into a relational database system, was the ease with which numeric analysis could be performed, either directly within the database, or though the use of external tools such as a spreadsheet — pivot tables were found to be particularly enlightening when looking at the relationship between different packet components.

The generation of basic statistical data on various packet parameters allows for the detection of anomalies. Some of the work already done in this regard ahas been published in Cowie and Irwin (2010a,b). This is closely related to the metric generation discussed in Chapter 8.

9.1.5 Metric Generation

Metrics provide a distilled representation of a particular dataset. Ideally metrics (and the production process) chosen for should be regarded as 'SMART', in that they should be Specific, Measurable, Actionable, Relevant and Timely. One of the key factors in satisfying these requirements is that the generation process should be automated as far as possible. Once suitable metrics have been identified and found to be of use, the manual process used to generate then should be codified, and automated.The automation process should ensure that the metrics can be produced timeously, as well as accurately.

When determining what metrics are relevant for a particular operational need or research project, one should resist to generate metrics for everything possible.

While it is recognised that part of the research process is exploring the data, metrics can be added iteratively as needed rather than as a blanket offering. Metrics are valuable in communicating trends and salient facts such as top ten lists, but are also likely to miss small (and possibly interesting events) due to their aggregate nature.

9.1.6 Detection and Analysis

The final phase of the analysis process is to interact directly with the packet captures. One should start looking at specific areas of interest that have been identified by the initial numeric or visual overview. More detailed mathematical analysis and graphing may prove sufficient, but in many cases, researchers will need to refer back to packet captures. These can be explored with a tool such as tcpdump if a quick overview of the packets are needed, or Wireshark if more in-depth analysis of smaller volumes of packet data is required.

The analysis process is iterative, and a finding may require additional processing of the data in order to correlate with other datasets, or even other activity happening in the same temporal vicinity. At this stage of the process, other tools can be used. Examples of this are the IP to ASN tool used in Section 6.4, online whois registries and geolocation tool sets. Once analysis is complete, it should be determined if the even is something of future interest, and if so how it can be identified in the future. The automation of this is particularly important for operational deployments.

9.2 Research Application

Network telescopes have gained in popularity since the inception of this research. Viewed as having a different goal to that filled by the various classes of honeypot and Honeynet systems, the network telescope is focussed on the Network and Transport layers of the OSI stack, rather than the application layer where the Honeypot technologies operate. Network Telescopes provide a very low risk means of data collection, provided they are correctly setup. They also scale well, as more address space is added, even on relatively large address spaces, modern commodity hardware should be more than sufficient. Long term data storage requirements are quite small given the lack of payloads on the majority of packets. In contrast IDS and IPS systems are frequently ignored due to the large volumes of information produced — appropriate configuration and tuning of the systems are key. Network telescopes can be used to gather a variety of information relating to the evolution of activity on the Internet at large.

The flexibility of the research that can be achieved with the data collected has been addressed in Section 2.8, and at key points throughout the analysis of the RUSCOPE1 dataset presented in this work. Even older datasets can be reprocessed with new tools, or with a different focus on the types of information be in evaluated, and may well yield surprising results. One of the key areas in future years is likely to be the comparative analysis of packet capturers and data from disparate systems, that cover the same temporal period. Being able to perform this kind of distributed analysis will help researches answer questions such as assessing the rate of spread, or the particular a rates of coverage of address space by particular netblocks or hosts. Further work with distributed datasets can help characterise the localised bias sensors my experience. A prime example motivating for distributed sensor placement was given in Section 7.4, where significantly different levels of traffic directed to 445/tcp were observed dependant on the address space being used. Certain telescopes such as the RUSCOPE2 sensor, observed almost no traffic relating to the Conficker spread.

Further exploratory work relating to network telescopes is described in Section 10.3.

9.3 Operational Application

The principle of a network telescope can be be applied to an operational context within an organisation, and incorporated into existent traditional operational network security components such as firewalls and IPS/IDS solutions. The deployment of a telescope in such an environment allows for the potential to help mitigate the flood of alerts that can be present from traditional security systems. A sample deployment is shown in Figure 9.1 where a telescope system is deployed in-line with traditional solutions.

In this case, data collected from the telescope address block(s) can be used to discriminate between plain sequential scanning, and possibly more targeted attacks. Similarly, persistently hostile networks can be identified, and pre-emptively

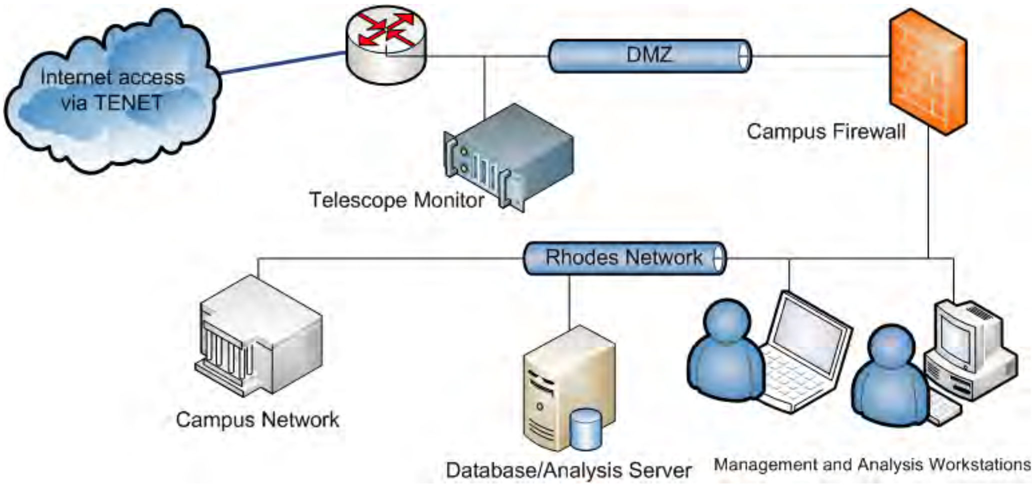


Figure 9.1: Sample Operational Telescope Deployment

filtered on border routers — alternately these networks could be routed to honeynet/sacrificial systems (much like the process of 'bait and switch' (Diebold *et al.*, 2005)). Another potential use is in the era of growing concern around economic and reputational risk, the analysis of backscatter traffic can provide an early warning to potential reputational damage due to spoofing of the organisation's address space.

9.3.1 Integration

Network telescopes can be integrated into an organisation's existing security framework in a number of ways. The first concern is where the telescope could be placed. While traditionally these have been outward looking, telescopes can also be applied with an inward focus (inside the firewall). Choosing to perform such a deployment can give early warning of potentially malicious activity, infections, or compromise of internal systems. The application of such a system byInternet Service Providers could lead to early identification and remediation, of malware infections on client systems. Free address blocks within the larger allocations made to providers would be a prime locality given the local-bias of much scanning activity, making the likelihood early detection and subsequent remediation much higher.

The data that a telescope collects can be used in a number of different ways. The recommendations below are generic, and would need to be tailored appropriately for an organisation performing an operational deployment.

Routing — Traffic observed matching given criteria could be aggregated and republished using protocols such as BGP. This in turn could allow for a number of different actions to be taken by suitable routing equipment at the organisational boundary, whether it be tagging of packets, or null-routing of offending blocks.

Firewall — The same principle as above can be applied, with the data being consumed by a firewall system, allowing for filtering of known noisy netblocks. Depending on actual configurations, data collected on a telescope can be used to determine the value of other security measures

IDS — Pre-filtering can be applied to IDS traffic, allowing for the removal of certain type of traffic observed on the telescope, but not relevant to the organisation. An example of this would be traffic destined to 445/tcp targeting a Unix web-server running on 80/tcp. If Telescopes are utilised across a number of sites, data can be aggregated, and flags raised if for example one address block scans distributed portions of IP address space operated by an organisation within a given period, as this could be indicative of targeted activity rather than just random scanning.

Email — The use of scoring mechanisms is already well established as part of the ongoing fight against SPAM. Collected information could be processed into RBL lists, or other formats suitable for consumption by mail-server infrastructure.

Overall is it critical to remember that addresses can and are easily spoofed. failure to compensate for this though the use of suitable threshold evaluation or scoring of inputs from multiple sources could easily result in a denial of service being performed by the organisation's own equipment.

9.3.2 Practical considerations

The primary practical consideration is what address space is available, and how it can be practically routed. Not all organisations may have complete control over the routing of their address space. Even in cases where direct routing of an address block cannot be achieved, a telescope can still be implemented using ARP, (either active, or with static ARP entries) to direct small numbers of addresses to a sensor.

The majority of organisations are likely to have small (/29) blocks of Address space available on their subnets. Suggested placement of these smaller blocks is at the start, end and possiibly the middle of larger allocations. Suitable policies will also need to be put in place regarding access to the data as well as the direct operational requirements of the sensor system.

9.4 Summary

This chapter has considered the application of network telescopes both as tools purely for research, as well as a means of augmenting an organisation's security posture. These two roles do not have to be mutually exclusive. Reflecting on the current state of IPv4 Addressing, where as of the time of writing only seven /8 network blocks remain free for allocation to regional registries, the likelihood of new telescope systems being deployed utilising large tracts of address space is fairly small. Approaches such as using greynets ((Baker *et al.*, 2010a), where a telescope operator makes use of smaller slices of space, and performs some kind of aggregation of the resulting data are far more likely to be adopted, particularly in the operational role.

This study concludes in the following Chapter, which revisits the research goals stated in Chapter 1, and considers how they have been addresses. Some reflection is also given to the work conducted and possible future research activities that can build on the this study.

I think computer viruses should count as life. I think it says something about human nature that the only form of life we have created so far is purely destructive. We've created life in our own image.

Stephen Hawking - Physicist

10 Conclusion

THE research presented in this work has assessed a large baseline of data comprising some 40 million captured events from the RUSCOPE1 dataset collected over 50 months, and which represents a relatively long period in 'Internet Time'. This dataset and the research based on it, is believed to be the first such large baseline study of a sensor network outside of the United States or Western Europe – the traditional 'homes' of the Internet. The continuous measurement over a long temporal baseline is another aspect that adds to the novel value of this study.

During the course of this study two novel graphical tools (InetVis and Hilbert Plots) were developed for the inspection and analysis of large IP traffic datasets. Work already being based on the research done in this thesis has been the development of high speed GPU based packet classifiers (Nottingham and Irwin, 2009a, 2010b) to aid in the problem of processing datasets of ever increasing size. The work done in the development of these tools, and the preliminary analysis of the data, have resulted in a number of publications at both a national and international level as mentioned in the preceding Chapters.

10.1 Document Recap

This section recaps what has been covered in the research presented so far.

Chapter 1 The scope of the research was detailed and an outline of the research methods and the specific areas of research focus provided.

Chapter 2 provided background in terms of other research that has been performed relating to Network Telescopes, as was the case for passive monitoring (Section 2.4.1) using these systems. Issues such as Active and Passive traffic classification (Section 2.3) and differing modes of telescope operation (Section 2.5) were discussed.

Chapter 3 Discussed the setup and configuration of the RUSCOPE1 network telescope system. Other primary data sources that contributed to this work were also discussed. Importantly this chapter also addressed some of the concerns expressed by the researcher's institution surrounding the establishment of the sensor.

Chapter 4 Describes a number of different tools and analytic methods that were used for the analysis of the collected data. While many of these were already existent, several new tools and techniques were developed specifically for this research. The three visualisation tools - Geopleth (Section 4.4), Hilbert Curve plots (Section 4.5) and InetVis (Section 4.6) were invaluable in gaining a high level appreciation of the large volumes of data under consideration.

Chapter 5 Explored the RUSCOPE1 dataset and analysed it from a number of points of view relating directly to the IP packet structure (Sections 5.1-5.4), looking at protocols, source addresses, datagram size and TTL value.

Chapter 6 Presented a higher level analysis of the meta data and ancillary attributes relating to the datagrams in the RUSCOPE1 dataset. The proposed distance score metric was evaluated along with the sources of recorded traffic analysed from geopolitical and topological viewpoints. An analysis of traffic originating from 'Bogon' IP addresses — both those that have been unallocated at the time of observation, or are classified as reserved — concludes the Chapter.

Chapter 7 Presented a detailed case study of traffic on a network telescope. The detailed discussion of traffic destined to 445/tcp covers two Internet worms Zotob/Rbot and Conficker/DownAdUp. The latter is covered in particular detail due to its significant impact on traffic volumes since its appearance, and the persistent presence that has been maintained.

Chapter 8 Proposed a number of metrics for use in describing both network telescopes (Section 8.1), and their resultant datasets (Section 8.2), by building on the analyses done in the previous Chapters. Various plotting schemes and graphical aids for communicating a high level overview of trend information were presented in Section 8.3.

Chapter 9 This Chapter presented an overview of the strategic and tactical applications of a network telescope in the context of a security solution within an organisation. The applications towards the use of such sensors in future Network and Systems Security related research is also considered.

10.2 Research objectives

The four primary research objectives that were stated in Chapter 1 are revisited below:

1. In support of the primary objective, the technologies and issues surrounding network telescopes were discussed and the merits of this means of network monitoring evaluated. These generally passive systems, when combined with active sensors such as intra-organisational Firewalls and Intrusion Detection Systems, and possibly in conjunction with inter-organisational information clearing systems such as Dshield, can provide for a comprehensive view of current threats facing the organisation.

2. A number of visual tools were developed for analysing the large volumes of input data from a variety of sensors. Static analysis tools and methods were provided that focused on geopolitical placement and network topology, which provided some insight into observed trends. Some evidence was also presented supporting the common 'wisdom' relating to much of the seemingly malicious traffic originating from China, United States (largely seen to be

netblocks allocated to residential providers), Western Europe, and South-East Asia. African networks scored particularly highly in packet prevalence on the captures analysed, but this was most likely largely due to the location of the network telescope within this address space. An interactive tool (InetVis) was also presented which allowed for time-sequence analysis of captured data and was used in the identification of a number of novel scanning techniques seen in the wild.

3. A high level framework for telescope management and operation was developed, drawing on the research areas above. Examples of improved mechanisms for information sharing were proposed.

10.3 Future Work

Much of the analysis and discussion relating to the value and impact of both this research and the associated technologies has been presented in Chapter 9. In conclusion however, it is felt that while many technological advances have been made in the ongoing fight against malicious digital code, the 'war' is far from over. For as long as humans develop software there will be flaws. And as long as there is an incentive to find these flaws, whether purely for 'street cred', academic interest, or possibly the most likely motivator — financial incentive — these flaws will be found and exploited. The solution is to understand the likelihood of this exploitation and prepare to mitigate and defend as best as possible in advance, while being cognisant of the fact that there is no perfect solution – other than possible total platform heterogeneity.

It is recognised that there are a number of areas touched on in this work which have not been fully explored and could form the basis of further research. In itself the research, as presented, could be repeated against other suitable datasets and comparative studies done. It is hoped that the metrics proposed in Chapter 8 will enable easier collaboration with other researchers performing this kind of analysis. Other specific significant and future research areas which can be drawn out of this work are:

- The sizing and placement of network telescopes will still need further exploration and experimental analysis. This does, however, require access to larger datasets, and increased flexibility in configuring sensors. This research would be of particular relevance to the use of network telescopes in an operational role, where larger netblocks of size /24 or greater may not be available.

- The issue of backscatter and passive traffic, which while addressed by much of the work done by CAIDA still requires further work. As discussed in Section 2.3, ICMP and TCP are relatively clear cut in determining the active/passive nature of traffic, but without protocol level enumeration UDP remains something of an enigma. The question around whether the SYN+ACK flag combination should be considered active or passive in a TCP context also needs further exploration and analysis.

- While constituting a very small portion of the overall traffic observed, research could be done into malformed and non protocol conforming datagrams that were observed. Some of these were discussed in Section 5.1.1. The initial investigation done on these showed that packet checksums matched the packets, suggesting that these datagrams may be formed by malfunctioning Network Address or Port Address translation gateways, or just poorly implemented software, rather than packet corruption 'on the wire'.

- As collaboration between researchers, organisations and intra-organisational use grows, a means needs to be developed to share suitable high level metrics between sensors in an automated manner. The metrics discussed in Chapter 8 may well serve as a basis, but the framework for performing the appropriate authorisation, authentication, anonymization and dissemination will still need to be constructed. In an operational context this same framework could be used as an additional means of defense against emerging threats.

In closing, Network Telescopes in their varying forms provide researchers working in the fields of information security and networking with insight into emerging trends on the global Internet, often allowing subtle variations and smaller scale incidents to be observed that would otherwise be missed using traditional methods such as IDS. Their value has been proven in the observation of global malware phenomena. One of the biggest challenges facing the application of this technology is the migration to the next generation of Internet Protocol Addressing – IP version 6. The sheer magnitude of the available address space, will probably force malware authors to reconsider scanning strategies. Nevertheless, the principles, skills and techniques learned by researchers operating on the current Internet will still be very much applicable.

References

Aben, E. *Conficker/Conflicker/Downadup as seen from the UCSD Network Telescope*. Online, CAIDA Network Telescope Project - Backscatter, February 2009.
URL http://www.caida.org/research/security/ms08-067/conficker.xml

Aben, E., Avila, S. C., and Claffy, K. *The caida ucsd network telescope two days in november 2008 dataset - 12th , 19th (collection)*. Online, CAIDA Network Telescope Project - Backscatter, 2008. Support for the UCSD Network Telescope "Two Days in November 2008" Dataset and the UCSD Network Telescope are provided by Cisco Systems, Limelight Networks, the US Department of Homeland Security, the National Science Foundation, and CAIDA Members.
URL http://www.caida.org/data/passive/telescope-2days-2008_dataset.xml

Acton, R., Friess, N., and Aycock, J. *Inverse geolocation: Worms with a sense of direction*. In *27th IEEE International Performance Computing and Communications Conference, IPCCC 07*, pages 487–493. New Orleans, LA, 2007. ISSN 1424411386 (ISBN); 9781424411382 (ISBN).

Albanna, Z., Almeroth, K., Meyer, D., and Schipper, M. *IANA Guidelines for IPv4 Multicast Address Assignments*. RFC 3171 (Best Current Practice), August 2001.
URL http://www.ietf.org/rfc/rfc3171.txt

Anagnostakis, K. G., Sidiroglou, S., Akritidis, P., Xinidis, K., Markatos, E., and Keromytis, A. D. *Detecting targeted attacks using shadow honeypots*. In *Proceedings of the 14th conference on USENIX Security Symposium - Volume 14*, pages 9–9. USENIX Association, Berkeley, CA, USA, 2005.
URL http://portal.acm.org/citation.cfm?id=1251398.1251407

Andreolini, M., Bulgarelli, A., Colajanni, M., and Mazzoni, F. *Honeyspam: honeypots fighting spam at the source*. In *SRUTI'05: Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop*, pages 11–11. USENIX Association, Berkeley, CA, USA, 2005.

Arkin, O. *ICMP Usage In Scanning*. Online, 2000. Black Hat Briefings, Amsterdam.
URL `http://www.blackhat.com/presentations/bh-europe-00/OfirArkin/OfirArkin.ppt`

Arkko, J. and Bradner, S. *IANA Allocation Guidelines for the Protocol Field*. Technical Report 5237, Internet Engineering Task Force, February 2008.
URL `http://www.ietf.org/rfc/rfc5237.txt`

Bailey, M., Cooke, E., Jahanian, F., Nazario, J., and Watson, D. *The Internet Motion Sensor: A distributed blackhole monitoring system*. In *Proceedings of Network and Distributed System Security Symposium (NDSS '05)*. San Diego, CA, February 2005a.

Bailey, M., Cooke, E., Jahanian, F., Provos, N., Rosaen, K., and Watson, D. *Data Reduction for the Scalable Automated Analysis of Distributed Darknet Traffic*. *Proceedings of the USENIX/ACM Internet Measurement Conference*, October 2005b.

Bailey, M., Cooke, E., Watson, D., Jahanian, F., and Nazario, J. *The Blaster Worm: Then and Now*. *IEEE Security & Privacy*, 3(4):26–31, 2005c.

Baker, F. *Requirements for IP Version 4 Routers*. RFC 1812 (Proposed Standard), June 1995. Updated by RFC 2644.
URL `http://www.ietf.org/rfc/rfc1812.txt`

Baker, F., Harrop, W., and Armitage, G. *IPv4 and IPv6 Greynets*. Technical Report 6018, Internet Engineering Task Force, September 2010a.
URL `http://www.ietf.org/rfc/rfc6018.txt`

Baker, F., Harrop, W., and Armitage, G. *IPv4 and IPv6 Greynets*. RFC 6018 (Informational), September 2010b.
URL `http://www.ietf.org/rfc/rfc6018.txt`

Barnett, R. and Irwin, B. *A framework for the rapid development of anomaly detection algorithms in network intrusion detection systems*. In *8th Annual Information Security South Africa (ISSA) Conference*. 6-9 July 2009. School of Tourism & Hospitality, University of Johannesburg, Auckland Park, Johannesburg, South Africa.

Barnett, R. J. and Irwin, B. *Towards a taxonomy of network scanning techniques*. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, SAICSIT '08, pages 1–7. ACM, New York, NY, USA, 2008. ISBN 978-1-60558-286-3. doi: http://doi.acm.org/10.1145/1456659.1456660.
URL `http://doi.acm.org/10.1145/1456659.1456660`

Beck, F., Festor, O., and State, R. *High Security Laboratory - Network Telescope*. Technical Report No. 9999, Institut National de Recherche en Informatique et en Automatiqu (INRIA), March 2007.
URL `http://hal.archives-ouvertes.fr/docs/00/33/75/68/PDF/Technical_Report_Network_Telescope.pdf`

Berghel, H. *Malware month*. *Commun. ACM*, 46:15–19, December 2003. ISSN 0001-0782. doi:http://doi.acm.org/10.1145/953460.953476.
URL `http://doi.acm.org/10.1145/953460.953476`

Bethencourt, J., Low, W. Y., Simmons, I., and Williamson, M. *Establishing darknet connections: an evaluation of usability and security*. pages 145–146, 2007. doi:http://doi.acm.org/10.1145/1280680.1280700.

Biddle, P., England, P., Peinado, M., and Willman, B. *The Darknet and the Future of Content Distribution*. In *ACM Workshop on Digital Rights Management*. November 2002.
URL `http://www.bearcave.com/misl/misl_tech/msdrm/darknet.htm#_ftn1`

Braden, R. *Requirements for Internet Hosts - Communication Layers*. RFC 1122 (Standard), October 1989. Updated by RFCs 1349, 4379, 5884, 6093.
URL `http://www.ietf.org/rfc/rfc1122.txt`

Cai, M., Hwang, K., Pan, J., and Papadopoulos, C. *Wormshield: Fast worm signature generation with distributed fingerprint aggregation*. *IEEE Transactions on Dependable and Secure Computing*, 4(2):88–104, 2007.

URL http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-33847742744&partnerID=40&rel=R7.0.0

Carnivore.IT. *Conficker does not like me?* Online Blog, 3 November 2009. Accessed 21 November 2010.
URL http://carnivore.it/2009/11/03/conficker_does_not_like_me

Carr, I., Clinton G. *Reverse Geographic Location of a Computer Node*. Master's thesis, Air Force Institute of Technology. Wright-Patterson Airforce Base OH, School of Engineering and Management, 2003.
URL http://handle.dtic.mil/100.2/ADA415378

Castaneda, F., Sezer, E. C., and Xu, J. *Worm vs. worm: preliminary study of an active counter-attack mechanism*. In *WORM '04: Proceedings of the 2004 ACM workshop on Rapid malcode*, pages 83–93. ACM Press, New York, NY, USA, 2004. ISBN 1-58113-970-5. doi:http://doi.acm.org/10.1145/1029618.1029631.

CERT. *CERT®Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL*. 2001. Http://www.cert.org/advisories/CA-2001-19.html.
URL http://www.cert.org/advisories/CA-2001-19.html

CERT/CC. *CERT®Advisory CA-2003-04 MS-SQL Server Worm*. online, January 2003.
URL http://www.cert.org/advisories/CA-2003-04.html

Chen, X., Andersen, J., Mao, Z., Bailey, M., and Nazario, J. *Towards an understanding of anti-virtualization and anti-debugging behavior in modern malware*. In *Dependable Systems and Networks With FTCS and DCC, 2008. DSN 2008. IEEE International Conference on*, pages 177 –186. 2008. doi: 10.1109/DSN.2008.4630086.

Chen, Z., Chen, C., and Ji, C. *Understanding localized-scanning worms*. In *27th IEEE International Performance Computing and Communications Conference, IPCCC 07*, pages 186–193. New Orleans, LA, 2007. ISSN 1424411386 (ISBN); 9781424411382 (ISBN).

Chen, Z. and Ji, C. *Importance-scanning worm using vulnerable-host distribution*. *48th Annual Global Telecommunications Conference, 2005. GLOBECOM '05. IEEE*, 3:6, 28 Nov.-2 Dec. 2005. doi:10.1109/GLOCOM.2005.1577955.

Chen, Z. and Ji, C. *Optimal worm-scanning method using vulnerable-host distributions*. *International Journal of Security and Networks: Special Issue on Computer and Network Security*, 2(1/2):71–80, March 2007. ISSN 1747-8405.
URL `http://portal.acm.org/citation.cfm?id=1359210.1359217`

Cheswick, B. *The design of a secure internet gateway*. In *in Proc. Summer USENIX Conference*, pages 233–237. 1990a. Http://www.cheswick.com/ches/papers/gateway.pdf.

Cheswick, B. *An evening with berferd in which a cracker is lured, endured, and studied*. In *In Proc. Winter USENIX Conference*, pages 163–174. 1990b.

Clark, D. *Subnetwork addressing scheme*. RFC 932, January 1985.
URL `http://www.ietf.org/rfc/rfc932.txt`

Conti, G. *Security Data Visualization: Graphical Techniques for Network Analysis*. No Starch Press, 1st edition, October 2007, 272 pages. ISBN 978-1593271435.

Conti, G. and Abdullah, K. *Passive visual fingerprinting of network attack tools*. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 45–54. ACM, New York, NY, USA, 2004. ISBN 1-58113-974-8. doi:http://doi.acm.org/10.1145/1029208.1029216.

Cooke, E., Bailey, M., Mao, Z., Watson, D., Jahanian, F., and McPherson, D. *Toward understanding distributed blackhole placement*. In *WORM'04 - Proceedings of the 2004 ACM Workshop on Rapid Malcode*, pages 54–64. Arbor Networks, 2004.
URL `http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-14944363571&partnerID=40&rel=R7.0.0`

Cooke, E., Jahanian, F., and McPherson, D. *The Zombie roundup: Understanding, detecting, and disrupting botnets*. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet (SRUTI 2005 Workshop)*. Cambridge, MA, July 2005.

Cooke, E. M. *Exposing Internet Address Use to Enhance Network Security*. Ph.D. thesis, The University of Michigan, Ann Arbor, MI, USA, May 2007.
URL `http://www.eecs.umich.edu/~emcooke/pubs/emcooke-thesis.pdf`

Cooperative Association for Internet Data Analysis (CAIDA) . *Measuring the use of ipv4 space with heatmaps*. Online, 6 October 2009. Last Accessed:

November 13 2010.
URL `http://www.caida.org/research/traffic-analysis/arin-heatmaps/`

Costa, M., Crowcroft, J., Castro, M., Rowstron, A., Zhou, L., Zhang, L., and Barham, P. *Vigilante: end-to-end containment of internet worms*. In *SOSP '05: Proceedings of the twentieth ACM symposium on Operating systems principles*, pages 133–147. ACM Press, New York, NY, USA, 2005. ISBN 1-59593-079-5. doi:http://doi.acm.org/10.1145/1095810.1095824.

Cowie, B. and Irwin, B. *Baseline statistical analysis of network telescope data*. In *Southern African Telecommunications Networks and Applications Conference (SATNAC)*. September 2010a. Spier Estate, South Africa.

Cowie, B. and Irwin, B. *Data classification for artificial intelligence construct training to aid in network incident identification using networktelescope data*. In *South African Institute of Computer Scientists and Information Technologists Conference (SAICSIT)*. October 2010b. Bela-Bela, South Africa.

DARPA. *Ethics and the Internet*. Technical Report 1087, Defense Advanced Research Projects Agency and Internet Activities Board and Internet Engineering Task Force, January 1989.
URL `http://www.ietf.org/rfc/rfc1087.txt`

Davis, M., Bodmer, S., and LeMasters, A. *Hacking Exposed: Malware & Rootkits Secrets & Solutions*. McGraw-Hill Osborne Media, 1st edition edition, September 2009. ISBN-10: 0071591184 ISBN-13: 978-0071591188.

DeNardis, L. *A history of internet security*. In **Leeuw, K. D. and Bergstra, J.**, editors, *The History of Information Security*, pages 681–704. Elsevier Science B.V., Amsterdam, 2007.
URL `http://www.sciencedirect.com/science/article/B8K61-4PW089R-D/2/703d307d1f58c533324f590aea4b6ab0`

Denning, P. J. *The internet worm*, pages 193–200. ACM, New York, NY, USA, 1990. ISBN 0-201-53067-8. doi:10.1145/102616.102629.

Diebold, P., Hess, A., and Schäfer, G. *A honeypot architecture for detecting and analyzing unknown network attacks*. In **Brauer, W., Müller, P., Gotzhein, R., and Schmitt, J.**, editors, *Kommunikation in Verteilten Systemen (KiVS)*,

Informatik aktuell, pages 245–255. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-27301-1.
URL `http://dx.doi.org/10.1007/3-540-27301-8_20`

eEye Digital Security. *ANALYSIS: Code Red II Worm*. Online, 4 August 2001a. Last Accessed: 2010-12-08.
URL `http://www.eeye.com/Resources/Security-Center/Research/Security-Advisories/AL20010804`

eEye Digital Security. *Analysis: .ida "code red" worm*. Online, 17 July 2001b. Last Accessed: 2009-11-08.
URL `http://www.eeye.com/Resources/Security-Center/Research/Security-Advisories/AL20010717`

eEye Digital Security. *Internet Security Systems PAM ICQ Server Response Processing Vulnerability*. Online, 18 March 2004. Last Accessed: 2009-11-08.
URL `http://www.eeye.com/Resources/Security-Center/Research/Security-Advisories/AD20040318`

Eisenberg, T., Gries, D., Hartmanis, J., Holcomb, D., Lynn, M. S., and Santoro, T. *The cornell commission: on morris and the worm*. *Commun. ACM*, 32:706–709, June 1989. ISSN 0001-0782.
URL `http://doi.acm.org/10.1145/63526.63530`

Ferguson, P. and Senie, D. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing*. Technical Report 2827, Internet Engineering Task Force, May 2000. Updated by RFC 3704.
URL `http://www.ietf.org/rfc/rfc2827.txt`

Ford, M., Stevens, J., and Ronan, J. *Initial results from an ipv6 darknet*. In *ICISP '06: Proceedings of the International Conference on Internet Surveillance and Protection*, page 13. IEEE Computer Society, Washington, DC, USA, 2006. ISBN 0-7695-2649-7. doi:http://dx.doi.org/10.1109/ICISP.2006.14.
URL `http://eprints.wit.ie/419/1/PID255945.pdf`

Fuller, V. and Li, T. *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*. RFC 4632 (Best Current Practice), August 2006.
URL `http://www.ietf.org/rfc/rfc4632.txt`

Fuller, V., Li, T., Yu, J., and Varadhan, K. *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*. Technical Report 1519, Internet Engineering Task Force, September 1993. Obsoleted by RFC 4632.
URL `http://www.ietf.org/rfc/rfc1519.txt`

Gardner, P. E. *The internet worm: What was said and when*. *Computers & Security*, 8(4):305–316, June 1989.
URL `http://www.sciencedirect.com/science/article/B6V8G-45K52VT-2V/2/7f7442324ee70556656b797e9af0ab3b`

Gerich, E. *Guidelines for Management of IP Address Space*. RFC 1466 (Informational), May 1993. Obsoleted by RFC 2050.
URL `http://www.ietf.org/rfc/rfc1466.txt`

Goebel, J., Holz, T., and Willems, C. *Measurement and analysis of autonomous spreading malware in a university environment*. 2007.
URL `http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-37849044998&partnerID=40&rel=R7.0.0`

Harder, U., Johnson, M. W., Bradley, J. T., and Knottenbelt, W. J. *Observing internet worm and virus attacks with a small network telescope*. *Electronic Notes in Theoretical Computer Science*, 151(3):47–59, June 2006. doi:http://dx.doi.org/10.1016/j.entcs.2006.03.011.
URL `http://www.sciencedirect.com/science/article/B75H1-4K674VG-4/2/edc43a47baf91551130e720c18d2a35e`

Harrop, W. and Armitage, G. *Defining and evaluating greynets (sparse darknets)*. In *LCN '05: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary*, pages 344–350. IEEE Computer Society, Washington, DC, USA, 2005a. ISBN 0-7695-2421-4. doi:http://dx.doi.org/10.1109/LCN.2005.46.

Harrop, W. and Armitage, G. *Greynets: a definition and evaluation of sparsely populated darknets*. In *MineNet '05: Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 171–172. ACM, New York, NY, USA, 2005b. ISBN 1-59593-026-4. doi:http://doi.acm.org/10.1145/1080173.1080177.

Hawkinson, J. and Bates, T. *Guidelines for creation, selection, and registration of an Autonomous System (AS)*. RFC 1930 (Best Current Practice), March 1996.
URL `http://www.ietf.org/rfc/rfc1930.txt`

Heidemann, J. and Pradkin, Y. *"mapping the internet address space" (poster)*. Online, August 2007.
URL `http://www.isi.edu/ant/address/`

Heidemann, J., Pradkin, Y., Govindan, R., Papadopoulos, C., and Bannister, J. *Exploring visible internet hosts through census and survey*. Technical Report ISI-TR-2007-640, USC/Information Sciences Institute, May 2007.
URL `http://www.isi.edu/~johnh/PAPERS/Heidemann07c.pdf`

Hick, P., Aben, E., Andersen, D., and Claffy, K. *The caida ucsd network telescope "three days of conficker" (collection)*. Online, CAIDA Network Telescope Project - Backscatter, 2009. Support for the UCSD Network Telescope "Three Days Of Conficker" Dataset and the UCSD Network Telescope are provided by Cisco Systems, Limelight Networks, the US Department of Homeland Security, the National Science Foundation, and CAIDA Members.
URL `http://www.caida.org/data/passive/telescope-3days-conficker_dataset.xml`

Highland, H. J. *The internet worm continued*. *Comput. Secur.*, 8:460–461, September 1989. ISSN 0167-4048.
URL `http://dx.doi.org/10.1016/0167-4048(89)90075-8`

Hu, J., Gao, J., and Rao, N. *Defending against internet worms using a phase space method from chaos theory*. In *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2007*, volume 6570, pages –. Orlando, FL, 2007. ISSN 0277786X (ISSN); 0819466921 (ISBN); 9780819466921 (ISBN).
URL `http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-35948933519&partnerID=40&rel=R7.0.0`

Hubbard, K., Kosters, M., Conrad, D., Karrenberg, D., and Postel, J. *Internet Registry IP Allocation Guidelines*. RFC 2050 (Best Current Practice), November 1996.
URL `http://www.ietf.org/rfc/rfc2050.txt`

Huston, G. and Michaelson, G. *Textual Representation of Autonomous System (AS) Numbers*. RFC 5396 (Proposed Standard), December 2008.
URL `http://www.ietf.org/rfc/rfc5396.txt`

IANA. *Special-Use IPv4 Addresses*. Technical Report 3330, Internet Engineering Task Force, September 2002.
URL `http://www.ietf.org/rfc/rfc3330.txt`

Internet Assigned Numbers Authority (IANA). *ICMP type numbers*. [online], 13 February 2008a. Accessed 2008-12-12.
URL `http://www.iana.org/assignments/icmp-parameters`

Internet Assigned Numbers Authority (IANA). *IPv4 Global Unicast Address Assignments*. [online], 11 2008b. Current Version Dated 2008-11-11 - Accessed 2008-11-30.
URL `http://www.iana.org/assignments/ipv4-address-space`

Internet Security Systems. *Snort RPC preprocessing vulnerability*. Online, 3 March 2003. Last Accessed: 2010-12-09.
URL `http://www.iss.net/threats/advise141.html`

Irwin, B. *TCP Sorcery*. Conference Presentation, 21 November 2009. 1st Annual ZaCon Security Conference, University of Johannesberg, South Africa.
URL `http://www.zacon.org.za/Archives/2009/slides/2009_zacon_Barry_Irwin.pdf`

Irwin, B. *Conficker: 687 days later*. Conference Presentation, 9 October 2010. 2nd Annual ZaCon Security Conference, University of Johannesberg, South Africa.
URL `http://www.zacon.org.za/Archives/2010/slides/2010_ZaCon_Barry_Irwin.pdf`

Irwin, B. and Barnett, R. *An analysis of logical network distance on observed packet counts for network telescope data*. In *Southern African Telecommunications Networks and Applications Conference (SATNAC)*. 31 Aug - 2 Sept 2009. Royal Swazi Spa, Swaziland, ISBN:978-0-620-44106-6.

Irwin, B. and Pilkington, N. *High level internet scale traffic visualization using hilbert curve mapping*. In **Conti, G., Goodall, J. R., and Ma, K.-L.**, editors, *VizSEC 2007 Proceedings of the Workshop on Visualization for Computer Security*, Mathematics and Visualization, pages 147–158. Springer, May 2008a. doi:10.1007/978-3-540-78243-8_10.

Irwin, B. and Pilkington, N. *Internet level visualization using hilbert curves*. In **Conti, G., Goodall, J. R., and Ma, K.-L.**, editors, *VizSEC 2007 Proceedings of the Workshop on Visualization for Computer Security*, Mathematics and Visualization, pages 147–158. Springer Berlin Heidelberg, May 2008b. doi: 10.1007/978-3-540-78243-8_10.

Irwin, B., Pilkington, N., Barnett, R., and Friedman, B. *A geopolitical analysis of long term internet telescope traffic*. In *10th Southern African Network and Applications Conference (SATNAC)*. Sugar Beach Resort, Mauritius, 2007.

Irwin, B. and van Riel, J.-P. *Inetvis: a graphical aid for the detection and visualisation of network scans*. In *2007 Workshop on Vizualisation for Cyber Security (VizSec2007)*. Hyatt Regency Hotel, Sacramento, California, 2007.

Janert, P. K. *Gnuplot in Action: Understanding Data with Graphs*. Manning Publications, August 2009, 396 pages. ISBN: 1933988398.
URL `http://www.manning.com/janert/`

Jaquith, A. *Security Metrics: Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley Professional, April 2007, 336 pages. ISBN 978-0321349989.

Josefsson, S. *The Base16, Base32, and Base64 Data Encodings*. RFC 4648 (Proposed Standard), October 2006.
URL `http://www.ietf.org/rfc/rfc4648.txt`

Karn, P. and Simpson, W. *ICMP Security Failures Messages*. Technical Report 2521, Internet Engineering Task Force, March 1999.
URL `http://www.ietf.org/rfc/rfc2521.txt`

Kessler, G. and Shepard, S. *A Primer On Internet and TCP/IP Tools*. RFC 1739 (Informational), December 1994. Obsoleted by RFC 2151.
URL `http://www.ietf.org/rfc/rfc1739.txt`

Klensin, J. *Reflections on the DNS, RFC 1591, and Categories of Domains*. RFC 3071 (Informational), February 2001.
URL `http://www.ietf.org/rfc/rfc3071.txt`

Komarnitsky, A. *nmap-web: port scanning made easy*. *Sys Admin*, 9(10):22–30, 2000. ISSN 1061-2688.

Kompella, R., Singh, S., and Varghese, G. *On scalable attack detection in the network. IEEE/ACM Transactions on Networking*, 15(1):14–25, 2007.
URL http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-33947507304&partnerID=40&rel=R7.0.0

Kumar, A., Paxson, V., and Weaver, N. *Exploiting underlying structure for detailed reconstruction of an internet-scale event*. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, IMC '05, pages 33–33. USENIX Association, Berkeley, CA, USA, 2005.
URL http://portal.acm.org/citation.cfm?id=1251086.1251119

Larkin, E. *The internet's public enemy number one. PC World (San Francisco CA)*, 25(12):67–68, 2007. ISSN 07378939 (ISSN).
URL http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-36649021087&partnerID=40&rel=R7.0.0

Lau, S. *The spinning cube of potential doom. Commun. ACM*, 47:25–26, June 2004. ISSN 0001-0782. doi:http://doi.acm.org/10.1145/990680.990699.
URL http://doi.acm.org/10.1145/990680.990699

Levy, E. *The making of a spam zombie army: Dissecting the sobig worms. IEEE Security and Privacy*, 1:58–59, July 2003. ISSN 1540-7993. doi:10.1109/MSECP.2003.1219071.
URL http://portal.acm.org/citation.cfm?id=939830.939937

Lu, N.-P. and Lin, S.-C. *An analysis of internet topology via traceroute sampling. Journal of the Chinese Institute of Engineers*, 32(1):123–128, 2009.
URL http://140.118.16.82/www/index.php/JCIE/article/viewFile/908/448

Lyon, G. F. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Nmap Project, 1 January 2009, 468 pages. ISBN: 978-0979958717.
URL http://nmap.org/book/

Marty, R. *Applied Security Visualization*. Addison-Wesley Professional, August 2008, 552 pages. ISBN 978-0321510105.

Mattsson, U. *Defending the database. Network Security*, 2007(7):14–17, July 2007.
URL http://www.sciencedirect.com/science/article/B6VJG-4P7P3KM-7/2/434b112158507de354717ee6240338b7

McCanne, S. and Jacobson, V. *The BSD packet filter: a new architecture for user-level packet capture*. In *Proceedings of the USENIX Winter 1993 Conference Proceedings on USENIX Winter 1993 Conference Proceedings*, pages 2–2. USENIX Association, Berkeley, CA, USA, 1993.
URL `http://staff.washington.edu/dittrich/papers/bpf-usenix93.ps`

McGraw, G. *Silver bullet talks with Mikko Hyppönen*. *IEEE Security and Privacy*, 5:8–11, November 2007. ISSN 1540-7993. doi:10.1109/MSP.2007.177.
URL `http://portal.acm.org/citation.cfm?id=1340076.1340092`

McRea, R. *Security visualization: What you don't see can hurt you*. *ISSA Journal*, 6(6):39–41, June 2008.
URL `http://holisticinfosec.org/toolsmith/docs/june2008.pdf`

Microsoft. *Microsoft Security Bulletin MS02-039: Buffer Overruns in SQL Server 2000 Resolution Service Could Enable Code Execution*. Online, Jjuly 2002.
URL `http://www.microsoft.com/technet/security/bulletin/ms02-039.mspx`

Microsoft. *MS03-026 : Buffer Overrun In RPC Interface Could Allow Code Execution (KB823980)*. Technical report, Microsoft, July 16 2003a. Originally posted: July 16, 2003 Revised: September 10, 2003.
URL `http://www.microsoft.com/technet/security/Bulletin/MS03-026.mspx`

Microsoft. *MS03-039 : Buffer Overrun In RPCSS Service Could Allow Code Execution (KB824146)*. Technical report, Microsoft, September 10 2003b.
URL `http://www.microsoft.com/technet/security/Bulletin/MS03-039.mspx`

Microsoft. *Virus alert about the Nachi worm (KB826234)*. Online, August 18 2003c.
URL `http://support.microsoft.com/kb/826234`

Microsoft. *MS04-011: Security Update for Microsoft Windows (KB835732)*. Technical report, Microsoft, April 13 2004. Updated: August 10, 2004.
URL `http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx`

Microsoft. *Microsoft Security Bulletin MS02-039: Vulnerability in Plug and Play Could Allow Remote Code Execution and Elevation of Privilege (899588)*. Online, August 9 2005.
URL `http://www.microsoft.com/technet/security/bulletin/ms05-039.mspx`

Microsoft. *MS06-040 : Vulnerability in Server Service Could Allow Remote Code Execution (KB921883)*. Technical report, Microsoft, September 12 2006.
URL `http://www.microsoft.com/technet/security/bulletin/ms06-040.mspx`

Microsoft. *[ms-msrp]: Messenger service remote protocol specification*. Online, March 2007. Last Revision 19 November 2010.
URL `http://msdn.microsoft.com/en-us/library/cc236303(PROT.13).aspx`

Microsoft. *MS08-067 : Vulnerability in Server Service Could Allow Remote Code Execution (KB958644)*. Technical report, Microsoft, Oct 23 2008a.
URL `http://www.microsoft.com/technet/security/Bulletin/MS08-067.mspx`

Microsoft. *Virus alert about the Win32/Conficker worm (KB962007)*. Online, August 18 2008b. Last Review: December 1, 2010 - Revision: 10.0.
URL `http://support.microsoft.com/kb/826234`

Microsoft. *Win32/conficker*. Online, 8 Jan 2009. Updated: Nov 10, 2010.
URL `http://www.microsoft.com/security/portal/Threat/Encyclopedia/Entry.aspx?Name=Win32/Conficker`

Minshall, G. *Tcpdpriv: a program for eliminating confidential information from packets collected on a network interface*. Online, 31 Oct 2005. Version 1.2.
URL `http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html`

Monsch, J. and Marty, R. *DAVIX: a live CD for data analysis and visualization*. online, Auugust 6 2008. Linux distribution.
URL `http://www.secviz.org/node/89`

Moore, D. *Network Telescopes: Observing Small or Distant Security Events*. Presentation, August 2002.
URL `http://www.caida.org/publications/presentations/2002/usenix_sec/`

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., and Weaver, N. *Inside the slammer worm*. *IEEE Security and Privacy*, 1(4):33–39, 2003. ISSN 1540-7993. doi:http://dx.doi.org/10.1109/MSECP.2003.1219056.
URL `http://portal.acm.org/citation.cfm?id=939830.939954`

Moore, D. and Shannon, C. *The CAIDA Dataset on the Code-Red Worms - July and August 2001, (collection)*. Online, August 2001.
URL `http://www.caida.org/data/passive/codered_worms_dataset.xml`

Moore, D., Shannon, C., and Claffy, K. *Code-red: a case study on the spread and victims of an internet worm*. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, IMW '02, pages 273–284. ACM, New York, NY, USA, 2002. ISBN 1-58113-603-X. doi:http://doi.acm.org/10.1145/637201.637244.
URL `http://doi.acm.org/10.1145/637201.637244`

Moore, D., Shannon, C., Voelker, G. M., and Savage, S. *Network telescopes*. Technical report, CAIDA, 2004.
URL `http://www.caida.org/publications/papers/2004/tr-2004-04/`

Moore, D., Voelker, G., and Savage, S. *Inferring internet denial-of-service activity*. In *In Proceedings of the 10th Usenix Security Symposium*, pages 9–22. 2001.
URL `http://www.caida.org/outreach/papers/2001/BackScatter/usenixsecurity01.pdf`

Mullen, T. M. *Defending Your Right to Defend*. In **Wyler, N. R.**, editor, *Aggressive Network Self-Defense*, chapter 10, pages 313–321. Syngress, Burlington, 2005. ISBN 9781931836203.
URL `http://www.sciencedirect.com/science/article/B85DB-4NY5T3W-4/2/9a54212be0645807028830449f04c674`

Munroe, R. *Map of the internet*. Online, 2006a.
URL `http://xkcd.com/195/`

Munroe, R. *Map of the internet*. Online, 11 December 2006b. Accessed: October 11 2010.
URL `http://blag.xkcd.com/2006/12/11/the-map-of-the-internet/`

Nahorney, B. *The downadup codex*. Online, March 2009.
URL `http://www.symantec.com/connect/sites/default/files/the_downadup_codex_ed1.pdf`

Nazario, J. *Defense and Detection Strategies against Internet Worms*. Artech House, Inc., Norwood, MA, USA, 2003. ISBN 1580535372.

Nottingham, A. and Irwin, B. *gpf: A GPU accelerated packet classification tool*. In *Southern African Telecommunications Networks and Applications Conference (SATNAC)*. 31 Aug - 2 Sept 2009a. Royal Swazi Spa, Swaziland, ISBN:978-0-620-44106-6.

Nottingham, A. and Irwin, B. *GPU Packet Classification using OpenCL: A Consideration of Viable Classification Methods*. In *South African Institute of Computer Scientists and Information Technologists Conference (SAICSIT)*. 13-14 October 2009b. Riverside Hotel and Conference Centre, Vaal River.

Nottingham, A. and Irwin, B. *Investigating the effect of genetic algorithms on filter optimisation within fast packet classifiers*. In *8th Annual Information Security South Africa (ISSA) Conference*. 6-9 July 2009c. School of Tourism & Hospitality, University of Johannesburg, Auckland Park, Johannesburg, South Africa.

Nottingham, A. and Irwin, B. *Conceptual design of a CUDA based packet classifier*. In *Southern African Telecommunications Networks and Applications Conference (SATNAC)*. September 2010a. Spier Estate, South Africa.

Nottingham, A. and Irwin, B. *Parallel packet classification using GPU co-processors*. In *South African Institute of Computer Scientists and Information Technologists Conference (SAICSIT)*. October 2010b. Bela-Bela, South Africa.

Oberheide, J., Karir, M., and Mao, Z. M. *Characterizing dark DNS behavior*. In *DIMVA '07: Proceedings of the 4th international conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 140–156. Springer-Verlag, Berlin, Heidelberg, 2007. ISBN 978-3-540-73613-4. doi: http://dx.doi.org/10.1007/978-3-540-73614-1_9.

Pang, R., Allman, M., Paxson, V., and Lee, J. *The devil and packet trace anonymization*. *SIGCOMM Comput. Commun. Rev.*, 36:29–38, January 2006. ISSN 0146-4833.
URL `http://doi.acm.org/10.1145/1111322.1111330`

Pang, R., Yegneswaran, V., Barford, P., Paxson, V., and Peterson, L. *Characteristics of internet background radiation*. In *IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 27–40. ACM, New York, NY, USA, 2004. ISBN 1-58113-821-0. doi:http://doi.acm.org/10.1145/1028788.1028794.

Paxson, V. *tcpslice - extract pieces of and/or merge together tcpdump files*. tcpdump.org. Manpage bundled as part of the software distribution.
URL `http://tcpdump.org/`

Paxson, V. *An analysis of the witty outbreak: exploiting underlying structure for detailed reconstruction of an internet-scale event*. In *Proceedings of the 2005 ACM workshop on Rapid malcode*, WORM '05, pages 51–51. ACM, New York, NY, USA, 2005. ISBN 1-59593-229-1.
URL `http://doi.acm.org/10.1145/1103626.1103636`

Pemberton, D. S. *An Empirical Study of Internet Background Radiation Arrival Density and Network Telescope Sampling Strategies*. Master's thesis, Victoria University of Wellington, Jan 2007.
URL `http://www.mcs.vuw.ac.nz/comp/graduates/archives/msc/Dean_Pemberton_MSC_Thesis.pdf`

Porras, P., Saidi, H., and Yegneswaran, V. *An analysis of conficker's logic and rendezvous points*. Technical report, SRI International, 4 February 2009. Last Update 19 March 2009.
URL `http://mtc.sri.com/Conficker/`

Portokalidis, G. and Bos, H. *Sweetbait: Zero-hour worm detection and containment using low- and high-interaction honeypots*. *Computer Networks*, 51(5):1256–1274, April 2007.
URL `http://www.sciencedirect.com/science/article/B6VRG-4M3RMF2-1/2/e2ea4b0494495451218a40dc6baed6c8`

Postel, J. *User Datagram Protocol*. RFC 768 (Standard), August 1980.
URL `http://www.ietf.org/rfc/rfc768.txt`

Postel, J. *Internet Control Message Protocol*. Technical Report 792, Internet Engineering Task Force, September 1981a. Updated by RFCs 950, 4884.
URL `http://www.ietf.org/rfc/rfc792.txt`

Postel, J. *Internet Control Message Protocol*. RFC 777, April 1981b. Obsoleted by RFC 792.
URL `http://www.ietf.org/rfc/rfc777.txt`

Postel, J. *Internet Protocol*. Technical Report 791, Internet Engineering Task Force, September 1981c. Updated by RFC 1349.
URL `http://www.ietf.org/rfc/rfc791.txt`

Postel, J. *Transmission Control Protocol*. RFC 793 (Standard), September 1981d. Updated by RFCs 1122, 3168, 6093.
URL `http://www.ietf.org/rfc/rfc793.txt`

Postel, J. *Domain Name System Structure and Delegation*. RFC 1591 (Informational), March 1994.
URL `http://www.ietf.org/rfc/rfc1591.txt`

Pouget, F., Dacier, M., and Pham, V. *Understanding threats: A prerequisite to enhance survivability of computing systems*. *Int. J. Crit. Infrastruct.*, 4(1-2):153–171, 2008. ISSN 14753219.
URL `http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-37849016863&partnerID=40&rel=R7.0.0`

Provos, N. *A virtual honeypot framework*. *Proceedings of the 13th USENIX Security Symposium*, 1:1–15, 2004.

Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and Lear, E. *Address Allocation for Private Internets*. Technical Report 1918, Internet Engineering Task Force, February 1996.
URL `http://www.ietf.org/rfc/rfc1918.txt`

Reynolds, J. *Assigned Numbers: RFC 1700 is Replaced by an On-line Database*. Technical Report 3232, Internet Engineering Task Force, January 2002.
URL `http://www.ietf.org/rfc/rfc3232.txt`

Reynolds, J. and Postel, J. *Assigned Numbers*. Technical Report 1700, Internet Engineering Task Force, October 1994. Obsoleted by RFC 3232.
URL `http://www.ietf.org/rfc/rfc1700.txt`

Richard, M. and Ligh, M. *making fun of your malware*. Conference Presentation Defcon 17, Las Vegas USA, August 2009.
URL `https://www.defcon.org/images/defcon-17/dc-17-presentations/defcon-17-michael_ligh-matt_richard-making_fun_of_malware.pdf`

Richter, J. P. F. *An Investigation into the Design and Implementation of an Internet-Scale Network Simulator*. Masters thesis, Department of Computer Science, Rhodes University, Grahamstown, South Africa, December 2008.
URL `http://eprints.ru.ac.za/1709/1/Richter-MSc-TR09-107.pdf`

Roberts, P. F. *ISS reports snort vulnerability*. 4 March 2003. Last Accessed 2010-12-04.
URL `http://www.infoworld.com/d/security-central/iss-reports-snort-vulnerability-984`

Sanders, C. *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems*. No Starch Press, San Fransisco, May 2007, 172 pages. ISBN: 978-1-59327-149-7.

Schneier, B. *The zotob storm*. *IEEE Security and Privacy*, 3:96–, November 2005. ISSN 1540-7993. doi:B330C7C3-614A-43A2-BCCA-586B1F5B0E2B.
URL `http://portal.acm.org/citation.cfm?id=1106275.1106309`

Schultz, E. E. *The msblaster worm: going from bad to worse*. *Network Security*, 2003(10):4 – 8, 2003a. ISSN 1353-4858. doi:DOI: 10.1016/S1353-4858(03)01005-5.
URL `http://www.sciencedirect.com/science/article/B6VJG-49TNTM1-5/2/97d615d1f4b019c212242b8b5c81a52a`

Schultz, E. E. *The sobig worm variants: Letter after letter from a-e*. *Network Security*, 2003(8):7–10, August 2003b.
URL `http://www.sciencedirect.com/science/article/B6VJG-4996KNF-7/2/1cf7b5f86324aa6fe43c6edf801f398c`

Schwagele, C. *dotNetVis: An enhancement and .NET re-implementation of the InetVis Data Visualisation Tool*. Honours thesis, Rhodes University, Department of Computer Science, November 2010.
URL `http://www.cs.ru.ac.za/research/g07s3491/downloads/WriteUp/Writeup.pdf`

SecuriTeam. *Tcpdump isakmp denial of service exploit release*. 10 March 2003. Last Accessed 2010-12-09.
URL `http://www.securiteam.com/exploits/5KP0J009FO.html`

SecuriTeam. *Tcpdump remote denial of service exploit*. 9 June 2005. Last Accessed 2010-12-09.
URL `http://www.securiteam.com/exploits/5GP012KG0S.html`

SecuriTeam. *Wireshark DNP3 dissector infinite loop vulnerability*. 30 August 2007. Last Accessed 2010-12-09.
URL `http://www.securiteam.com/securitynews/5LP0V00MAI.html`

SecuriTeam. *Wireshark RMI packet dissector information disclosure*. 6 August 2008. Last Accessed 2010-12-09.
URL `http://www.securiteam.com/securitynews/5YP011PP5W.html`

ShadowServer. *Conficker*. 2010.
URL `http://www.shadowserver.org/wiki/pmwiki.php/Stats/Conficker`

Shannon, C. and Moore, D. *The spread of the witty worm*. *IEEE Security and Privacy*, 2:46–50, July 2004a. ISSN 1540-7993. doi:10.1109/MSP.2004.59.
URL `http://portal.acm.org/citation.cfm?id=1018027.1018275`

Shannon, C. and Moore, D. *The CAIDA Dataset on the Witty Worm - March 19-24, 2004, (collection)*. Online, March 2004b.
URL `http://www.caida.org/data/passive/witty_worm_dataset.xml`

Shannon, C., Moore, D., and Aben, E. *The CAIDA Backscatter-2004-2005 Dataset - May 2004 - November 2005, (collection)*. Online, , CAIDA Network Telescope Project - Backscatter, 2005.
URL `http://www.caida.org/data/passive/backscatter_2004_2005_dataset.xml`.

Shannon, C., Moore, D., and Aben, E. *The CAIDA Backscatter-2006 Dataset - February 2006 - November 2006, (collection)*. Online, CAIDA Network Telescope Project - Backscatter, 2006.
URL `http://www.caida.org/data/passive/backscatter_2006_dataset.xml`

Shannon, C., Moore, D., and Aben, E. *The CAIDA Backscatter-2007 Dataset - January 2007 - November 2007, (collection)*. Online, CAIDA Network Telescope Project - Backscatter, 2007.
URL `http://www.caida.org/data/passive/backscatter_2007_dataset.xml`

Shinoda, Y., Ikai, K., and Itoh, M. *Vulnerabilities of passive internet threat monitors*. In *SSYM'05: Proceedings of the 14th conference on USENIX Security Symposium*, pages 14–14. USENIX Association, Berkeley, CA, USA, 2005.

Sinha, S., Bailey, M., and Jahanian, F. *Shedding light on the configuration of dark addresses*. In *Proceedings of the Network and Distributed System Security Symposium, (NDSS '07)*. Internet Society, San Diego, California, USA,, 28th February - 2nd March 2007. doi:doi=10.1.1.110.6327.
URL `http://www.eecs.umich.edu/~mibailey/publications/ndss07_final.pdf`

Spafford, E. H. *Crisis and aftermath*. *Commun. ACM*, 32:678–687, June 1989a. ISSN 0001-0782.
URL `http://doi.acm.org/10.1145/63526.63527`

Spafford, E. H. *The internet worm program: an analysis*. *SIGCOMM Comput. Commun. Rev.*, 19:17–57, January 1989b. ISSN 0146-4833.
URL `http://doi.acm.org/10.1145/66093.66095`

Spitzner, L. *To Build A Honeypot*. online, 4 August 1999.
URL `http://www.spitzner.net/honeypot.html`

Stavrou, A., Cook, D. L., Morein, W. G., Keromytis, A. D., Misra, V., and Rubenstein, D. *Websos: an overlay-based system for protecting web servers from denial of service attacks*. *Computer Networks*, 48(5):781–807, August 2005.
URL `http://www.sciencedirect.com/science/article/B6VRG-4FH5GVF-1/2/46fb8cb3ac64c3e042fcefaeb4661fad`

Stevens, W. R. *TCP/IP illustrated (vol. 1): The Protocols*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1993. ISBN 0-201-63346-9.

Stoll, C. *The cuckoo's egg: tracking a spy through the maze of computer espionage*. Doubleday, New York, NY, USA, 1st edition, September 1989. ISBN 0-385-24946-2, 326 pages.

Sun, L., Ebringer, T., and Boztag, S. *An automatic anti-anti-vmware technique applicable for multi-stage packed malware*. In *Malicious and Unwanted Software, 2008. MALWARE 2008. 3rd International Conference on*, pages 17 –23. 2008. doi:10.1109/MALWARE.2008.4690853.

Team Cymru. *The bogon reference*. [online], 2008. Accessed 2008-11-30.
URL `http://www.team-cymru.org/Services/Bogons/`

Team Cymru. *The team cymru bogon list v5.0*. [online], November 2009.
URL `http://www.cymru.com/Documents/bogon-list.html`

Teoh, S. T., Ma, K.-L., Wu, S. F., and Zhao, X. *Case study: Interactive visualization for internet security*. In *VIS '02: Proceedings of the conference on Visualization '02*. IEEE Computer Society, Washington, DC, USA, 2002. ISBN 0-7803-7498-3.

The Electronic Souls Crew. *tcpdump exploit*. Fulldisclosure Mailing list, 29 November 2002.
URL `http://seclists.org/fulldisclosure/2002/Nov/381`

Tsuchiya, P. *On the assignment of subnet numbers*. Technical Report 1219, Internet Engineering Task Force, April 1991.
URL `http://www.ietf.org/rfc/rfc1219.txt`

Tufte, E. *Beautiful Evidence*. Graphics Press, 2004.

van Riel, J.-P. *Internet Traffic Visualisation: IDS traffic and Darknets*. Honours Thesis, Rhodes University, Department of Computer Science, November 2005.

van Riel, J.-P. and Irwin, B. *Identifying and investigating intrusive scanning patterns by visualizing network telescope traffic in a 3-d scatter-plot*. In *Proceedings of 6th Annual Information Security South Africa (ISSA)*. Balalaika Hotel, Sandton, South Africa, 5–7 July 2006a.

van Riel, J.-P. and Irwin, B. *Inetvis, a visual tool for network telescope traffic analysis*. In *Proceedings of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, AFRIGRAPH '06, pages 85–89. ACM, New York, NY, USA, 2006b. ISBN 1-59593-288-7.
URL `http://doi.acm.org/10.1145/1108590.1108604`

van Riel, J.-P. and Irwin, B. *Toward visualised network intrusion detection*. In *Proceedings of 9th Annual Southern African Telecommunication Networks and Applications Conference (SATNAC2006)*. Spier Wine Estate, Western Cape, South Africa, 3-6 September 2006c. Poster Paper.

Vanderavero, N., Brouckaert, X., Bonaventure, O., and Le Charlier, B. *The honeytank: A scalable approach to collect malicious internet traffic*. *International Journal of Critical Infrastructure*, 4(1-2):185–205, 2008. ISSN 14753219 (ISSN).

Vohra, Q. and Chen, E. *BGP Support for Four-octet AS Number Space*. RFC 4893 (Proposed Standard), May 2007.
URL `http://www.ietf.org/rfc/rfc4893.txt`

VUPEN Security. *Wireshark LWRES dissector multiple buffer overflow vulnerabilities*. 28 January 2010. Last Accessed 2010-12-09.
URL `http://www.vupen.com/english/advisories/2010/0239`

Weaver, N., Hamadeh, I., Kesidis, G., and Paxson, V. *Preliminary results using scale-down to explore worm dynamics*. In *WORM '04: Proceedings of the 2004 ACM workshop on Rapid malcode*, pages 65–72. ACM Press, New York, NY, USA, 2004. ISBN 1-58113-970-5. doi:http://doi.acm.org/10.1145/1029618.1029628.

Wei, S. and Mirkovic, J. *Correcting congestion-based error in network telescope's observations of worm dynamics*. In *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, IMC '08, pages 125–130. ACM, New York, NY, USA, 2008. ISBN 978-1-60558-334-1.
URL `http://doi.acm.org/10.1145/1452520.1452536`

Wessels, D. *Ipv4 heatmaps: BGP route advertisements*. Online, 2007.
URL `http://maps.measurement-factory.com/gallery/Routeviews/2007`

White, D. *Ms05-039 and the zotob summary*. Online, 18 August 2005. Last accessed 2010-12-01.
URL `http://singe.za.net/blog/archives/510-MS05-039-and-the-Zotob-summary.html`

Wustrow, E., Karir, M., Bailey, M., Jahanian, F., and Huston, G. *Internet background radiation revisited*. In *Proceedings of the 10th annual conference on Internet measurement*, IMC '10, pages 62–74. ACM, New York, NY, USA, 2010. ISBN 978-1-4503-0483-2. doi:http://doi.acm.org/10.1145/1879141.1879149.
URL `http://doi.acm.org/10.1145/1879141.1879149`

Xu, J., Fan, J., Ammar, M., and Moon, S. B. *On the design and performance of prefix-preserving IP traffic trace anonymization*. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, pages 263–266. ACM, New York, NY, USA, 2001. ISBN 1-58113-435-5. doi:http://doi.acm.org/10.1145/505202.505234.
URL `http://doi.acm.org/10.1145/505202.505234`

Zou, C., Duffield, N., Towsley, D., and Gong, W. *Adaptive defense against various network attacks*. *IEEE Journal on Selected Areas in Communications*, 24(10):1877–1887, 2006a.
URL `http://www.scopus.com/scopus/inward/record.url?eid=2-s2.0-33749830235&partnerID=40&rel=R7.0.0`

Zou, C., Towsley, D., and Gong, W. *On the performance of internet worm scanning strategies*. *Performance Evaluation*, 63:700–723, 2006b. doi:doi:10.1016/j.peva.2005.07.032.

Zou, C. C., Gong, W., Towsley, D., and Gao, L. *The monitoring and early detection of internet worms*. *IEEE/ACM Trans. Netw.*, 13(5):961–974, 2005. ISSN 1063-6692. doi:http://dx.doi.org/10.1109/TNET.2005.857113.

Glossary

ACK Acknowledgement - a flag used in TCP communications

AfriNIC African Network Information Centre

API Application Programing Interface

APNIC Asia-Pacific Network Information Centre

ARIN American Registry for Internet Numbers

ARP Address Resolution Protocol

ARPANET Advanced Research Projects Agency Network - a precurser to the Internet

AS Autonomous System

ASN Autonomous System Number

backscatter The constant stream of traffic receievd by computers on the internet that is not part of the operational or production network traffic.

BCP Best Current Practice

BGP Border Gateway Protocol

BPF BSD Packet Filter

CAIDA Cooperative Association for Internet Data Analysis

CERT Computer Emergency Resonse Team

CSS Cascading Style Sheets

CSV Comma Separated Value

CWG Conficker Working Group - www.confickerworkinggroup.org

DARPA Unites States of America's Defense Advanced Research Projects Agency

DCE/RPC Distributed Computing Environment / Remote Procedure Calls

DDoS Distributed Denial of Service

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

dotted-quad IP addresses written in the form AAA.BBB.CCC.DDD

DRM Digital Rights Management

GMT Grenwich Mean Time

GPU Graphics Processing Unit

HSV Hue, Saturation and Value - a cylindrical colour co-ordinate system

IANA Internet Assigned Numbers Authority

IBR Internet Background Radiation

ICMP Internet Control Message Protocol

IDS Intrusion Detection System

IETF Internet Engineering Task Force

IPS Intrusion Prevention System

ISP Internet Service Providers

LACNIC Latin America and Caribbean Network Information Centre

lzma Lempel-Ziv-Markov chain algorithm (LZMA) is an algorithm used to perform data compression

MD Message Digest - a cryptographic hash family.

MTU Maximum Transmission Unit

NAT	Network Address Translation
NIDS	Network Intrusion Detection System
NTP	Network Time Protocol
OSI	Open Systems Interconnect
p2p	Peer to Peer
ppd	Packets per Day
ppm	Packets per Minute
pps	Packets per Second
RBL	Real-time Block List - commonly also referred to DNS Block Lists
RFC	Request for Comments
RIPENCC	Réseaux IP Européens Network Coordination Centre, also known as RIPE
RST	Reset - a flag used in TCP communications
SAST	South African Standard Time
SHA	Secure Hash Algorithim - a cryptographic hash family.
sha1	Secure Hash Algorithim
SME	Small and Medium Enterprise
SMTP	Simple Mail Transport Protocol
SPAM	Unsollicited communication, usually of a commercial nature.
SQL	Structured Query Language
SVG	Scalable Vector Graphic
SYN	Synchronise - a flag used in TCP communications
TCP	Transmission Control Protocol
TTL	Time to Live

UCSD	University of California San Diego
UDP	User Datagram Protocol
URN	Uniform Resource Name
UTC	Universal Co-ordinated Time

Appendices

A

Major Worms

This timeline is intended to provide a selected overview of Malware that has spread via the Internet, with the distinction that the spread was primarily via a network, but using using protocols other than email. Much of this has been sourced from the community maintained page on Wikipedia[1], and augmented with other resources the researcher has come across. Readers are referred to the aforementioned page for a much larger and more detailed list.

Several of these worms, target the RPC/DCOM stack on Microsoft Windows platforms. Readers are referred to the discussion around these vulnerabilities in Chapter 7.

- **1988**
  - *November 2* : The Morris worm, created by Robert Tappan Morris Junior, infects DEC VAX and Sun machines running BSD UNIX connected to the Internet, and becomes the first worm to spread extensively "in the

[1] `http://en.wikipedia.org/wiki/Timeline_of_notable_computer_viruses_and_worms` - Accessed 2010-11-27

wild", and one of the first well-known programs exploiting buffer overrun vulnerabilities.

- **2001**
  - *May 8*: The Sadmind worm spreads by exploiting holes in both Sun Solaris and Microsoft IIS.
  - *July*: The Sircam worm is released, spreading through e-mails and unprotected network shares.
  - *July 13*: The Code Red worm attacking the Index Server ISAPI Extension in Microsoft Internet Information Services is released.
  - *August 4*: A complete re-write of the Code Red worm, Code Red II begins aggressively spreading, primarily in China.
  - *September 18*: The Nimda worm is discovered and spreads through a variety of means including vulnerabilities in Microsoft Windows and backdoors left by Code Red II and Sadmind worm.
  - *October 26*: The Klez worm is first identified.
- **2002**
  - *July*: Scalper Worm - Infected Vulnerable versions of the Apache webserver on FreeBSD systems. Maintined a list of infected nodes in memory to allow for remote command and control. One of the first to provide a self update/upgrade ability. Wile not widespread it was the first significant worm to target FreeBSD
  - *September*: Slapper Worm attacked linux systems though an vulnerability in OpenSSL and Apache. The mode of operation was similar the Scalper worm and Code Red.
- **2003**
  - *January 24*: The SQL slammer worm, also known as the Sapphire worm, attacks vulnerabilities in Microsoft SQL Server and MSDE causes widespread problems on the Internet. Exploited port 1434/udp. Entire worm fitted in a single 418 byte packet.
  - *August 12*: The Blaster worm, also known as the Lovesan worm, spreads rapidly by exploiting a vulnerability in system services present on Microsoft Windows computers.

- *August 18*: The Welchia (Nachi) worm is discovered. The worm tries to remove the Blaster worm and patch Windows.
- *August 19*: The Sobig worm (technically the Sobig.F worm) spreads rapidly via mail and network shares.
- *October 24*: The Sober worm is first seen and maintains its presence until 2005 with many new variants.

- **2004**

  - *March 19*: The Witty worm is a record-breaking worm in many regards. It exploited holes in several Internet Security Systems (ISS) products. It was the fastest disclosure to worm, it was the first internet worm to carry a destructive payload and it spread rapidly using a pre-populated list of ground-zero hosts.
  - *May 1*: The Sasser worm emerges by exploiting a vulnerability in LSASS and causes problems in networks, even interrupting business in some cases.
  - *December*: Santy, the first known "webworm" is launched. It exploited a vulnerability in phpBB and used Google in order to find new targets. It infected around 40000 sites before Google filtered the search query used by the worm, preventing it from spreading.

- **2005**

  - *August 16*: The Zotob worm and several variations of malware using the same vulnerability are discovered. The effect was overblown because several United States media outlets were infected.
  - *October 13*: The Samy Javascript worm using XSS as an infection vector became the fastest spreading virus by some definitions as of 2006, spreads on Myspace.com, placing servers under severe load.

- **2006**

  - *Late Septembe*r: Stration or Warezov worm first discovered.

- **2007**

  - *Jun 29*: An XSS Worm known as JTV.worm was initiated by a security group known as n0ths affecting Justin.tv, infecting 2,525 profiles within 24 hours. The worm was used for research purposes and the security team released detailed information never-before researched about the factors that affect XSS worms.

- **2008**

  - *July 31*: The Koobface computer worm targets users of Facebook and MySpace. New variants constantly appear
  - *November 20*: Conficker A targets systems running Microsoft Windows family operating systems, exploiting a flaw in the RPC/DCOM stack. This was in response to the publication of the MS08-067 Security advisory on 23rd October 2009. This in turn was in response to the xploit being seein in the wild. A detailed chronology is contained in Chapter 7.
  - *December 28*: Conficker B emerges

- **2009**

  - *February 20*: Conficker C emerges
  - *March* 4: Conficker D emerges
  - *April* 8: Conficker E emerges

- **2010**

  - *June 17*: Stuxnet, a Windows trojan, was detected. It is the first worm to attack SCADA systems. Some suggest targets Iranian nuclear facilities. It is cryptographically signed with a valid certificate from Realtek, used by the company to sign its hardware drivers. This indicates compromise within this organisation. The worm used a number of 0-day and legacy vulnerabilities in windows systems to spread.

B

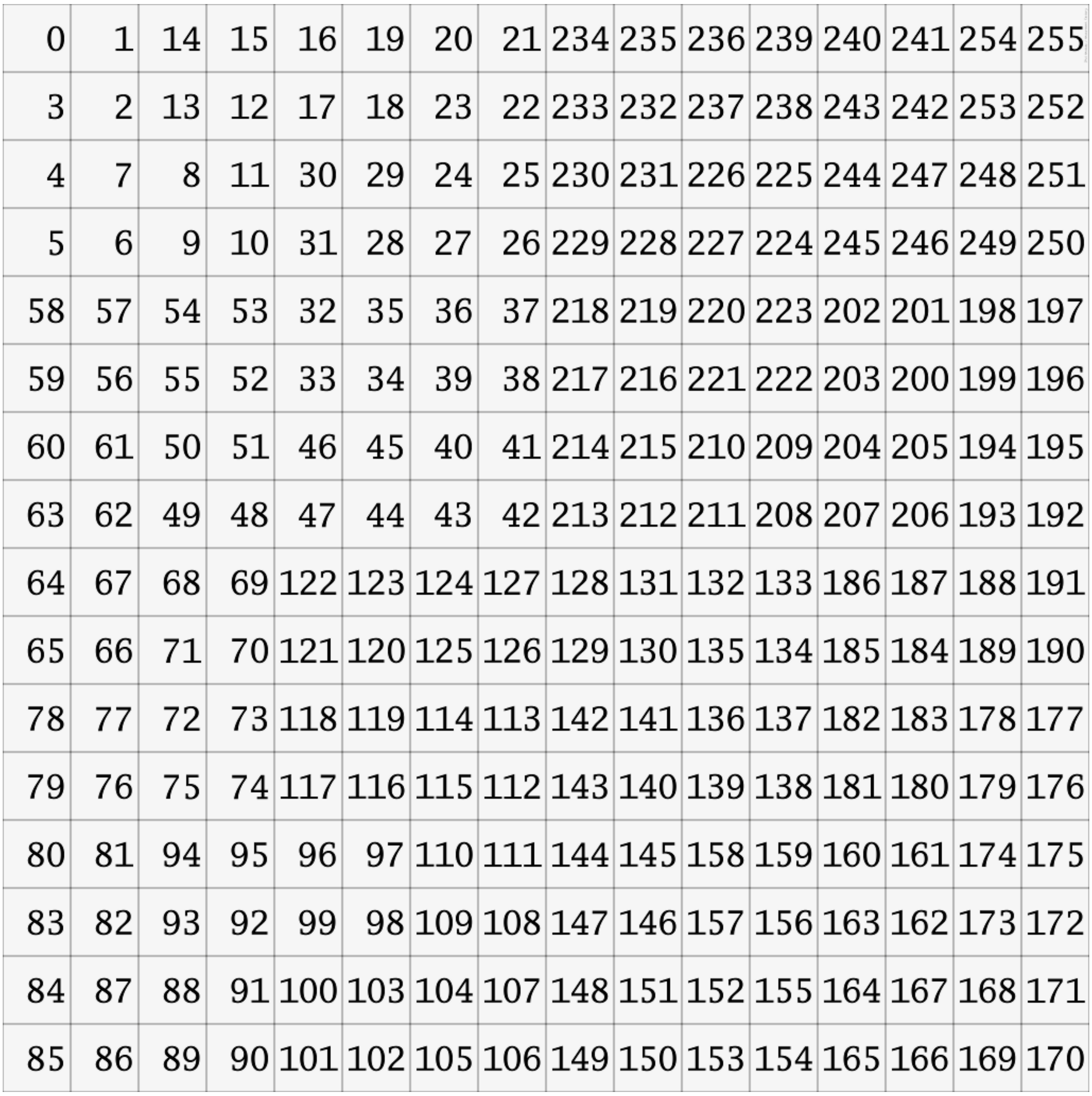
Hilberts

This section provides a quick overview of the interpretation of Hilbert Curve plots. The most important thing to consider is the orientation of the plot, as different tools (as discussed in Section 4.5) may use different starting points within the grid. Care should also be taken when interpreting subsections of the curve a the order increases as within each grid block, the orientation of the entry and exit nodes may change. The overall shape of the curve however remains the same.

For the purposes of this study, all curves are oriented with the equivalent of 0 (or 0.0.0.0) being in the top left-hand corner, and the curve progressing to complete wit 255 (or 255.255.255.255) in the top right. An overview of the numeric placements of the blocks on a 4^{th} order Hilbert Curve are shown in Figure B.1. This figure can be used as a guideline even when interpreting higher order curves, and the overall allocation of the high level blocks stays the same. As discussed, a 4^{th} order Hilbert Curve passes though 256 nodes (these are conveniently numbered from 0 to 255). This maps cleanly to the number of /8 networks present on the IPv4 Internet.

Should a higher order curve be used, each of these blocks in turn can be broken down into 256 component blocks. Should a 8^{th} order Hilbert Curve, be used, a total of 65 536 blocks will be present on the grid, which correlates to the total number of netblocks of size /16 (each /8 having 256 component blocks). The map can be re-applied on a smaller scale, taking rotation into account. The same principle can be applied to higher order curves.

Figure B.2 shows the actual path that the curve of 4^{th} order takes though a grid of blocks. Sub-blocks in a higher order curve would be oriented to conform to the entry and exit points of the lower order curve.



0	1	14	15	16	19	20	21	234	235	236	239	240	241	254	255
3	2	13	12	17	18	23	22	233	232	237	238	243	242	253	252
4	7	8	11	30	29	24	25	230	231	226	225	244	247	248	251
5	6	9	10	31	28	27	26	229	228	227	224	245	246	249	250
58	57	54	53	32	35	36	37	218	219	220	223	202	201	198	197
59	56	55	52	33	34	39	38	217	216	221	222	203	200	199	196
60	61	50	51	46	45	40	41	214	215	210	209	204	205	194	195
63	62	49	48	47	44	43	42	213	212	211	208	207	206	193	192
64	67	68	69	122	123	124	127	128	131	132	133	186	187	188	191
65	66	71	70	121	120	125	126	129	130	135	134	185	184	189	190
78	77	72	73	118	119	114	113	142	141	136	137	182	183	178	177
79	76	75	74	117	116	115	112	143	140	139	138	181	180	179	176
80	81	94	95	96	97	110	111	144	145	158	159	160	161	174	175
83	82	93	92	99	98	109	108	147	146	157	156	163	162	173	172
84	87	88	91	100	103	104	107	148	151	152	155	164	167	168	171
85	86	89	90	101	102	105	106	149	150	153	154	165	166	169	170

Figure B.1: 4^{th} order Hilbert Curve Map

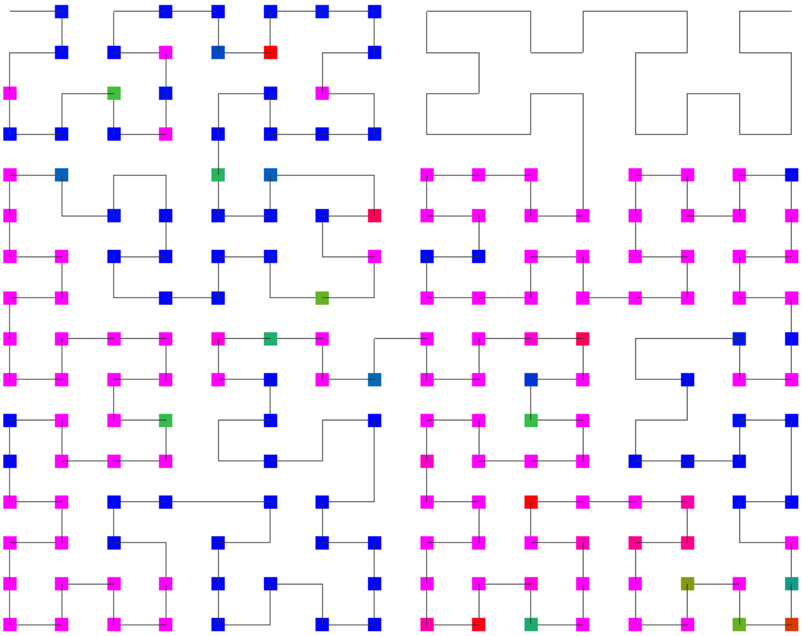


Figure B.2: 4^{th} order Hilbert Curve showing plotting path for class A (/8) network blocks.

C
Networking Overview

This appendix contains an overview of the common networking protocols used in TCP/IP networking and some packet structures.

The basic structure of the common network protocols is covered in Section C.1, followed by details of the actual packet structures in Section C.2. Detail pertaining to the ICMP protocol and its use of Types and Codes is presented in Section C.3. IPv4 Address assignments are discussed in Sections C.4 and C.5, the latter dealing with so-called 'Bogon' addresses. The Appendix concludes with a selection of ISO 3661 country codes as used in DNS allocations at a country level in Section C.6.

C.1 Protocols

The three most common protocols observed in this study have been presented in Table C.1 as an extract of the list of assigned protocols provided by the Internet Numbers Assignment Authority (IANA) current as of December 2009. This authoritative list is described in RFC5237 (Arkko and Bradner, 2008) and the current version can be found at `http://www.iana.org/assignments/protocol-numbers/`.

Protocol Number	Keyword	Name	References
1	ICMP	Internet Control Message	RFC792
6	TCP	Transmission Control	RFC793
17	UDP	User Datagram	RFC768

Table C.1: Primary IP packet payloads

C.2 Packet Structures

Packet datagram structures are presented in this section by means of block diagrams. Internet Protocol version 4 (IPv4) (Postel, 1981c) is presented first, as the remainder are all payloads to this structure. This is then followed by TCP (Postel, 1981d), UDP (Postel, 1980) and ICMP (Postel, 1981b). For all of these only a high level overview relevant to the research contained in this document is given. For more detail, readers are encouraged to make use of Stevens (1993), and refer to the RFC documents mentioned in Table C.1. TCP in particular has a number of extensions to the original protocol described in RFC793, which are in common use on the modern Internet.

C.2.1 IP

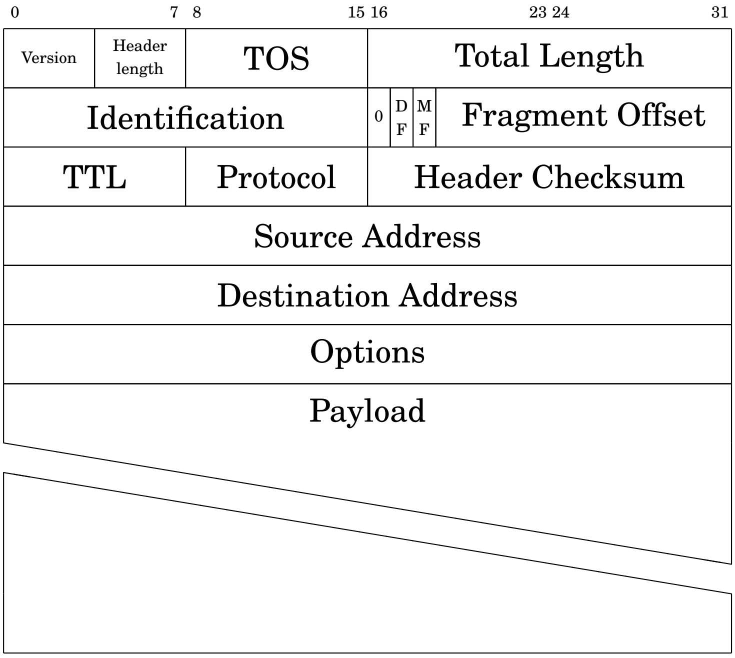


Figure C.1: IP Datagram

C.2.2 TCP

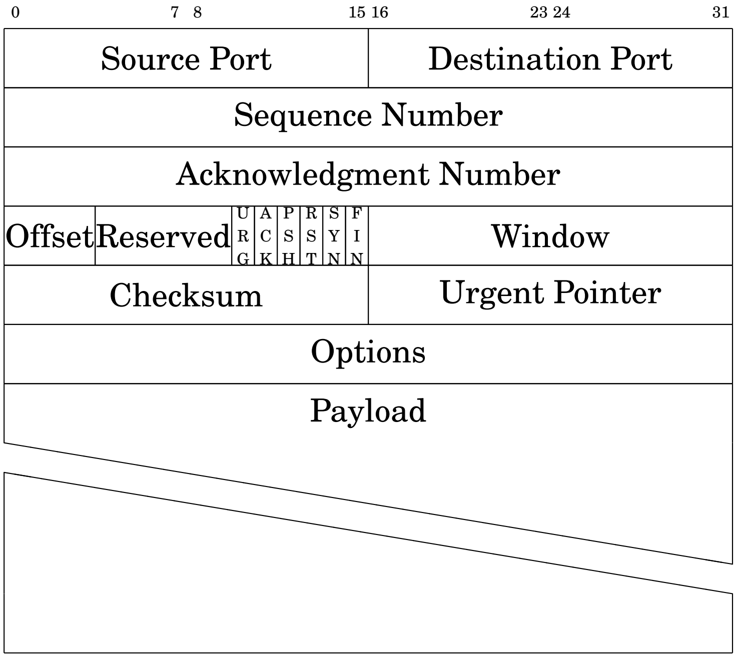


Figure C.2: TCP Datagram

C.2.3 UDP

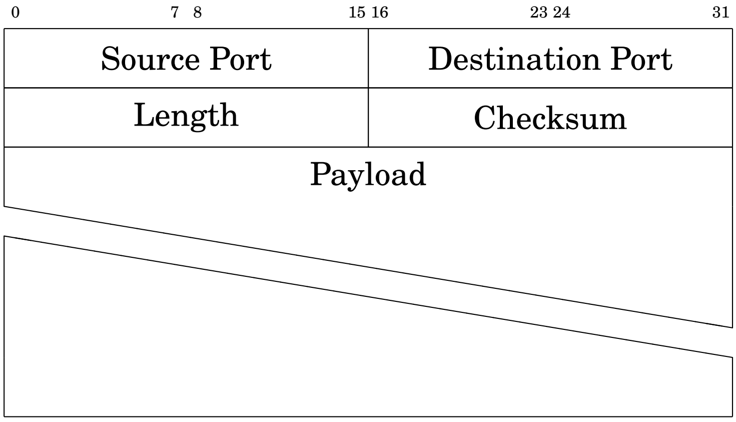


Figure C.3: UDP Datagram

C.2.4 ICMP

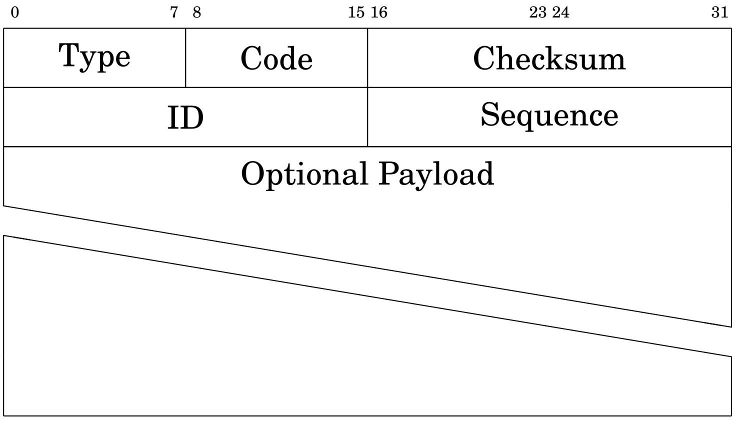


Figure C.4: ICMP Datagram

C.3 ICMP

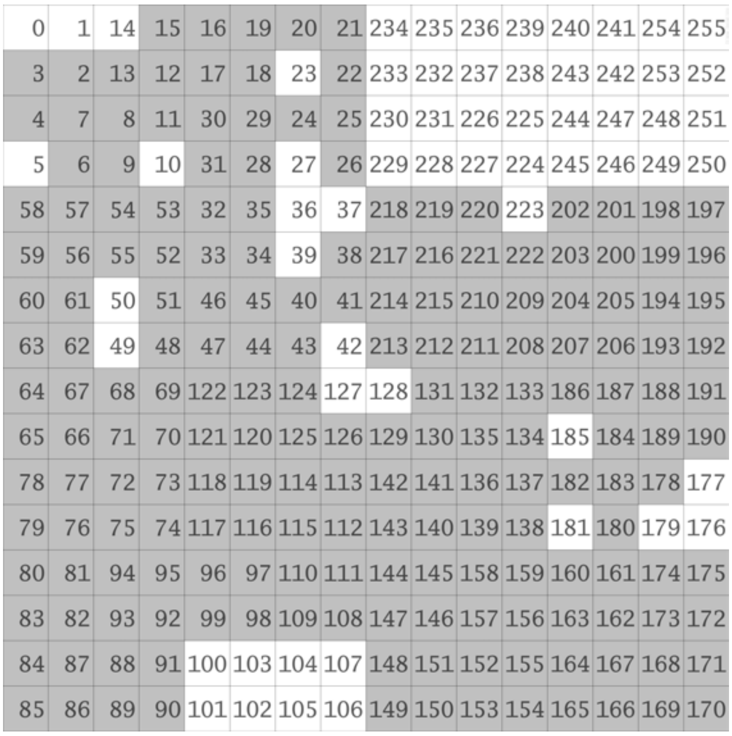
The Internet Control Message Protocol (ICMP) as defined in RFC792/STD005 (Postel, 1981a) has a rich set of features for providing out of band communications on IP networks. The following data in Table C.2 is taken as an extract from RFC 1700 (Reynolds and Postel, 1994) and RFC 2521 (Karn and Simpson, 1999), and the current online IANA assigned list (Internet Assigned Numbers Authority (IANA), 2008a), which replaced RFC1700 as of January 2002 (Reynolds, 2002). This list reflects only Types and Codes relevant to this research and in common use. A current complete list can be found at `http://www.iana.org/assignments/icmp-parameters`.

C.4 Address Assignments

IP address space assignment is managed by IANA (Internet Assigned Numbers Authority (IANA), 2008b) and delegated to Regional Internet Registries (RIR) in each geographic region for onwards assignment to end users. The zones of operation for the five RIR's is shown in Figure C.6, and detailed in Table C.3. The Hilbert

Table C.2: Selected ICMP Types

Type	Code	Name
0	-	Echo Reply
3	0	Net Unreachable
	1	Host Unreachable
	2	Protocol Unreachable
	3	Port Unreachable
	4	Fragmentation with Do not Fragment (DF) bit set
	5	Source Route Failed
	6	Destination Network Unknown
	7	Destination Host Unknown
	8	Source Host Isolated
	9	Communication with destination network is administratively prohibited
	10	Communication with destination host is administratively prohibited
	11	Destination network unreachable for TOS
	12	Destination host unreachable for TOS
	13	Communication Administratively Prohibited
	14	Host Precedence Violation
	15	Precedence cutoff in effect
4	-	Source Quench
5	0	Redirect Datagram for the Network (or subnet)
	1	Redirect Datagram for the Host
	2	Redirect Datagram for the Type of Service and Network
	3	Redirect Datagram for the Type of Service and Host
6	0	Alternate Address for Host
8	-	Echo Request
9	0	Normal router advertisement
	16	Does not route common traffic
10	-	Router Selection
11	0	Time to Live exceeded in Transit
	1	Fragment Reassembly Time Exceeded
12	0	Parameter Problem - Pointer indicates the error
	1	Parameter Problem - Missing a Required Option
	2	Parameter Problem - Bad Length
13	-	Timestamp Request
14	-	Timestamp Reply
15	-	Information Request
16	-	Information Reply
17	-	Address Mask Request
18	-	Address Mask Reply
30	-	Traceroute



0	1	14	15	16	19	20	21	234	235	236	239	240	241	254	255
3	2	13	12	17	18	23	22	233	232	237	238	243	242	253	252
4	7	8	11	30	29	24	25	230	231	226	225	244	247	248	251
5	6	9	10	31	28	27	26	229	228	227	224	245	246	249	250
58	57	54	53	32	35	36	37	218	219	220	223	202	201	198	197
59	56	55	52	33	34	39	38	217	216	221	222	203	200	199	196
60	61	50	51	46	45	40	41	214	215	210	209	204	205	194	195
63	62	49	48	47	44	43	42	213	212	211	208	207	206	193	192
64	67	68	69	122	123	124	127	128	131	132	133	186	187	188	191
65	66	71	70	121	120	125	126	129	130	135	134	185	184	189	190
78	77	72	73	118	119	114	113	142	141	136	137	182	183	178	177
79	76	75	74	117	116	115	112	143	140	139	138	181	180	179	176
80	81	94	95	96	97	110	111	144	145	158	159	160	161	174	175
83	82	93	92	99	98	109	108	147	146	157	156	163	162	173	172
84	87	88	91	100	103	104	107	148	151	152	155	164	167	168	171
85	86	89	90	101	102	105	106	149	150	153	154	165	166	169	170

Figure C.5: Hilbert Curve of IPv4 assignments as of September 2009

Curve plot in Figure C.5[1] shows the current state of IP address block assignments as of December 2008.

A number of major assignments blocks changed their allocation status during the period of study, the majority of these were from unallocated to being allocated for management by one of the Regional Registries. These are listed in the Table C.4, which has been constructed using the IANA allocation list[2] and the *Team Cymru Bogon List*[3] (Team Cymru, 2009). Noting the dates of change is important when evaluating telescope data for Bogons. Two specific network blocks worth noting are 14/8 and 46/8 which were returned to the pool of available addresses block. The latter was re-allocated in September 2009.

C.5 Bogons

> *A bogon prefix is a route that should never appear in the Internet routing table. A packet routed over the public Internet (not including*

[1] Sourced from `http://en.wikipedia.org/wiki/File:Regional_Internet_Registries_world_map.svg` under Creative Commons Share-Alike Licence

[2] `http://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xml`

[3] The current version may be obtained at *http://www.cymru.com/Documents/bogon-list.html*

RIR	**Name**	**Operational Zone**	**URL**
AfriNIC	African Network Information Centre	Africa	http://www.afrinic.net/
ARIN	American Registry for Internet Numbers	United States, Canada, parts of the Caribbean	http://www.arin.net/
APNIC	Asia-Pacific Network Information Centre	Asia, Australiasia	http://www.apnic.net/
LACNIC	Latin America and Caribbean Network Information Centre	Latin America, parts of the Caribbean	http://www.lacnic.net/
RIPE NCC	Réseaux IP Européens Network Coordination Centre	Europe, the Middle East, Central Asia	http://www.ripe.net/

Table C.3: Regional Internet Registries (RIR)

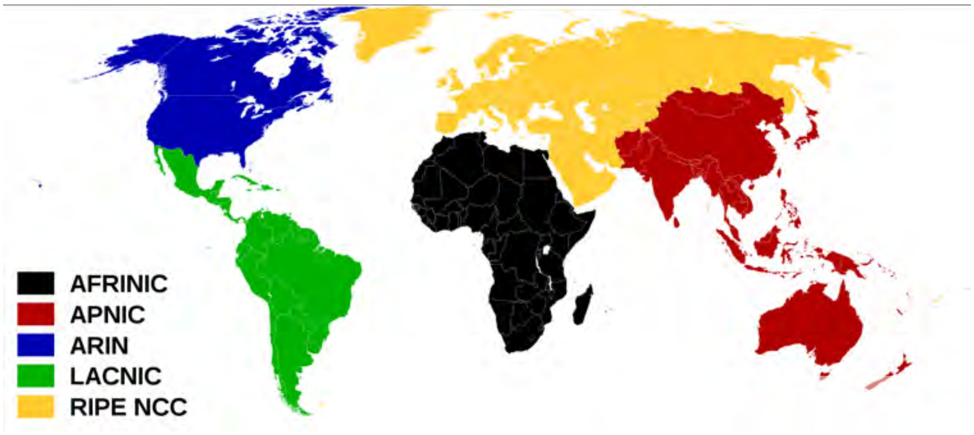


Figure C.6: Regional Internet Registries: Zones of Operation

> *over VPNs or other tunnels) should never have a source address in a bogon range. These are commonly found as the source addresses of DDoS attacks* — Team Cymru[4].

Bogons are defined as 'Martian'[5] packets in terms of RFC 1918 (Rekhter *et al.*, 1996) and include both unallocated address space from the IANA perspective[6] and Special-Use IPv4 Addresses as currently defined in RFC 3330 (IANA, 2002). In terms of this designation, packets with address sources originating from such network blocks should not be present on the global Internet. Listing 10 shows the current Bogon list as maintained by Team Cymru (Team Cymru, 2009) — a Non Profit organisation focussed on Internet Security — current as of version v5.0[7]. Aggregation has been performed between adjacent network blocks for the sake of brevity. The latest copy of this list can be obtained from `http://www.cymru.com/Documents/bogon-list.html` in a number of different formats. These lists are intended for use with some kind of bogon-filter[8] such as a router or firewall.

C.6 Country Codes

Country Code (CC) top level domains (ccTLD) are used and allocated to appropriate DNS management authorities within a particular country or autonomous territory

[4] `http://www.team-cymru.org/Services/Bogons/` Accessed 2009-11-13
[5] `http://en.wikipedia.org/wiki/Martian_packet`
[6] Address blocks allocated to regional registries but not assigned to end-users are not included
[7] Published 18 September 2009
[8] `http://www.catb.org/jargon/html/B/bogon-filter.html`

Table C.4: Changed IPv4 Address allocations 2005-2009

Prefix (/8)	**RIR**	**Date**	**Status**	**Prefix (/8)**	**RIR**	**Date**	**Status**
2	RIPE	2009-09	Allocated	112	APNIC	2008-05	Allocated
14	***IANA***	***2008-01***	***Reserved***	113	APNIC	2008-05	Allocated
41	AFRINIC	2005-05	Allocated	114	APNIC	2007-10	Allocated
46	***IANA***	***2007-04***	***Reserved***	115	APNIC	2007-10	Allocated
46	RIPE	2009-09	Allocated	116	APNIC	2007-01	Allocated
73	ARIN	2005-03	Allocated	117	APNIC	2007-01	Allocated
74	ARIN	2005-06	Allocated	118	APNIC	2007-01	Allocated
75	ARIN	2005-06	Allocated	121	APNIC	2006-01	Allocated
76	ARIN	2005-06	Allocated	122	APNIC	2006-01	Allocated
77	RIPE	2006-08	Allocated	123	APNIC	2006-01	Allocated
78	RIPE	2006-08	Allocated	124	APNIC	2005-01	Allocated
79	RIPE	2006-08	Allocated	125	APNIC	2005-01	Allocated
89	RIPE	2005-06	Allocated	126	APNIC	2005-01	Allocated
90	RIPE	2005-06	Allocated	173	ARIN	2008-02	Allocated
91	RIPE	2005-06	Allocated	174	ARIN	2008-02	Allocated
92	RIPE	2007-03	Allocated	175	APNIC	2009-08	Allocated
93	RIPE	2007-03	Allocated	178	RIPE	2009-01	Allocated
94	RIPE	2007-07	Allocated	180	APNIC	2009-04	Allocated
95	RIPE	2007-07	Allocated	182	APNIC	2009-08	Allocated
96	ARIN	2006-10	Allocated	183	APNIC	2009-04	Allocated
97	ARIN	2006-10	Allocated	184	ARIN	2008-12	Allocated
98	ARIN	2006-10	Allocated	186	LACNIC	2007-09	Allocated
108	ARIN	2008-12	Allocated	187	LACNIC	2007-09	Allocated
109	RIPE	2009-01	Allocated	189	LACNIC	2005-06	Allocated
110	APNIC	2008-11	Allocated	190	LACNIC	2005-06	Allocated
111	APNIC	2008-11	Allocated	197	AFRINIC	2008-10	Allocated

in order provide a localised domain space for their Internet community. The two letter codes used by IANA are those as defined in ISO 3166-1, per RFC1591 and RFC3071 (Postel, 1994; Klensin, 2001). Notable exceptions to this are the use of .UK rather than .GB in the United Kingdom and the .SU domain which was designated for the Soviet Union, but has been absent from ISO3661 since 2001, yet continues to persist in DNS. The .GB domain is valid, but is as of the time of writing listed as reserved by IANA and closed for any further registrations. An abbreviated list is provided in Table C.5, which includes a number of legacy and additional domain codes (such as .GB, .SU). Omitted from this Table are the majority of small island nations.

Table C.5: Selected ISO 3166 Country Codes

AE	United Arab Emirates
AF	Afghanistan
AI	Anguilla
AL	Albania
AM	Armenia
AO	Angola
AQ	Antarctica
AR	Argentina
AS	American Samoa
AT	Austria
AU	Australia
AW	Aruba
AX	Aland Islands
AZ	Azerbaijan
BA	Bosnia & Herzegovina
BB	Barbados
BD	Bangladesh
BE	Belgium
BF	Burkina Faso
BG	Bulgaria
BH	Bahrain
BI	Burundi
BJ	Benin
BM	Bermuda
BO	Bolivia
BR	Brazil
BS	Bahamas
BT	Bhutan
BW	Botswana
BY	Belarus
BZ	Belize
CA	Canada
CD	Congo, Dem Rep
CF	Central African Rep
CG	Congo
CH	Switzerland
CI	Cote D'Ivoire
CK	Cook Islands
CL	Chile
CM	Cameroon
CN	China
CO	Colombia
CR	Costa Rica
CS	Czechoslovakia*
CU	Cuba
CV	Cape Verde
CX	Christmas Island
CY	Cyprus
CZ	Czech Republic
DE	Germany
DK	Denmark
DM	Dominica
DO	Dominican Republic
DZ	Algeria
EC	Ecuador
EE	Estonia
EG	Egypt
EH	Western Sahara
ER	Eritrea
ES	Spain
ET	Ethiopia
EU	European Union
FI	Finland
FJ	Fiji
FR	France
GA	Gabon
GD	Grenada
GE	Georgia
GG	Guernsey
GH	Ghana
GI	Gibraltar
GL	Greenland
GM	Gambia
GN	Guinea
GQ	Equatorial Guinea
GR	Greece
GT	Guatemala
GU	Guam
GW	Guinea-Bissau
GY	Guyana
HK	Hong Kong
HN	Honduras
HR	Croatia
HU	Hungary
ID	Indonesia
IE	Ireland
IL	Israel
IM	Isle of Man
IN	India
IQ	Iraq
IR	Iran
IS	Iceland
IT	Italy
JM	Jamaica
JO	Jordan
JP	Japan
KE	Kenya
KG	Kyrgyzstan
KH	Cambodia
KP	Korea (North)
KR	Korea (South)
KW	Kuwait
KZ	Kazakhstan
LA	Laos
LB	Lebanon
LC	Saint Lucia
LI	Liechtenstein
LK	Sri Lanka
LR	Liberia
LS	Lesotho
LT	Lithuania
LU	Luxembourg
LV	Latvia
LY	Libya
MA	Morocco
MC	Monaco
MD	Moldova
ME	Montenegro
MG	Madagascar
MK	Macedonia
ML	Mali
MM	Myanmar
MN	Mongolia
MR	Mauritania
MT	Malta
MU	Mauritius
MV	Maldives
MW	Malawi
MX	Mexico
MY	Malaysia
MZ	Mozambique
NA	Namibia
NE	Niger
NG	Nigeria
NI	Nicaragua
NL	Netherlands
NO	Norway
NP	Nepal
NZ	New Zealand
OM	Oman
PA	Panama
PE	Peru
PH	Philippines
PK	Pakistan
PL	Poland
PR	Puerto Rico
PS	Palestine
PT	Portugal
PY	Paraguay
QA	Qatar
RO	Romania
RS	Serbia
RU	Russian Federation
RW	Rwanda
SA	Saudi Arabia
SB	Solomon Islands
SC	Seychelles
SD	Sudan
SE	Sweden
SG	Singapore
SH	St. Helena
SI	Slovenia
SK	Slovak Republic
SL	Sierra Leone
SM	San Marino
SN	Senegal
SO	Somalia
SR	Suriname
SU	USSR *
SV	El Salvador
SY	Syria
SZ	Swaziland
TD	Chad
TG	Togo
TH	Thailand
TJ	Tajikistan
TK	Tokelau
TM	Turkmenistan
TN	Tunisia
TO	Tonga
TP	East Timor
TR	Turkey
TT	Trinidad & Tobago
TW	Taiwan
TZ	Tanzania
UA	Ukraine
UG	Uganda
UK	United Kingdom
US	United States
UY	Uruguay
UZ	Uzbekistan
VA	Vatican City State
VN	Viet Nam
WS	Samoa
YE	Yemen
YU	Serbia & Montenegro*
ZA	South Africa
ZR	Zaire*
ZM	Zambia
ZW	Zimbabwe

* These are depracated allocations which were formerly assigned

Listing 10 Aggregated Bogon List - version v5.0 (18 September 2009)

```
0.0.0.0/7
5.0.0.0/8
10.0.0.0/8
14.0.0.0/8
23.0.0.0/8
27.0.0.0/8
31.0.0.0/8
36.0.0.0/7
39.0.0.0/8
42.0.0.0/8
49.0.0.0/8
50.0.0.0/8
100.0.0.0/6
104.0.0.0/6
127.0.0.0/8
169.254.0.0/16
172.16.0.0/12
176.0.0.0/7
179.0.0.0/8
181.0.0.0/8
185.0.0.0/8
192.0.2.0/24
192.168.0.0/16
198.18.0.0/15
223.0.0.0/8
224.0.0.0/3
```

This list was obtained from the Team Cymru Bogon list (`http://www.cymru.com/Documents/bogon-list.html`) current as of 2009-12-27.

D

Database design and operation

The PostgreSQL[1] database system was chosen for this project. One of the primary drivers was its excellent native support for dealing with Internet addresses[2]. The scalability[3] of this database platform was another consideration, as the datasystem was envisaged to have useful life exceeding that of just this research project.

D.1 Design

The database was constructed with the schema as shown in Figure D.1. As discussed in Section 3.2.2, the database was intended to hold just the relevant packet header information. The PACKETS table holds information that is relevant and common to all IP datagrams. tables TCP, UDP and ICMP hold protocol specific information for the three most common IP payloads. The reader is referred to

[1] `http://www.postgresql.org/`

[2] `http://www.postgresql.org/docs/8.2/static/functions-net.html`

[3] `http://www.postgresql.org/about/`

Section Section C.2 for details on the protocol specific packet structures. Checksums were omitted from the table designs. The final two tables contain ancillary information relating to the source IP addresses from collected datagrams. The OS table contains a mapping of probably source operating system of a given IP address. This process was performed my making use of the p0f passive traffic analysis tool on the raw packet captures, and processing the resultant output. Geographical mappings of the probably origin of source traffic are stored in the LOCATION table. Mappings were performed using the GeoIPLite database from Maxmind[4], and the similar database maintained at HOSTIP.INFO[5]. The four primary tables make use of a unique id field generated for each datagram on insertion into the database, in order to provide links. The table structure can be seen in Figure D.1.

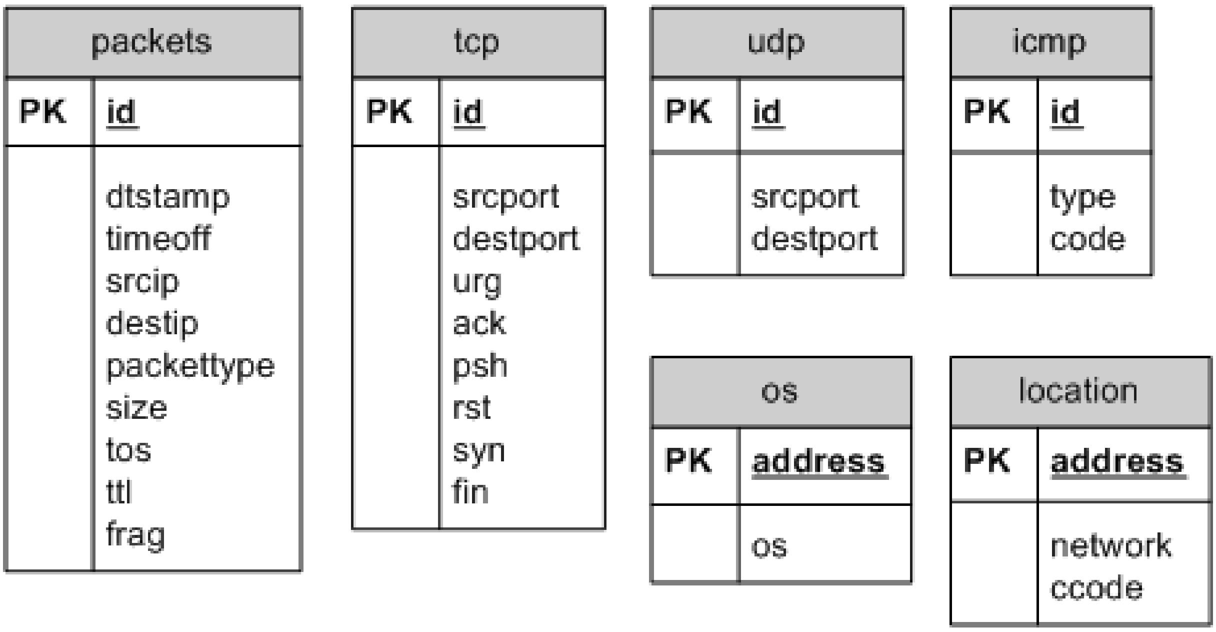


Figure D.1: Database schema overview

D.2 Considerations

Separate database instances were used for the Rhodes Telescope Dataset and the samples portions of the CAIDA backscatter dataset as they were separate logical instances. A third database instance was created for the ad-hoc processing of other captures.

[4] http://www.maxmind.com/app/geoip_country

[5] http://www.hostip.info/

Table D.1: Load speeds

Tool	Wall Time	Rate ($Packet/s$)	CPU Time
sharkbait	16m44.769s	3147	3m23.979s
tcpdump	47m25.455s	1110	47m26.030s

Timing examples are given from the processing of a 3 162 989 packet capture file from the CAIDA backscatter collection

File: backscatter-20070223-0000-clean.pcap.lzo

D.3 Population

A custom C++ application was developed which allowed for relatively rapid loading of pcap dump files into the database. This was able to achieve mean rates of around 190 000 packets per minute. This proved around three times faster than simply parsing the files with Tcpdump - which in turn would still require further processing. The CPU utilisation was also markedly decreased with a 21 times reduction in utilisation and CPU time consumed. Sample comparative information for this tool is shown in Table Table D.1 on page 292.

This allowed for loading of even the large CAIDA sets in most cases of 15-20 minutes per capture file - representing a speedup of around 3.5-4x over the real-time capture rate. Some of the larger files were between 11 and 15% slower than the capture rate. This only became apparent with capture files exceeding 12million packets/hour.

This tool could be further optimised, but was found to be sufficient for the needs of this project. Loading of capture files with the sharkbait tool developed, resulted in the PACKETS, TCP, UDP and ICMP tables being populated with data. This loading was periodically performed. Loading of the OS and LOCATION tables was done using two utilities providing wrapper services around the p0f operating system classifier, and Geo-location tool-chains respectively. The tools both made use of the 'raw' pcap files recorded by the telescope. These two tables were updated following loads of data into the primary tables.

D.4 Operation

Database operation was relatively simple. Periodically rotated raw captures were copied from the live Telescope sensor onto the DB server. Post transfer, checksums were verified, and the files were then carved and repackaged as needed using the tcpslice and tcpdump utilities. Data processing scripts were then run to load the datagrams from the resultant pcap files into the database. After the initial loading of the pcap data into the appropriate database, the scripts for the geolocation and operating system identification were run. Database snapshots were regularly taken for backup purposes.

E

Other known Telescope Dataset Repositories

This appendix lists other projects, or network traffic repositories that have made use of Network Telescope technology. Of those listed below, it is believed that CAIDA.org and RIPE are the only organisation making datasets relatively freely available. PREDICT datasets are available to researchers inside the USA. Both CAIDA and PREDICT have a vetting process for researchers wanting access to data.

While the majority of these datasets are not currently active, researchers may have some success contacting the host organisations. Datasets known to be active and available at the time of writing are: CAIDA, PREDICT, Rhodes and RIPE. It is hoped that more researchers consider making data available, either as suitably processed raw captures, or by publishing summary data using methods such as those described in Chapter 8.

E.1 CAIDA

NAME	CAIDA
Location	University of California San Diego, USA
Description	A large /8 network telescope operated by the Cooperative Association for Internet Data Analysis (CAIDA). Largely backscatter telescope data. The project also contains lots of other Internet Networking related data.
Operational Dates	Established 2002, datasets from 2004. Currently active
Access policy	Vetting required
URL	Primary: `http://www.caida.org/` Data overview: `http://www.caida.org/data/overview/` PREDICT Datasets mirrored at CAIDA `http://www.caida.org/projects/predict/`

E.2 CSIR/DPSS

NAME	CSIR-DPSS
Location	South Africa
Description	A new network telescope comprising three numerically distant /24 netblocks. Operated by the Council of Scientific and Industrial Research (CSIR) DPSS Unit. This telescope was only established in Late 2010.
Operational Dates	Late 2010, Active
Access policy	Unknown
URL	Unknown

E.3 IUCC/IDC Internet Telescope

NAME	IDC Network Telescope
Location	Israel
Description	The Israel Inter-university Computation Center (IUCC)[a] has assigned a /16 (former Class B - with 65,536 IP addresses), which is "dark space", as a place where the InterDisciplinary Center (IDC) [b] in Herzliya Israel, has been able to install a network monitor, which receives "backscatter" packets from all over the Internet. Before the IDC, Riverhead Networks operated the telescope before being bought out by Cisco. Summary data is available from 28-Sep-2004 though to 08-Aug-2005. More data may be available [a] http://www.iucc.ac.il/ [b] http://www.idc.ac.il/
Operational Dates	Sep 2004-Aug 2005, Defunct ?
Access policy	Unknown
URL	Primary http://noc.ilan.net.il/research/telescope/

E.4 ICIR Network Telescope Project

NAME	International Computer Science Institute Network Telescope Project
Location	Berkeley, California, USA
Description	The project is described as building a large-scale, diverse telescope, portions of which also include active honeypots. The front-end sensors are spread across a large number of address blocks. We aim to leverage not just unallocated blocks but also unallocated subblocks within allocated blocks (greynets).
Operational Dates	circa 2004 - 2006 Now Defunct ?
Access policy	Unknown
URL	Primary http://www.icir.org/vern/telescope.html

E.5 LOBSTER

NAME	LOBSTER
Location	Research and Academic Sites Across Europe
Description	Comprises approximately forty sensors in twelve countries in three continents. Primarily Located in Europe
Operational Dates	Established mid 2006, Currently active?
Access policy	Unknown
URL	`http://www.ist-lobster.org/publications/`

E.6 PREDICT

NAME	PREDICT
Location	United States of America
Description	Protected Repository for the Defence of Infrastructure Against Cyber Threats (PREDICT) a repository of data for cyber security research. An Initiative of the Department of Homeland Security Science and Technology directorate, the virtual center provides a common framework for managing datasets from various data providers. PREDICT also formalizes the process of gaining access to these datasets. This common framework benefits both the data providers, as they no longer have to review, approve and monitor individual researchers who approach them for access to various datasets; as well as security researchers, as they no longer need to rely on ad hoc and often arbitrary policies of each data provider that they wish to obtain datasets from. PREDICT is a collaborative project, with Merit and the University of Michigan as the lead organizations. Other organizations involved in the effort are the University of Wisconsin, University of Washington, Internet2 and XO Communications. Includes some of the data available in the UWISC, UMICH repositories. Some of the PREDICT Data is also available on CAIDA.
Operational Dates	Established 2005, Currently active.
Access policy	Limited to researchers within the USA. All research and. work involving PREDICT datasets must be carried out at locations within the 50 United States.
URL	https://www.predict.org/ Data via CAIDA http://www.caida.org/projects/predict/

E.7 RIPE

NAME	RIPE Labs
Location	Located in Europe
Description	The RIPE Data Repository is a large data store (currently ~100TB) that holds a diverse set of data of interest to the scientific and operator community. Notable is the mirror of the Waikato Internet Traffic Storage (WITS) passive datasets[a] collected in New Zealand.
Operational Dates	Currently in operation
Access policy	Registered users
URL	Primary http://labs.ripe.net/datarepository/ Datasets: http://labs.ripe.net/datarepository/data-sets

[a] https://labs.ripe.net/datarepository/data-sets/the-waikato-internet-traffic-storage-wits-passive-datasets

E.8 Rhodes University

NAME	RUSCOPE
Location	South Africa
Description	This is the data discussed in this study. Current data comprises August 2005 to present for RUSCOPE1 adn September 2009 to present for RUSCOPE2. System consistes of two /24 netblocks, numerically distant.
Operational Dates	Aug 2005-present
Access policy	Open to researchers
URL	Email b.irwin@ru.ac.za

E.9 SWITCH

NAME	SWITCH Internet Background Noise (IBN)
Location	Switzerland
Description	SWITCH operates their Internet Background Noise (IBN) sensor which has a maximum size of three /17 networks ($\simeq$ 98304 IP addresses),, divided up over small netblock (similar to greynet operation). Only summary information and graph images appear to be provided.
Operational Dates	Operationasl since June 2003
Access policy	Unknown, possibly only to project members
URL	Primary `http://www.switch.ch/security/IBN/`

E.10 UMICH/IMS

NAME	Internet Motion Sensor
Location	University of Michigan
Description	An early network telescope, established by the Department of Electrical Engineering and Computer Science. Data collected was most notably used by Cooke and Bailey in a number of publications. Datasets may be available via PREDICT.
Operational Dates	Established circa 2006 ? Defunct.
Access policy	Unknown
URL	Primary `http://ims.eecs.umich.edu/` (possibly offline)

E.11 UWISC

NAME	UWISC
Location	University of Wisconsin, USA
Description	Established and operated by Paul Barfordas part of the research conducted as projects in the Wisconsin Advanced Internet Laboratory (WAIL). Some of this data may be available via PREDICT. The iSink project may have data available, and dates back to 2003.
Operational Dates	Established 2004 ? Defunct ?
Access policy	Unknown
URL	Primary `http://pages.cs.wisc.edu/~pb/` iSink: `http://www.potaroo.net/iepg/july-2003/isink.pdf`

E.12 WOMBAT

NAME	WOMBAT
Location	Primarily Located in Europe
Description	Worldwide observatory of malicious behaviors and attack threats (WOMBAT). European Union funded research project that aims at providing new means to understand the existing and emerging threats that are targeting the Internet economy and the net citizens. The approach carried out by the partners include a data collection effort as well as some sophisticated analysis techniques. Projected Total cost: 4,422,746 €
Operational Dates	Established beginning of 2008, Currently active
Access policy	Unknown, possibly only to project members
URL	Primary `http://www.wombat-project.eu/` Project info: `http://www.ist-world.org/ProjectDetails.aspx?ProjectId=b22607bfe1c8416a8a2b92e16486fba2`

F

Contents of Multimedia CD

The accompanying CD contains the following directories:

Telescope Analysis

- **TopIP** The top twenty graphs as generated for IP addresses at the /8, /16, /24 and 32 aggregation levels. CSV files are included.
- **TOPTCP** The top twenty graphs as generated for TCP ports, filtered by SYN and RST flags. CSV files are included.
- **TOPUDP** The top twenty graphs as generated for UDP ports. CSV files are included.

Final Images High resolution verions of images used in this text.

When it comes to software, I much prefer free software, because I have very seldom seen a program that has worked well enough for my needs, and having sources available can be a life-saver

Linus Torvalds - Linux Architect

Colophon

This work was produced largely using Open Source Software. The researcher wishes to express his thanks to the authors and development teams involved in producing the software systems used both in the processing of the data in this research and in the actual formatting of this document.

This document was compiled using the LyX environment, which provides a front end to the LaTeX document processing system. The MikTeX port of LaTeX for Microsoft Windows, with development led by Christian Schenk, was used for the final production. In support of this, a number of other crucial tools were used by the back-end, notably ImageMagik and Ghostscript. Specific LaTeX packages used for the layout of this work are BYTEFIELD, LETTRINE and QUOTCHAP. The HYPERREF package was used to make the PDF version of this document 'hot linked' in terms of references to sections and URLs.

Data was stored in, and processed, using a PostgreSQL database system residing on a FreeBSD server; the latter also being used as the operating system on the capture device, and data-processing server. PgAdminIII was used for the development, debugging and performance tuning of the many database queries used in the analysis and exploration. The majority of the graphs in this document, the electronic appendix, and those used in the general exploration were produced using gnuplot. These were often further processed and annotated using two excellent graphics packages: GIMP and Inkscape, respectively providing both bitmap and vector based image manipulation.

Data manipulation utilities were largely implemented in Python, PHP and C, the latter making use of GCC, the GNU Compiler Collection. Packet specific tools used were the venerable tcpdump/libpcap, WireShark and the libtrace toolkit produced by the University of Waikato Computer Science Department.

While this is not an exhaustive list of every software component used, it represents the major packages utilised in conducting this research. Not to be forgotten are the myriad of additional libraries on which these all depend. Finally, while the final production was done on a Microsoft Windows platform, the bulk of the data analysis, development and data processing was conducted using systems running Open Source operating systems: FreeBSD and Ubuntu Linux.

A list of these software components, along with their URLs is listed below.

Tool	**URL**
LyX	`http://www.lyx.org/`
LaTeX	`http://www.latex-project.org/`
MikTeX	`http://miktex.org/`
ImageMagik	`http://www.imagemagick.org/`
Ghostscript	`http://pages.cs.wisc.edu/~ghost/`
PostgreSQL	`http://www.postgresql.org/`
FreeBSD	`http://www.freebsd.org/`
Ubuntu Linux	`http://www.ubuntu.com/`
PgAdmin III	`http://www.pgadmin.org/`
gnuplot	`http://www.gnuplot.info/`
GIMP	`http://www.gimp.org/`
Inkscape	`http://www.inkscape.org/`
Python	`http://www.python.org/`
PHP	`http://www.php.org/`
GCC	`http://gcc.gnu.org/`
tcpdump/libpcap	`http://www.tcpdump.org/`
WireShark	`http://www.wireshark.org/`
libtrace	`http://research.wand.net.nz/software/libtrace.php`