Creating a flexible data processing engine for large packet capture datasets

Submitted in partial fulfilment
of the requirements of the degree of

Bachelor of Science (Honours)

of Rhodes University

Ignus van Zyl

Grahamstown, South Africa
October 30, 2014

Abstract

This project produced a prototype system that could analyse and report on packet capture files generated by network telescopes. This incorporated prior research has gone into both the analysis of packet captures as well as the reporting of results. This research considers a variety of basic numeric analysis and reporting techniques in an attempt to gain a greater understanding of the requirements of the system. The design and implementation of the system was based on the need of the system to have a standardised infrastructure to allow for comparison of results between darknets. Another focus of development was to include an element of flexibility to the reporting output. The results are then considered and expanded upon using reporting output, in an attempt to better understand the packet activity captured by the dataset and represented by the reporting infrastructure.

ACM Computing Classification System Classification

ACM Computing Classification System Classification (2012 version, valid through 2014)
I.7.3 [Network Security] Web protocol security
I.7.5 [Network Security] Denial-of-Service attacks
I.8.4 [Database and storage security] Database activity monitoring
General Terms pcap, Network telescope, Darknet, analysis, report generation

Acknowledgements

I would like to acknowledge the people and organisations that helped me realise the completion of this paper. My gratitude extends first to Professor Barry Irwin, whose advice and support has been instrumental to the completion of the project.

I also thank my family for their support and understanding during the year. Their belief in me allowed me to focus on the completion of my Honours Degree.

I am also grateful to my colleagues in the Honours Lab, without whom my year would not have been as interesting nor as fulfilling as it has become.

I would also like to acknowledge Absa Bank, which has continued to support me financially during my Honours year.

This research makes use of GeoLite data created by MaxMind.

This work was undertaken in the Distributed Multimedia CoE at Rhodes University, with financial support from Telkom SA, Tellabs, Genband, Easttel, Bright Ideas 39, THRIP and NRF SA (TP13070820716). The authors acknowledge that opinions, findings and conclusions or recommendations expressed here are those of the author(s) and that none of the above mentioned sponsors accept liability whatsoever in this regard.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This introductory chapter deals with introducing the problem that is handled by the project. It will also include an overview of the project considerations as well as a breakdown of the chapters that are to follow. This chapter will focus on introducing the context of the system. This chapter will also outline the structure of the document. The system itself is based around packet capture files created by network telescopes. Network telescopes create a new avenue of study for computer scientists, by observing the packets collected by them we can gain greater knowledge about security threats and attacks prevalent on the Internet (Moore, Shannon, Voelker, and Savage, 2004). Traffic captured by network telescopes are inherently interesting because none of the captured packets have a legitimate reason for arriving at network telescopes to be recorded as they are all unsolicited (Pang, Yegneswaran, Barford, Paxson, and Peterson, 2004).The analysis and study of network telescope traffic captures will allow us to better understand the presence of network scans, worm activity and DDoS activity through reflected traffic (Harder, Johnson, Bradley, and Knottenbelt, 2006). The nature of Internet Background Radiation is constantly changing (Wustrow et al., 2010). The analysis of this data can reveal the trends and activity of malicious packets and their hosts (Allman, Paxson, and Terrell, 2007).

1.1 Problem Statement

Network telescopes observe millions of packets of Internet traffic through the monitoring of unused IP address space (Moore et al., 2004). The data gathered by these telescopes is important to the study of Internet Background Radiation (Pang et al., 2004) and also

more specifically to the monitoring and study of potentially malicious IP traffic across the Internet (Irwin, 2011). With increasingly large datasets being created by telescopes across multiple studies, the distribution of analysis results through the scientific community could be improved by creating a flexible yet standardised reporting engine (Irwin, 2013). Such a standardised reporting format will also provide researchers with an overview of trends in the analysed data-sets.

1.2 Research Goals

Information security has recently become an important field both academically and economically. As a result there is much research and analysis performed on the packet captures of network telescopes (Moore et al., 2004), greynets (Harrop and Armitage, 2005) and honeynets (Francois, Festor, et al., 2009). An issue currently faced is a lack of standardisation with the reporting of results. This in turn makes it difficult for researchers to compare analysis results across different studies, or even across different darknets. A reporting engine that analyses and represents the data from different darknets in a standardised and comparable format would enable researchers not only to analyse the traffic in their own datasets, but also to quickly compare results and findings with other researchers. This will also allow for an acceleration in the publication of reported findings in the research community.

The goals of the research are as follows:

- To investigate the feasibility of a flexible yet standardised reporting engine that reports on data generated from packet captures.
- To build a prototype system that would be able to analyse pcap datafiles.
- To buld a prototype reporting framework for the analysis results.

Section 3.5.3 of this paper will determine whether or not, and to what extent, these goals have been met by the prototype system.

1.3 Research Scope

The scope of the research must be defined to give context to the research and documentation that comprises the rest of the document.

The datasets were five packet capture files from five separate darknets. Each dataset was a part of larger datasets that have been collected for academic research. The datasets are discussed in section 3.1.

This system cannot send or receive packets across a network, even though some of the system libraries have such capabilities. The reporting system is designed to receive formatted pcap datafiles.

The system will not be expected to create textual analysis to support the visual report as this was considered to be outside the scope of an Honours project. The core focus of the system is to produce graphical and tabular output as a framework for an analyst; room is left in the generated report for the analyst to comment on findings presented in the report.

1.4 Document Structure

- Chapter two is a literature review on the analysis and reporting of packet capture data. This chapter looks at more clearly defining the concepts needed to develop a larger context for the project. Various analysis approaches are also investigated along with techniques for representing packet-related data.
- Chapter three covers the design and implementation of the prototype system. The fundamental components of the system are identified, after which the challenges and changes during implementation are discussed. This chapter also contains a critical evaluation of the system.
- Chapter four contains case studies that showcase the analysing and reporting capabilities of the document. Anomalies in initial darknet result comparisons are broken down and analysed in isolation.
- Chapter five holds the findings as well as the final evaluation on the project. A section is also reserved for possible future work on the project.
- An appendix follows the conclusion chapter that contains output from both the general and ancillary reports, as well as information on reaching the project program files through GitHub.

Chapter 2

Literature Review

2.1 Introduction

The pursuit of the study of network security is almost as old as the Internet itself (Irwin, 2011). Real interest in the topic only developed after the discovery of the Morris worm in November of 1988 (Spafford, 1989). Network telescopes are a valuable tool in gaining an understanding about activity across a network. They give researchers the ability to collect large amounts of packet data across parts of the physical Internet (Moore et al., 2004). This raw data is inherently meaningless, but through analysis much can be learned about the behaviour of malicious Internet traffic (Bailey, Cooke, Jahanian, Provos, Rosaen, and Watson, 2005b). With large datasets being created by telescopes across multiple studies, the distribution of analysis results through the scientific community could be improved by creating a flexible yet standardised reporting engine (Irwin, 2013). The need for tools in the field of digital forensics is increasing rather than decreasing, and there is scope for improvement and availability of these tools (Garfinkel, 2010).

This literature review will introduce the origin of the data in sections 2.2 - 2.6. Sections 2.7 - 2.10 will examine analysis techniques for packet captures. Data visualisation and reporting techniques will be examined across sections 2.11 - 2.13.

2.2 Network Telescopes

A network telescope is some part of assigned and address routed IP space. The IP addresses are used but not utilized by any host system (Moore et al., 2004). A network

telescope makes use of unallocated IP addresses by monitoring the activity at these addresses (Irwin, 2013). This allows researchers to study the characteristics and behaviour of Internet Background Radiation without having to worry about distinguishing between IBR and legitimate network traffic (Irwin, 2013). Any traffic received can be defined as unsolicited (Moore et al., 2004). Studying traffic captured by these telescopes provide researchers with interesting insights into network security events such as denial-of-service attacks, Internet worm packets and network scanning attempts (Moore et al., 2004). Network telescopes do not respond to requests, nor send traffic within the IP address block that it observes. The telescope does not record packet data outside of the observed net block (Du and Yang, 2011). Figure 2 gives a simple overview of a network telescope infrastructure. The probe targets in this diagram, following normal network telescope practices, will be unused IP addresses (Moore et al., 2004). The probes are randomly sent across the network from an active end-host, and some of the randomly generated IP addresses fall into the observed network block; as such the probing packets are recorded by the network telescope.

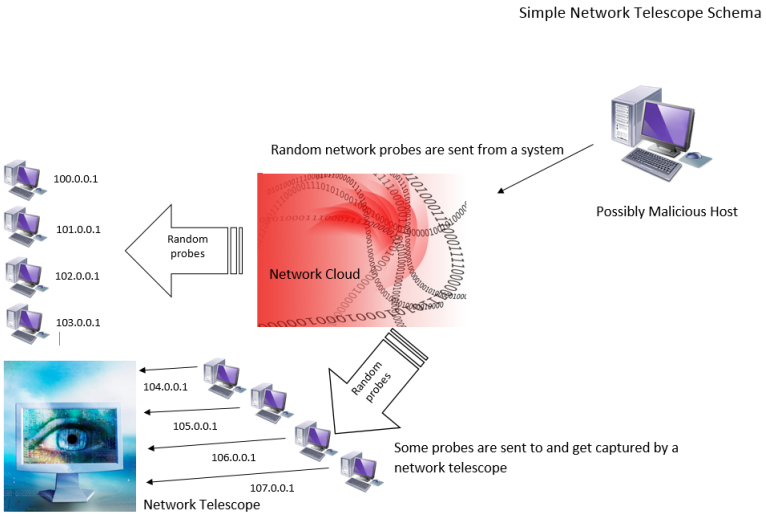


Figure 2.1: Basic schema of a Network Telescope

2.3 Classification of Network Telescopes

Darknets, or non-populated network blocks, have been favoured for studying malicious packet activity across networks because all packet data is unsolicited (Irwin, 2013). A non-populated network block is an IP address block for which there are no active end-point hosts. This means that there is a high level of confidence that received packets or network activity is not legitimate (Moore et al., 2004). The different types of observed network are discussed below.

2.3.1 Darknet

A darknet, also referred to as a Blackhole or a Sink, is described as a completely passive packet sink (Irwin, 2011). As a result the IP traffic across this network is completely unidirectional (Zseby, King, Fomenkov, and Claffy, 2014) as darknets do not respond to any traffic received across the unused IP address block (Moore et al., 2004). It is important to remember that the malicious and misconfigured activity on any two darknets is rarely the same; the difference coming from the position of the darknet on the network as well as how the darknet responds to incoming traffic (Bailey, Cooke, Jahanian, Myrick, and Sinha, 2006).

2.3.2 Lightnet

A lightnet has no canonical definition, but if a darknet can be said to be an IP address block that is completely unpopulated (Bailey et al., 2006) then a lightnet, or lit network, would be a network address block that is mostly or completely populated (Harrop and Armitage, 2005). Darknet data offers valuable analysis opportunities because most if not all of the intercepted traffic is unsolicited (Bailey et al., 2005b). The same rule does not apply to lightnets, where most or all of the users are active and each end-host could potentially be sending and receiving packet data across the network. This makes analysis of the data more difficult, and as such lightnet monitoring is more prevalent in cases where malicious packets have to be identified in a timely manner (Dhillon and Ansari, 2012). A smaller IP block will be observed in such cases, and as a result there is a trade-off between the amount of data gathered and the ability to react to it (Bailey et al., 2005a).

2.3.3 Greynet

A greynet is a network block that holds smaller IP blocks that fall into two categories, dark or lit (Harrop and Armitage, 2005). The dark blocks within the greynet are identical in action to a darknet, and do not respond to Internet traffic. The lit blocks within the greynet grant the telescope the ability to observe otherwise unaccessible traffic across the monitored network block (Irwin, 2011). While some lit network blocks may consist of active hosts, they may also be running a TCP-connection spoofer. This allows the receiving host to spoof a TCP SYN/ACK handshake, allowing the network telescope to capture the TCP packet data that would come after the handshake, before the target host sends a TCP RST and drops the connection (Irwin, 2011). This allows the telescope to capture packet data that would not be available on a pure darknet as the TCP SYN from the possibly attacking source would not be responded to (Harrop and Armitage, 2005).

2.4 Network Telescope traffic type

A typical network telescope will collect both relevant and irrelevant traffic. One task of telescope data-set analysis is to sort packets so that those that offer no intelligible information can be identified and ignored. The packets worth analysing can be grouped as follows:

2.4.1 Active Traffic

Active traffic can be determined to be packets which are expected to elicit a response when processed by the TCP/IP stack of the receiving host (Irwin, 2011). TCP packets that have the SYN flag set, indicating that a response is expected, are counted for analysis purposes. TCP packets that have the ACK or RST flags set could be the result of backscatter traffic. Backscatter traffic is intercepted when monitored address are being used to spoof packets as part of a security attack (Irwin, 2013). ICMP traffic packets that attempt to elicit any response will also be analysed (Harder et al., 2006, Irwin, 2011). Since UDP traffic is stateless, no expectations for initiation or response can be inferred from the headers. As such, deeper analysis of the UDP payload is required to determine if the UDP traffic is active (Irwin, 2011). An example of network traffic considered active can be seen in Table 2.1 (a).

2.4.2 Passive Traffic

Passive traffic is traffic that, when processed by the TCP/IP stack of the receiving host will give no legitimate response. It is considered unlikely that potentially malicious software would use these packets to determine information about the host system (Irwin, 2011). Passive traffic is usually the result of scanning activity; which is typically the result of the monitored IP range being spoofed, denial-of-service flooding, misconfiguration of network addresses or mangled and unintelligible packets (Irwin, 2011). An example of network traffic considered passive can be seen in Table 2.1 (b).

Table 2.1: Classification of active and passive traffic

(a) Active network packet classification (Irwin, 2011)

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`

(b) Passive network classification (Irwin, 2011)

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`

2.5 Internet Background Radiation

Internet background radiation (IBR) is unsolicited traffic (Bailey et al., 2005b) sent received by an IP address block with no active hosts (Wustrow et al., 2010). This traffic can also be described as nonproductive, as it is received by unused IP addresses, nonfunctioning servers and servers that are not intended to receive traffic (Pang et al., 2004). IBR is collected across network telescopes (Pang et al., 2004), sometimes referred to as darknets, as will be discussed later (Bailey et al., 2005b). This is done to remove the possibility of legitimate traffic being captured with the IBR, as no legitemate traffic would be requested by a darknet (Irwin, 2013).It is possible to divide such unwarranted packets into two classes, malicious and benign (Pang et al., 2004).

2.5.1 Benign background radiation

There are few reasons that benign packets are destined for an unused IP block. As the monitored IP block is empty of live hosts, none of the packets received have been solicited, nor are any of them expected (Bailey et al., 2005b). Very few functioning end-hosts would send unsolicited data to an unused IP block, unless the sending host was somehow miscalibrated (Irwin, 2011). This leaves the few mangled or misconfigured packets who end up at the IP block by chance, along with reflected reply packets from Internet activity in other blocks, known as backscatter (Moore et al., 2006).

2.5.2 Misconfigured Packets

One reason is misconfiguration of the packet, which most likely comes about through an error occurring when entering the destination address. This could occur at an end-point system or in a Network Address Translation gateway (Irwin, 2011).

2.5.3 Backscatter

Backscatter can be defined as traffic that has arrived at the network telescope as a result of activities which caused reflection of traffic from the originating machine (Irwin, 2013).Most of this radiation is made up from falsely addressed as a result of an address in the observed block being spoofed (Pang et al., 2004). Packet Spoofing is when an

end-system generates a source IP for the packet header that is not the actual IP of the end-host (Harder et al., 2006). A source of backscatter is a spoofed SYN-flood Denial of Service (DDoS) attack (Pang et al., 2004). As a result, the network telescope will receive ACK (acknowledgement) packets as that unused IP address is the spoofed source that the attacker is using, as the DDoS is flooding the target end-host with SYN packets to disrupt service (van Riel and Irwin, 2006a). The following TCP packet combinations are usually classified as backscatter, as they are the packets most likely to be generated from an end-host attempting to respond to a spoofed source: TCP ACK(acknowledge), RST(reset), SYN(synchronise)+ACK, and RST+ACK (Wustrow et al., 2010).

2.5.4 Potentially malicious background radiation

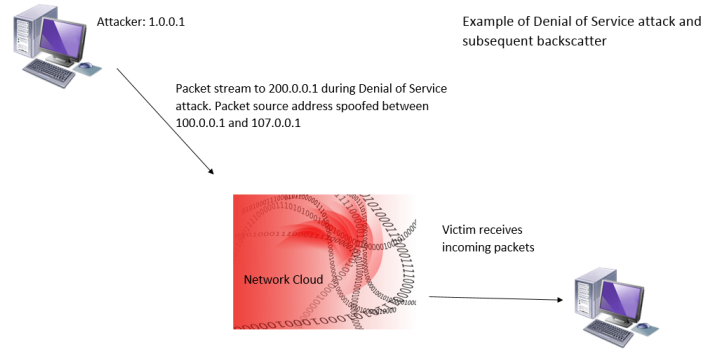
Malicious packets can be considered as unsolicited network traffic that attempts to gain a response from, or interact with, the end-point system of an unused IP address block (Irwin, 2011). This includes network scans by worms, Denial of Service (DDoS) attempts, ICMP packet scans and TCP SYN+ACK requests (Irwin, 2011, Pang et al., 2004). Potentially malicious traffic attempts to connect or interact with an end-host, and as such can be classified as active traffic. Traffic that does not attempt to elicit a response from the host is considered passive, and usually classed as benign traffic. These classifications will be discussed in depth in the Network Telescope section.

2.6 Sources of IBR

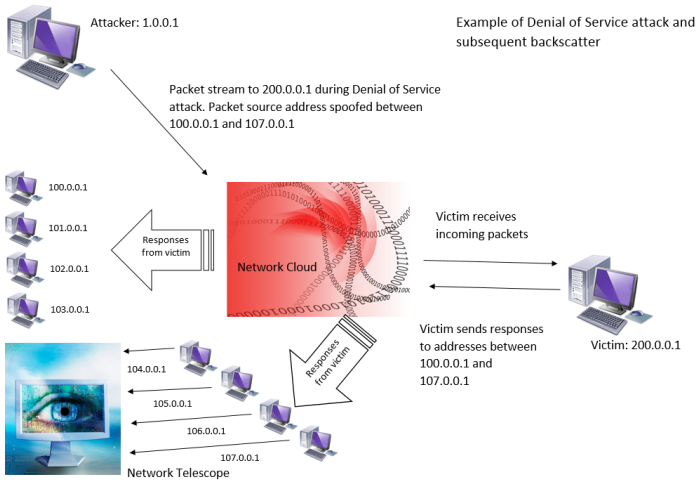
There are different network activities that produce IBR. Some of them are active in nature, i.e. worms or botnets actively searching for vulnerable end-point hosts; scanning a network is usually the first step in exploiting a network (Muelder, Ma, and Bartoletti, 2005). Other packets are received as a result of address spoofing. To make it difficult for a target end-host or the host's ISP to detect DoS attacks, attackers will spoof source IP addresses of packets intended for the victim. As the target end-host cannot easily differentiate malicious traffic from legitimate traffic, it will attempt to respond to all packets received (CAIDA, 2012).

2.6.1 Packet Spoofing

If the attacker spoofs IP addresses in the Network Telescope address block, reply packets from the attack will be observed as they travel to a host that does not exist (CAIDA, 2012). Figure 1a shows a Denial of Service attack taking place on a target system. Figure 1b explains how backscatter, or reply packets(Moore et al., 2006), are captured by the darknet as a result of packet spoofing. The packets received from the attacker have a range of spoofed addresses; some of the reply packets are then sent to an observed but unused IP address block and intercepted by the telescope (Bailey et al., 2005b). A diagrammatic representation of this can be seen in Figure 2.2 below. Apart from the TCP-SYN flood mentioned above, an ICMP Type 8 ping-flood can result in ICMP Type 0 responses which consume bandwidth on both the uplink and downlink of a network (Irwin, 2011).



(a) DoS attack on a target end-host



(b) DoS backscatter catpured by network telescope

Figure 2.2: DoS attack on a target end-host and resultant backscatter

2.6.2 Denial of Service attacks

A Denial of Service (DoS) attack on a target host is initiated with the expected result that the target system will crash or become overwhelmed by packets and/or connections, making the formation of legitimate connections difficult if not impossible (Kurose and Ross, 2010). This is done by consuming the network resources available to the host (Moore et al., 2006) or by causing the crash of the host system (Kurose and Ross, 2010).

2.6.3 Vulnerability Attack

This DoS attack focuses on sending packets designed to cause malfunctions to a vulnerable application or operating system, in an attempt to make the host crash or stop a service (Kurose and Ross, 2010).

2.6.4 Connection Flooding

This DoS attack relies on creating TCP connections with a target host in an attempt to prevent later legitimate connections by leaving the connection open (Kurose and Ross, 2010). Some of the DoS backscatter observed will be TCP SYN-ACK packets, to show that a connection has been established with the target host (Kurose and Ross, 2010). This type of DoS generates the most darknet data as the target host will send replies to the received packets, usually in the form of ACK, SYN-ACK or RST packets (Wustrow et al., 2010).

2.6.5 Bandwidth flooding

This DoS attack relies on rapidly flooding the target host with a large number of packets (Moore et al., 2006). In doing so the host cannot accept legitimate packets simply because a packet bottleneck has been created at the target host (Kurose and Ross, 2010).

2.6.6 Malicious Network Scans

Scans can range from fully manual to fully automated in nature. The purpose of a network scan is to identify active hosts in that network block, and then to find vulnerabilities on

those hosts. As such, network scans can take the form of simple ping requests across a network to determine live hosts, sequential port scans across a single IP address to search for vulnerabilities, or the scanning of one or a small number of possibly vulnerable ports across an IP network (CAIDA, 2012). Network scans will usually take place as port scans, i.e. where a port or group of ports is scanned over multiple IP addresses (Harder et al., 2006), or it will be in the form of a host scan, where all open ports of a single IP are scanned before moving to the next (van Riel and Irwin, 2006b). Worm activity forms part of the total scanning activity observed on a network (Muelder et al., 2005).

2.6.7 Internet Worms

Internet worms will attempt to spread by targeting randomly generated IP addresses (Irwin, 2012b). When randomised IP addresses correlate to addresses within the IP block observed by the telescope, those packets will be recorded (CAIDA, 2012). One such worm is the Conficker worm of November 2008 (Irwin, 2012b), which will be discussed with other worm examples in the analysis section.

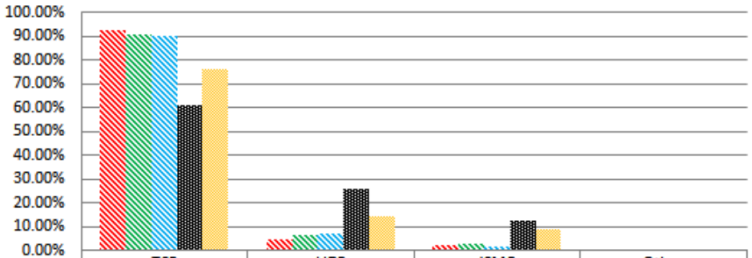
2.7 Dataset Analysis

In the present network security environment, patterns and behaviour of denial-of-service attacks and self-propagating worms have become some of the most important security concepts to understand (Wustrow et al., 2010). The reason for this being that these types of network activity pose one of the greatest threats to network integrity (Kim, Reddy, and Vannucci, 2004). This section will deal with the different analysis techniques and approaches used in the study of network telescope data. Analytical approaches will be discussed, as will some of the interesting statistics or information revealed by this analysis. Part of this section will also deal with highlighting important statistics and identifiers that should be considered in the analysis.

Research on and analysis of network traffic reveals many characteristics about possibly malicious network traffic across the Internet (Wustrow et al., 2010). The interest in this field has led to the development and utilisation of many different methods for packet analysis (Muelder et al., 2005). Some strategies and designs for network traffic analysis are highlighted below.

2.7.1 Network telescope analysis

Time is an important characteristic of packet analysis, and the temporal interaction between IP addresses and ports across a network telescope can reveal a wealth of data (Nkhumeleni, 2014). The analysis of temporal patterns within datasets allows researchers to observe and correlate packet activity patterns (Pang et al., 2004), even allowing analysis of discrete events across multiple darknets (Irwin, 2013). Systematic grouping of certain packet characteristics, such as the protocol used by the received packet, allow for a more abstract analysis, and highlight key trends (Nkhumeleni, 2014). Figure 4 describes the popularity of protocols used in potentially malicious packet exchange (Nkhumeleni, 2014). The graph shows that TCP is the most popular protocol for attempting network interaction, and is used much more frequently than the UDP and ICMP protocols (Nkhumeleni, 2014).



	TCP	UDP	ICMP	Other
196-a	92.49%	5.17%	2.33%	0.011%
196-b	90.66%	6.56%	2.77%	0.021%
196-c	90.45%	7.42%	2.12%	0.011%
146-a	61.34%	25.82%	12.84%	0.004%
155-a	76.34%	14.79%	8.86%	0.004%

Figure 2.3: Breakdown of packet protocol usage across five darknets (Nkhumeleni, 2014)

Aggregate analysis of port activity also yields interesting results. Large amounts of port traffic was observed over port 445/tcp (Nkhumeleni, 2014). This traffic is most likely generated by the Conficker worm, which targets port 445/tcp (Irwin, 2012b). It is possible that the Sasser worm is responsible for some part of the observed traffic as it also exploits port 445/tcp (Harder et al., 2006). There was also a large amount of observed traffic over port 1434/udp, indicating SQL Slammer activity(Bailey et al., 2005b) across the network block.

2.7.2 Analysis across multiple network telescopes

By analysing and comparing data from five distinct network telescopes, it becomes easier to discern patterns in global Internet background radiation (Irwin, 2013). This is because there is less interference from backscatter radiation which is potentially focused on a specific address block (Bailey et al., 2005b). Having data from multiple address blocks allowed for comparison between the results. An analysis of the top 10 TCP and UDP ports across the telescopes revealed that the same targeted ports were almost consistently present across all telescopes (Irwin, 2013). The data also revealed the presence of the same hosts across multiple network blocks, a notable example of this being that more than ten thousand unique hosts were discovered sending packets to 3389/tcp, the Microsoft Remote Desktop Protocol, across all five observed network blocks (Irwin, 2013). Analysing data from distributed darknets gives researchers a more global perspective on unsolicited network traffic (Bailey et al., 2005b). One notable difficulty with using a distributed darknet to gather packets is to make sure that all the darknet clocks are synchronised (Nkhumeleni, 2014). Without temporal accuracy, it becomes more difficult to draw correlations between observed network events across different network telescopes (Nkhumeleni, 2014).

2.7.3 Identification of trends across multiple datasets

The characteristics of malicious IBR are not restricted to specific network blocks. Valuable analysis results can be found by comparing analyses of separate network blocks as it will effectively filter and highlight the re-occurrence of possibly malicious packets (Bailey et al., 2005b).

Analysis was done comparing the results of five distinct /24 network blocks that were contained within the TENET network (Irwin, 2013). A study of the protocols carried by the packets revealed that most of the potentially malicious packets were TCP protocol packets. It was further seen that 99.97% of traffic across these blocks was as a result of either TCP, UDP or ICMP traffic (Irwin, 2013). The observation of various address blocks allows for the aggregation of the top ten source and target destination ports of potentially malicious IBR (Irwin, 2013). Another important aspect of observing multiple datasets is the ability to compare activity across networks to isolate important data, such as worm activity on a single port across address blocks (Nkhumeleni, 2014).

Another study that spanned over 60 darknets in 30 organisations found that the majority of IBR packets stem from relaitvely few unique IP addresses. They also found that some

packet behaviour consists of sources, and to some extent the services being targeted, that are not observable through darknet analysis (Bailey et al., 2005b). Two main types of historic network analysis were presented, the observation of live networks, and the observation of darknet packet data. The issue was raised that neither form of analysis was broad enough to gather all of the available data. As such a scalable hybrid analysis architecture was created that utilised two components. Internet Motion Sensors (IMS) were formed through a collection of distributed darknet sensors (Bailey et al., 2005b). Any new activity recorded on the darknet was then sent to the second component, the Host Motion Sensor (HMS) for further in-depth analysis. This was done in the hope of creating a scalable architecture that could detect new worm behaviour or attacks across the global network in a timely manner(Bailey et al., 2005b). A notable point raised in the paper was that it was more efficient to consider the distributions of source IP addresses rather than the unique source IP addresses themselves when analysing and comparing data from different darknets; especially when an active response is time-critical (Bailey et al., 2005b).

2.8 Packet analysis

Packet headers hold important information about the packet, such as the source and destination addresses, source and target ports, and packet length (Kurose and Ross, 2010). As a network telescope passively captures traffic (Nkhumeleni, 2014) in almost all cases, it becomes difficult to intercept active TCP packets (Irwin, 2011). While UDP and ICMP traffic can be passively gathered (Bailey et al., 2005a), the TCP packets which have a considerably larger presence on the network cannot be analysed as their payloads are never delivered (Irwin, 2013). The packet header of the first TCP-SYN connection attempt, as well as other packet headers, afford researchers the opportunity to better estimate the likelihood that a captured packet had malicious intent.

2.8.1 Analysis of packet size and attack pattern

The growing trend of peer-to-peer network communication across the Internet has made it more difficult to analyse packets for reliable data (Lin, Lu, Lai, Peng, and Lin, 2009). One interesting analysis value is the packet size distribution in the data (Lin et al., 2009). An example of this is the TCP packet size, as TCP is the most common application layer protocol (Kurose and Ross, 2010) observed in many darknet repositories (Irwin, 2013).

A TCP packet cannot be smaller than 60 bytes, and if the mean packet size distribution lies close to this value, it can be inferred that the SYN packets captured on the darknet are connection attempts rather than misconfigured packets (Irwin, 2011). Another way of identifying suspicious packet activity is to look for a correlation between destination IP address and port numbers present in the headers (Kim et al., 2004). An statistically unlikely number of packets with the same ports and destination addresses may indicate that an attack is occurring or that a target machine has been compromised (Kim et al., 2004).

2.8.2 Analysis of Time To Live data in header

Another interesting form of analysis included the systematic study of the Time to Live (TTL) field in packet headers to determine if a scan was a legitimate security threat or a decoy scan to hide the actual scan (O'Connor, 2013). This was achieved as a result of the TTL header decrementing as it travels across routers. Every packet header holds a TTL value, which is decremented as it travels across routers, to prevent (faulty) packets looping endlessly across the network and consuming network resources (Kurose and Ross, 2010). This means that if the scans were unnecessarily bounced across routers they would have a lower TTL than the malicious scan; as a result the origin and location of the malicious scan could be determined (O'Connor, 2013). Figure 5 illustrates the change of TTL information in the packet header as it travels across routers.

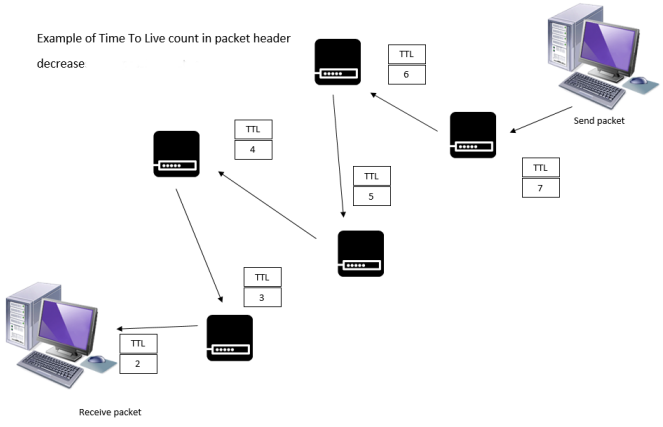


Figure 2.4: TTL header decrements after travelling through router

2.9 Worm Traffic Analysis

In recent years there has been an increase in activity of worms (Harder et al., 2006). Worms are self-propagating malware that seek to take advantages of vulnerabilities of the target system, and then use that compromised system to propagate further (Irwin, 2011). Worm activity will usually be noticeable as the packet data is focused on a single, or small range of, ports as worms seek to exploit specific vulnerabilities (Muelder et al., 2005). Widespread infection creates many dangers for network users, including the possibility of launching unmanageable DDoS attacks from the large number of infected hosts (Staniford et al., 2002).

2.9.1 Conficker

The Conficker worm exploits a netBIOS vulnerability in some Windows operating systems (Shin and Gu, 2010). Systems identified as critical (level of vulnerability) are Windows 2000, XP and 2003; Vista and Server 2008 were identified as important (Bortnik, 2010). Conficker targeted port 445/tcp, through which the vulnerability identified in MS08-067[1] could be exploited (Irwin, 2012b). An observation of captured packet sizes received at port 445/tcp show the majority of them to be 62 bytes in size, which correlate strongly to initial Conficker propagation traffic capture (Irwin, 2012b). Conficker also updates itself by generating new domain names, and then connecting to said names to download a newer version of itself to the compromised host (Shin and Gu, 2010). By cracking the domain-generation algorithm, researchers have been able to use DNS sinkholing, similar to network telescope practice, to intercept packets from the Conficker worm and study them (Shin and Gu, 2010). Analysis of Conficker activity highlighted another interesting trend in worm activity, a flaw in the propagation generation algorithm. Conficker was limited to IPv4 address blocks 0-127 on a /24 subnet in the second and fourth octets of IPv4 target generation (Irwin, 2013). The result of the flaw can be seen in figure 6, where worm activity drops after reaching IP x.x.x.128 for certain subnets, while remaining active in other subnet blocks (Irwin, 2013).

[1] https://technet.microsoft.com/en-us/library/security/ms08-067.aspx

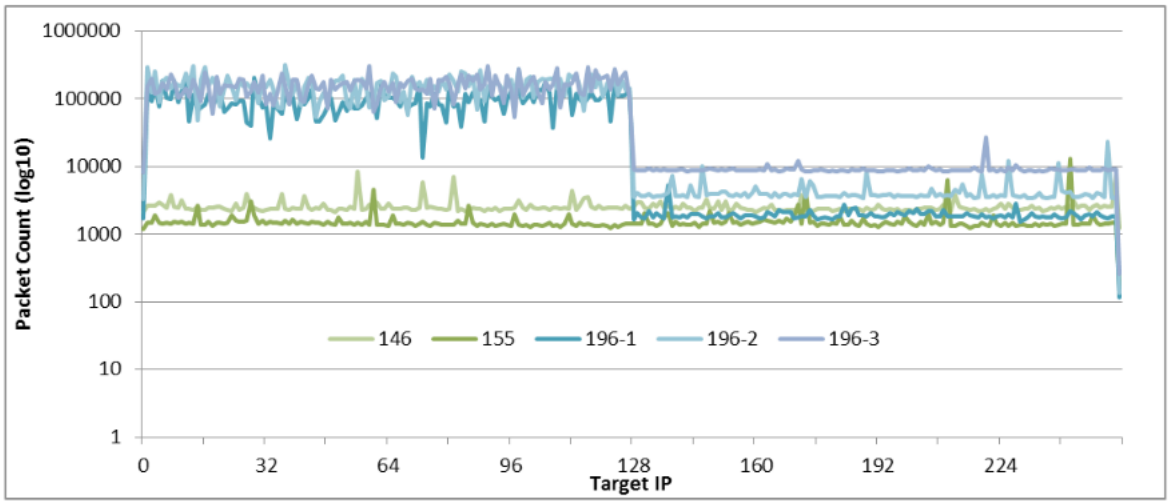


Figure 2.5: Conficker activity across 5 subnets (Irwin, 2013)

2.9.2 Code Red

The Code Red worm was discovered in July of 2001, and exploited hosts by compromising Microsoft IIS web servers (Staniford, Paxson, Weaver, et al., 2002). The vulnerability used is listed under CVE number CVE-2001-0500[2]. The first version of the worm, CRv1, had a flaw similar to that seen in Conficker (Irwin, 2013), where it used a fixed seed for random address generation (Staniford et al., 2002). As a result all newly propagated instances of the worm attempted to compromise the same range of IP addresses. The random number generator was fixed with the release of CRv2, or Code Red I, into the wild on 19th July 2001 (Staniford et al., 2002). Another note is that the newer CRv2 contained a DDoS payload targeting the web server of the White House[3] (Staniford et al., 2002).

2.9.3 Slammer

The Slammer worm is still alive, and produced all of the background radiation observed at port 1434/udp in 2005 (Bailey et al., 2005b). The Slammer worm is an interesting case as the majority of the vulnerable population of end-point systems were compromised in less than 30 minutes (Bailey et al., 2006). The targeted vulnerable systems are those running unpatched versions of SQL Server 2000 and Microsoft Desktop Engine (MSDE) 2000 (Irwin, 2011), through which the worm exploits an overflow vulnerability (Nkhumeleni,

[2]http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2001-0500
[3]http://www.whitehouse.gov/

2014). The Slammer worm activity is characterised by packets that are 404 bytes in size and, as previously mentioned, focus on port 1434/udp (van Riel and Irwin, 2006a).

2.9.4 Sasser

The Sasser worm relies on a two step design to infect hosts: The first relies on buffer-overflow to inject a piece of shell code, which then attempts to download and run an .exe file as the second step, completing the exploit (Bailey et al., 2005b). Port 445/tcp is also the target port of the Sasser worm (Harder et al., 2006). The Sasser worm selects class C networks and performs scans on all hosts of that subnet (/24) (Harder et al., 2006).

2.9.5 Witty

Witty was first observed on the 19th of march 2004, released only a day after the security vulnerability was declared (Shannon and Moore, 2004). The worm exploited a buffer overflow vulnerability, caused by the decoding of ICQ packets (Irwin, 2011), in numerous Internet Security Systems products (Shannon and Moore, 2004). Witty was also the first worm to carry a destructive payload, erasing parts of the victim's hard-disk drive until the machine was reset or the worm caused a fatal system error (Shannon and Moore, 2004). To study the spread of the worm, bandwidth measurement was used as an indicator to the number of infected and active hosts (Kumar, Paxson, and Weaver, 2005). By analysing the frequency and number of Witty packets intercepted by the database, it becomes possible to extrapolate an accurate value for the number of hosts that are infected and active on the network (Kumar et al., 2005).

2.10 Analysis of Patch Tuesday dataset

Patch Tuesday refers to the second Tuesday of each month(Bortnik, 2010); Microsoft releases accumulated security patches on this day (Zseby, King, Brownlee, and Claffy, 2013).

CAIDA researchers put together an analysis tutorial using three tools. Corsaro, pcap trace processing software; Octave scripts, which have numerical computation and graphics capabilities; and tcpdump (Zseby et al., 2014). The dataset was created from data

created by the UCSD Network Telescope (Zseby, King, Fomenkov, and Claffy, 2014) run by CAIDA (CAIDA, 2012). Instead of pushing the pcap into a database format, the researchers chose to represent them in a FlowTuple format (Zseby et al., 2014). From there they started packet analysis, looking at the following identifiers: Number of packets per hour; Number of unique source IP addresses (per hour); Protocol analysis of packets; Port number analysis of packets; Temporal analysis of packet behaviour (Zseby et al., 2014).

The importance of temporal analysis of packet count and unique source IP address hits is further emphasised in another CAIDA paper studying the changes in IBR with relation to Patch Tuesday (Zseby et al., 2013). One significant but circumstantial result of this analysis is the observation of DNS backscatter packets from a DNS name server. The telescope recorded between 4 and 6.5 million DNS backscatter packets in 45 hours before the Patch Tuesday of January 2012. Within two hours after Patch Tuesday, the backscatter stopped (Zseby et al., 2013). One theory is that the released patch prevented the possibly compromised hosts (of a botnet) from participating further in the DDoS of the name server from which the backscatter traffic was being received (Zseby et al., 2013).

2.11 Dataset reporting

Packet capture datasets hold millions of individual packet hits. As such it becomes nearly impossible to manually search through all the data. The sheer quantity of the data also increases the difficulty of packet recognition. Representation of and reporting on datasets enables researchers to abstract the data and infer patters. It also enables the sharing of analysis results between researchers (Irwin, 2013).

2.11.1 Network Scans

Many packets captured through network telescopes are the result of network scanning in an attempt to exploit end-hosts (Muelder et al., 2005). Vertical scans are scans that occur across most or all ports on a single IP address, where as horizontal scans are scans that comprise one or a few ports across multiple IP addresses (van Riel and Irwin, 2006b). An early attempt at visualising network activity, including scans, is the Spinning Cube of Potential Doom (Muelder et al., 2005)(van Riel and Irwin, 2006b), which represents network scanning activity as a line in three-dimensional space. An important note here is that RST traffic, generally considered as backscatter, could be a scan to infer Firewall

policy from an end-host (Moore et al., 2006). Creating fingerprint identifiers of scanning patters simplifies scan recognition, which can be difficult as a result of the pseudo-random nature of some network scan activity(Muelder et al., 2005).

2.11.2 The start of a reporting framework

August 2012 saw the release of a paper documenting the implementation of a simple network analysis and reporting framework. The framework relied on the use of the Winpcap tool; a packet capturing API (Dhillon and Ansari, 2012). The study was run on a lit enterprise network, with the aim of implementing a system that would be able to differentiate legitimate network traffic and attack attempts. The main issue raised was the loose binding between users and their traffic. It is difficult to to accurately determine the source of a packet, as addresses have become dynamic, and are also relatively easily spoofed (Dhillon and Ansari, 2012). Use of the Winpcap[4] tool and the Network Traffic Monitoring application enabled the capture of frames over the network. The captured fields included the source MAC address of each packet. It was analysis on these addresses that determined whether a packet was legitimately created within the network or if it was spoofed from an outside source (Dhillon and Ansari, 2012). From there a simple tabular interface would represent suspicious packets and include information such as the packet number, the date-time the packet was recorded and the log event that identified the packet. Every time a packet with an unknown MAC address was identified it would be logged and brought to the attention of the user (Dhillon and Ansari, 2012).

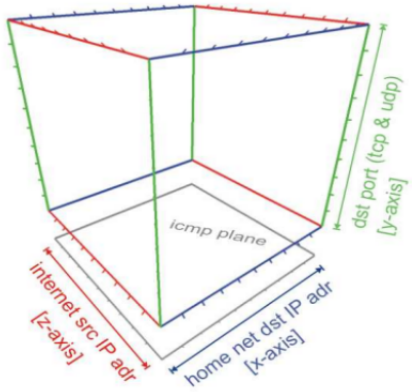
2.11.3 Representing the data

Information is most readily represented by either words or images. People are more readily able to distinguish patterns through visual analysis of data, and will be more likely to observe unexpected patters as well (van Riel and Irwin, 2006a). As such an attempt was made to produce a viable visual analysis tool (van Riel and Irwin, 2006a). The key features of the tool identified include a log plot function to introduce spacial expansion in clusters of data that would have been obscured in a linear plot (van Riel and Irwin, 2006a). The tool is also intuitively time-animated as the temporal order of network events is important in identifying and understanding scanning patterns (van Riel and Irwin, 2006a). The tool also allows for manipulation of the data in the form of

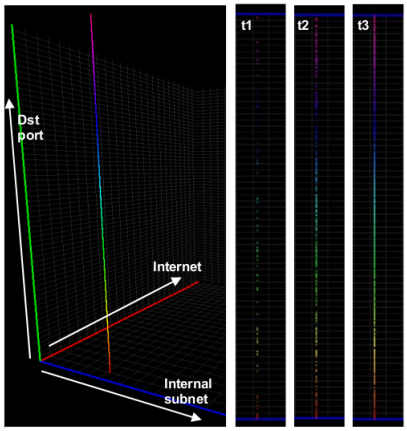
[4]http://www.winpcap.org/

translation, rotation and scaling, ensuring that regions of interest within the visualisation can be isolated (van Riel and Irwin, 2006a). The analysis tool was able to isolate and describe both vertical and horizontal scanning attempts and describe instances of 'Creepy Crawly' horizontal scan attempts from the Slammer worm, as well as random distributed scan activity (van Riel and Irwin, 2006a).

InetVis successfully represents network packet events, and its functionality is improved by its ability to filter events, as well as the fact that you can interact with the represented data temporally (van Riel and Irwin, 2006b). One possible extension that was raised is the implementation of a connection-flow representation as well as the already present packet event representation; a connection-flow being an aggregate of packet events (van Riel and Irwin, 2006b). The InetVis plotting scheme can be seen in figure 7a (van Riel and Irwin, 2006b). Figure 7b shows the graphical output produced by the InetVis tool in a 3d environment, and represents a vertical port scan on a singular IP address of the network (van Riel and Irwin, 2006b).



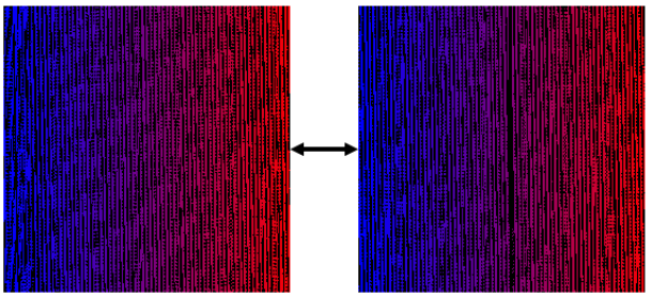
(a) Graphic plotting scheme of InetVis



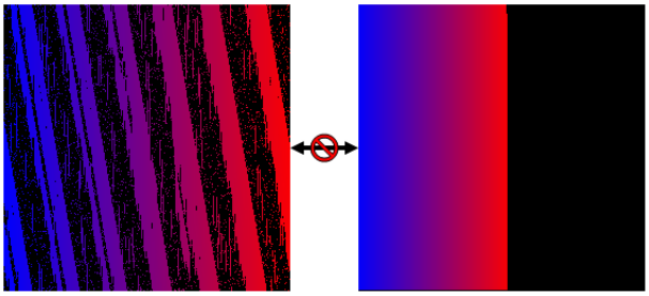
(b) Visualisation of vertical port scan using the InetVis tool

Figure 2.6: Graphical representation using InetVis (van Riel and Irwin, 2006b)

Visualisation of network scan activity enables the viewer to quickly identify network scan patterns (Muelder et al., 2005). The visualisation also allows for easier classification and comparison of different scanning activity across the network block (Muelder et al., 2005). Figure 8a represents two separate scans on the network block that have a similar pattern, which tells the viewer that the scanning algorithms were extremely similar Muelder et al. (2005). Figure 8b shows two separate scan patterns that utilised different scanning algorithms and techniques (Muelder et al., 2005).



(a) Pattern visualisation of similar scans



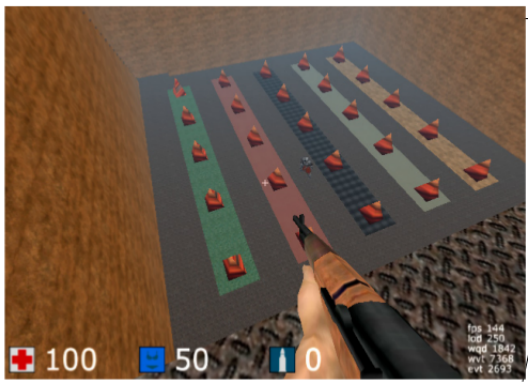
(b) Pattern visualisation of different scans

Figure 2.7: Pattern representations of network scans (Muelder et al., 2005)

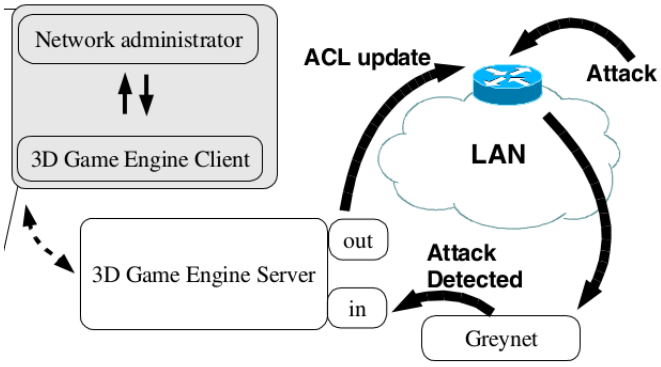
2.11.4 Network monitoring and control through an interactive 3D game-engine

There are many different approaches to network monitoring, and the visualisation of this data is usually in a 2D or 3D format, and may contain interaction capability (Harrop and Armitage, 2006). The Spinning Cube of Potential Doom is a good example of creative representation (van Riel and Irwin, 2006b) and served as a basis for the development of the more interactive InetVis (van Riel and Irwin, 2006a). Creating a more intuitive and interactive representation of network data would reduce network-specific training required for those identifying and suppressing malicious traffic across a network (Harrop and Armitage, 2006). The system relies on an effective translation of network events into virtual-world avatars, and to then translate interaction with the avatars into network reconfiguration events (Harrop and Armitage, 2006). Simple visual metaphors were introduced to describe different network variables and activities. These were then mapped onto network metrics (Harrop and Armitage, 2006). The system allows interaction with the network through a game character, which can perform certain basic actions like shooting an avatar to interact with the network in some way (Harrop and Armitage, 2006). Added functionality comes from the ability to set multi-user acceptance of reactive net-

work events before they are carried out (Harrop and Armitage, 2006). The system will create avatar-bots to represent certain network metrics, including IP address and port number - represented by location; and the content type of the data - represented by the colour and texture of the object. User defined alerts would be represented by oscillation about a fixed point (Harrop and Armitage, 2006). It was posited that the human mind was better suited to pattern recognition, as opposed to computer algorithms; which would make this visualisation of network data more beneficial to and efficient for users (Harrop and Armitage, 2006). Figure 9a shows the graphical interface of the monitoring tool, while figure 9b shows a simple representation of the subsequent network interaction in the background (Harrop and Armitage, 2006).



(a) Representation of user interface



(b) Diagram of system interaction with network

Figure 2.8: Interactive network activity mapping (Harrop and Armitage, 2006)

2.12 Internet Motion Sensor

The Internet Motion Sensor (IMS) is a system that was designed to give new insight into current Internet security threats (Bailey et al., 2005a). The system trades in-depth

information for global visibility, taking in less data overall but using distributed darknets for added network visibility (Bailey et al., 2005a). An important aspect of the IMS is that it makes an active attempt to complete a TCP connection(Cooke, Bailey, Watson, Jahanian, and Nazario, 2004) by sending a TCP-ACK request and listening(Kurose and Ross, 2010), in order to capture the payload of the possibly malicious packet sent across the connection(Bailey et al., 2004). TCP packets hold the only data that need to be actively collected as UDP and ICMP packets can be passively gathered (Bailey et al., 2005a). The lightweight responder used to gather TCP payload data is designed to take an MD5 hash of the gathered payload, only storing payloads with a unique hash (Bailey et al., 2004). Analysis revealed an interesting trend in network scan activity. Some worms, after infecting the hosts, install backdoors into the system (Cooke et al., 2004). It was observed across networks that known backdoor ports of previous worms were being scanned, possibly by opportunistic attackers seeking to create botnets from previously compromised hosts (Bailey et al., 2005a). A DoS attack was observed December 10 2003; as a result of address spoofing some of the backscatter was observed by the distributed darknet (Cooke et al., 2004). The packets were received in five discrete network events, three targeting the web server of www.sco.com at port 80, one DoS attack against the FTP server at port 21 and another attack on the SMTP mail server, port 25 (Cooke et al., 2004).

2.13 Hilbert Curves

The Hilbert curve, first described by David Hilbert in 1891, represents a space-filling curve that fills more space as its order is increased (Cowie and Irwin, 2010).The Hilbert Curve visualisation tool was developed with the intention of providing a high-level analysis tool for large network packet captures; including analysis among a distributed network telescope array (Irwin, 2011). Hilbert curves of higher orders can be used to group certain classes of network block (Irwin, 2011), and curves of the order 4, 8, 12 and 16 hold 256, 65536, 16,777,216 and 4,294,967,296 represented points (Irwin and Pilkington, 2008). This holds value as a method of representation as they correspond to the natural grouping of network blocks /8 (Class A) /16 (Class B) and /24 (Class C), where the class is representative of the size of the measured network block correlates to the number of points; with Class C holding 256 unique IP addresses, Class B holding 65536 unique IPs and Class A holding 16,777,216 unique IPs (Kurose and Ross, 2010). Hilbert curves allow for the evaluation and comparison of the efficacy of network telescopes with regards to the

size of the observed IP block (Irwin and Pilkington, 2008). Hilbert curves similarly allow researchers to visualise the effectiveness of network worm propagation algorithms (Irwin and Barnett, 2009) across different network block sizes (Irwin and Pilkington, 2008).

2.14 Analysis approaches in other fields

This section covers work that, while not entirely relevant to the topic in question, represent different approaches to similar problems.

2.14.1 Waikato Environment for Knowledge Analysis Data Mining Software

The Waikato Environment for Knowlege Analysis (WEKA) software tool was conceived in 1992 with the aim of supplying researchers with a unified data-mining workbench with machine learning capabilities (Hall, Frank, Holmes, Pfahringer, Reutermann, and Witten, 2009). WEKA incorporates many learning schemes, developing and adding to the orignial schema available to researchers since 1992 (Hall et al., 2009). Some of the features include data classification, data cluster detection, attribute selection and filtering, as well as association rule discovery (Du, 2010). WEKA also allows the user to specify preprocessing filters as part of an enhanced preporcessing tool package introduced with the new release (Hall et al., 2009). WEKA is used by many disciplines as a data-mining tool; some of the projects to use WEKA include: Systems for natural language processing; Distributed and parallel data mining; Open source data mining systems and integration with the Kepler open source workflow platform[5] as part of the Kepler Weka project (Hall et al., 2009).

2.14.2 Passive IP traceback

It is difficult to deploy an Internet-scale IP traceback system, in part because of the need for cooperation between Internet Service Providers (Yao, Bi, and Zhou, 2010). Yao et al. (2010) introduces a system that relies on passively gathering ICMP message backscatter, which is generated by routers as the packets travel from attacker to victim. While the

[5]https://kepler-project.org/

reflection routers, routers that reflect ICMP traffic to network telescopes like CAIDA (CAIDA, 2012), could be pinpointed through IP address mapping, it becomes impossible to accurately trace the packets to the source as a result of a lack of information on the current state of the Internet topology (Yao et al., 2010).

2.14.3 Geo-location of received packets

The need for network visualisation software became apparent after Operation Aurora, a targeted network attack focused on Google, Adobe and over 30 Fortune 100 companies (O'Connor, 2013). Particularly, it was considered useful to correlate IP traffic with geographical locations. This would have vastly increased the response time against the Aurora attack, as it would immediately have highlighted a frequent and seemingly unnecessary connection from multiple end-points to a web-server in Taiwan and another server in China (O'Connor, 2013). The book "Violent Python" discusses what steps would need to be taken to implement such a system. The MaxMind open source GeoLiteCity database was used to correlate source IP addresses to city-based accuracy (O'Connor, 2013). The database was queried using the PyGeoIP library, produced by Jennifer Ennis (O'Connor, 2013). Dpkt was used to parse the captured pcap data, and from there the PyGeoIP script was run on the isolated source addresses. KML[6] files were then used to create markers on Google Maps relevant to the source and destination IP address hits, as a means of visualising the information. If the coordinates returned a value found on Google Maps, the marker was created. If the coordinates returned a result such as "location does not exist", the KML file returned an empty string instead (O'Connor, 2013).

2.15 Summary

The analysis of Internet Background Radiation has the ability to reveal a wealth of information about the practices, trends and characteristics of malicious network traffic. Packet capture, the first step of IBR analysis, can be completed on a localised or distributed network telescope. This in turn allows researchers to compare packet capture results between darknets, as long as there is a standardised temporal frame for the different telescope data-sets. The analysis of darknet data-sets is not limited to any one methodology or focus; as such there is a wealth of information gained and extrapolated from packet capture analysis

[6]https://developers.google.com/kml/documentation/

across multiple fields in the Network Security industry. The diversity of analysis methods and results is presented for the reader's consideration in this literature review. While this approach allows for innovation within the field of Network Security, it limits the ability to compare results between data-sets as there is little to no standardisation on the reporting of the analysis results. While a standardised reporting framework would possibly narrow the scope of new analysis techniques, it would provide a frame of reference within which results from isolated darknets could be easily compared. The second obstacle within the field of Network Security is that it is a time-sensitive pursuit. New vulnerabilities, as well as the programs that exploit them, need to be discovered and countered as quickly as possible to minimise damage to end-hosts across the network. The creation of a reporting framework would hopefully allow for the timely identification of new trends among more recent darknet packet captures. The next chapter hopes to introduce a system design for such a reporting prototype. Chapter four will look at reporting output generated by the system.

Chapter 3

Design

This chapter presents the design of the system and will have three main sections. Section 3.1 discusses the isolated parts that make up the system, as well as how they interact with one another. Section 3.2 deals with actually building the system, and includes challenges and constrains faced during system implementation. This section will also look at the reasons for and development of the ancillary reporting system created during the implementation phase of the project. Section 3.3 deal with the evaluation of system performance. The alignment of the system with the research goals is also evaluated.

3.1 Datasets

Five datasets were used to test the reporting capabilities of the system. The sensors used to gather the datasets have been divided into two categories. This is not only because of their address distance, but also as a result of the similarity of traffic across these networks, and the differences seen in the packet count (Nkhumeleni, 2014). The five monitored IPv4 address blocks are contained within the TENET[1] (AS2018) network (Irwin, 2013). The observed blocks exist in three distinct top-level IPv4 network address blocks: 146/8, 155/8 and 196/8. All five datasets were taken from the period 04 July 2013 to 12 February 2014, a total of 224 days of packet captures.

[1] http://www.tenet.co.za/

Name	First packet received	Last packet received	Total number of packets	Category
146.x.x/24	04-07-2013 08:53:01	12-02-2014 14:55:23	8663883	A
155.x.x/24	04-07-2013 08:52:48	12-02-2014 14:54:59	9256741	A
196.21.x/24 (1)	04-07-2013 08:52:47	12-02-2014 14:54:48	16364801	B
196.21.x/24 (2)	04-07-2013 08:52:47	12-02-2014 14:53:30	16398051	B
196.24.x/24	04-07-2013 08:52:51	12-02-2014 14:54:15	15523596	B

Table 3.1: Breakdown of datasets

3.1.1 Category A

The 146.231.x/24 and 155.232.x/24 data captures are included in the same category because of the characteristics that they have in common. They have a similar number of recorded packets across the timeframe. They also share similarities in the logged traffic. They are also much closer to one another logically on the IPv4 address list than the 196 subsets. The similarity of these darknets is apparent in the comparison of darknets in section 4.1. The breakdown of the packet caputres in Table 3.1 also give evidence for the similar spread of traffic across the various datasets.

3.1.2 Category B

196.21.x/24 (1), 196.21.x/24 (2) and 196.24.x/24 are grouped into this category also because they have approximate packet counts and are logically close. They also share this category as a result of the influx of traffic they receive on port 445 as a result of Conficker (Nkhumeleni, 2014).

3.2 Tools and Techniques

This section looks at the tool choices made during the design stage of the project. These tools and techniques are then used in collaboration to create the process components described in the next section.

3.2.1 JSON

JavaScript Object Notation[2] (JSON) is a human-readable, lightweight data format. JSON is a text based data format that is built on two structures. The first is a collection of name/value pairs, similar to the structure of a dictionary or hash table. The second is an ordered list of values, in this case an ordered list of the name/value pairs. This data structure was selected because of the inherent closeness between it and Python dictionaries. This simplifies the parsing of the data which is then processed and manipulated using Python. It was also selected because JSON is an almost universally readable format, and can be utilised not only by various languages, but also incorporated into JavaScript in a web-based platform.

3.2.2 Ipv4-heatmap

Ipv4-heatmap[3] is a mapping tool that graphically represents IP addresses on a map of the Internet (Irwin and Pilkington, 2008). The map of the internet is represented by a twelfth order Hilbert curve (Irwin and Pilkington, 2008). Hilbert curves have been previously introduced in section 2.13 and will not be dealt with here. Ipv4-heatmap was developed by The Measurement Factory. It allows not only the mapping of IPs to a Hilbert curve, it also allows the user to create a graphical overlay of where IP addresses are currently allocated. The output created is a 4096 x 4096 pixel block, where recorded IP addresses appear as coloured pixels while non-recorded IPs remain blank. At this resolution each pixel represents 256 hosts present in a single /24 network block. An example of the IPv4-heatmap output can be seen in Figure 3.1.

[2] json.org
[3] http://maps.measurement-factory.com/software/ipv4-heatmap.1.html

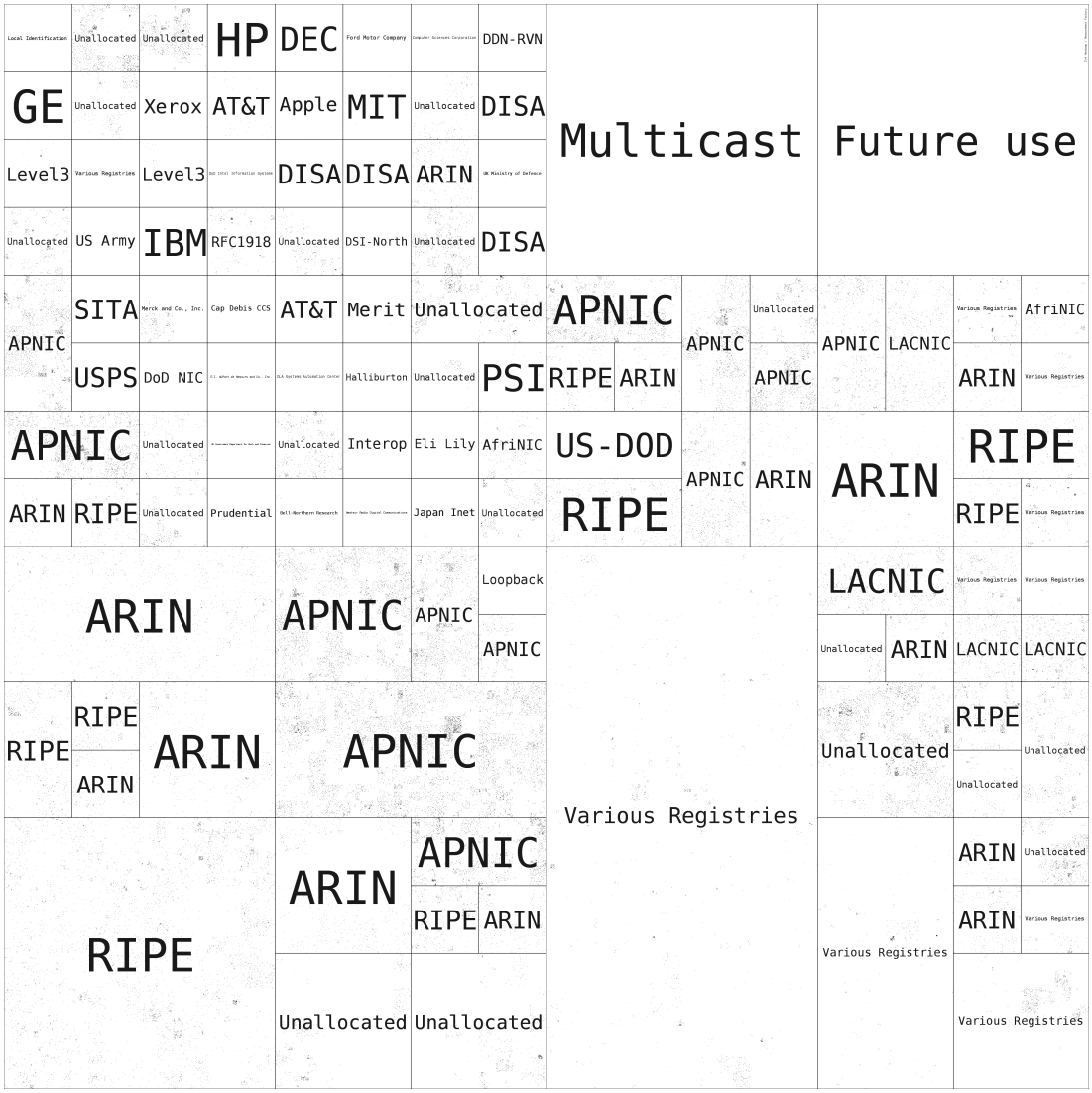


Figure 3.1: Example Ipv4-heatmap output

3.2.3 Python

The selection of Python as the programming language was the first step in designing the system itself. Python integrates exceptionally will with Linux, this itself a major reason for selecting the language. Python also offers an almost limitless amount of user created libraries, allowing the language to be flexible to the needs of the user. As a scripted language it also gives excellent performance with regards to string manipulation and dictionary searches, integral concepts which would enable the development of the system (Prechelt, 2000).

3.2.4 Dpkt

Dpkt[4] is an open-source, stand-alone Python module that can be used to create and parse packet capture (pcap) files. Pcap files, as previously mentioned, are files that hold the information of packets that have been captured through the use of a network telescope. While Dpkt can be used to create pcap files, this functionality has been largely ignored as it exists outside the scope of the project. It has been used here as a result of its ability to parse pcap files, after which the information can be manipulated and stored in a more human-readable format. Dpkt was selected for two main reasons. The decision to create the system using Python led to the first advantage of Dpkt, that it is a Python module. It would thus be easier to integrate the different components if they functioned using the same language. The second advantage is that Dpkt is fast. The drawback here is that completeness is sacrificed for speed. Dpkt decodes single network packets, which raises two key issues, namely that the module cannot deal with the fragmentation of packets at the IP level or the TCP level. Dpkt does, as a result of fragmented and corrupted packets, throw errors as it is not comprehensive. This is not considerably alarming, as the system is meant to highlight trends and unexpected abnormalities, not create a complex breakdown of packets or report the results in real-time.

3.2.5 Pandas

Pandas[5] is an open-source library that offers easy to use data structures and data analysis tools for Python (McKinney, 2011). pandas is built on top of the Python matplotlib library, and can be incorporated with other libraries such as iPython and numpy to increase its flexibility as a data reporting library (McKinney, 2011). Pandas was chosen as a tool because of the flexibility of both its data structures and its graphical output abilities. Speed of processing was also a consideration when selecting the pandas library for graphical output.

[4]https://code.google.com/p/dpkt/downloads/list
[5]http://pandas.pydata.org/

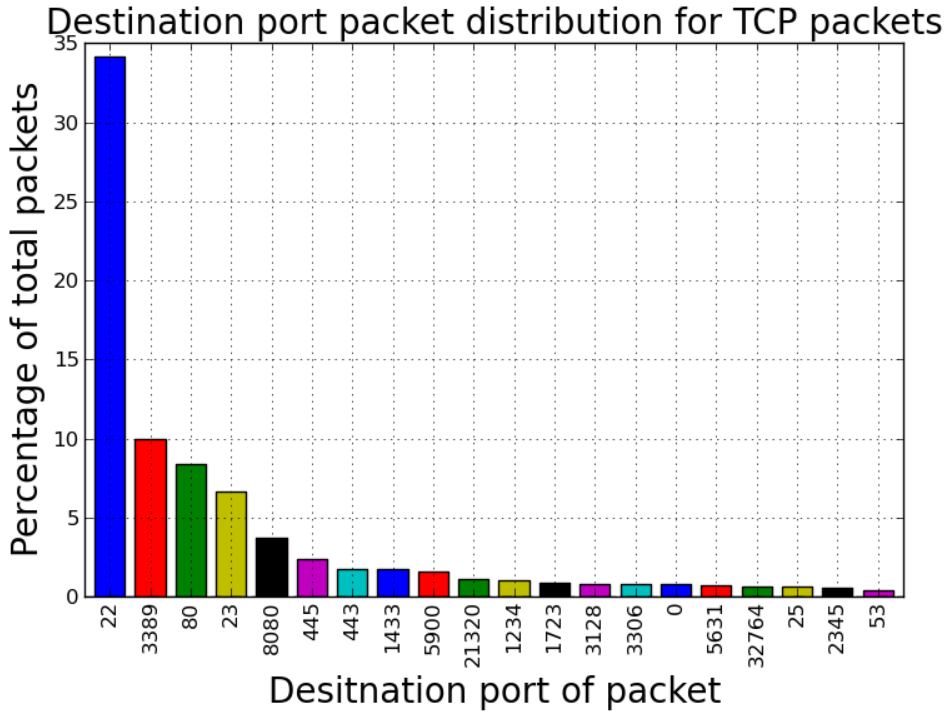


Figure 3.2: Example pandas output

3.2.6 PyGeoIP

PyGeoIP[6] is a python library based on MaxMind's GeoIP C API[7]. PyGeoIP uses the freely available MaxMind database[8]. The database is then queried by the library and returns information, including the country of origin, related to the IP address queried against it.

3.2.7 Latex

Latex[9] is a document preparation and formatting system that is used to create professional documents (Kopka and Daly, 1995). It has the ability to create documents that contain both graphics and neatly formatted mathematical formulae (Mittelbach, Goossens, Braams, Carlisle, and Rowley, 2004). Latex can be used to create documents in a number of styles including pdf files (Kopka and Daly, 1995).

[6]https://github.com/appliedsec/pygeoip
[7]https://github.com/maxmind/geoip-api-c
[8]http://dev.maxmind.com/geoip/legacy/install/country/
[9]http://latex-project.org/ftp.html

3.3 System components

This section looks at the various components as well as how they interact to achieve the goals of the system. Many of the system parts are interrelated as a result of requiring output from a previous component. There are no components that are required to run in parallel. Each of the components are referenced as a labeled process that produces output. The labels on Figure 3.3 correspond to the program components which are listed below.

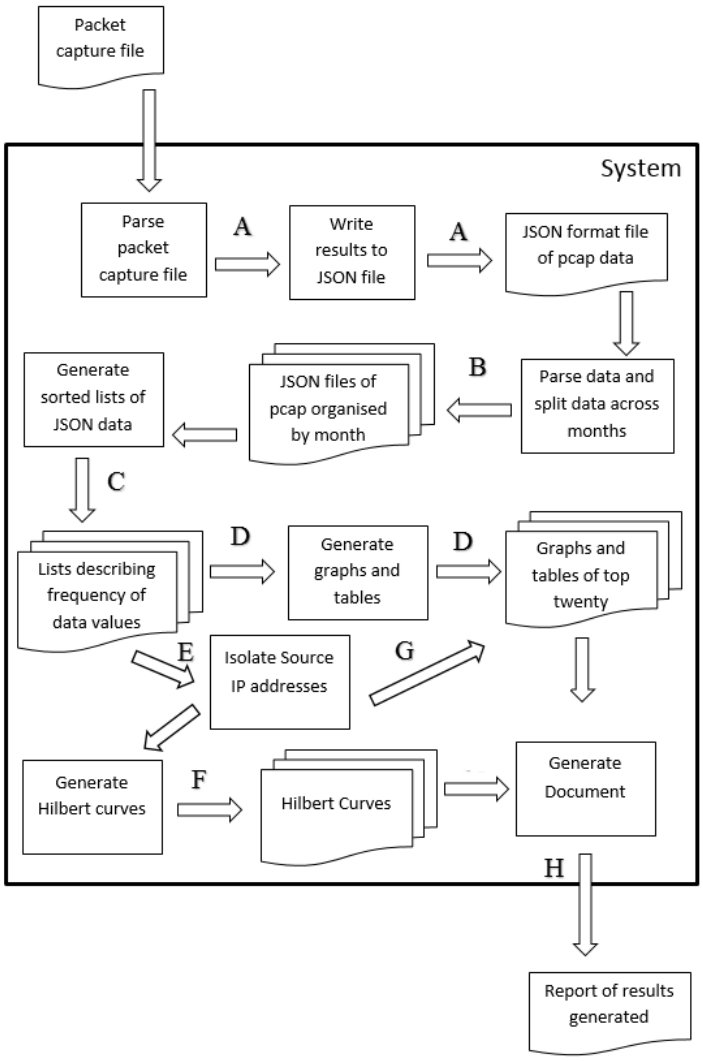


Figure 3.3: System diagram

3.3.1 Dpkt to JSON parser

The first component of the system, represented by process (A) of Figure 3.3, converts the pcap file into a JSON data format file. The file is first parsed by the Dpkt module. Dpkt methods then return valuable information that the packet contains. These returned values are as follows: Timestamp of the packet arrival at darknet; Source IP address of the packet; Destination IP address; Length of the packet; Source port; Destination port and the Protocol that the packet uses. These values are then encapsulated into a JSON format text file, where each packet is represented by an object and the values returned by Dpkt make up the values of the name/value pairs. While the parser currently reads all packets, the program can also filter only desirable packets by using certain criteria.

3.3.2 JSON datafile splitter

Process (B) is the second component, which parses the JSON file and reads the timestamp on the packet as it parses. It also creates a JSON file named after the darknet and the packet month. If the month present in the packet timestamp is not the same as that of the previous packet, the created JSON file being written to closes and a new month JSON file is created and written to. The splitter is set to use months as the unit of separation, as it gives a more detailed overview of packet activity than an analysis of the entire pcap JSON file, but does not create such a data overload that it would be difficult to see correlations and discrepancies created by packet behaviour. The splitter can however be set to split the parsed packets across any timeframe, be it days, weeks, years etc. This is useful as it will allow us to isolate timeframes where packet activity has been observed to be of interest.

3.3.3 Pandas series data formatter

Component (C) parses the JSON files, using each name/value pair to generate separate dictionaries for every name in the packet object. Each dictionary is then populated with unique name values, and each name in the dictionary is followed by the count of that particular name, i.e. how many times that unique value appears in the dataset. This will create a structure as follows: {10.0.0.1: 15} for source IP 10.0.0.1 appearing 15 times. When all of the dictionaries have been compiled from the available data they are converted into a list of sorted tuples. The tuples are sorted from most frequent to least frequent

and then written to a text file, with each list of tuples generating a new text file. These text files are essentially lists of ordered tuples with a unique first value.

3.3.4 Graph and table generation

The fourth component, (D), parses the list of tuples and isolates the top twenty most frequent occurrences in the list. Here every tuple value that is parsed is also summed to give the total number of packet hits in the list It then divides the tuples into a series of values (the y axis of the graph) and a series of indices (the x axis of the graph). The pandas library is then used to generate graphical representations of the current data. At this time tables are also generated containing the name of the tuple, the frequency of the tuple hits and the percentage of total packets attributed to this name value. Individual tables and graphs are created from each of the tuple lists.

3.3.5 Source IP isolation

Component (E) is formed of a simple python program that strips the first tuple values of the Source IP list and writes them to a text file now containing all unique IP addresses in the currently analysed block of the pcap file.

3.3.6 Hilbert graph generation

The sixth component locates the list of IPs present in the target folder and uses the python OS library to execute ipv4-heatmap in the terminal and feed it the list of unique IP values, as well as a graphical overlay of IP distribution based on IANA records, and an output name. A .png format graphical representation of the Hilbert curve is then generated and saved to the folder. This is represented by process (F).

3.3.7 Geolocation

Process (G) of Figure 3.3 reads in the list of IPs created by the fifth, and uses PygeoIP and the Maxmind IP Country database to correspond IP addresses to the country that has been allocated their IP block. This output is then written to a list in the format required by the fourth component, which can then also render the data graphically.

3.3.8 Document Generation

The last process, (H), generates a Latex skeleton for the report. Graphs and tables are then isolated in the folder and displayed in chronological order grouped into sets identifying the trends in the darknet on a monthly basis as well as then giving a holistic overview of the darknet traffic. The overall analysis also includes a Hilbert curve plotted from the source IP addresses that contact the darknet over the course of the packet capture.

3.4 Implementation

This section discusses key packet information highlighted in the literature review that would be useful for analysis purposes. It explores the problems encountered during the implementation of the system itself. Section 3.4.3 will discuss the design of an ancillary system, which was created as a result of difficulties identified in 3.4.2. The last section discusses the requirements and constraints of running the system.

3.4.1 Key Statistics and Identifiers

During the literature review there were many avenues of analysis discussed with repsect to network packet analysis, as well as statistics and identifiers that could give the researcher a greater understanding of the packet capture data. There is a large amount of relevant security information that can be gathered from packet analysis (Irwin, 2012a). The isolation of unique IP addresses, as well as the analysis of payloads from the source, i.e. analysing the source and target ports as well as the protocol used, can give us an idea of the size of an aggressive botnet or the degree to which a worm or other virus has spread (Zseby et al., 2014). Creating a top-ten listing of the statistics of potentially malicious IBR can give insight into trends or patterns within the traffic (Irwin, 2013), more so if data is available from a distributed darknet (Nkhumeleni, 2014). Horizontal and vertical scans (van Riel and Irwin, 2006a). Studying the size of the packet can lead to possibly identifying the intent behind the packet (Kim et al., 2004). Lin et al. (2009) identified a useful grouping of metrics in the form of a tuple that holds Source IP, Destination IP, protocol, source port, destination port . Irwin (2012a) defined a more complete set of metrics broken down into three categories: Top item trends within the packet capture

must be highlighted; temporal aspects of the packet capture must be well documented; ratio of active traffic to backscatter (Irwin, 2012a).

3.4.2 Difficulties encountered

The first issue that affected the design of the system arose early. The Dpkt module threw AttributeError exceptions when handling certain rare packets. Some of the information of the captured packets could not be recovered by Dpkt, and the program would in turn return an error. This was overcome by using try-catch statements and replacing the irrecoverable values with a placeholder, in this case "-1", which would then be filtered out by later components.

The second issue was the choice between CSV and JSON as the data format. Originally CSV had been chosen, as the created files are smaller than their JSON counterpart. The pandas library also had methods designed to interact with CSV files. In the end JSON was chosen for three reasons. The first was that JSON was more human readable. The second was that JSON is a more flexible data format, and integrates well with JavaScript. The third is that JSON data structures are easier to manipulate in python because of their similarity to python data structures.

Another area of difficulty was the lack of Dpkt documentation available. This made it difficult to identify and correctly use the methods in the Dpkt library to retrieve the needed data from the pcap file.

Scope change and scope creep management were problematic throughout the course of development. The system could be built around practically anything. It has the ability to showcase multiple different combinations of analysis results. This lends itself to being easily changed during the course of development, which makes it difficult to decide which areas of the system are more important, and which should be discarded. It is also inherently difficult on choosing a focus for both the data in the report, as well as the style in which it is displayed.

One of the greatest challenges was introducing flexibility to the report while retaining a report structure that was generic enough to allow comparisons between reports. It was also a challenge to find the balance between too little information versus an information overload. The decision was then made to create an ancillary system. This is discussed in more detail in section 3.4.3.

3.4.3 Development of an ancillary system

The report serves as a summary of the packet activity across the sensor set under study. It does not have the ability to analyse and graph an element of interest within the dataset that it identifies. To perform more in-depth analysis on the JSON datasets, the code used for the sorting and graphing of the data was adapted, and some new code written to produce different graphs. These programs used an identified filter to isolate the information from relevant packets while ignoring the 'noise' of the rest of the Internet background radiation. These programs require user input as the report generation software does not have the ability to identify areas of interest within the analysis and automatically deconstruct them. They filter the dataset using an object value identified as important in the original report, and create a report where the output is focused on the filtered value and its packet traffic captured by the darknet. The programs produce a timeseries of all packets received as well as a cumulative time series. Both of these outputs can be seen in Figure 3.4.

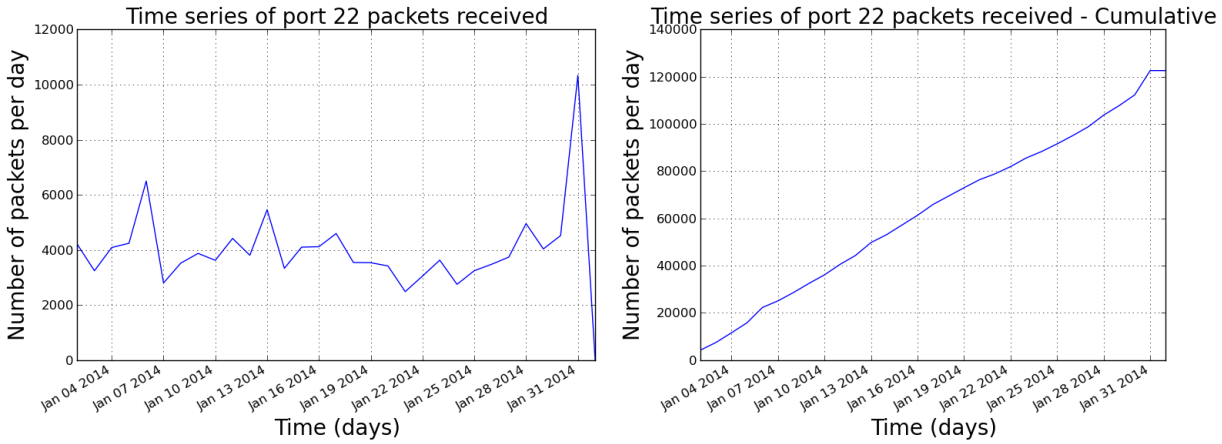


Figure 3.4: Example Time-series output

Two other graphs are also produced, a scatter plot of Destination port vs Time, an example of which can be seen in Figure 3.6, as well as a time-series of a specific port, both with reference to a single source or destination IP address.

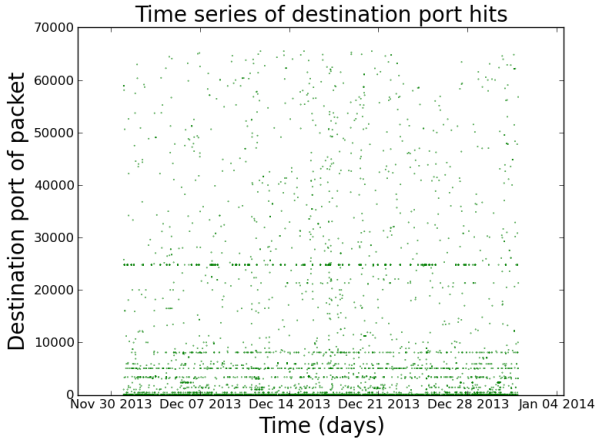


Figure 3.5: Example scatter plot

Many of the graphs created will be useful in helping to explain the results of the analysis, and will be included to better illustrate the packet capture data. These programs were then incorporated into an ancillary system to the main project. The letters on figure 3.6 refer to the same parts of the system found in Section 3.3. Parts I and J are new editions and will be discussed below.

Process (I) of Figure 3.6 is a program that parses the JSON format dataset and creates separate list files for each present object. These files are then read in by process (J) which plots the packet information against time, to better represent the temporal nature of the packet traffic intercepted by the darknet. Processes (D,E,F,G) act as expected. Processes (C) and (I) also filter the packet data based on an identified value of interest, e.g. a single destination IP address in the dataset. This value, identified through the report, is then used to generate another report that filters against one or more values for a specific object in the packets of the dataset. Process (C) differs from the original document generation process only through the value filter which is applied to the packets, ensuring that only the relevant analysis packets are recorded. Processes (I) and (J) are completely unique to the ancillary report, creating lists of packet values that are then used to generate time related graphs. These temporal representations of the packet data are unique to the ancillary report and do not appear in the main report.

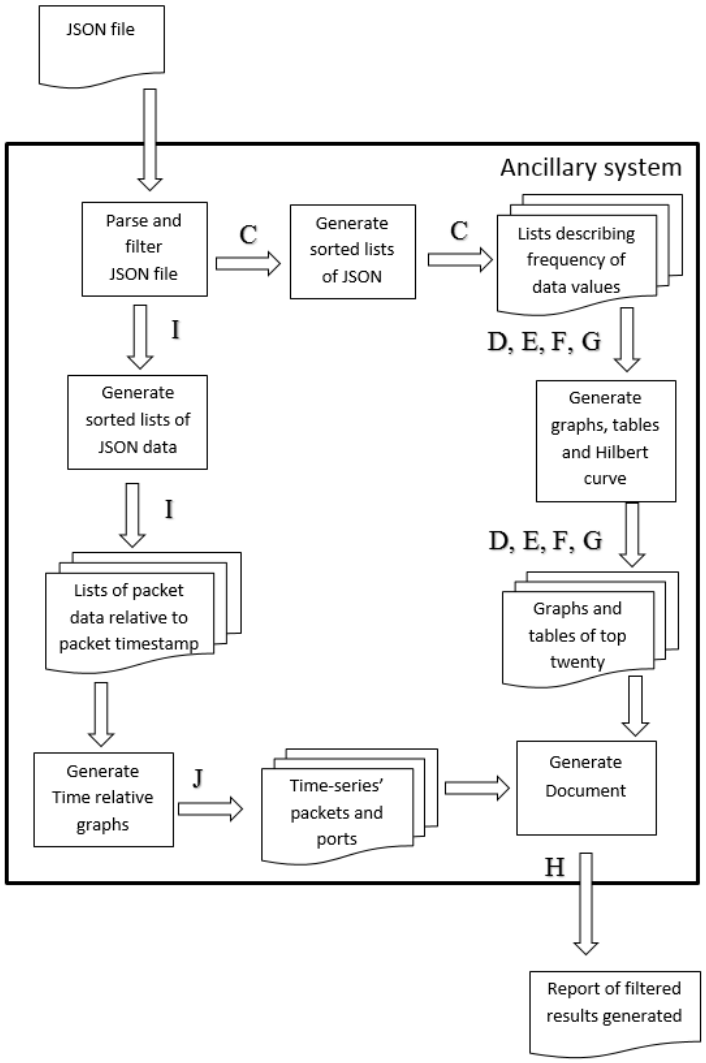


Figure 3.6: Ancillary system diagram

3.4.4 System constraints

It is important to note that the system will strain the memory resources of the user's PC if the input packet capture file is too large. A pcap file that is larger than the available memory will crash the system, as the ability to manage extremely large pcap files was not included in the system specifications. Pcap files that are found to be too large should be split into multiple files before serving as input for the system.

It is recommended that the host running the system have twice as much memory available as the size of the pcap file to ensure that the analysis and report generation happen in a timely manner.

3.5 Evaluation

This section will deal with evaluating the ability of the system to achieve its goals. It will compare these goals to those set out at the beginning of the paper before concluding whether or not the system has remained relevant.

3.5.1 Output of the system

The system creates a document that describes the activity of packets recorded in a pcap file. The length of the report is dependent on the number of months that the pcap scans. While the system can be set to weeks, days etc. a monthly summary is the default setting. For each month the document will contain a summary of the most frequent packet activity for a number of information objects: Destination IP, TCP destination port, UDP destination port, Source IP, Source port, Protocol used, Geolocation on source IPs. All of these graphs and tables The final analysis results summarise the entire dataset and include a Hilbert curve of the source IPs recorded for the period of dataset capture. The ancillary report produces a report that focuses on a particular filter, and only for the dataset that is being filtered; be it monthly or the total dataset. This secondary report also contains a time-serises, cumulative time series, destination port vs time plot, as well as time-series plots of specific ports.

3.5.2 Goals that the system achieves

The initial analysis performed by the system targets quantitative results based on information that can be derived from the packet header. Each packet is processed and the data that has been identified as useful is stored. The system then summarises, tabulates and graphs the data obtained from the packets, comparing the frequency of packets based on certain fields. The system gives both a breakdown of the total packet capture as well as a breakdown by month. These results are then displayed in chronological order, each set of data grouped into its month. The generated document also leaves space for the user to write observations or comments on the document. The ancillary system filters the dataset using an identified value. It produces a drill-down of the behaviour of packets related to the identified value within the dataset.

3.5.3 Achievement of research goals

The system is able to generate report documents using Latex. A general report of packet activity throughout the packet capture, as well as a drill down report that can filter packet values of interest. The system is able to analyse a pcap file and return tabular and graphical output representing the data. The system is also able to produce standardised reports irrespective of the pcap. The second and third research goals have been met with regards to the prototype system. It is felt that, while the system does exhibit a certain amount of flexibility with regards to its reporting output, it is still largely limited by human interaction with the system, both in the identification of results that require further investigation and in the ability to automatically create a more case-specific report. This is mitigated however by the existence of the ancillary report generation, which allows a greater degree of flexibility for analysis than the original report framework. The achievement of these goals as well as possible future work on the system will be addressed in section 5.3.

3.6 Summary

This chapter has looked at the design and implementation phases of the system. The design elements of the system were affected by the findings in chapter two, which in turn had effects on the implementation of the system. The parts of the system have been documented, as well as the interaction between those parts and the output created by the system. Challenges faced during the implementation are discussed, including the decision to create an ancillary system based on the perceived lack of flexibility exhibited by the first report. The system is also critically evaluated in this chapter. It was found that the system completed two of the three research goals, and reached a level of competence in giving the report flexibility with regards to its analysis and output. Chapter four uses the graphical and tabular output generated by the report generation systems to perform case studies and further analysis to better showcase the strengths of the system.

Chapter 4

Analysis

This chapter looks at selected case studies identified by the reporting system data. Interesting trends that have been identified are deconstructed in an attempt to gain greater insight into what is happening in the packet capture while also demonstrating the capabilities of the reporting system. The first case study is a comparison of results for the 146 and 155 (Category A) darknets. This first case study is conducted in the attempt to show the strengths of the report with regards to trend identification across a darknet, as well as its ability to compare results from different darknets. The other case studies focus on interesting packet activity identified by the comparisons conducted in the first case study, in an attempt to show the flexibility of the system, as well as its ability to describe more discrete packet activity as opposed to the general flow of packet activity across a darknet.

4.1 Case Study: Comparison of the 146 and 155 darknets using reporting output

The 146 and 155 datasets have been placed in the same category as a result of the similarity of their packet captures (Nkhumeleni, 2014). This is useful to illustrate the value of the system output with regards to comparing results for two separate datasets. This section will focus on introducing some of the initial analysis results as well as create a contextual setting for the analysis of the following sections. The analysis of the following case studies are all generated by filtering packet activity by values determined to be of interest in the original report.

4.1.1 Destination IP and port activity for July and August 2013

Rank	Destination IP	Number of packets received	Percentage of total packets
1	146.x.x.110	13667	1.620
2	146.x.x.105	11676	1.384
3	146.x.x.114	8599	1.019
4	146.x.x.118	8359	0.991
5	146.x.x.119	6461	0.766
6	146.x.x.36	5701	0.676
7	146.x.x.5	5233	0.620
8	146.x.x.50	4806	0.570
9	146.x.x.62	4628	0.549
10	146.x.x.57	4611	0.547

(a) 146 07/13 dataset

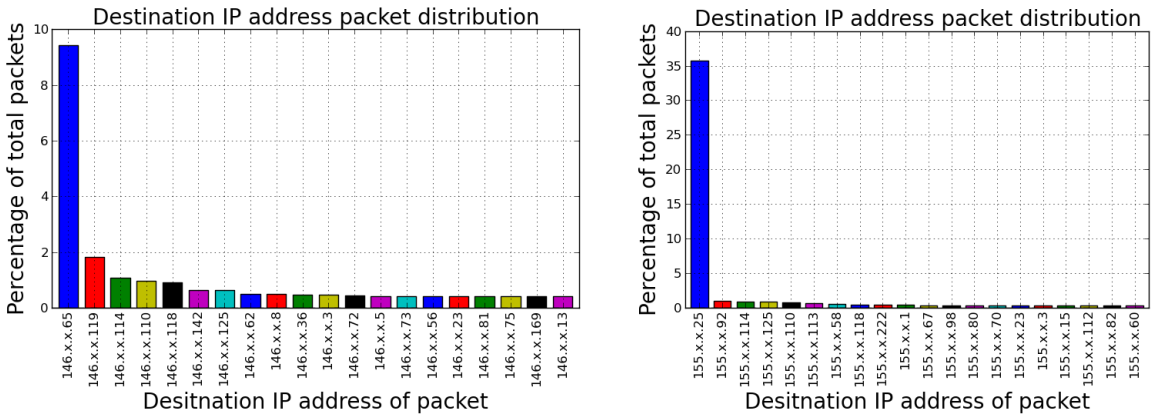
Rank	Destination IP	Number of packets received	Percentage of total packets
1	155.x.x.25	488243	35.423
2	155.x.x.30	54948	3.987
3	155.x.x.210	24029	1.743
4	155.x.x.114	8988	0.652
5	155.x.x.113	8945	0.649
6	155.x.x.102	8361	0.607
7	155.x.x.110	7882	0.572
8	155.x.x.222	7206	0.523
9	155.x.x.118	5907	0.429
10	155.x.x.14	5415	0.393

(b) 155 07/13 dataset

Table 4.1: Destination IP results for July 2013

The first thing that is noticeable is the large packet spike for destination IP 155.x.x.25, rank 1, in the 155 dataset in Table 4.1 (b). The 146 dataset shows a more evenly spread ratio of packets to destination IPs, with no destination IP receiving even two percent of the dataset packets. The 155 dataset however has one IP address that receives 35 percent of the total packets of the dataset. It has been noticed over the course of this research that if one IP address receives a noticeably larger share of the total packet count then the

IP itself has most likely been spoofed (Mirkovic and Reiher, 2004). The attack packets were then reflected by the target host and captured by the darknet IP; the same IP that was spoofed as the source address of the attack packets.



(a) Destination IP results for 146 dataset August 2013(b) Destination IP results for 146 dataset August 2013

Figure 4.1: Comparison of destination IP and TCP port activity for 146 and 155

A look at the graphical output in Figure 4.1 (b) from the following month reveals that there is still a large amount of activity on the 155.x.x.25 destination IP address. The source of this traffic will be looked at in section 4.2. A spike is also present on the 146 dataset, on IP 146.x.x.65 of Figure 4.1 (a). There were no large discrepancies between IP packet frequency in the previous month. This anomaly will also be further analysed in section 4.3.

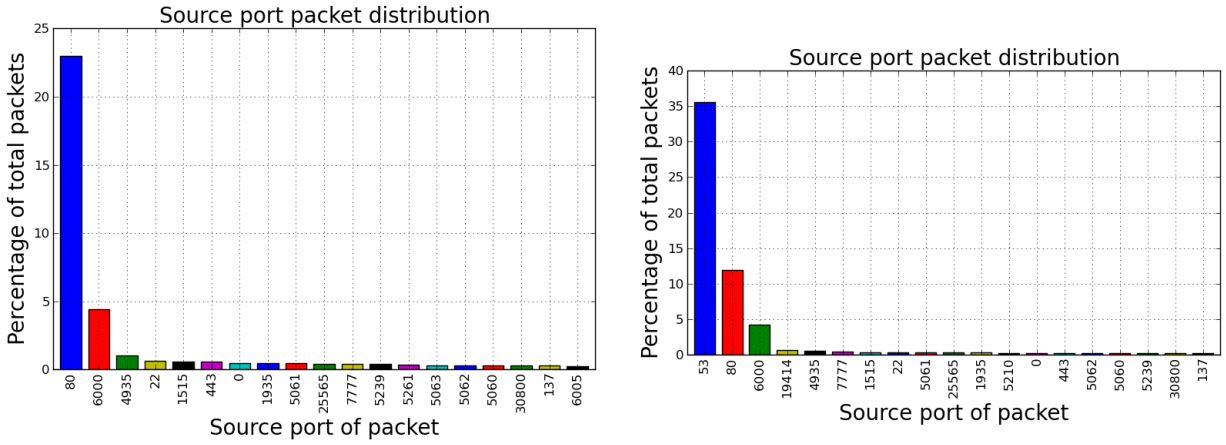
Rank	Source port	Number of packets sent	Percentage of total packets
1	80	126231	14.965
2	6000	44212	5.242
3	30800	19189	2.275
4	4935	8852	1.049
5	22	7308	0.866
6	12200	3701	0.439
7	3001	3403	0.403
8	5109	3266	0.387
9	5061	2980	0.353
10	25565	2867	0.340

(a) 146 07/13 dataset

Rank	Source port	Number of packets sent	Percentage of total packets
1	53	486264	35.279
2	80	127746	9.268
3	6000	48539	3.522
4	30800	19802	1.437
5	4445	8202	0.595
6	22	7548	0.548
7	4935	6353	0.461
8	5343	3724	0.270
9	3001	3592	0.261
10	12200	3157	0.229

(b) 155 07/13 dataset

Table 4.2: Source port results for July 2013



(a) 146 08/13 dataset

(b) 155 08/13 dataset

Figure 4.2: Source port results for August 2013

An interesting correlation is that source port 53 is responsible for 35 percent of the traffic to the 155 darknet in Table 4.2 (b), and 35 percent of the traffic is sent to a specific destination IP in Table 4.1 (a). It could then be assumed that most of the packets traveling to the IP in question were sent from port 53. The popularity of port can also be seen in Figure 4.2 (b). The distribution of source port activity seen in Figure 4.2 (a) is closer to the expected IBR results for the early months of the dataset. This also indicates that it may be an attack related to DNS amplification (Deshpande, Katsaros, Basagiannis, and Smolka, 2011), and will be explored further in section 4.2.

4.1.2 SSH activity starts December 2013

Rank	Destination port	Number of packets received	Percentage of total packets
1	22	129851	15.735
2	80	92347	11.190
3	3389	61771	7.485
4	445	38227	4.632
5	8080	33773	4.092
6	23	25653	3.108
7	443	24609	2.981
8	1433	19162	2.321
9	5900	12549	1.520
10	3128	9391	1.137

(a) 146 12/13 dataset

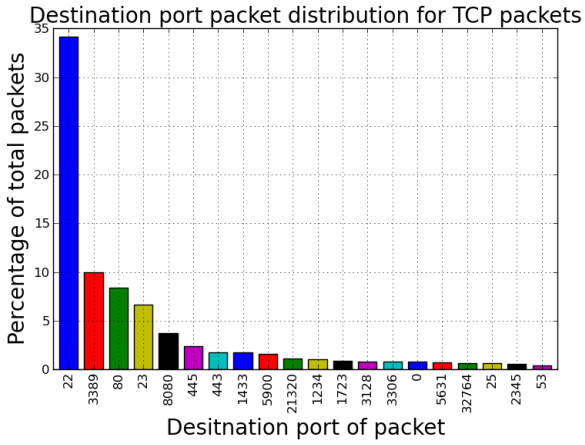
Destination port	Number of packets received	Percentage of total packets
22	99002	12.672
80	97104	12.429
3389	60621	7.759
8080	34482	4.414
1433	26519	3.394
443	25478	3.261
445	19036	2.437
5900	16537	2.117
23	13146	1.683
3128	10480	1.341

(b) 155 12/13 dataset

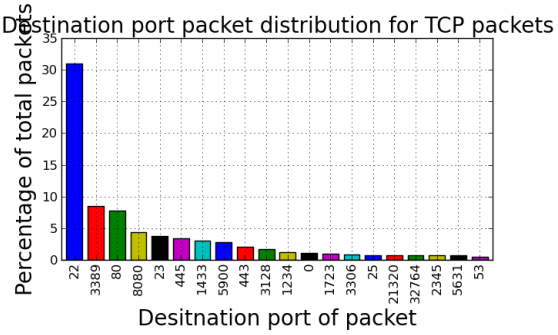
Table 4.3: TCP destination port results

December is the first month that SSH packets create noticeably larger traffic than in previous months. Tables 4.3 (a) and (b) clearly show that traffic to destination port 22 has the highest frequency for both darknets respectively. Note that the traffic on port 22 makes up over 15 percent of darknet 146 (4.3 a) and over 12 percent of the TCP traffic of the 155 darknet (4.3 b). These percentages are much lower than in the month of January 2014, as a sharp rise in packets to port 22 is recorded near the middle of the month.

Figure 3.4 (a) shows that the TCP SSH traffic recorded by the 146 darknet in January now comprises nearly 35 percent of all TCP traffic. Similarly Figure 3.4 (b) shows a rise to just over 30 percent of all TCP traffic comprising of TCP SSH traffic. The rise in SSH traffic will be further dealt with in section 4.4.



(a) 146 darknet 01/14



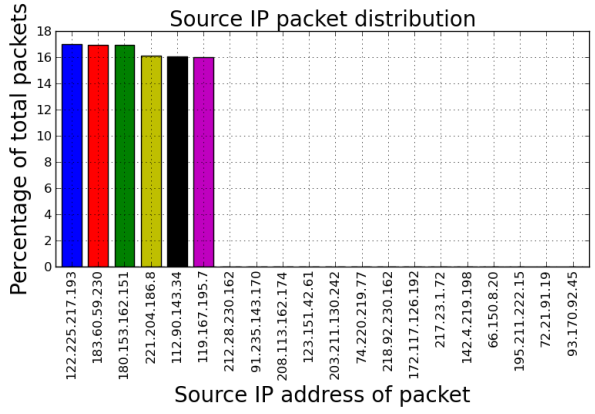
(b) 155 darknet 01/14

Figure 4.3: Destination port results for the 146 and 155 darknets for January 2014

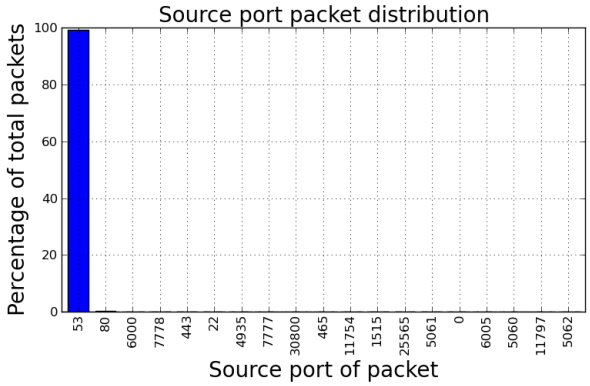
4.2 Case Study: DNS amplification attack

A breakdown of the 155.x.x.25 IP mentioned in section 4.1.1 revealed that six unique source IP addresses were responsible for almost all of the traffic collected at the aforementioned destination IP address. It can therefore be concluded that the hosts at these IP addresses are the target of the attack. It was found that the attack continued throughout the months of July and August.

Figure 4.4 shows the packet frequency from the top twenty source IPs sending packets to 155.x.x.25. It also shows a breakdown of source ports of the recorded packets. Almost all of the source port 53 traffic seen in Figure 4.4 (b) can be attributed to the six source IPs that are the dominant contributors in Figure 4.4 (a). The large presence of source port 53 traffic from the target hosts suggests that they were the victim of a DNS amplification (DDoS) attack (Kambourakis, Moschos, Geneiatakis, and Gritzalis, 2007). Further analysis was then done by isolating the traffic that each IP was responsible for. Figure 4.5 is a collection of destination port scatter plots and time-series plots that tracks the activity of one source IP over two months.

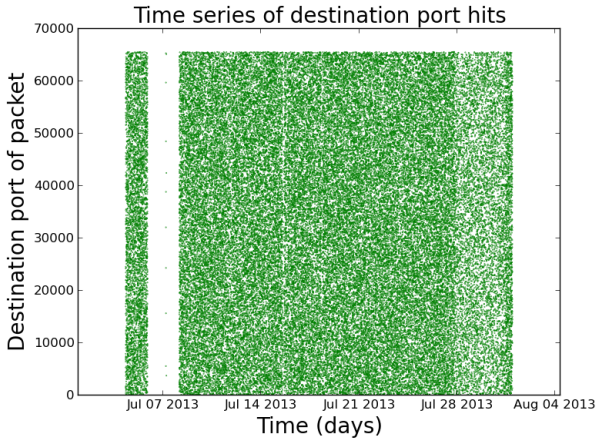


(a) Source IPs responsible for traffic

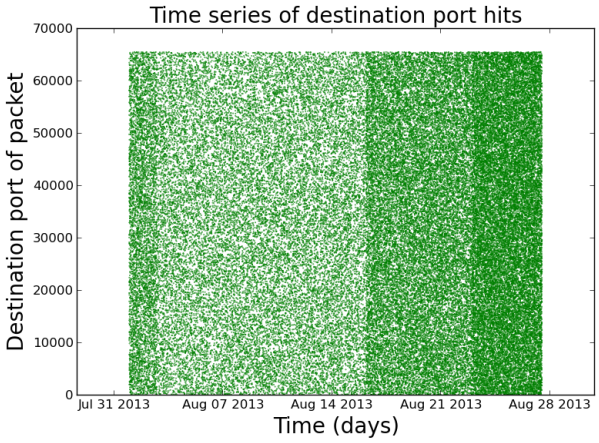


(b) Source ports of attack

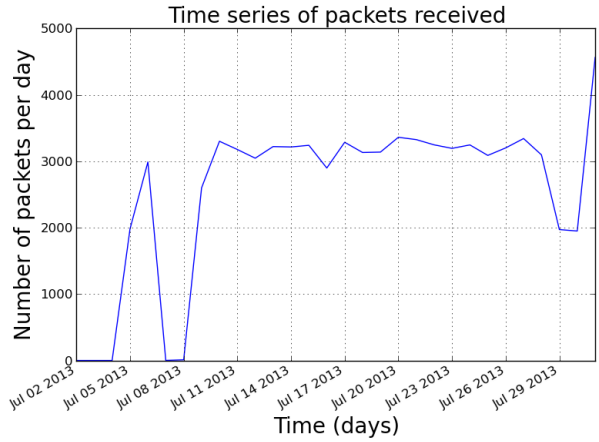
Figure 4.4: Breakdown of IP 155.x.x.25



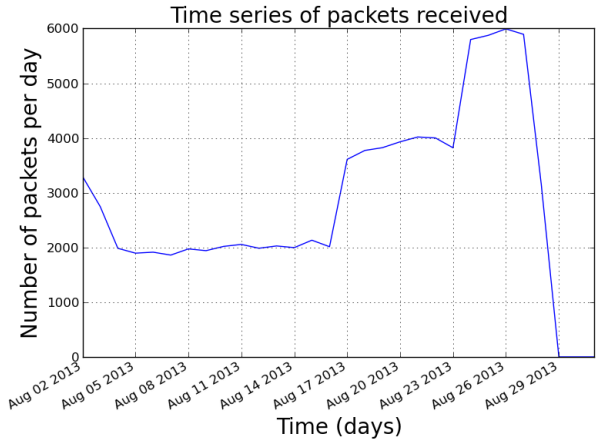
(a) Destination port of packet vs Time 07/13



(b) Destination port vs Time 08/13



(c) Time-series of packets received 07/13

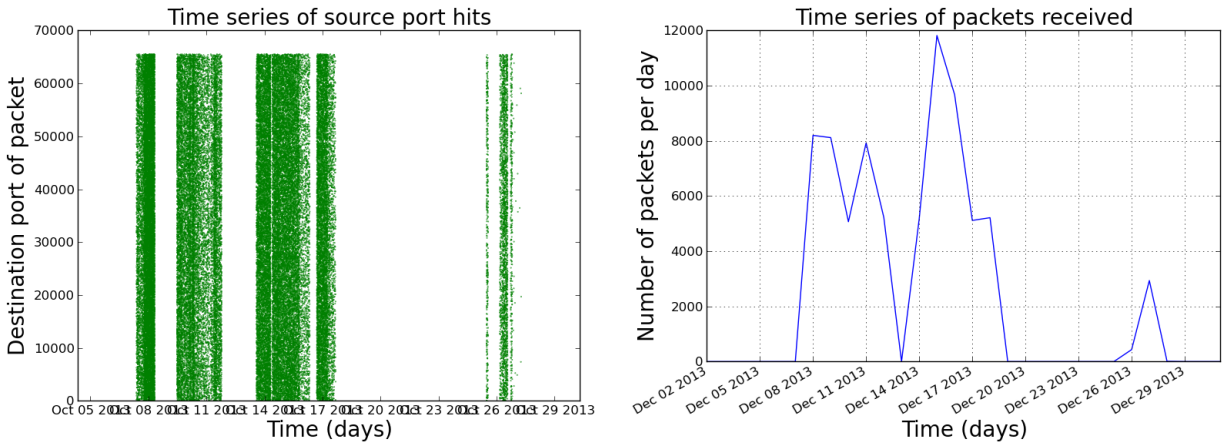


(d) Time-series of packets received 08/13

Figure 4.5: Packets from IP 122.225.217.193

Figure 4.5 (a) and 4.5 (b) are plots that represent the destination ports of the packets sent by the source IP in question. We can see that the attack was across all ports, and continued throughout the two months. DNS amplification attacks aim to consume the bandwidth of a system, which is done by flooding a target host with DNS response packets that it hasn't requested (Deshpande et al., 2011). This is most likely a reflected zombie botnet attack (Kambourakis et al., 2007).

A similar attack was recorded on the 155 darknet during October 2013. The attack was shorter in duration than the July-August attack.



(a) Source port vs Time 155 10/13

(b) Time-series 155 10/13

Figure 4.6: Recorded DNS attack October 2013

It is interesting to note that this DNS attack was much shorter in duration than the one recorded in July/August. It is also interesting to note that the banding occurs on the source ports in Figure 4.6 (a) similar to the banding in Figure 4.5 (a & b) but the duration is not as long as that noted in Figure 4.5 (c & d). Looking at Figure 4.6(b), the transmission of packets is not as consistent as the traffic recorded in the July/August period. This would suggest that this is a group with access to a different zombie botnet.

4.3 Case Study: DDoS found in 08/13

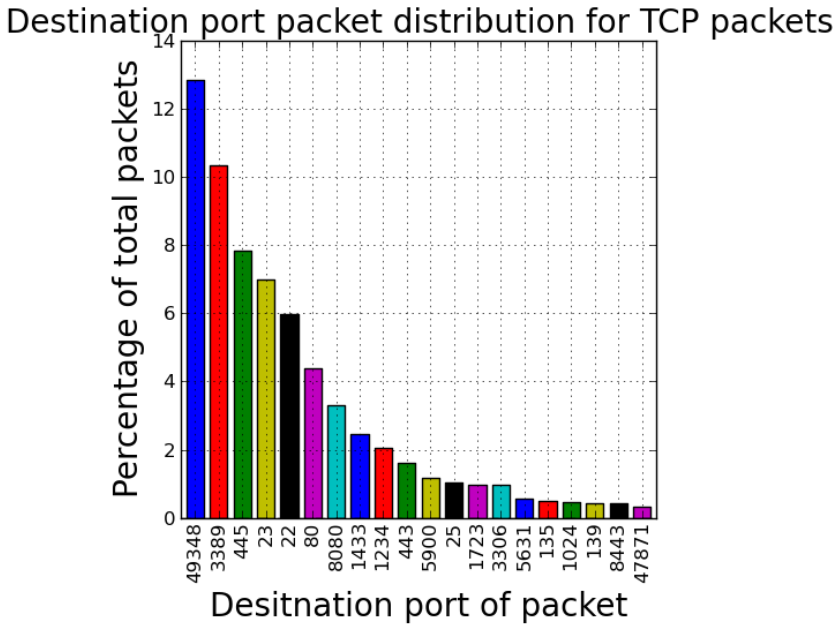


Figure 4.7: TCP destination port results for 146 darknet August 2013

Analysis run on the 146.x.x/24 dataset revealed that 49348 was the port at which the most traffic was received. This was also confirmed by the results presented in Figure 4.5, which gives a destination port breakdown of the month of August for the 146 darknet. It was the only such instance of the port appearing in the top port results and was investigated as a result. A port lookup revealed that 49348 is inside a block of dedicated ports used for Apple's Xsan Filesystem Access (Apple, 2014) with respect to Apple products. The port is simply listed as dynamic and/or unallocated otherwise. This information seemed to point to an attack that targeted Apple users, and the results were then filtered using port 49348.

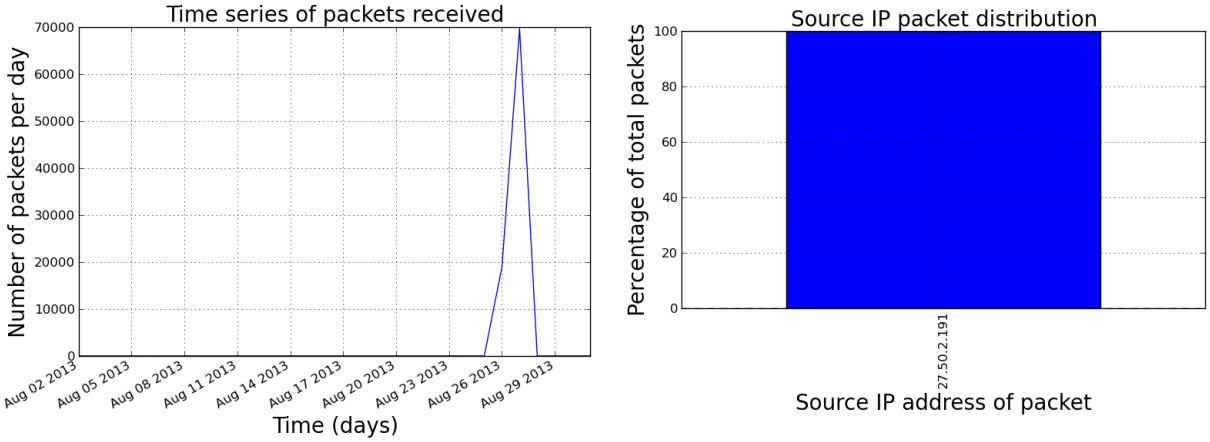


Figure 4.8: Results of filter on 49348

The results show that all of the traffic was received by a single IP address, and all of the traffic originated from the same address. It is also clear that the traffic started on the 26th of August and ended after the 27th of August, only two days of activity. The spike is also noteworthy, from 0 to 20000 on the first day of the attack and from 20000 to 70000 on the second day. These packets are not sent from the attacker, but from the receiver. The DDoS attack results in backscatter (Irwin, 2013) from the target host as it attempts to reply to the attacking packets. These packets are then received at the darknet as a result of IP spoofing (Mirkovic and Reiher, 2004), where one of the spoofed source IP addresses in the attack was 146.231.x.65. All of the packets that were sent to port 49348 were sent by 27.50.2.191, who is assumed to be the target of the DDoS. 27.50.2.191 was then used as the filter parameter for all of the other darknet datasets for the month of 08/2013 in an attempt to validate the presence of a DDoS. 196.21 (1) returned nearly identical results to the 146.231 dataset.

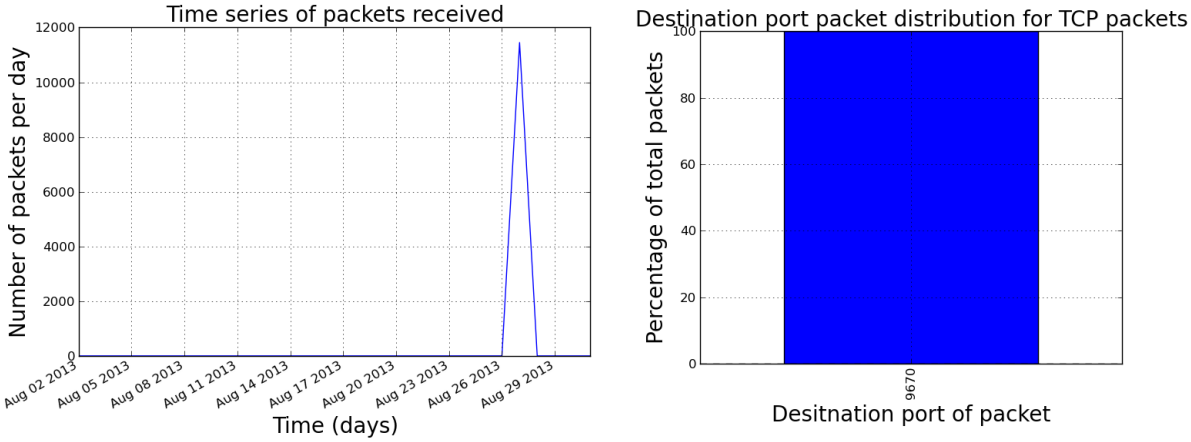


Figure 4.9: 196.21 (1) results on IP filter

The packet spike is much smaller than the backscatter captured by the 146 dataset, and occurs later as well. This could indicate that this attacker may have started after the 146.231 spoofer, or that this attacker may have been cycling through spoofed IP addresses. There is still undeniably a change in activity on the 27th of August on both darknets. The same IP is responsible for both spikes but otherwise does not feature in the dataset activity. Further DDoS activity was reported on a sub-list of the GMANE mailing list[1], where user Mike Wrigth said that he was receiving constant traffic from 27.50.2.191:80 for port 4460 of his host. This was posted 02:17 27 August 2013[2] and coincides with the captured traffic on the darknets.

4.4 Case Study: SSH/port 22 activity in the datasets

While activity on port 22 (SSH protocol) had been increasing since the middle of December 2013, there was a large, maintained activity spike in January of 2014. This is particularly interesting as the spike was seen across all five darknets. All five darknets will be examined in this case study. Special attention will be given to the month of January in particular across all of the darknets. The month of February 2014 will not be focused on as the packet captures only continue for twelve days of the month.

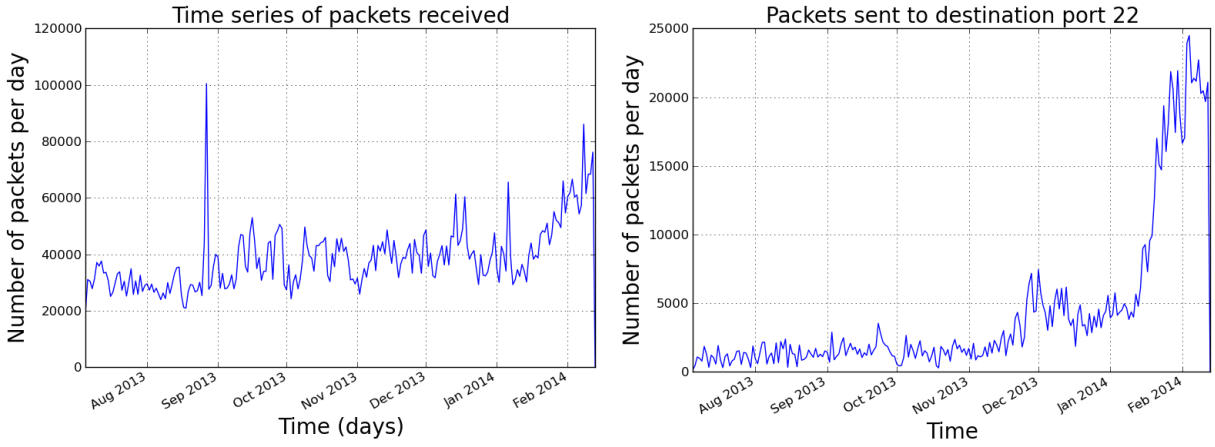


Figure 4.10: Time series of 146.231 dataset

[1] http://gmane.org/
[2] http://comments.gmane.org/gmane.comp.security.firewalls.netfilter.general/46291

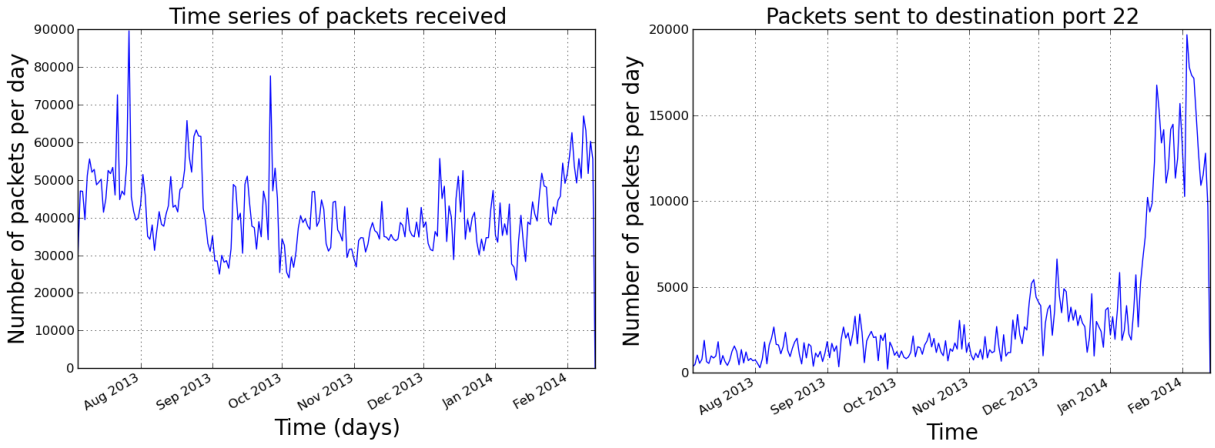


Figure 4.11: Time series of 155.232 dataset

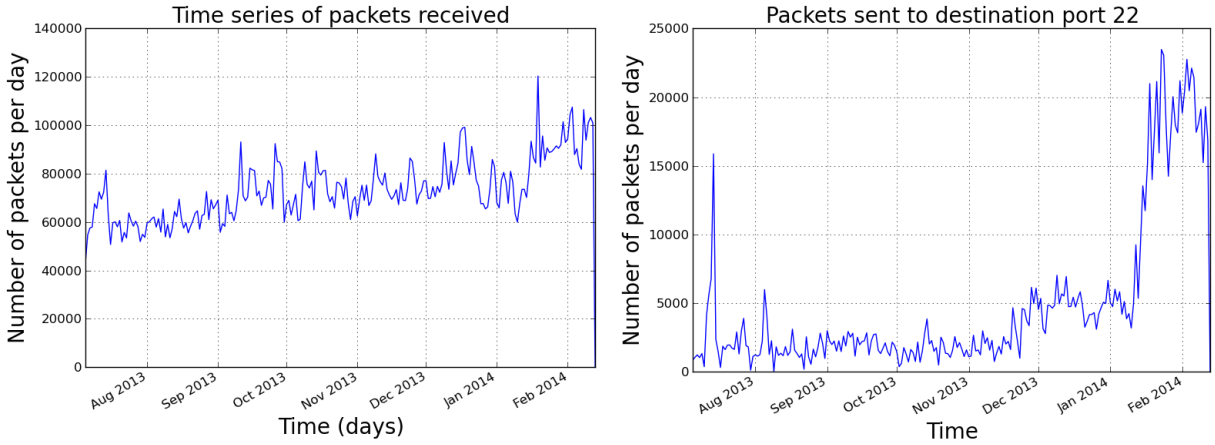


Figure 4.12: Time series of 196.21 (1) dataset

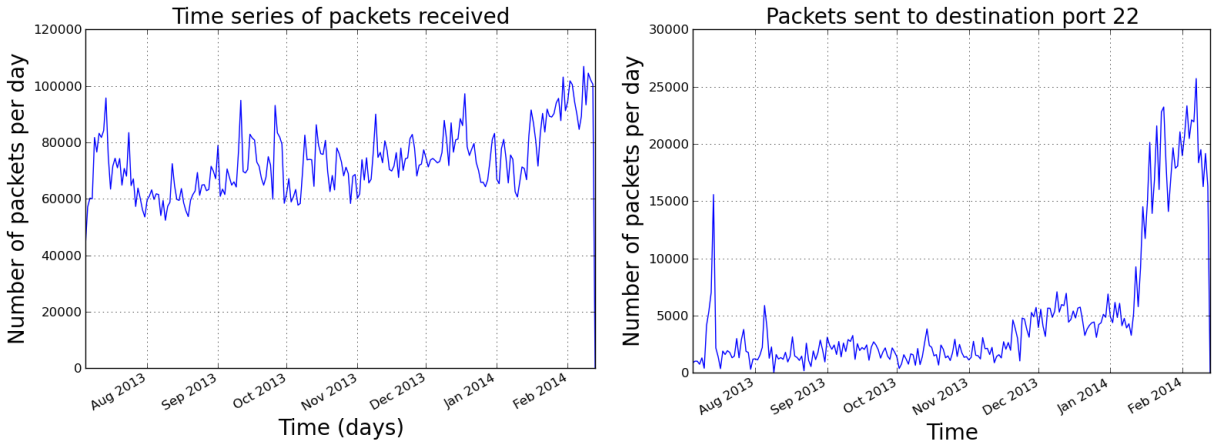


Figure 4.13: Time series of 196.21 (2) dataset

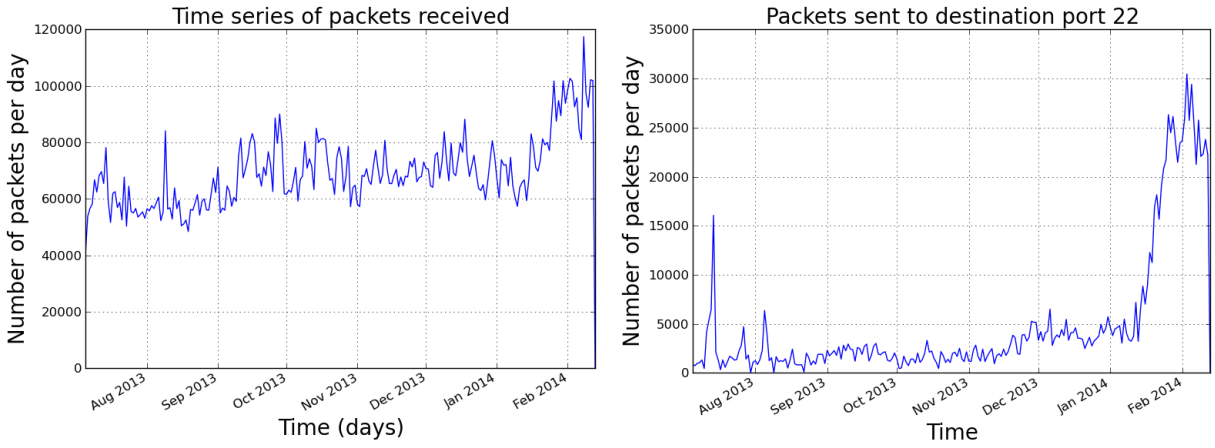


Figure 4.14: Time series of 196.21 (3) dataset

It is clear that, looking at figures 4.1-5, that there was an increase in SSH activity at roughly the same time across all the darknets. Of these five darknets, the 146 and 196.24 darknets are the most interesting cases. SSH traffic accounts for 25% of the traffic for the 146 darknet during the month of January. Of all the darknets this is the greatest percentage of port 22 traffic per total packets recorded. 196.24 is of interest because it logged the most SSH packets in a single day, across all darknets, and despite receiving less overall SSH packets, has a higher packet percentage than the other 196 datasets.

Darknet	SSH TCP packets	SSH UDP packets	Dataset Total packets	% of total packets
146	338648	5	1352396	25.041
155	266832	9	1246840	21.401
196.21 (1)	395192	8	2582831	15.301
196.21 (2)	394116	7	2508143	15.714
196.24	389682	4	2373478	16.418

Table 4.4: Packet frequency across port 22 for the month of January 2014

Note that Table 4.13 includes packets that were sent to or received at port 22. This is also the only table in Chapter 4 that was not created by the system, and is instead an amalgamation of two tables.

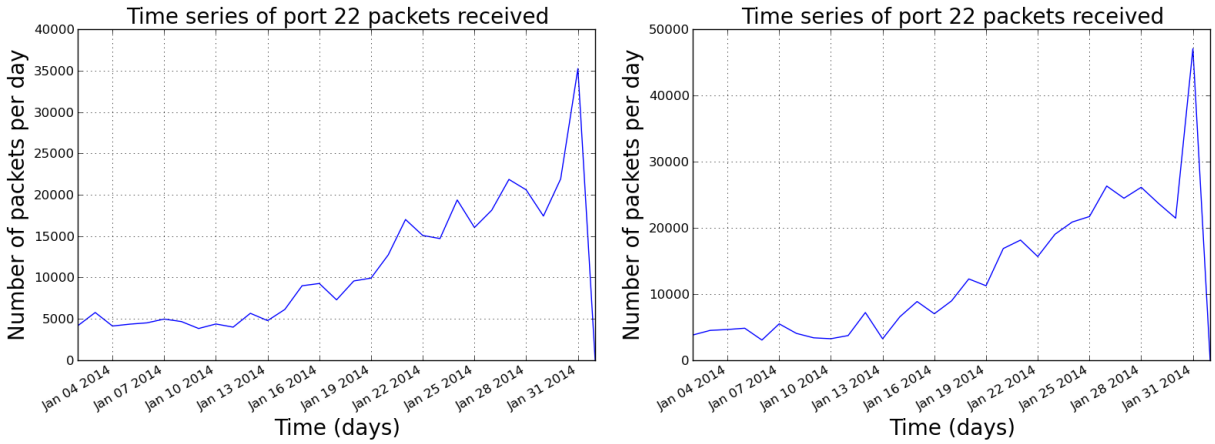


Figure 4.15: Time series packets received at port 22 for 146 and 196.24 for January 2014

Looking at the time series for these two darknets, it is clear that SSH traffic started climbing around the 15th of January. It is also interesting that the 146 dataset saw a sudden drop on the 28th of January, whereas the 196.24 dataset saw a drop on the 30th. Both datasets then show a similar peak that registers far above the 5000 packets per day average that lasts until the 13th of the month.

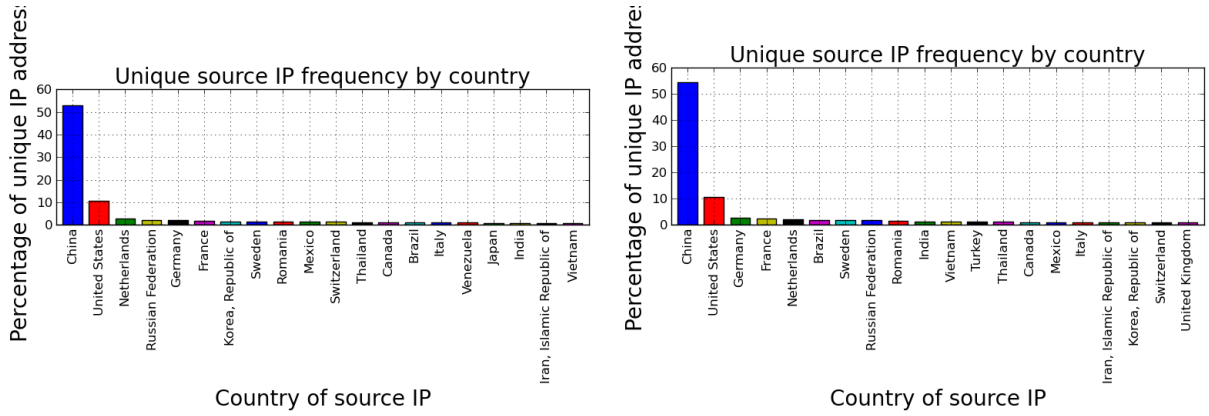


Figure 4.16: Unique Source IPs related to SSH traffic for 146 and 196.42

Of the IP addresses responsible for sending SSH packets, more than 50% belong to IPs registered by China. An additional 10% is generated from the United States. It is interesting to note that the distribution of unique IPs responsible for the traffic would be so similar in different darknets. IP spoofing is a reality however, and these results should not be considered entirely accurate. The general trend can be assumed to be accurate even

with the possibility of IP spoofing, as IP spoofing is most commonly observed in DDoS backscatter (Irwin, 2013) (Mirkovic and Reiher, 2004).

A study by Allman et al. (2007) looked at scanning data over the period 1994 - 2006. Over this period SSH scanning traffic was minimal and unnoteworthy (Allman et al., 2007). 2007 is a notable year for SSH traffic as it is the first year that SSH packets made up a significant percentage of Internet Background Radiation (Wustrow et al., 2010). It is also interesting that traffic to port 23 increased in 2007 also(Wustrow et al., 2010). SSH became a major scanning packet contributor in 2014 (Durumeric, Bailey, and Halderman, 2014). It was also noted that SSH was the most targetted port in large scans, but only the seventh most targeted in smaller scans (Durumeric et al., 2014). SSH traffic, along with RDP, ICMP and MYSQL traffic showed similar patterns with regards to their country of origin. All of these protocols had the largest number of scanning packets come from China, with the U.S.A being the second highest contributor (Durumeric et al., 2014).

Chapter 5

Conclusion

This last chapter will present a summary of the paper itself. It will reiterate some of the key points of each chapter. There will also be some concluding remarks about the project itself. Finally a consideration will be given to future possible development of the project itself.

5.1 Summary of the research

The main focus of the project, as outlined in chapter one, was the creation of a system that had analysis and reporting capabilities with respect to packet capture files created by network telescope sensors. The system was also expected to create a standardised form of reporting output. Chapter two looked at three key areas of literature. The first was the fundamental work from which the system would get its context. The second key area dealt with the possible analysis routes of packet capture data. The third and final area handled possible reporting styles and responses to generated analytical data. The integration of findings in the literature becomes apparent in the third chapter, which covers the design and implementation of the system, as well as an evaluation of the system itself. This is then followed by chapter four, that uses output from the system to perform analysis. The first part of the chapter focuses on a broader analysis and comparison of datasets while the second focuses on isolating interesting packet behaviour. This paper was written with the intent to document the prototype system while also showcasing the functionality and usefulness of the system to researchers in the Information Security field.

5.2 Concluding remarks

Network traffic analysis is falls under the banner of Information Security in the field of Computer Science. Internet background radiation gives researchers a better understanding of how both active and passive malicious traffic interacts with the larger Internet. An understanding of the network traffic captured by the datasets is fundamental in the pursuit of creating a reporting infrastructure. This project explores the nature of unsolicited network traffic through analysis, which enables the creation of reports as a result of the tabular and graphical output. The system can create reports of the same style and format from a supplied pcap file. Examples of system outputs can be seen in the appendix, which also holds details of the Git repository that hosts the code created during the implementation of the system. It should be noted here that the ancillary report produces all of the graphs and tables present in the first report. These have been left out for the sake of brevity.

5.3 Future work

There are different avenues that can be explored in an attempt to improve the functionality or increase the uses of the system. Some ideas towards a more complete reporting system are listed below.

- Development of system to allow for near real-time packet sniffing analysis and reporting that updates regularly to keep information as well as summarised results current and relevant.
- Implement a data carving functionality, whereby the system can separate parts of the larger pcap file for analysis without the need to process the pcap to another format.
- Increase the scope of the reporting output to focus on a broader collection of data categories.
- Introduce the ability to create textual output that serves as an interpretation of the analysis and output produced by the report.
- Create a component within the system that automatically compares reports from different datasets after generation. This component would then highlight areas of

both reports that have a high and low similarity, which is usually an indicator of interesting packet traffic.

References

Mark Allman, Vern Paxson, and Jeff Terrell. A brief history of scanning. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pages 77–82. ACM, 2007.

Apple. Well known tcp and udp ports used by apple software products, May 2014. URL `support.apple.com/kb/HT6175?viewlocale=en_US`.

Michael Bailey, Evan Cooke, Tim Battles, and Danny McPherson. Tracking global threats with the internet motion sensor. In *32nd Meeting of the North American Network Operators Group*, 2004.

Michael Bailey, Evan Cooke, Farnam Jahanian, Jose Nazario, and David Watson. The internet motion sensor: A distributed blackhole monitoring system. In *In Proceedings of Network and Distributed System Security Symposium (NDSS)*, pages 167–179, 2005a.

Michael Bailey, Evan Cooke, Farnam Jahanian, Niels Provos, Karl Rosaen, and David Watson. Data reduction for the scalable automated analysis of distributed darknet traffic. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 21–21. USENIX Association, 2005b.

Michael Bailey, Evan Cooke, Farnam Jahanian, Andrew Myrick, and Sushant Sinha. Practical darknet measurement. In *Information Sciences and Systems, 2006 40th Annual Conference on*, pages 1496–1501. IEEE, 2006.

Sebastián Bortnik. Conficker by the numbers, 2010. URL `http://www.eset.com/us/resource/papers/white-papers/`. Accessed 16 May 2012.

CAIDA. The USCD network telescope. Online, August 2012. URL `http://www.caida.org/projects/network_telescope/`. Accessed Wed 28 May 2014.

Evan Cooke, Michael Bailey, David Watson, Farnam Jahanian, and Jose Nazario. The internet motion sensor: A distributed global scoped internet threat monitoring system.

Technical report, Technical Report CSE-TR-491-04, University of Michigan, Electrical Engineering and Computer Science, 2004.

Bradley Cowie and Barry Irwin. A baseline numeric analysis of network telescope data for network incident discovery. In *Southern African Telecommunications Networks and Applications Conference (SATNAC)*, 2010.

Tushar Deshpande, Panagiotis Katsaros, Stylianos Basagiannis, and Scott A Smolka. Formal analysis of the dns bandwidth amplification attack and its countermeasures using probabilistic model checking. In *High-Assurance Systems Engineering (HASE), 2011 IEEE 13th International Symposium on*, pages 360–367. IEEE, 2011.

Navneet Kaur Dhillon and Mrs Uzma Ansari. Enterprise network traffic monitoring, analysis, and reporting using winpcap tool a packet capturing api. *International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE)*, 1(6):pp–19, 2012.

Haitao Du and Shanchiech Jay Yang. *Discovering Collaborative Cyber Attack Patterns Using Social Network Analysis*. Social Computing, Behavioral-Cultural Modeling and Prediction. Springer Berlin Heidelberg, March 2011.

Hongbo Du. *Data Mining Techniques and Applications/ An Introduction*. Cengage Learning EMEA, 2010.

Zakir Durumeric, Michael Bailey, and J Alex Halderman. An internet-wide view of internet-wide scanning. In *USENIX Security Symposium*, 2014.

Jérôme Francois, Olivier Festor, et al. Activity monitoring for large honeynets and network telescopes. *International Journal On Advances in Systems and Measurements*, 1(1): 1–13, 2009.

Simson L. Garfinkel. Digital forensics research: The next 10 years. *Digital Investigation*, 7, Supplement(0):S64 – S73, 2010. ISSN 1742-2876. doi: http://dx.doi.org/10.1016/j.diin.2010.05.009. URL `http://www.sciencedirect.com/science/article/pii/S1742287610000368`. The Proceedings of the Tenth Annual DFRWS Conference.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutermann, and H. Witten, I. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1): 10–18, June 2009.

Uli Harder, Matt W. Johnson, Jeremy T. Bradley, and William J. Knottenbelt. Observing internet worm and virus attacks with a small network telescope. *Electronic Notes in Theoretical Computer Scinece*, 151(3):47 – 59, May 2006.

Warren Harrop and Grenville Armitage. Greynets: a definition and evaluation of sparsely populated darknets. In *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 171–172. ACM, 2005.

Warren Harrop and Grenville Armitage. Real-time collaborative network monitoring and control using 3d game engines for representation and interaction. In *Proceedings of the 3rd international workshop on Visualization for computer security*, pages 31–40. ACM, 2006.

Barry Irwin. *A FRAMEWORK FOR THE APPLICATION OF NETWORK TELESCOPE SENSORS IN A GLOBAL IP NETWORK*. Phd thesis, Rhodes University, January 2011.

Barry Irwin. Network telescope metrics. In *Southern African Telecommunications and Applications Conference (SATNAC)*, 2012a.

Barry Irwin. A network telescope perspective of the conficker outbreak. In *Information Security for South Africa (ISSA), 2012*, pages 1–8. IEEE, 2012b.

Barry Irwin. A baseline study of potentially malicious activity across five network telescopes. In *5th International Conference on Cyber Conflict*, 2013.

Barry Irwin and R Barnett. An analysis of logical network distance on observed packet counts for network telescope data. In *Southern African Telecommunications Networks and Applications Conference (SATNAC)*, volume 31, 2009.

Barry Irwin and Nick Pilkington. High level internet scale traffic visualization using hilbert curve mapping. In *VizSEC 2007*, pages 147–158. Springer, 2008.

Georgios Kambourakis, Tassos Moschos, Dimitris Geneiatakis, and Stefanos Gritzalis. A fair solution to dns amplification attacks. In *Digital Forensics and Incident Analysis, 2007. WDFIA 2007. Second International Workshop on*, pages 38–47. IEEE, 2007.

Seong Soo Kim, AL Narasimha Reddy, and Marina Vannucci. Detecting traffic anomalies through aggregate analysis of packet header data. In *NETWORKING 2004. Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, pages 1047–1059. Springer, 2004.

Helmut Kopka and Patrick W Daly. *A guide to LATEX*. Harlow, England, 1995.

Abhishek Kumar, Vern Paxson, and Nicholas Weaver. Exploiting underlying structure for detailed reconstruction of an internet-scale event. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, pages 33–33. USENIX Association, 2005.

James F. Kurose and Keith W. Ross. *Computer Networking A Top-Down Approach*. Addison Wesley, 2010.

Ying-Dar Lin, Chun-Nan Lu, Yuan-Cheng Lai, Wei-Hao Peng, and Po-Ching Lin. Application classification using packet size distribution and port association. *Journal of Network and Computer Applications*, 32(5):1023–1030, 2009.

Wes McKinney. pandas: a foundational python library for data analysis and statistics. In *PyHPC 2011: Python for High Performance and Scientific Computing*, 2011.

Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.

Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, and Chris Rowley. *The LATEX companion*. Addison-Wesley Professional, 2004.

David Moore, Colleen Shannon, Geoffrey M Voelker, and Stefan Savage. Network Telescopes: Technical Report. Technical report, Cooperative Association for Internet Data Analysis (CAIDA), Jul 2004.

David Moore, Colleen Shannon, Douglas J Brown, Geoffrey M Voelker, and Stefan Savage. Inferring internet denial-of-service activity. *ACM Transactions on Computer Systems (TOCS)*, 24(2):115–139, 2006.

Chris Muelder, Kwan-Liu Ma, and Tony Bartoletti. A visualization methodology for characterization of network scans. In *Visualization for Computer Security, 2005.(VizSEC 05). IEEE Workshop on*, pages 29–38. IEEE, 2005.

Thizwilondi Moses Nkhumeleni. CORRELATION AND COMPARATIVE ANALYSIS OF TRAFFIC ACROSS FIVE NETWORK TELESCOPES. Masters thesis, Rhodes University, April 2014.

TJ O'Connor. *Violent Python A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers*. Elsevier, 2013.

Ruoming Pang, Vinod Yegneswaran, Paul Barford, Vern Paxson, and Larry Peterson. Characteristics of internet background radiation. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 27–40. ACM, 2004.

Lutz Prechelt. An empirical comparison of c, c++, java, perl, python, rexx and tcl. *IEEE Computer*, 33(10):23–29, 2000.

Colleen Shannon and David Moore. The spread of the witty worm. *IEEE Security and Privacy*, 2(4):46–50, July 2004. ISSN 1540-7993. doi: 10.1109/MSP.2004.59. URL `http://dx.doi.org/10.1109/MSP.2004.59`. Accessed 25 May 2014.

Seungwon Shin and Guofei Gu. Conficker and beyond: a large-scale empirical study. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 151–160. ACM, 2010.

Eugene H Spafford. The internet worm program: An analysis. *ACM SIGCOMM Computer Communication Review*, 19(1):17–57, 1989.

Stuart Staniford, Vern Paxson, Nicholas Weaver, et al. How to own the internet in your spare time. In *USENIX Security Symposium*, pages 149–167, 2002.

Jean-Pierre van Riel and Barry Irwin. Identifying and investigating intrusive scanning patterns by visualizing network telescope traffic in a 3-d scatter-plot. In *ISSA*, pages 1–12, 2006a.

Jean-Pierre van Riel and Barry Irwin. 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, New York, NY, USA, 2006b. ACM. ISBN 1-59593-288-7. doi: 10.1145/1108590.1108604. URL `http://doi.acm.org/10.1145/1108590.1108604`.

Eric Wustrow, Manish Karir, Michael Bailey, Farnam Jahanian, and Geoff Huston. Internet background radiation revisited. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 62–74. ACM, 2010.

Guang Yao, Jun Bi, and Zijian Zhou. Passive ip traceback: capturing the origin of anonymous traffic through network telescopes. In *ACM SIGCOMM Computer Communication Review*, volume 40, pages 413–414. ACM, 2010.

Tanja Zseby, Alistair King, Nevil Brownlee, and KC Claffy. The day after patch tuesday: Effects observable in ip darkspace traffic. In Matthew Roughan and

Rocky Chang, editors, *Passive and Active Measurement*, volume 7799 of *Lecture Notes in Computer Science*, pages 273–275. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-36515-7. doi: 10.1007/978-3-642-36516-4_32. URL http://dx.doi.org/10.1007/978-3-642-36516-4_32.

Tanja Zseby, Alistair King, Marina Fomenkov, and KC Claffy. Analysis of unidirectional ip traffic to darkspace with an educational data kit, 2014. URL http://www.caida.org/publications/papers/2014/analysis_unidirectional_ip_traffic/.

Appendix A

Additional materials

A.1 Example reporting output

Report on activity for network telescope 155.x.x/24

For the period 04/07/2013 - 01/02/2014

Analysis has been performed on datasets extracted from the packet catpure to produce the graphical and tabular output of the report. There has however been no interpretation of the results generated by the report. A latex document of the report has also been created however that can be altered to include an interpretation of resutls.

Contents

1 07/2013

1.1 Destination IP

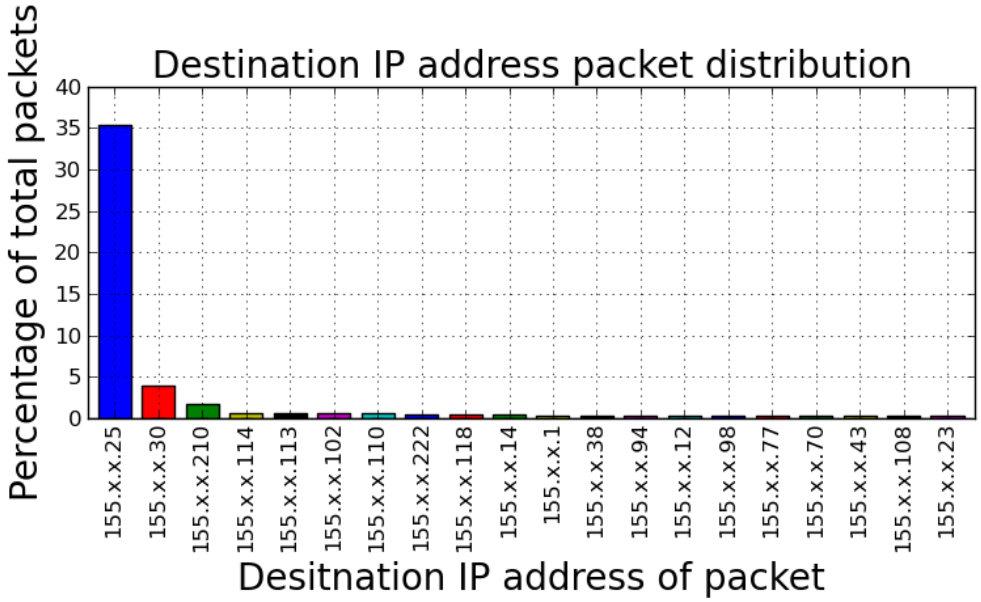


Figure 1: Destination IP packet count

Table 1: Destination IP packets

Destination IP	Number of packets recorded	Percentage of total packets
155.x.x.25	488243	35.42
155.x.x.30	54948	3.986
155.x.x.210	24029	1.743
155.x.x.114	8988	0.652
155.x.x.113	8945	0.648
155.x.x.102	8361	0.606
155.x.x.110	7882	0.571
155.x.x.222	7206	0.522
155.x.x.118	5907	0.428
155.x.x.14	5415	0.392
155.x.x.1	4738	0.343
155.x.x.38	4675	0.339
155.x.x.94	4642	0.336
155.x.x.12	4486	0.325
155.x.x.98	4375	0.317
155.x.x.77	4340	0.314
155.x.x.70	4252	0.308
155.x.x.43	4232	0.307
155.x.x.108	4159	0.301
155.x.x.23	4119	0.298
Total:	663942	39

Number of unique hits: 1378320

1.2 Destination Port TCP

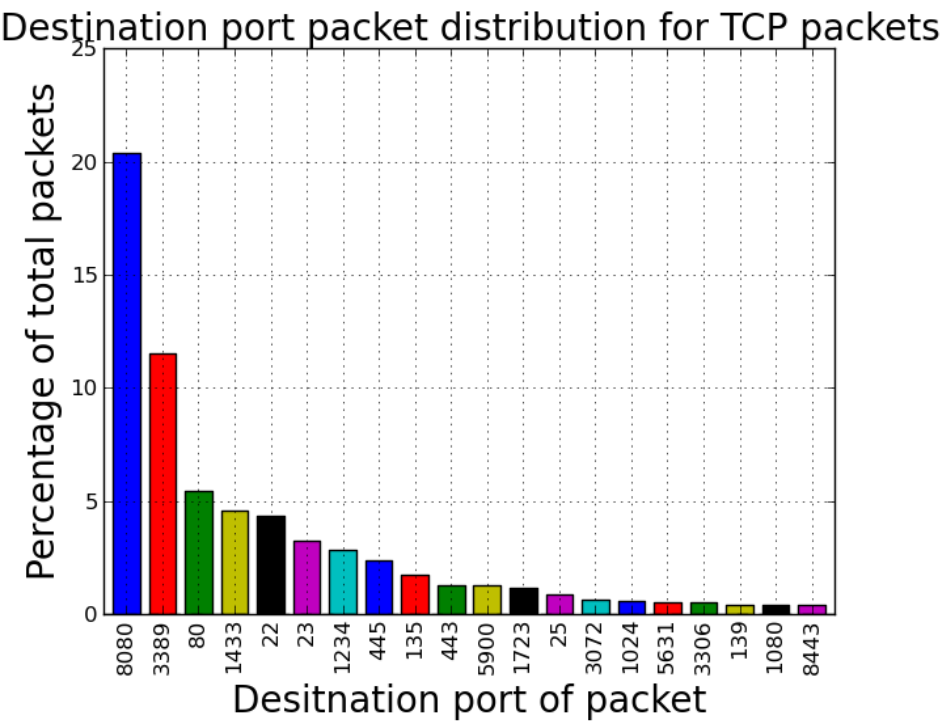


Figure 2: Destination IP packet count

Table 2: Destination Port TCP packets

Destination Port TCP	Number of packets recorded	Percentage of total packets
8080	117397	20.36
3389	66330	11.50
80	31342	5.437
1433	26531	4.602
22	25067	4.348
23	18785	3.258
1234	16309	2.829
445	13848	2.402
135	9947	1.725
443	7341	1.273
5900	7246	1.257
1723	6611	1.146
25	4991	0.865
3072	3664	0.635
1024	3524	0.611
5631	3210	0.556
3306	2985	0.517
139	2403	0.416
1080	2319	0.402
8443	2283	0.396
Total:	372133	55

Number of unique hits: 576406

1.3 Destination Port UDP

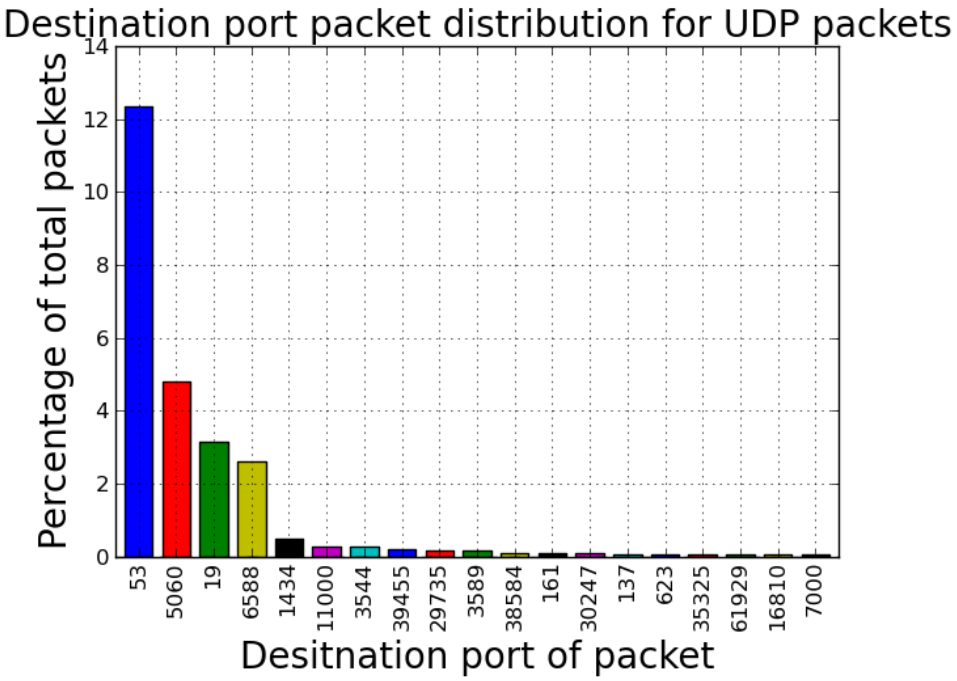


Figure 3: Destination IP packet count

Table 3: Destination Port UDP packets

Destination Port UDP	Number of packets recorded	Percentage of total packets
53	98965	12.34
5060	38619	4.815
19	25168	3.138
6588	21066	2.626
1434	4147	0.517
11000	2249	0.280
3544	2144	0.267
39455	1821	0.227
29735	1472	0.183
3589	1339	0.166
38584	948	0.118
161	836	0.104
30247	709	0.088
137	581	0.072
623	515	0.064
35325	458	0.057
61929	452	0.056
16810	450	0.056
7000	444	0.055
Total:	202383	21

Number of unique hits: 801914

1.4 Source IP

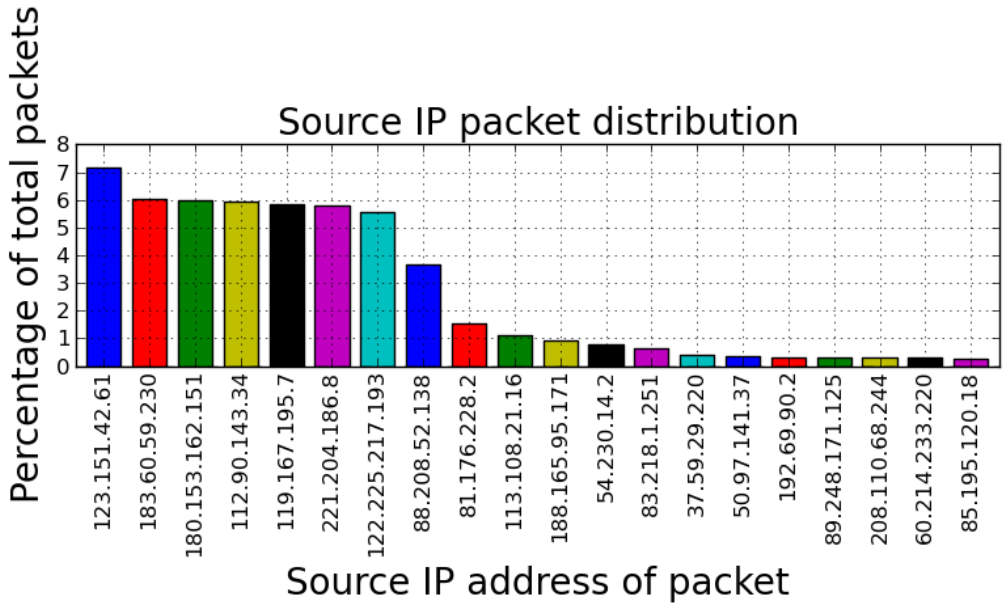


Figure 4: Destination IP packet count

Table 4: Source IP packets

Source IP	Number of packets recorded	Percentage of total packets
123.151.42.61	98547	7.149
183.60.59.230	83032	6.024
180.153.162.151	82777	6.005
112.90.143.34	81568	5.917
119.167.195.7	80195	5.818
221.204.186.8	80056	5.808
122.225.217.193	76843	5.575
88.208.52.138	50611	3.671
81.176.228.2	21057	1.527
113.108.21.16	15575	1.129
188.165.95.171	12737	0.924
54.230.14.2	10953	0.794
83.218.1.251	9126	0.662
37.59.29.220	5430	0.393
50.97.141.37	4837	0.350
192.69.90.2	4234	0.307
89.248.171.125	4215	0.305
208.110.68.244	4096	0.297
60.214.233.220	3912	0.283
85.195.120.18	3723	0.270
Total:	733524	44

Number of unique hits: 1378320

1.5 Source Port

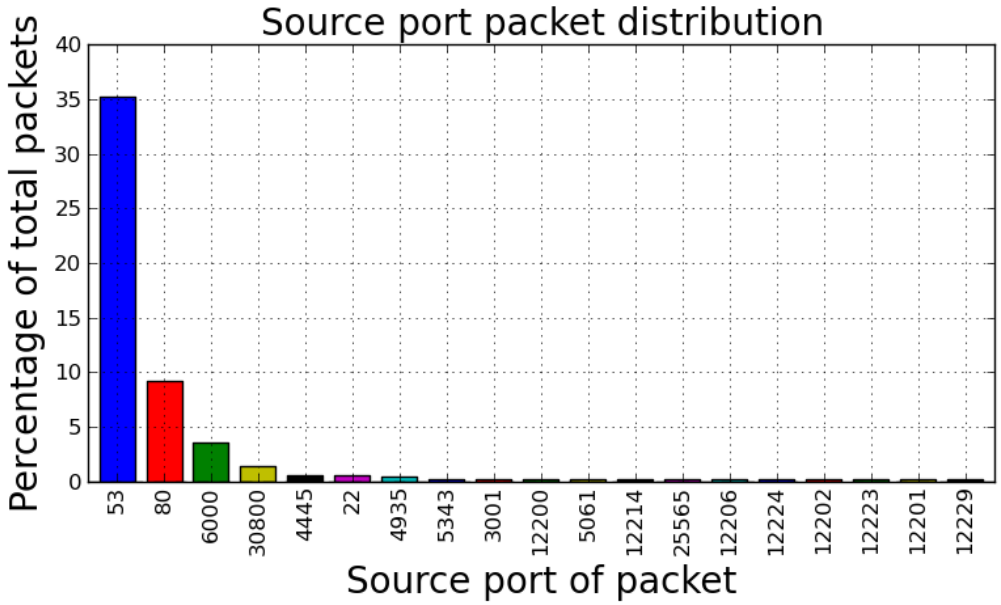


Figure 5: Destination IP packet count

Table 5: Source Port packets

Source Port	Number of packets recorded	Percentage of total packets
53	486264	35.27
80	127746	9.268
6000	48539	3.521
30800	19802	1.436
4445	8202	0.595
22	7548	0.547
4935	6353	0.460
5343	3724	0.270
3001	3592	0.260
12200	3157	0.229
5061	3064	0.222
12214	2913	0.211
25565	2892	0.209
12206	2859	0.207
12224	2836	0.205
12202	2779	0.201
12223	2778	0.201
12201	2776	0.201
12229	2775	0.201
Total:	740599	48

Number of unique hits: 1378320

1.6 Geolocation results of source IPs

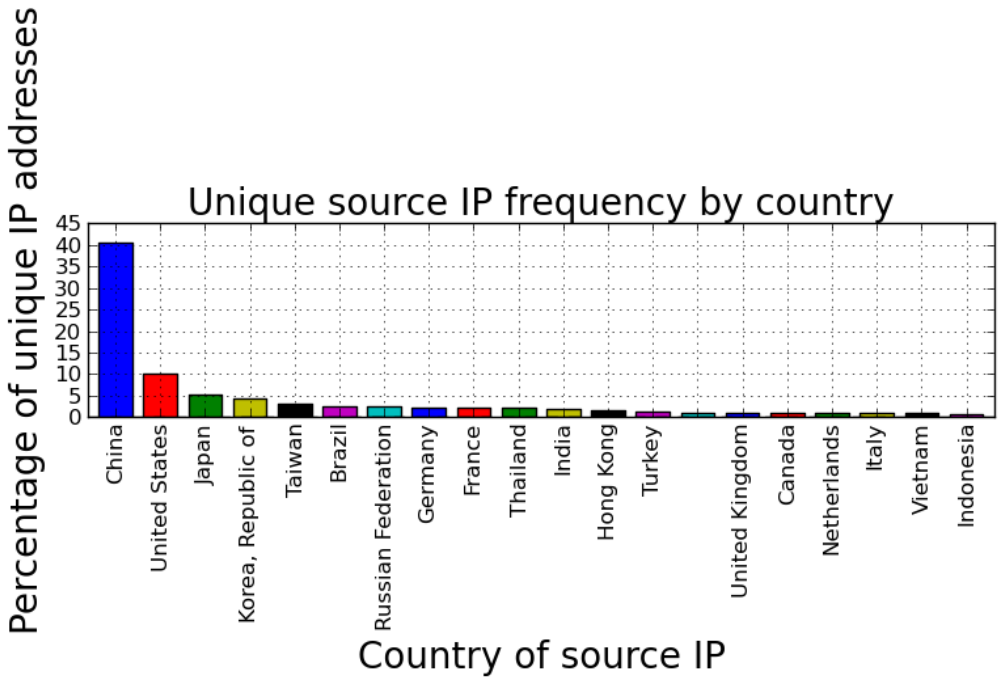


Figure 6: Destination IP packet count

Table 6: Geolocation results of source IPs packets

Geolocation results of source IPs	Number of packets recorded	Percentage of total packets
China	21347	40.48
United States	5368	10.18
Japan	2822	5.352
Korea, Republic of	2212	4.195
Taiwan	1646	3.121
Brazil	1352	2.564
Russian Federation	1303	2.471
Germany	1224	2.321
France	1219	2.311
Thailand	1181	2.239
India	935	1.773
Hong Kong	877	1.663
Turkey	726	1.376
	584	1.107
United Kingdom	550	1.043
Canada	534	1.012
Netherlands	523	0.991
Italy	496	0.940
Vietnam	483	0.916
Indonesia	410	0.777
Total:	45792	78

Number of unique hits: 52726

9 Entire Dataset

9.1 Destination IP

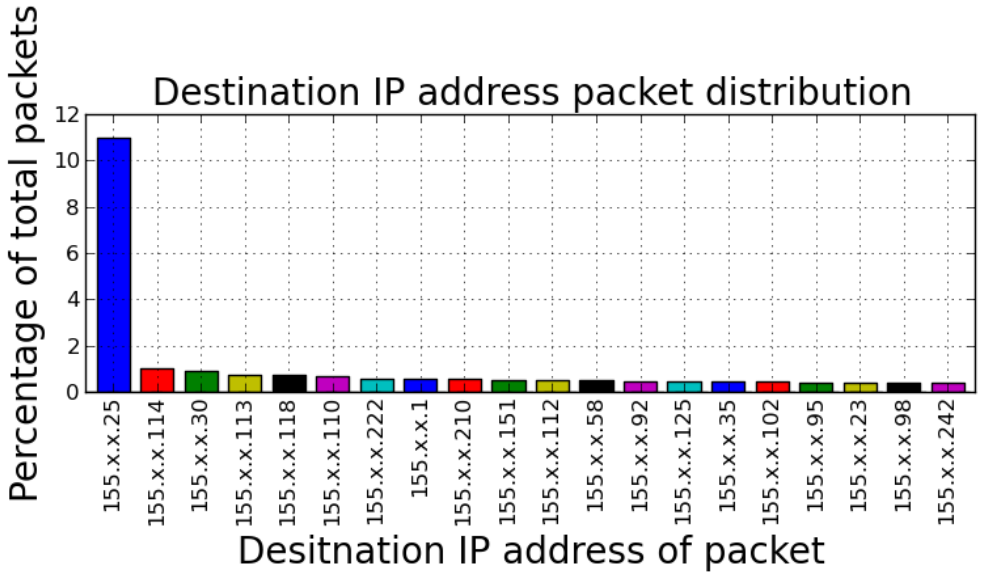


Figure 49: Destination IP packet count

Destination IP	Number of packets recorded	Percentage of total packets
155.x.x.25	1017157	10.98
155.x.x.114	94921	1.025
155.x.x.30	84967	0.917
155.x.x.113	69459	0.750
155.x.x.118	68582	0.740
155.x.x.110	63217	0.682
155.x.x.222	55263	0.597
155.x.x.1	54411	0.587
155.x.x.210	51836	0.559
155.x.x.151	49822	0.538
155.x.x.112	47758	0.515
155.x.x.58	44991	0.486
155.x.x.92	44124	0.476
155.x.x.125	41280	0.445
155.x.x.35	40729	0.439
155.x.x.102	39835	0.430
155.x.x.95	39074	0.422
155.x.x.23	37025	0.399
155.x.x.98	36869	0.398
155.x.x.242	36623	0.395
Total:	2017943	11
Number of unique hits:	9256741	

Table 49: Destination IP packets

9.2 Destination Port TCP

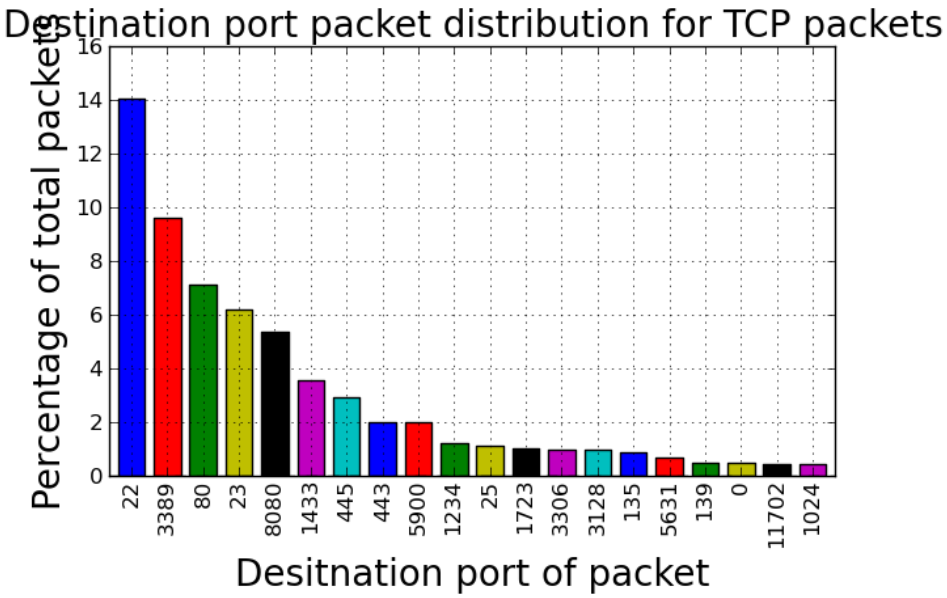


Figure 50: Destination IP packet count

Destination Port TCP	Number of packets recorded	Percentage of total packets
22	753999	14.08
3389	515411	9.624
80	381739	7.128
23	332224	6.203
8080	286188	5.344
1433	189615	3.540
445	155197	2.898
443	107228	2.002
5900	106041	1.980
1234	63541	1.186
25	59295	1.107
1723	54545	1.018
3306	50963	0.951
3128	50952	0.951
135	45709	0.853
5631	34756	0.649
139	26432	0.493
0	25075	0.468
11702	22567	0.421
1024	21546	0.402
Total:	3283023	52
Number of unique hits:	5355045	

Table 50: Destination Port TCP packets

9.3 Destination Port UDP

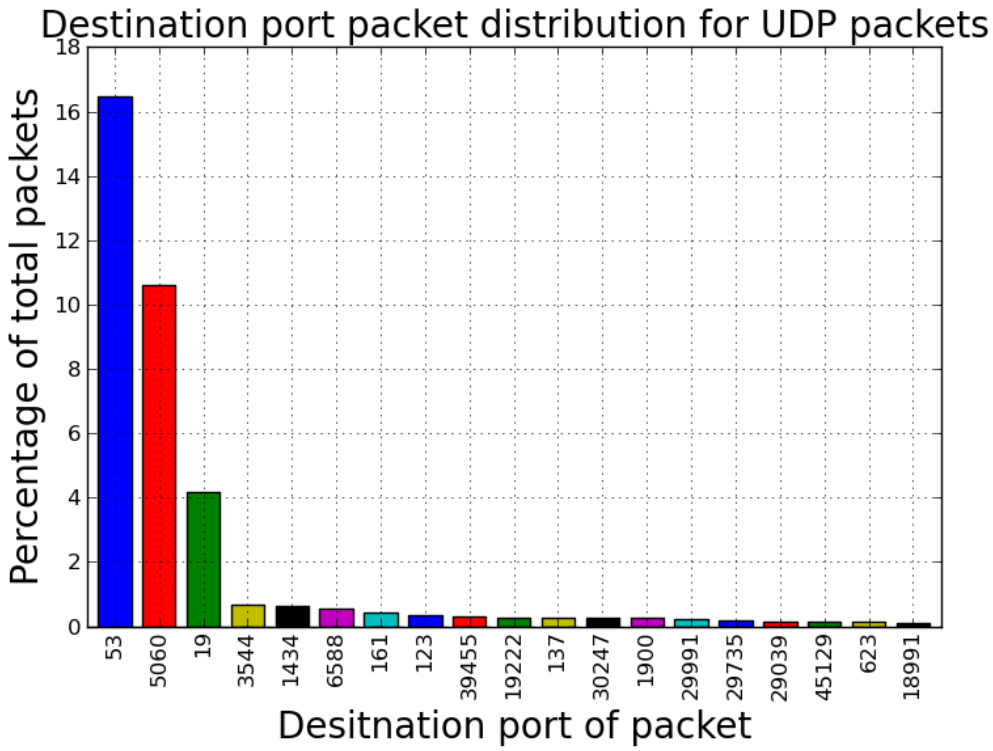


Figure 51: Destination IP packet count

Destination Port UDP	Number of packets recorded	Percentage of total packets
53	642608	16.46
5060	413325	10.59
19	162463	4.163
3544	26600	0.681
1434	24466	0.627
6588	21074	0.540
161	16080	0.412
123	13250	0.339
39455	12120	0.310
19222	11116	0.284
137	10987	0.281
30247	10692	0.274
1900	10096	0.258
29991	7967	0.204
29735	7407	0.189
29039	6046	0.154
45129	5349	0.137
623	4941	0.126
18991	3881	0.099
Total:	1410468	30
Number of unique hits:	3901696	

Table 51: Destination Port UDP packets

9.4 Source IP

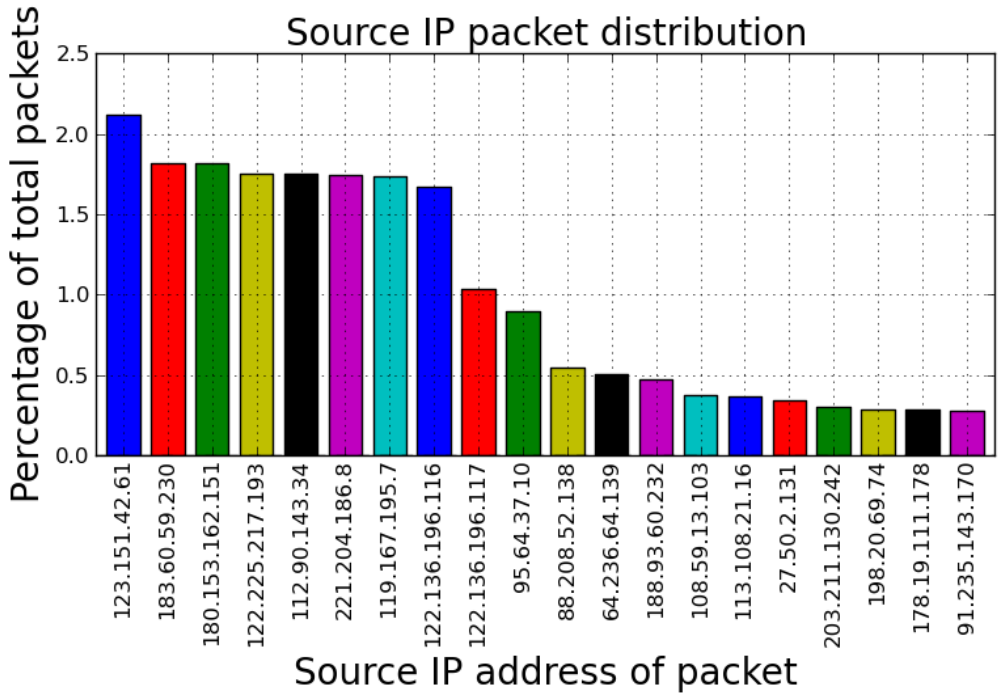


Figure 52: Destination IP packet count

Source IP	Number of packets recorded	Percentage of total packets
123.151.42.61	196594	2.123
183.60.59.230	168315	1.818
180.153.162.151	168048	1.815
122.225.217.193	162377	1.754
112.90.143.34	162332	1.753
221.204.186.8	161182	1.741
119.167.195.7	160546	1.734
122.136.196.116	154683	1.671
122.136.196.117	95925	1.036
95.64.37.10	83143	0.898
88.208.52.138	50611	0.546
64.236.64.139	46485	0.502
188.93.60.232	43890	0.474
108.59.13.103	34895	0.376
113.108.21.16	33841	0.365
27.50.2.131	31544	0.340
203.211.130.242	27909	0.301
198.20.69.74	26406	0.285
178.19.111.178	26389	0.285
91.235.143.170	25802	0.278
Total:	1860917	10
Number of unique hits:	9256741	

Table 52: Source IP packets

9.5 Source Port

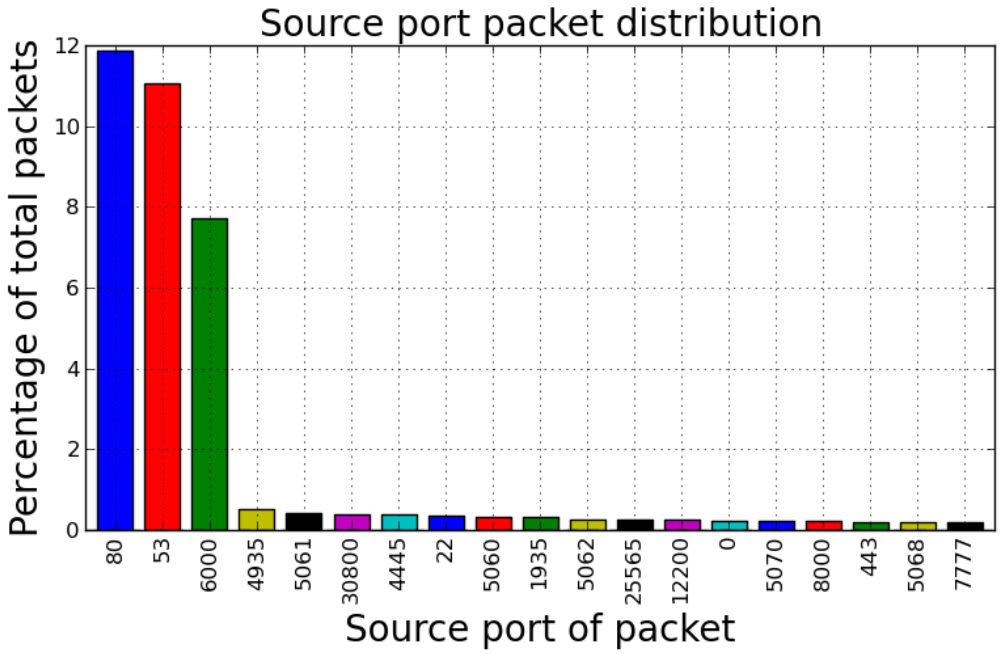


Figure 53: Destination IP packet count

Source Port	Number of packets recorded	Percentage of total packets
80	1098911	11.87
53	1024632	11.06
6000	713763	7.710
4935	48608	0.525
5061	38180	0.412
30800	37277	0.402
4445	35668	0.385
22	32254	0.348
5060	30500	0.329
1935	30471	0.329
5062	25903	0.279
25565	24987	0.269
12200	23546	0.254
0	22199	0.239
5070	22180	0.239
8000	20998	0.226
443	19811	0.214
5068	19692	0.212
7777	19570	0.211
Total:	3289150	29
Number of unique hits:	9256741	

Table 53: Source Port packets

9.6 Geolocation results of source IPs

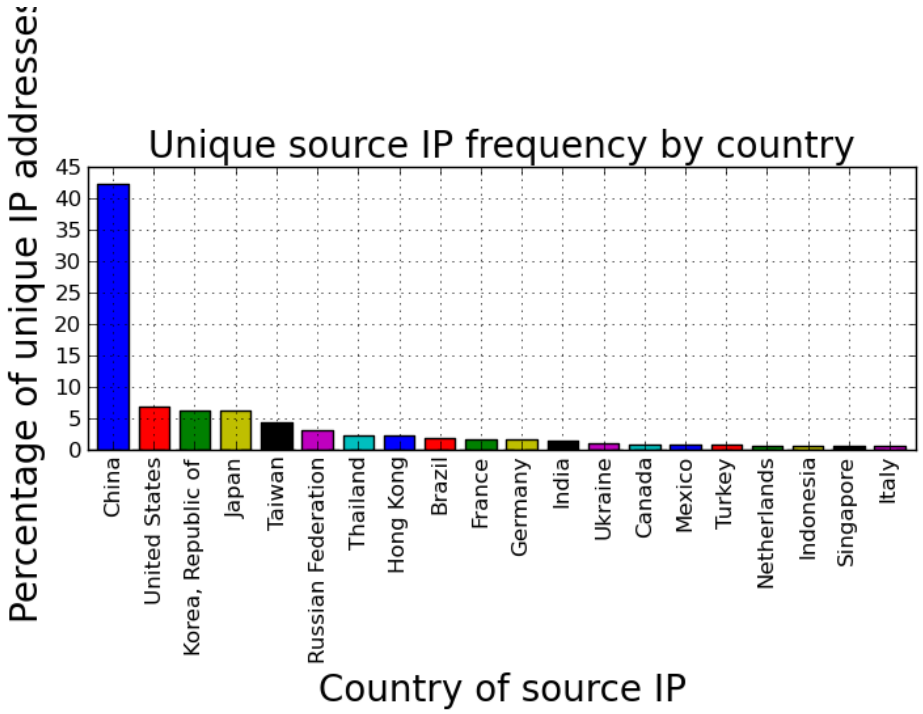


Figure 54: Destination IP packet count

Geolocation results of source IPs	Number of packets recorded	Percentage of total packets
China	188151	42.33
United States	30601	6.884
Korea, Republic of	28415	6.392
Japan	27733	6.239
Taiwan	19539	4.395
Russian Federation	14273	3.211
Thailand	10085	2.268
Hong Kong	10004	2.250
Brazil	8758	1.970
France	7637	1.718
Germany	7620	1.714
India	6873	1.546
Ukraine	4650	1.046
Canada	4393	0.988
Mexico	3996	0.899
Turkey	3676	0.827
Netherlands	3457	0.777
Indonesia	3402	0.765
Singapore	3370	0.758
Italy	3195	0.718
Total:	389828	76
Number of unique hits:	444484	

Table 54: Geolocation results of source IPs packets

9.7 Hilbert curve of darknet

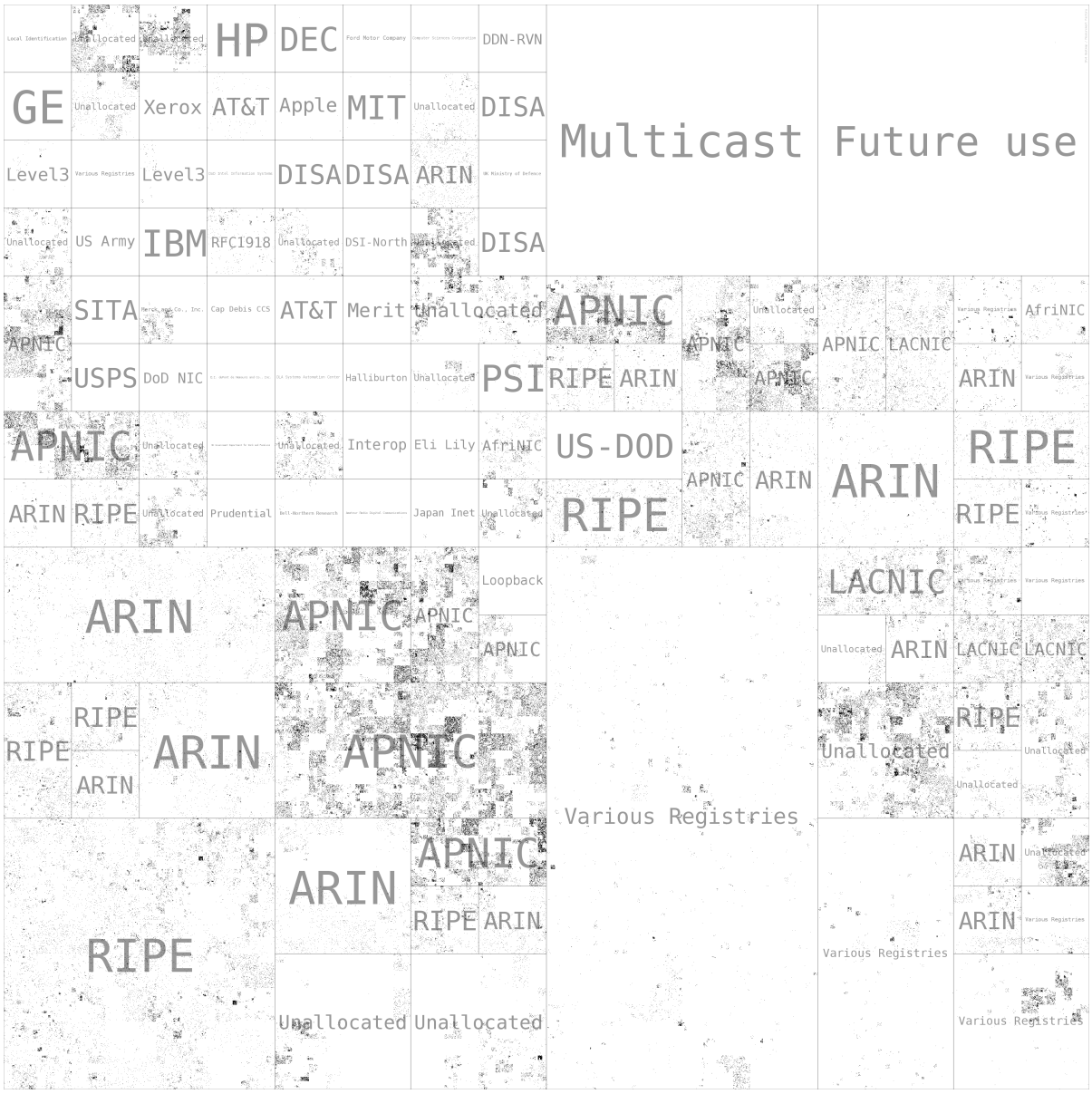


Figure 55: Source IP packet count

A.2 Example ancillary report output

Report on activity for destination IP 146.x.x.65 for the month of December 2013

For the period 01/12/2013 - 31/12/2014

Analysis has been performed on datasets extracted from the packet catpure to produce the graphical and tabular output of the report. There has however been no interpretation of the results generated by the report. A latex document of the report has also been created however that can be altered to include an interpretation of resutls.

Contents

1.6 Time Series

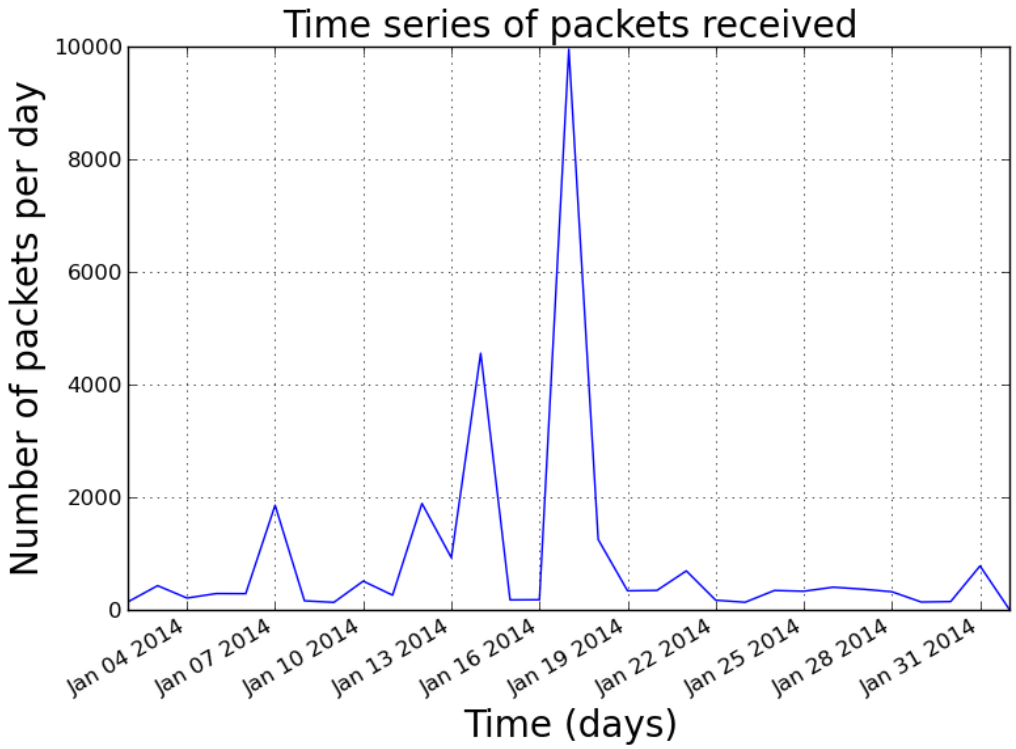


Figure 6: Destination IP packet count

1.7 Cumulative time series

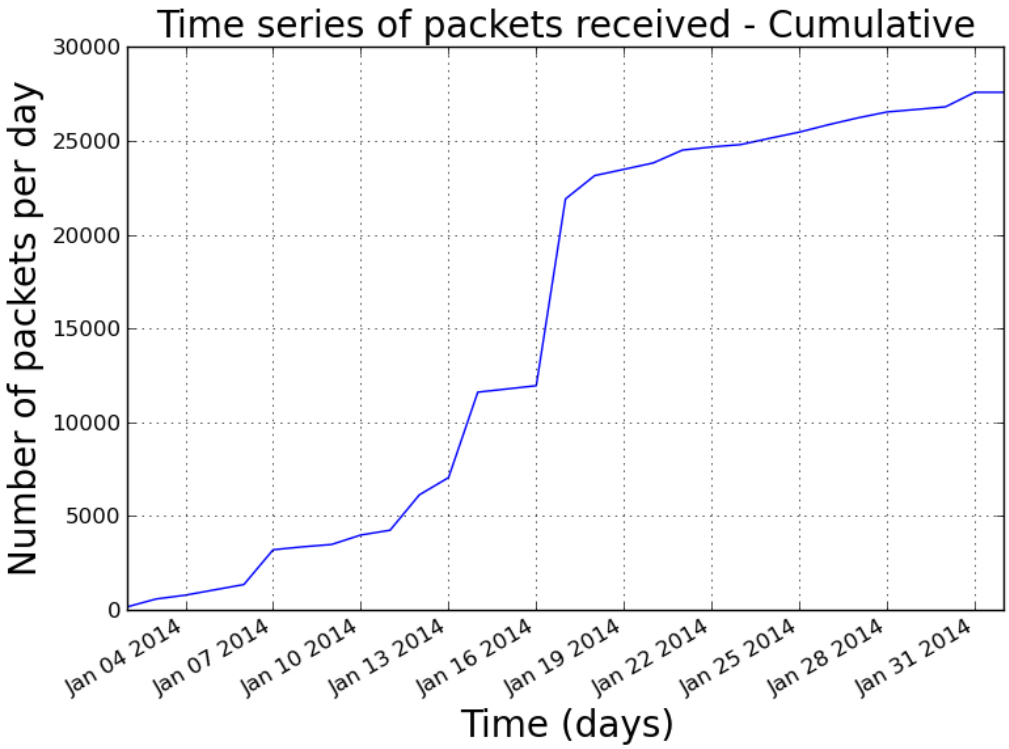


Figure 7: Destination IP packet count

1.8 Scatter plot of destination ports against time

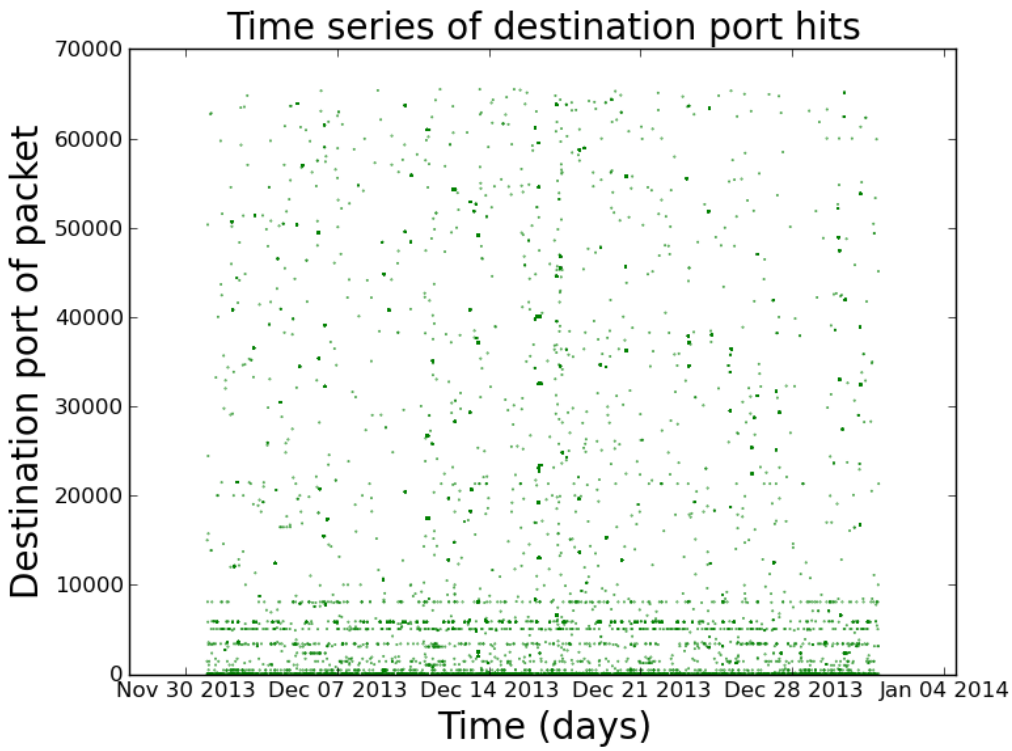


Figure 8: Destination IP packet count

1.9 Time series of port 22

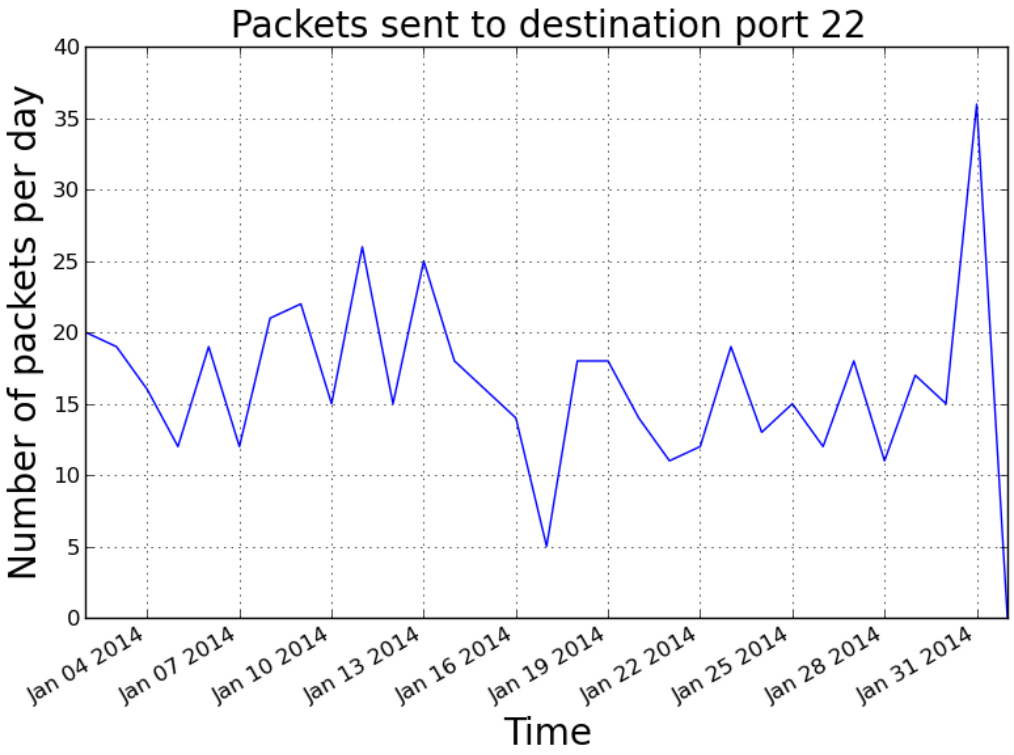


Figure 9: Destination IP packet count

A.3 Project Repository

All of the code has been made available at git clone https://github.com/29LetterAlphabet/Towards_document_generation

It includes all of the original code used throughout the project as well as readme files.