An Analysis of Internet Background Radiation within an African IPv4 Netblock

Submitted in partial fulfilment
of the requirements of the degree of

Master of Science

of Rhodes University

Wadeegh Hendricks

Grahamstown, South Africa
January 2019

Abstract

The use of passive network sensors has in the past proven to be quite effective in monitoring and analysing the current state of traffic on a network. Internet traffic destined to a routable, yet unused address block is often referred to as Internet Background Radiation (IBR) and characterised as unsolicited. This unsolicited traffic is however quite valuable to researchers in that it allows them to study the traffic patterns in a covert manner. IBR is largely composed of network and port scanning traffic, backscatter packets from virus and malware activity and to a lesser extent, misconfiguration of network devices. This research answers the following two questions: (1) What is the current state of IBR within the context of a South African IP address space and (2) Can any anomalies be detected in the traffic, with specific reference to current global malware attacks such as Mirai and similar.

Rhodes University operates five IPv4 passive network sensors, commonly known as network telescopes, each monitoring its own /24 IP address block. The oldest of these network telescopes has been collecting traffic for over a decade, with the newest being established in 2011. This research focuses on the in-depth analysis of the traffic captured by one telescope in the 155/8 range over a 12 month period, from January to December 2017. The traffic was analysed and classified according the protocol, TCP flag, source IP address, destination port, packet count and payload size. Apart from the normal network traffic graphs and tables, a geographic heatmap of source traffic was also created, based on the source IP address. Spikes and noticeable variances in traffic patterns were further investigated and evidence of Mirai like malware activity was observed. Network and port scanning were found to comprise the largest amount of traffic, accounting for over 90% of the total IBR. Various scanning techniques were identified, including low level passive scanning and much higher level active scanning.

Keywords: Internet Background Radiation, Network Telescopes, Darknet, Network Scanning

Acknowledgements

"Seek knowledge from the cradle to the grave" - Prophet Muhammad *pbuh*

I would firstly like to thank the Almighty Creator for giving me health, strength and ability to get through these 2 years. At times it was very tough, but I know it was my spirit, faith and family support that pulled me through.

To my wife, the force that drives me to better myself. Thank you for all the support and encouragement you have given me over the years. The sacrifices that you have made for our family by giving me the time to do this, by keeping our two boys occupied and entertained while I did my work; and in the last month, playing the role of an almost single parent to our newborn baby. I would not have made it through these 2 years were it not for you. To my 3 beautiful children, I know you have missed spending time with me while I sat locked in my study typing away. Its finally over and daddy can spend all the time with you now.

To my parents, the ones who always encouraged me to continue my education and be the best. Thank you for the good example that you have set for me and for raising me to be the person that I am. To the rest of my family (siblings, inlaws and friends), thank you for always supporting me, inspiring me and having faith in me. It meant so much over these years.

To my supervisor, Prof Barry Irwin, thank you for all the guidance, support and assistance you have given me these past 2 years. Sorry about all the missed deadlines and thanks for understanding. To the larger Rhodes Computer Science department, thanks for everything, especially the coffee machine. To my classmates, its been an awesome ride together. Thanks for all the help and support.

I would like to thank the many creators and contributors of the applications I used for my data analysis and presentation of results. The applications include Wireshark, Tshark, Maxmind, PostgreSQL, Tableau, P0f, NetworkMiner and Kali Linux.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Network telescopes are passive sensors which typically sit on the internet side of a firewall and captures all traffic destined for a particular IP address range. The netblock is unused, thereby making the traffic reaching it unsolicited. This unsolicited traffic is often referred to as Internet Background Radiation (IBR). This IBR is essentially one-way traffic, consisting of the connection attempts being made from the internet. No acknowledgements or replies are sent. There are many sources for IBR, which ranges from IP address and port scanning, denial of service attacks (Moore *et al.*, 2004), internet worm and virus activity, internet censorship (Dainotti *et al.*, 2011) and potentially misconfiguration of network devices. Irwin (2011) states that this traffic can be classified as hostile as it is unsolicited and unwanted.

Pang *et al.* (2004) state that by monitoring unallocated IP address segments, as is the case with network telescopes, is a much better way to measure IBR than by simply measuring all unsuccessful connections. The latter would actually produce false data if there was a host down at the time which was unable to respond to valid connections. Research done by Zou *et al.* (2005), showed that IBR was very useful as an early warning system when detection systems tested for certain anomalies in network traffic.

1.1 Problem Statement

The Center for Applied Internet Data Analysis[1] (CAIDA) runs one of the largest IPv4 network telescopes, which is located at the University of California in San Diego. It monitors a full /8 IPv4 segment (2^{24} addresses), which equates to $\frac{1}{256}$ of all available IPv4 addresses (Moore *et al.*, 2004). The work done at CAIDA were some of the first to use network telescopes as a tool to study internet traffic patterns. The research done at CAIDA has provided valuable insight into the following areas:

- Network worm and virus activity
- Malicious network and port scanning
- Denial of service attacks

Over the past few years there has been a great increase in malware activity across the internet, such as Conficker (Irwin and Nkhumeleni, 2015a), Mirai (Antonakakis *et al.*, 2017) and Code Red (Moore *et al.*, 2002). These came in the form of viruses, self replicating worms and more recently, ransomware. Understanding how the malware replicates, infects hosts and spreads itself is the first step towards stopping it. Security researchers in particular, draw on this knowledge and understanding to assist in the development of security systems aimed at protecting networks. Information security encompasses so many interesting areas of research, with network security being one of the most important. The world has evolved to such a technological extent, that many devices and systems are interconnected, sharing information across networks and over the internet. Understanding the way internet traffic has evolved is an important part of learning how to optimise networks, developing safer and more efficient protocols and making the internet more secure (Fomenkov *et al.*, 2004; Wang *et al.*, 2010).

Although there already exits a large body of research in the area of network telescopes and internet background radiation, the majority of it has been done on IP address ranges residing outside of South Africa. Little is actually known about the unsolicited traffic targeting IP addresses in South Africa and Africa in general. The limited research available for South African IP address spaces is largely due to the work done by Rhodes University, where they operate five /24 IPv4 network telescopes. A detailed analysis of the captured traffic from these telescopes have not been done since Irwin (2011); Nkhumeleni (2014), with little known of how the traffic patterns have changed over the years. This research therefore aims to investigate it through the objectives that follow.

[1] https://www.caida.org/projects/network_telescope/

1.2 Research Objectives and Goals

In order to get an better understanding of the makeup of unsolicited internet traffic targeting South African IPv4 addresses, this research will focus on the following goals and objectives.

Primary Objectives

- How has the Internet Background Radiation changed over the last 10 years when compared with the analysis done by Irwin (2011); Yates (2014); Irwin and Nkhumeleni (2015a) and others.
- Are there any significant anomalies that can be detected in the traffic, with specific reference to recent global attacks such as the Mirai botnet[2] (Antonakakis *et al.*, 2017) and EternalBlue[3] (Nakashima and Timberg, 2017) based malware variants such as WannaCry (Mohurle and Patil, 2017) and Petya (Richardson and North, 2017).

Secondary Objectives

- Analysis of IBR traffic according to protocol, port, source address, geolocation of source traffic, etc.
- What conclusions can be derived from the analysis of the data.
- What observations can be made regarding the findings.
- What trends can be reported.

1.3 Scope of this Research

Rhodes University has been collecting IBR on five network telescopes for many years and much research has been conducted on the data already. There have been comparative analysis of traffic data across the five telescopes (Nkhumeleni, 2014), traffic analysis to identify specific worms and malware activity (Irwin, 2012) and general traffic analysis and taxonomy of traffic done (Barnett and Irwin, 2008; Cowie and Irwin, 2010).

[2] https://en.wikipedia.org/wiki/Mirai_(malware)

[3] A security vulnerability in Micosoft Server Message Block found within Microsoft Windows operating systems

This research will focus on an in depth analysis of traffic from a single /24 network telescope in the 155/8 range, collected over the 12 month period of 2017 by the Computer Science Department of Rhodes University. The network telescope is a passive sensor and does not respond to any incoming packets, rather it logs the packet and then drops it. For the 12 month period there were no interruptions in service or downtime due to hardware or network issues, hence it was a full and complete data set.

1.4 Methodology

The methods to employ in order to successfully achieve the target objectives will be to firstly become familiar with the data by viewing it in various tools and by gathering statistics on the data. Understanding the data and its contents allows a researcher to get a feel for the data, thereby having a better knowledge of how and what information to extract from it. A sizeable portion of the analysis will be to identify and extract statistics from the data in the form of “top 10” results. As the data set is quite large, grouping similar events together and then computing statistics and drawing visualisations from it would make it easier to view patterns and trends. Once a pattern or trend is identified, the data will be analysed to a more granular level in order to fully understand the trends.

This research will have aspects of both quantitative and qualitative methods. The largest portion of the research will focus on the quantitative aspects when measuring the statistics and metrics from the data analysis, but it will stray a bit into qualitative methods when interpreting some of the trends and patterns.

1.5 Document Structure

This document is divided into a sequence of chapters, each covering a different step of the research, with each one supporting the contents of the previous chapter and continuing on where the previous one left off. The layout of the document is as follows:

Chapter 1
This chapter introduces the subject matter, explains the objectives and goals of the research and discusses the scope and methods employed in the research.

Chapter 2
A review of previous literature and research is covered in this chapter. The statistics from

previous research that are relevant to this one are also highlighted and compared. The relevant key concepts are explained and the chapter sets the tone for what will be done in this research paper.

Chapter 3
This chapter focuses on the data and analysis methods. Firstly, it covers a detailed account of the source and structure of the data. It then proceeds to discuss the various methods available for the analysis of the data and which ones were eventually used. The tools used in the analysis of the data are also covered. The initial results and overview of the data is provided in this chapter as well.

Chapter 4
This chapter presents the results and findings of the research. It compares the results with previous research that was done. Detailed statistics and metrics are provided in this chapter too.

Chapter 5
This chapter further investigates interesting findings from the analysis and results of the previous chapter. These findings are handled as individual case studies.

Chapter 6
This chapter is dedicated to comparing the findings of this research with that of previous work. The main focus is towards previous research conducted on the Rhodes University telescope data, but concludes with a brief comparison to other work as well.

Chapter 7
This chapter concludes the research and reflects on the objectives that were initially set. It discusses whether the research questions were answered and looks at potential future work in this field of study.

CHAPTER 2

Literature Review

This chapter takes a look at the previous research done in the field of network telescopes and IBR. The methods used in the previous studies and the key concepts derived from it will be applied to this research paper. This chapter also serves to explain the key concepts and results that were obtained in previous studies and to give a more detailed description of the subject matter.

Section 2.1 starts by introducing the key concepts of network telescopes, darknets, IBR and the pioneering work done in this field of study. Some of the benefits of IBR analysis are briefly mentioned and it also discusses a taxonomy used to distinguish between various the types of network telescopes found. Section 2.2 continues by discussing the impact of network telescope sizes on the probability of observing random packets on the internet. The TCP/IP suite is discussed in great detail in Section 2.3, with reference to lower level and application level protocols. This section contains three sub-sections which are used to discuss the lower level protocols (TCP, UDP and ICMP) in more granularity. Section 2.4 takes a look at Internet Protocol (IP) and how it is currently implemented on the internet. The differences between IPv4 and IPv6 are discussed. Current research being done on IPv6 network telescopes are discussed in this section. Section 2.5 discusses SIP traffic due to its high prevalence within IBR as well as its usage as a scanning and denial of service vector. The last two, Sections 2.6 and 2.7, deals with the classification and analyses techniques used on IBR and the malicious network activity that is found in IBR respectively. Section 2.7 examines network and port scanning, denial of service attacks and worm, viruses and malware activity on the internet, each being discussed in its own sub-section. Section 2.8 summarises and concludes the chapter.

2.1 Background on Network Telescopes and IBR

Using passive sensors to study IBR have in the past proven to be quite effective for many researchers. Often many network security researchers are only concerned with malicious activities aimed at actual systems or valid IP addresses. They tend to ignore traffic specifically targeted at the unused network segments. Moreover, much of the network analysis done after a targeted attack will show traffic patterns for that particular attack, not necessarily the reconnaissance work done prior to the attack. Moore *et al.* (2004) were some of the first researchers to actually study IBR, referring to it as "*backscatter analysis*". Their research focused primarily on understanding denial of service attacks. Among the earlier work done on IBR are the analysis of source IP addresses (Barford *et al.*, 2006) and the general categorisation and classification of IBR (Pang *et al.*, 2004; Wustrow *et al.*, 2010).

Irwin (2011) further expands the taxonomy of network telescopes by classifying them according to the netblock being monitored. The classification scheme, as stated by Irwin (2011), is:

- **Blackholes, Darknets and Sinks** - These are network segments that contain no active hosts on them. All the IP addresses in the segment are not in use and any traffic destined to it forms part of the data set.

- **Dimnets** - This refers to a network segment that has scattered active hosts in it. The network segment is usually quite large and any traffic generated to and from these active hosts are excluded from telescope data set.

- **Greynets** - These consist of multiple non-contiguous network segments which contain active hosts in between them. Though quite similar to *Dimnets*, these differ in that fact that they are made up of non-contiguous network segments and have many more active hosts located between the network segments.

Antonakakis *et al.* (2017) observed in their study that their network telescope was able to identify large scale Mirai botnet attacks. In the seven months of traffic that they analysed, they were able to identify over 116 billion Mirai connections coming from more than 55 million unique source IP addresses. According to Antonakakis *et al.* (2017), when the Mirai worm does a scan, it performs it in a very unique manner, leaving a type of fingerprint. With the telescope, they were able to detect these fingerprints quite easily. Irwin (2012) stated that distinct trends were observed which were related to the Conficker worm. The researcher was able to identify the exact dates that the attacks took place as well as identifying the different variants of the worm. Similar observations were made by Jonker *et al.* (2017) with regards to telescope data vs distributed honeypot data when it came to DDOS[1]. In the two years of data, as describes by Jonker *et al.* (2017), the network telescopes received over 12 million attacks compared to about 5.5 million attacks directed at the honeypots.

2.2 Network Telescope Size

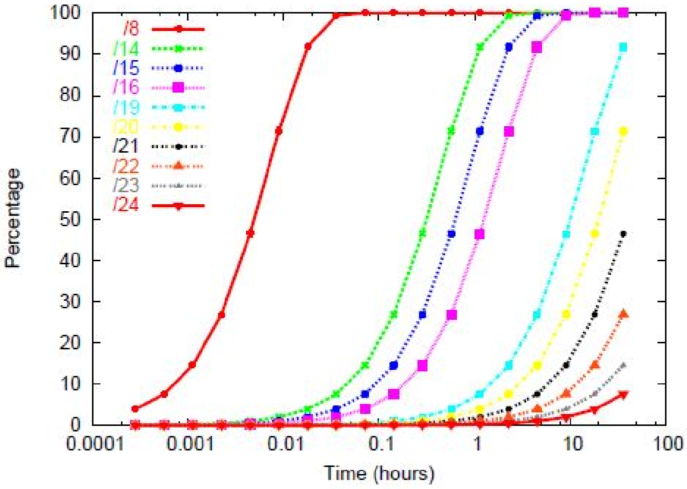


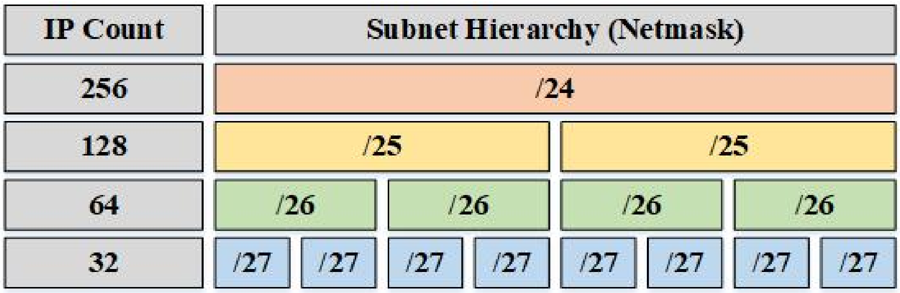
Figure 2.1: The probability of observing a random packet from a particular host (Moore *et al.*, 2004)

Moore *et al.* (2004) and Pemberton *et al.* (2007) states that the accuracy of observing large scale global internet events is dependant on the size of the telescope. The larger the IP range, the more accurate the deductions would be. The probability of observing traffic from a particular host is therefor greatly influenced by the size of the telescope. Moore *et al.* (2004) shows the percentage probability versus time of observing traffic from a specific host. In a full /8 network, the percentage probability is much higher than on

[1] Distributed Denial of Service

a single /24 network and the time to wait for the packet is also greatly reduced. Moore *et al.* (2004) states that according to their research, the probability of observing a random packet on a /8 network within the first hour is 100%, whereas on a /24 network it would drop down to around 0-1% for the same time period. The scaling between detection time and the network segment size is not linear. Moore *et al.* (2004) further states that for a random packet to be observed on a /24 telescope with the same confidence in detection as that of a /8 telescope, it would take 65 664 times longer.

Although the size of a network telescope affects the probability of observing a random packet, it has little impact on targeted scans or attacks. Smaller telescopes have however still been useful in providing valuable insight into darknet traffic. Woodhead (2012) analysed traffic from a small /24 IPv4 telescope over a 30 day period. The research showed that despite there only being 256 IP addresses in the range, each unique destination IP address received at least one packet within a 60 minute window. Chindipha *et al.* (2018) did a study on the effectiveness of using smaller network telescopes to study IBR. The study used two non-contiguous /24 IPv4 network telescope data sets, collected in February 2018. The collected data was divided into sample sizes based on destination IP addresses and then further analysed. Each of the /24 netblocks were split into equal /25 netblocks, who were then further split into equal /26 and then /27 netblocks (Fuller and Li, 2006), as illustrated in Figure 2.2.



IP Count	Subnet Hierarchy (Netmask)							
256	/24							
128	/25				/25			
64	/26		/26		/26		/26	
32	/27	/27	/27	/27	/27	/27	/27	/27

Figure 2.2: Breakdown of /24 subnet into sample sizes (Chindipha *et al.*, 2018)

The results showed that the count of unique source IP addresses targeting the various samples were fairly consistent throughout, with only minor variances observed. For example, there was only a difference of 0.69% between the count of unique source IP addresses received between the two /25 subnets. With the eight /27 subnets, the percentage count for unique IP addresses for each of the subnets differed by less than 1%. The same similarities were observed within both of the /24 data sets.

The benefit of using a larger network telescope is obvious, in that it allows for a greater percentage of the internet monitored, but with the availability of usable IPv4 addresses become scarcer, it has become nearly impossible to obtain an unused /8 range for these purposes.

2.3 TCP/IP Suite and Network Protocols

TCP/IP is a suite of protocols originally developed by DARPA[2]. It is essentially made up of two protocols, TCP (Transmission Control Protocol) and IP (Internet Protocol) (Forouzan and Fegan, 2002). The OSI[3] Reference Model was developed by the International Standards Organisation to create a standardised architecture of the implementation of networks and telecommunications (Wetteroth, 2002). The model consists of seven individual layers, each one controlling and taking responsibility for a different portion of the communication. The seven layers of the OSI model, along with their functionality, are summarised as follows.

- **Physical** - The first layer in the model, which deals with the physical media that will carry the data or signal between the different points. It covers the physical characteristics of the media (copper wire, fiber optic, wireless, etc), the bit rate within the media, the different topologies of the media and the transmission that is supported (speed and duplex).

- **Data Link** - The Data Link layer is responsible for the "organisation" of the data stream into manageable frames, performs error detection on the frames, controls the flow of the frames and is responsible for the transfer of frames between different networks or systems by means of using hardware addressing.

- **Network** - Internet Protocol is found within this layer and it adds in the routing and IP addressing information to create the packet.

- **Transport** - This layer is responsible for the delivery of the data between sender and recipient. It performs error control through retransmission, flow and connection control and is responsible for the reassembly of packets at the destination. TCP and UDP can be found in this layer.

- **Session** - The Session layer provides extra services to the previous three layers in maintaining and synchronising the connection between the sender and receiver.

[2] Defence Advanced Research Projects Agency - A military agency of the United States responsible for defence research.

[3] Open Systems Interconnectivity - A conceptual networking framework used to characterise communication between IT systems

- **Presentation** - This layer formats the exchanged information between the sender and receiver. Often the two systems are not the same and the information needs to be translated on either end. This layer also takes care of encryption and compression of the packets. Protocols such as SSL, TLS and MPEG can be found in this layer.
- **Application** - The final layer in the OSI model, this layer makes the information accessible to the end-user applications. The accessibility is provided through various interfaces, using specific ports and protocols applicable to the applications.

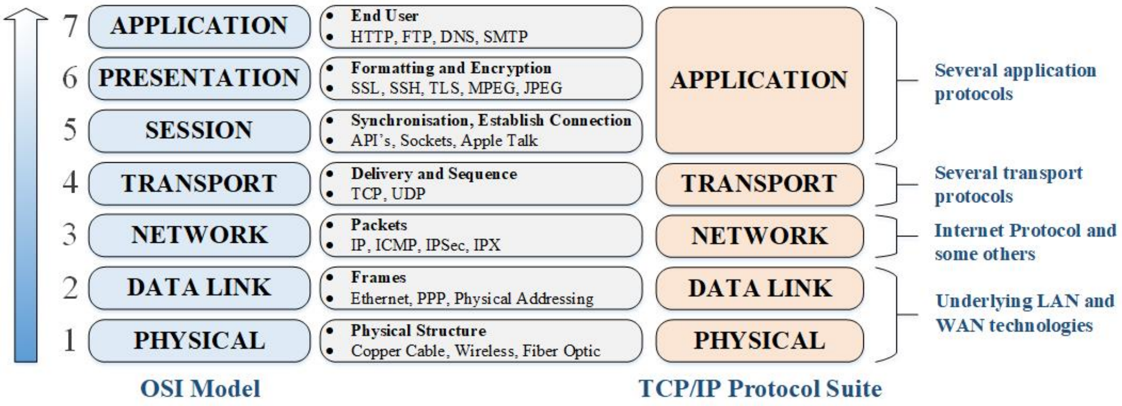

7 APPLICATION
End User
HTTP, FTP, DNS, SMTP
6 PRESENTATION
Formatting and Encryption
SSL, SSH, TLS, MPEG, JPEG
5 SESSION
Synchronisation, Establish Connection
API's, Sockets, Apple Talk
4 TRANSPORT
Delivery and Sequence
TCP, UDP
3 NETWORK
Packets
IP, ICMP, IPSec, IPX
2 DATA LINK
Frames
Ethernet, PPP, Physical Addressing
1 PHYSICAL
Physical Structure
Copper Cable, Wireless, Fiber Optic
APPLICATION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL
Several application protocols
Several transport protocols
Internet Protocol and some others
Underlying LAN and WAN technologies
OSI Model
TCP/IP Protocol Suite

Figure 2.3: OSI Reference Model compared to the TCP/IP Suite (Forouzan and Fegan, 2002)

A comparison between the OSI layers and TCP/IP layers can be seen in Figure 2.3. Although TCP/IP was in development and operational before the OSI model was accepted, it does exhibit many similarities in its designs. Figure 2.3 compares the OSI Reference Model and TCP/IP suite. TCP/IP combines the top three layers of the OSI model (Session, Presentation and Application) into a single "Application" layer. Within this layer, all of the session, presentation and application protocols can be found.

Network protocols related to traffic can be loosely divided into two types, namely lower level protocols and higher level protocols. Lower level protocols can be found within the Network and Transport layers and higher level protocols can be found in the Session, Presentation and Application levels. The levels referred to here are the levels within the OSI Reference Model. There are many lower level protocols, but the most common ones found in network telescope packets are TCP, UDP and ICMP (Forouzan and Fegan, 2002). The higher level protocols are associated with the applications and ports used and they rely on the lower level protocols for the way in which they are transported. Examples of higher levels protocols and their ports are are HTTP (80/tcp), TLS (443/tcp), FTP (21/tcp), SIP (5060/udp) and SMTP (25/tcp).

Much of the network telescope research that has previously been done concerns itself with a deep analysis of the protocols and ports in the traffic. The protocols and ports being used provides an indication of the type of traffic arriving at the network telescope and to some extent, the possible sources of the traffic. Research done by Antonakakis *et al.* (2017) was able to track the traffic pattern flows of the *Mirai* botnet to get an understanding of how it propagated and infected other devices. Their research analysed the connections made on 23/tcp and 2323/tcp as an indication of Mirai activity.

Studies often compare the percentage of TCP, UDP and ICMP traffic that makes up IBR. Table 2.1 shows the findings of some of the previous research done on network telescope data. As can be seen in the table, TCP is by far the most prevalent traffic type. It surpasses both UDP and ICMP by quite a substantial amount.

Table 2.1: Results of previous research done using network telescope data

Research Paper	**Telescope Size**	**Research Findings**		
		% TCP	**% UDP**	**% ICMP**
Irwin (2011)	/24	81.57	12.62	5.89
Woodhead (2012)	/24	97.9	1.4	0.7
Fachkha *et al.* (2012)	/16	91.9	5.5	2.9
Czyz *et al.* (2013)	/8	81.7	15.8	2.3
Fachkha and Debbabi (2016)	/8	76.6	19.9	2.8

2.3.1 TCP

TCP is the most widely used transport layer protocol on the internet today and it has many benefits over UDP. TCP is a stateful protocol, meaning that between the sender and receiver a session needs to be established first. Only then can communication take place. Once the session is established, the connection is maintained throughout the session and once the communication is completed, the session is closed. Traditional TCP communication uses a three step handshake method to establish the communication channel, as depicted in Figure 2.4. The source will send a SYN (synchronization) packet to the destination, who will then in turn respond with a SYN-ACK (synchronization-acknowledgement) packet. The source will then reply with an ACK (acknowledgement) packet to complete the handshake (Postel *et al.*, 1981b).

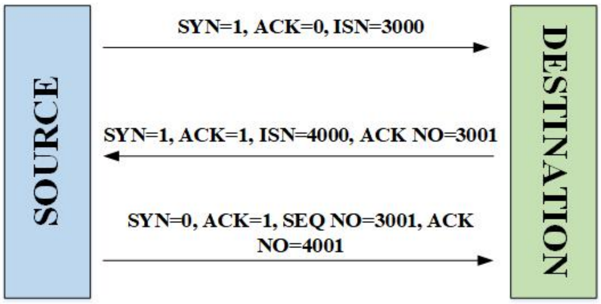

SOURCE
DESTINATION
SYN=1, ACK=0, ISN=3000
SYN=1, ACK=1, ISN=4000, ACK NO=3001
SYN=0, ACK=1, SEQ NO=3001, ACK NO=4001

Figure 2.4: TCP/IP 3 way handshake (Postel *et al.*, 1981b)

Each TCP packet contains a header with various fields and settings. These settings are used in the establishment of the communication session, maintaining the session and eventually closing the session. Of the many fields found in the TCP header, only ten of them are mandatory. Figure 2.5 depicts the TCP header, showing all the mandatory fields. Three of the mandatory fields used in the analysis of network telescope traffic, along with its description, are listed below:

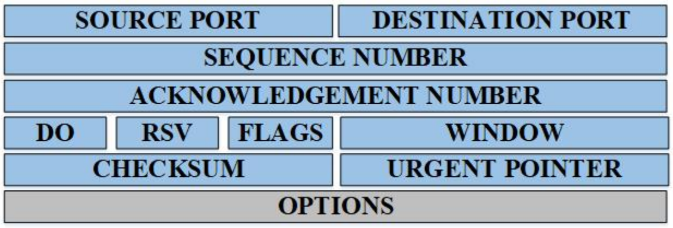


Figure 2.5: TCP header showing all mandatory fields (Postel *et al.*, 1981b)

- **Source Port** - The sender's port used to initiate the connection or respond to one. It is 16 bits in length and is often randomly generated.

- **Destination Port** - The field is 16 bits in length and is the port associated with the service being connected to, i.e. when connecting to a web page, the server will usually listen and accept connections on port 80 or 443.

- **Flags** - This field is also referred to as the *Control Bits*. It is six bits in length and each bit can be either a "1" or a "0", turning the control bit on or off. The six control bits are *URG* (Urgent Pointer), *ACK* (Acknowledgement), *PSH* (Push Function), *RST* (Connection Reset), *SYN* (Synchronize Sequence Numbers) and *FIN* (Final Packet).

Due to its reliability, flow control, congestion control and error checking, TCP is the preferred protocol for many types of data transmission and applications. Applications such as FTP, SMTP, Telnet and SSH all uses TCP. It is therefore understandable that

TCP generally makes up the majority of traffic on the internet.

2.3.2 UDP

UDP differs in comparison to TCP in that it is a stateless protocol and does not concern itself with whether packets get delivered or not. There is no complicated establishment of sessions or three way handshaking as with TCP (Forouzan and Fegan, 2002). UDP packets or datagrams also has a header on every one, but it is much less complex than the header of TCP. The header contains four fields, with only two of them being mandatory. All fields are 16 bits in length. Figure 2.6 depicts the UDP header, with mandatory fields marked in blue.

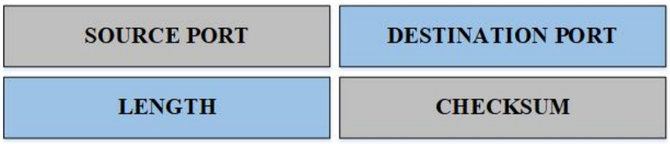


Figure 2.6: UDP header showing all fields

The list of UDP header fields, as outlined by the UDP RFC (Postel, 1980), are further described below:

- **Source Port** - This field contains the randomly generated sender port.
- **Destination Port** - This is a mandatory field and the value is associated to the application being connected to and listening port of the destination address.
- **Length** - This field is mandatory and contains the value in bytes of the header and data.
- **Checksum** - This is an optional field which could be used for datagram error checking.

Due to the fact that UDP does not have any error checking or congestion checking functionality, packet loss is often experienced. For applications that do not require these functionalities, but instead requires a lightweight option, this protocol is often used. Examples of applications that use UDP are multimedia streaming and voice over IP services. The Session Initiation Protocol (SIP), which is widely used in voice over IP systems, can be encapsulated in either TCP or UDP, but UDP is the preferred option in most systems.

2.3.3 ICMP

Whereas TCP and UDP are transport layer protocols, ICMP is situated just below it in the network layer and encapsulated within Internet Protocol. Due to the fact that IP lacks certain functionality such as error detection and error correction, ICMP provides a type of support role to IP (Forouzan and Fegan, 2002). ICMP datagrams are used to determine the route to a destination host, check if gateways are operational and to determine the shortest path to the destination host (Postel *et al.*, 1981a). Like TCP and UDP, each ICMP datagram has a header with required fields. Figure 2.7 depicts the ICMP header with its fields.

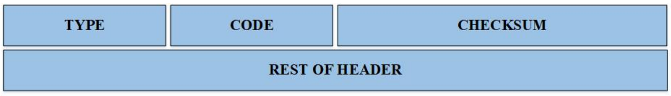


Figure 2.7: ICMP header showing all fields (Postel *et al.*, 1981a)

The main fields of the ICMP header are *Type*, *Code* and *Checksum*. The type field is eight bits in length and sets the type of ICMP datagram being sent. There are many types of ICMP datagrams, but they can be split into two basic groups, namely query messages and error reporting messages (Forouzan and Fegan, 2002). Table 2.2 summarises the main type fields and classifies them according to the groups.

Table 2.2: ICMP datagram types (Forouzan and Fegan, 2002)

Category	**Type**	**Description**
Error Reporting	3	Destination unreachable
Error Reporting	4	Source quench due to flow control
Error Reporting	11	Time to live has expired
Error Reporting	12	Issue with a parameter
Error Reporting	5	Message redirection
Query Message	0	Echo reply
Query Message	8	Echo request
Query Message	12	Timestamp reply
Query Message	13	Timestamp request

The code field is eight bits in length and works along with the type field to further identify the datagram. As an example, when an ICMP datagram has type field "3" set, it can have the code set to a number between zero and 15. When the code is "0", the datagram indicates that the destination network is unreachable, but when the code is set to “3”, it indicates that the destination port is unreachable. All 16 codes for type 3 have different descriptions for the datagram. The checksum field is used for error checking and is 16 bits in length. The rest of the header field contains additional information and may vary according to which type and code is set. This field is 32 bits in length.

There are many applications available that uses ICMP packets to test network connectivity issues and response times to destination hosts. *Ping*[4] (Packet Internet Gopher) is a very common software utility, available as part of most operating systems, that can be used to perform such tests. It is specifically used for message queries and not error reports. An echo request, which has an ICMP type set to “8” is used in this way. The response would be in the form of an acho reply, which has an ICMP type set to “0”. This is discussed further in Section 2.6, which deals with active and passive traffic. A major portion of ICMP traffic on the internet is generated by applications like this, as it is a quick and easy way to check if a host is active or not.

2.4 Internet Protocols

IPv4 is still the most commonly used internet protocol today, but when IPv6 eventually gets implemented globally, each networked device could potentially have its own unique IP address. IPv6 uses 128bit addressing as opposed to IPv4 which uses 32bit, giving a potential of 3.4 x 10^{34} usable addresses (Stallings, 1996). This means that network address translation will no longer be required and that devices could potentially be connected to one big “public” network, with the internet being the backbone of it. Table 2.3 below summarises the key differences between IPv4 and IPv6 as it applies to this discussion.

Table 2.3: Comparison between IPv4 and IPv6

	IPv4	**IPv6**
Address Space	4.29 x 10^{9}	3.4 x 10^{34}
Address Length	32 bit	128 bit
Representation	Dotted Quad Decimal	Hexadecimal

Since IPv4 is the most widely used internet protocol at the moment, it is understandable that most of the network telescope and internet background radiation research would

[4] https://en.wikipedia.org/wiki/Ping_(networking_utility)

focus their efforts in this area. There has however been a small amount of research done on IPv6. In one of the first IBR studies done on IPv6, Ford *et al.* (2006) reported that their research yielded very little data, stating that only 12 packets had been received by the network telescope for the period that they were capturing. The capturing was done on a small /48 network from December 2004 up to and including March 2006. Ford *et al.* (2006) states that other similar research to this, but utilising a larger IPv6 address space, yielded similar results to theirs. In research conducted by Irwin (2011), it is stated that over an 18 month monitoring period of a /48 network, the only packets recorded were the status probes sent to the network telescope.

In a later study done by Czyz *et al.* (2013), they performed a collection and analysis of IPv6 IBR. The study was meant to build on the previous research done by Ford *et al.* (2006). Their network telescope had a much greater address space than previous studies, utilising five /12 IPv6 networks segments, which equates to a very large portion of the total IPv6 addresses currently allocated to the internet. At the same time, they configured an IPv4 telescope with a network range slightly smaller than a /8 network, to compare the results against each other. Table 2.4 shows the comparison in low level protocol breakdown between the observed IPv4 and IPv6 network telescope traffic.

Table 2.4: Comparison of traffic breakdown between IPv4 and IPv6 (Czyz *et al.*, 2013)

Network Segments	**% TCP**	**% UDP**	**% ICMP**	**% Other**
IPv4 [/8]	81.7	15.8	2.3	0.2
IPv6 [5 x /12]	3.3	2.9	93.8	< 0.1

IPv6 is increasingly being adopted within academia and organisations, with provision being made for the support of both IPv4 and IPv6, according to Li *et al.* (2014). They also mention that Chinese researchers have developed a full IPv6 only backbone which is currently being used for security testing and other research.

2.5 Session Initiation Protocol

SIP scans have become quite commonplace on the internet (Raftopoulos *et al.*, 2015), often accounting for the largest portion of UDP traffic (Fachkha *et al.*, 2012; Irwin, 2012) within IBR.

Session Initiation Protocol operates as a signalling protocol, used in the streaming of voice and video applications. It is responsible for the initiation, maintenance and termination of the session. Voice over IP services typically uses SIP in their communications (Johnston, 2015). SIP uses a standard method of creating, maintaining and terminating the session. Fachkha *et al.* (2012) noted that SIP has become one of the most prevalent applications found within darknet traffic, with it currently being used for denial of service attacks against voice of IP systems.

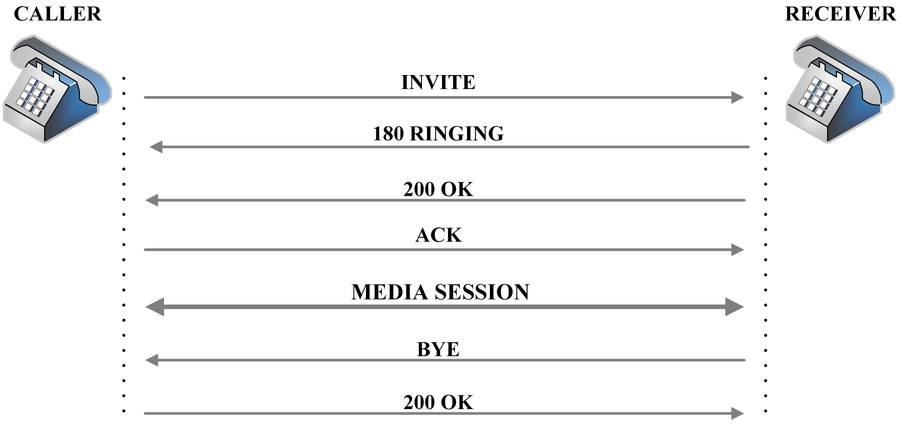


Figure 2.8: Establishment of a SIP session (Johnston, 2015)

SIP uses a text based encoding for its messages, which is readable within Wireshark. Figure 2.8 shows a simple establishment of a SIP session. When the session is initiated via the “INVITE”, certain header fields are set according to the requirements. As noted by Rosenberg *et al.* (2002), the header begins with the word INVITE, which is the method name. It then follows with a list of header fields, some of which are required and some of which are optional. Below are a list of header fields which are required.

- **Via** - The header field contains the DNS or IP address to where the connection is being sent. A second parameter called ”branch” is used to identify the session. An example would be “Via: SIP/2.0/UDP sipserver.ru.ac.za;branch=qkdyxi730smdtqlx”.

- **To** - The field contains a name to which the connection is being sent to. Along with this is a URI[5], which is usually in a format similar to an email address. An example of this field is “To: John <sip:john@ru.ac.za>”.

- **From** - This field is very similar to the “To” field but instead of a “branch” parameter, it has a “tag” parameter. An example of this field is “From: Sarah <sip:sarah@uct.ac.za;tag=396265386638>”.

- **Call-ID** - This fields consists of a globally unique identifier consisting of a random

[5] Universal Resource Identifier

string and the caller's name or IP address. An example of this field is "Call-ID: 40133313839303@server99.labserver.ac.za".

- **CSeq** - The Command Sequence field is made up of a random integer along with the method name. An example of this is "CSeq: 284540296836 INVITE".
- **Contact** - This field will contain a SIP URI and will contain either a fully qualified domain name (FQDN) or an IP address. It is used to show the exact route in which the destination server or person is to be reached. An example of this is "<sip:jane@servername.domain.com>".
- **Content-Type** - This field will contain a description of the message content.
- **Content-Length** - This field contains in bytes the size of the message body.

Unlike the previously discussed protocols (TCP, UDP and ICMP), SIP is an application level protocol which is usually encapsulated by a lower level protocol. Although it can be encapsulated in either TCP, UDP or SCTP, it is most often uses UDP. Wustrow *et al.* (2010) observed in their research a high presence of SIP traffic on udp/15206. For a full /8 network range monitored, SIP traffic accounted for 34% of the total packets and nearly 50% of the total data. On further investigation, the SIP traffic was found to be a maliciously crafted SIP invite request. Recent studies (Irwin, 2013) have shown that high counts udp/5060 traffic, the port often used by SIP, have been recorded. Even SIP botnets have been reported in a study by Dainotti *et al.* (2012). The botnet traffic, which ran for approximately 12 days, was analysed and found to have 20 million SIP connections over the period. These connections came from three million unique IP addresses. This botnet was believed to have scanned the full IPv4 address space over the 12 days (Raftopoulos *et al.*, 2015). In the study conducted by Fachkha *et al.* (2012), they analysed a /16 IPv4 darknet data set captured over an eight month period. SIP (udp/5060) accounted for the largest portion of UDP traffic and was also the top application layer protocol according to packet count.

2.6 Traffic Analysis and Classification

The analysis of internet traffic plays a crucial role in understanding how the internet is evolving and what applications are being used most often. This is important knowledge to have when it comes to network design, network management and network security (Caceres *et al.*, 2000). While the use of network telescope data has proven to be useful in identifying worm (Irwin, 2012) and DDOS attacks (Moore *et al.*, 2006), much research has also focused on its use to assist in the identification and classification of internet traffic. Wustrow *et al.* (2010) stated that the type of traffic, protocol and applications were easily observable from their results.

The various SYN, SYN-ACK, ACK and RST flags that are set on the packets are very important in understanding where the packets and connections originated from. Wustrow *et al.* (2010) classifies SYN as scanning traffic, with SYN-ACK, ACK and RST being classified as reflected. With IBR being unidirectional network traffic, it should theoretically only contain the initial SYN packet as there are no active hosts to reply with the SYN-ACK response. According to the 3 way handshake of TCP/IP as illustrated in Figure 2.4, if there are no outgoing SYN connections being made to the Internet from the network telescope, there should be no incoming SYN-ACK or ACK packets observed within the data. However, SYN packets are not the only ones that are found in IBR, as is evident in Figure 2.9.

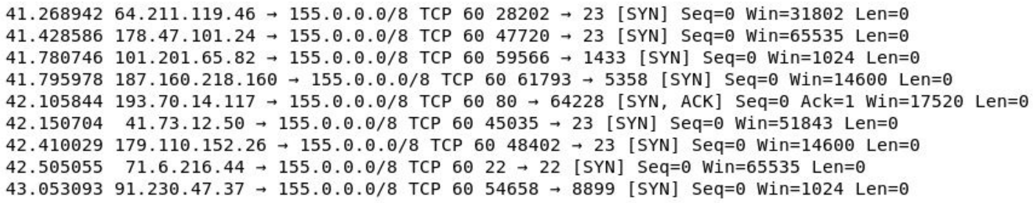
The SYN packet does however contain valuable information which is useful for researchers to study and understand the patterns of traffic flow. Some of the information contained in this traffic is as follows:

- Time and date stamp that the packet arrived
- Source IP Address of the sender
- Destination IP Address of the targeted node
- Lower Level Protocols (TCP, UDP, ICMP)
- Higher Level Protocols (HTTP, SNMP, SIP)
- Source port, which is randomly generated at the time
- Destination port, which coincides with the service being requested (HTTP, SMTP, SSH, HTTPS, etc)
- Other generic info including packet size, sequence number, TCP Flag, etc.

Figure 2.9 is an sample of a TCPDUMP[6] packet capture from a network telescope. It shows the one-way traffic without any reply packets.

The study by Irwin (2011) stated that IBR can be classified into either Active or Passive

[6] https://www.tcpdump.org/tcpdump_man.html



```
41.268942 64.211.119.46 → 155.0.0.0/8 TCP 60 28202 → 23 [SYN] Seq=0 Win=31802 Len=0
41.428586 178.47.101.24 → 155.0.0.0/8 TCP 60 47720 → 23 [SYN] Seq=0 Win=65535 Len=0
41.780746 101.201.65.82 → 155.0.0.0/8 TCP 60 59566 → 1433 [SYN] Seq=0 Win=1024 Len=0
41.795978 187.160.218.160 → 155.0.0.0/8 TCP 60 61793 → 5358 [SYN] Seq=0 Win=14600 Len=0
42.105844 193.70.14.117 → 155.0.0.0/8 TCP 60 80 → 64228 [SYN, ACK] Seq=0 Ack=1 Win=17520 Len=0
42.150704  41.73.12.50 → 155.0.0.0/8 TCP 60 45035 → 23 [SYN] Seq=0 Win=51843 Len=0
42.410029 179.110.152.26 → 155.0.0.0/8 TCP 60 48402 → 23 [SYN] Seq=0 Win=14600 Len=0
42.505055  71.6.216.44 → 155.0.0.0/8 TCP 60 22 → 22 [SYN] Seq=0 Win=65535 Len=0
43.053093 91.230.47.37 → 155.0.0.0/8 TCP 60 54658 → 8899 [SYN] Seq=0 Win=1024 Len=0
```

Figure 2.9: Sample IBR using tcpdump (destination address blinded)

traffic. Active traffic is understood to be that traffic where a legitimate reply is expected from the sender, whereas passive traffic is is that which the sender does not expect a legitimate reply. Active traffic is associated with various network and port scanning activities and passive traffic is associated with denial of service attacks where the source address is spoofed. Ping floods are often carried out as passive attacks, where the sender sends multiple echo requests from a spoofed source IP address.

2.7 Malicious Network Activity

Darknet traffic has effectively been used by researchers to identify and study malicious network activity (Moore *et al.*, 2003; Hick *et al.*, 2009; Irwin, 2013; Antonakakis *et al.*, 2017).

2.7.1 Network and Port Scanning

The acts of network and port scanning are quite similar and only differs in the way the actions are carried out. Both are a form of reconnaissance work, with the end goal possibly being a malicious activity. Scanning can be the work of a human attacker who is specifically targeting a host or network Raftopoulos *et al.* (2015), or it can be the work of a self propagating virus or worm (Yegneswaran *et al.*, 2003).

Network scanning that is done on a large scale can have quite a detrimental effect on a network and its services Raftopoulos *et al.* (2015). An example of this is the case of the *Slammer worm*[7] in 2003. When a host became infected with slammer, it started scanning random IP addresses across the network and the internet, looking for other vulnerable machines. Within three minutes of the host being infected, it started scanning and was

[7] https://en.wikipedia.org/wiki/SQL_Slammer

able to scan over 55 million hosts per second, crippling the network that it was on (Moore *et al.*, 2003).

Port scanning is when the attacker probes an active host to see which ports are responding. Scanning is carried out over TCP, UDP and ICMP echo requests (Durumeric *et al.*, 2013). Once a list of active ports or services are found, the attacker will try and use known vulnerabilities associated with those services to compromise the host (Liu and Fukuda, 2014). One of the most common applications that can do port scanning is NMAP[8]. Port scanning is often detectable in network logs by observing multiple connections to a single host across a large range of ports and protocols. Figure 2.10 depicts a typical port scan.

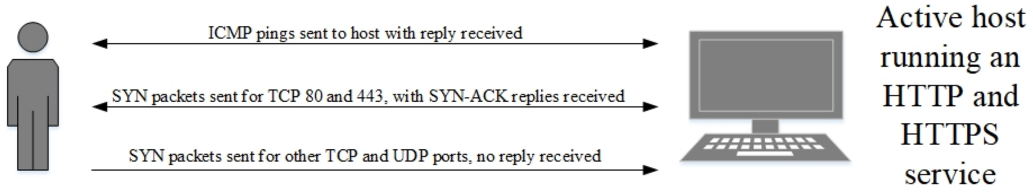

ICMP pings sent to host with reply received
SYN packets sent for TCP 80 and 443, with SYN-ACK replies received
SYN packets sent for other TCP and UDP ports, no reply received
Active host running an HTTP and HTTPS service

Figure 2.10: Typical port scan of an active host

Network scanning on the other hand is generally performed in one of two manners. Firstly, it can take the form of a general reconnaissance of active hosts. This could include the pinging of a range of hosts on a particular network segment with the intention of determining which hosts are active and which are not. ZMap[9], a popular open source network scanner, was used by Adrian *et al.* (2014) to scan the entire IPv4 address space and completed it in under five minutes. The second method of doing network scanning leans a bit towards port scanning. Here the attacker will attempt connections to a range of hosts on a particular network segment on a specific, common network port. For instance, the attacker could try and make HTTP connections to all the IP addresses on a particular network segment in order to determine which of them are running web servers (Liu and Fukuda, 2014).

Previous researchers (Yegneswaran *et al.*, 2003; Barnett and Irwin, 2008) categorises network scanning into four types, as discussed further below:

- **Vertical Scan** - This type of scan is performed on a single host, usually testing five or more ports on that host over a single hour. This scan is usually performed from a single source. The aim of this scan is to determine which vulnerabilities may be present on the host.

[8] https://nmap.org/

[9] https://zmap.io/

- **Horizontal Scan** - This type of scan is performed by a single host or source on five or more targets in the same subnet. The attacker scans these hosts on the same port, looking for the same vulnerability in them.
- **Coordinated Scans** - These are also referred to as *distributed scans* and are performed by multiple hosts on multiple targets. This more aggressive type of scanning targets a number of ports across the hosts on the network segment.
- **Stealth Scans** - As the name suggests, these scans are done is such a way to avoid detection. The targets are scanned at a very low frequency and can be in the form of either a vertical or horizontal scan.

2.7.2 Internet Worms, Viruses and Malware

Computer malware, whether it is a virus, worm, bot or the currently more popular ransomware, is basically malicious code intended to do harm to the system that it is executed on. This harm is often in the form of disrupting the normal operations of the system, damaging the operating system or data (encryption of data via ransomware) or the theft of information. All of them work in similar ways, with worms and ransomware having the ability to self replicate and self propagate across the network or internet to spread and infect other systems. This type of malware will perform network and port scanning to search for a particular vulnerability. Once a system containing the vulnerability has been found, the machine will be attacked and infected.

As the malware scan for target hosts or spread across the internet, they leave a type of fingerprint behind that is easily identifiable by security researchers and practitioners. In mid 2001, Moore *et al.* (2002) were able to study the effects of the *Code Red* worm from the data they captured on their passive network sensors. The passive sensors were configured on unused network segments comprising of a /8 and two /16 networks. Code Red exploited a buffer overflow flaw in Microsoft's IIS[10] server. This made it easy to track as the connections were all coming on TCP port 80. Moore *et al.* (2002) were able to detect 359'000 unique hosts who were all infected with Code Red and who were trying to infect other machines. This information was extracted from a 24hr window within the telescope data. Code Red used a specific algorithm to randomly generate IP addresses which it used to scan for vulnerabilities. Figure 2.11 below shows the rate of infection of Code Red.

[10] Internet Information Services - https://en.wikipedia.org/wiki/Internet_Information_Services

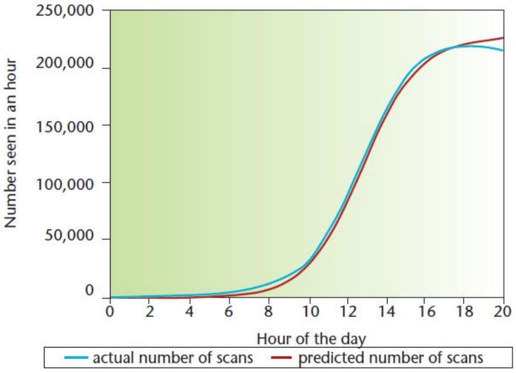


Figure 2.11: Code Red rate of infection (Moore *et al.*, 2003)

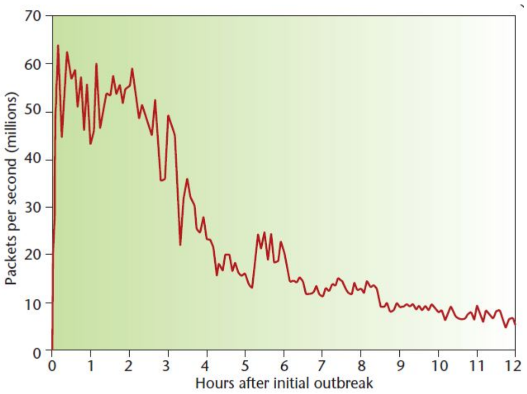


Figure 2.12: Scanning rate of Slammer worm (Moore *et al.*, 2003)

The *Slammer or Sapphire* worm began infecting vulnerable systems in January of 2003. This worm exploited a vulnerability is Microsoft's SQL Server. It spread at a much faster rate than Code Red and at the time was the fastest self replicating and propagating worm known. The worm eventually infected approximately 75 000 hosts, which is much less than Code Red, but had a much greater effect on infected systems. When a machine became infected, within three minutes it would start scanning for vulnerable targets at a rate of 55 million scans per second Moore *et al.* (2003), as seen in Figure 2.12. This caused a huge disruption of services across the internet.

Besides the malware that has previously been discussed, there has been a vast amount of other research done in the area of malware characterisation and distribution, with some of the most prominent studies in this field summarised in Table 2.5.

Table 2.5: Malware research using network telescope data

Malware Name	**Research and Year**
Witty Worm	Shannon and Moore (2004), Weaver *et al.* (2004)
Conficker Worm	Hick *et al.* (2009), Wustrow *et al.* (2010), Irwin (2012), Irwin (2013)
Mirai Botnet	Antonakakis *et al.* (2017), Liu and Fukuda (2018)

2.7.3 Distributed Denial of Service

A Denial of Service (DoS) is a type of cyber attack where the resources of a service gets over utilised, thereby causing an interruption or degradation of the service to legitimate users. In the case of a network DoS, this would take the form of a flood of network packets. When the attack originates from more than one source, it is referred to as a Distributed Denial of Service (DDoS). When the Slammer worm hit the internet in January of 2003, it scanned for vulnerable hosts and spread at such an alarming rate that it caused a global network DDoS. This included certain air flights, ATM's and the American 911 emergency services (Moore *et al.*, 2003). According to Mirkovic and Reiher (2004), a DoS attack can broadly be categorised into one of the following two:

- **Vulnerability Attack** - This type of attack generally exploits a known vulnerability in an application or operating system and allows the attacker to use up all the available computational resources (CPU, memory).

- **Flooding Attack** - This type of attack is related to the network and happens when the attacker floods the network with packets, causing an overload of the bandwidth and network resources.

Moore *et al.* (2006) produced some of the seminal work in the field of DDoS research using network telescopes. They coined the term *backscatter analysis*, which will be discussed below. Pang *et al.* (2004) further builds on the concept of backscatter analysis proposed by Moore *et al.* (2006) and uses it in the analysis of their data.

With the continuous growth of the internet and internet related ecommerce, availability of services is integral to the success of these online businesses. According to Orendorff (2017), the global ecommerce sales market is currently worth about 2.8 trillion USD and expected to grow to 4.5 trillion USD over the next three years. Because most internet based web applications are open to the public and requires no authentication or validation when a connection is made, it makes it very difficult to prevent DoS attacks. Also, there

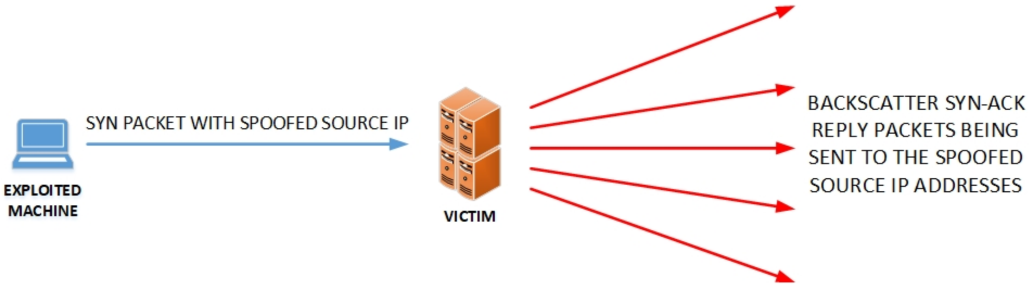

SYN PACKET WITH SPOOFED SOURCE IP
EXPLOITED MACHINE
VICTIM
BACKSCATTER SYN-ACK REPLY PACKETS BEING SENT TO THE SPOOFED SOURCE IP ADDRESSES

Figure 2.13: Backscatter Analysis

is no requirement in place that forces end users or server administrators to implement the latest security patches on their machines, there by leaving them vulnerable to aid in DoS attacks. Although DoS prevention systems have been implemented extensively across the internet, it can be an administrative nightmare to manage and cause the denying of legitimate traffic due to false positive results (Sachdeva *et al.*, 2010).

As mentioned above, there are different methods of performing a DoS attack, but the network flooding method is the most commonly used. When an attacker targets a specific host or number of hosts, it is usually done through some compromised machine. In a DDoS attack, the attacker will use numerous compromised machines to carry out the actual attack. The attacks are performed using a single or a combination of TCP, UDP or ICMP protocols (Sachdeva *et al.*, 2010). The compromised machines would quite often spoof the source address, making each packet appear to be coming from some other third party. Essentially the attacker sends a SYN packet to the victim with a spoofed source address. The victim responds with a SYN-ACK packet to the spoofed address. When these SYN-ACK packets are detected on a network telescope, it is referred to as backscatter (Moore *et al.*, 2006). Figure 2.13 demonstrates how backscatter analysis works.

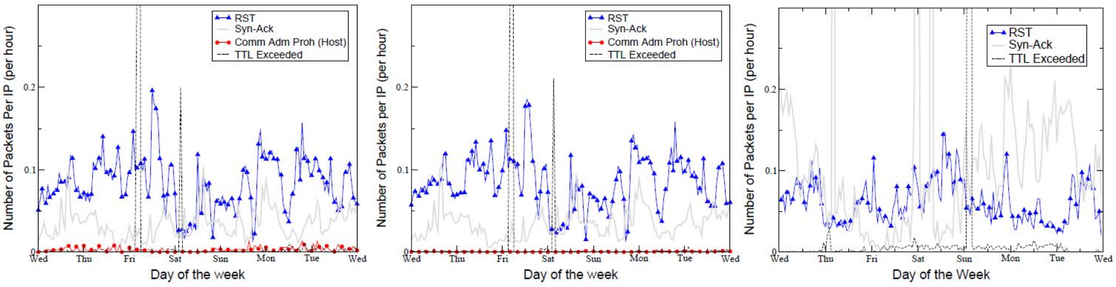
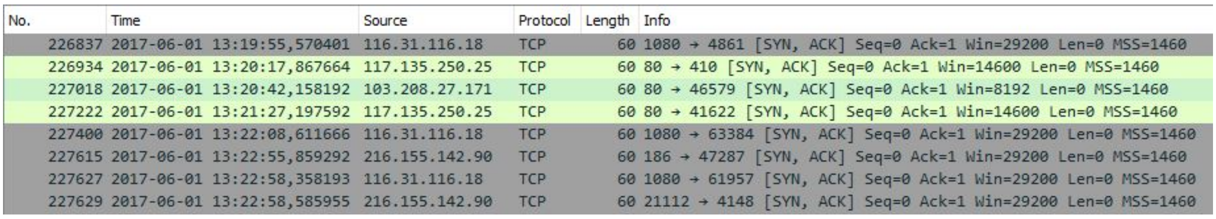


Figure 2.14: Backscatter analysis graphs from three network telescopes (Pang *et al.*, 2004)

Backscatter analysis has successfully been used by Moore *et al.* (2006) to identify the victim of the attack, the duration of the attack and some information regarding the method



No.	Time	Source	Protocol	Length	Info
226837	2017-06-01 13:19:55,570401	116.31.116.18	TCP	60	1080 → 4861 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
226934	2017-06-01 13:20:17,867664	117.135.250.25	TCP	60	80 → 410 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
227018	2017-06-01 13:20:42,158192	103.208.27.171	TCP	60	80 → 46579 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
227222	2017-06-01 13:21:27,197592	117.135.250.25	TCP	60	80 → 41622 [SYN, ACK] Seq=0 Ack=1 Win=14600 Len=0 MSS=1460
227400	2017-06-01 13:22:08,611666	116.31.116.18	TCP	60	1080 → 63384 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
227615	2017-06-01 13:22:55,859292	216.155.142.90	TCP	60	186 → 47287 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
227627	2017-06-01 13:22:58,358193	116.31.116.18	TCP	60	1080 → 61957 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460
227629	2017-06-01 13:22:58,585955	216.155.142.90	TCP	60	21112 → 4148 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460

Figure 2.15: Sample of SYN-ACK packets viewed in Wireshark

of the attack. Pang *et al.* (2004) was able to graph the backscatter by measuring the SYN-ACK and TCP RESET packets observed on their network telescope. Figure 2.14 shows the backscatter graphs taken from three of their network telescopes. Figure 2.15 shows a sample of SYN-ACK packets from a network telescope viewed in Wireshark.

2.8 Summary

This chapter took a look at the previous research in the field of IBR, with specific focus on work that was applicable to this study. The chapter starts by providing background and context to the use of network telescopes and what internet background radiation is. It describes the the classification of different types of telescopes and how the size of a telescope affects results. The chapter then continues with a detailed description of the TCP/IP suite and the lower and higher level protocols associated with it, including the analysis and classification of traffic. SIP is discussed in detail, as it is a major contributor to IBR in many studies, including this research. Lastly, the chapter concludes by discussing the malicious network traffic that is detectable within IBR. The malicious traffic section covers port and network scanning, various types of malware activity and distributed denial of service attacks.

CHAPTER 3

Dataset, Tools and Analysis Method

This chapter discusses the dataset, tools and analysis methods which were used in the course of conducting this research. Section 3.1 starts by discussing the source and structure of the data, the data collection method used by the network telescope and lastly, explains the reasons for using the particular dataset. Section 3.2 provides a brief overview of the data and some basic results from the initial analysis performed. Section 3.3 covers the tools used in the analysis and filtering of the data, the storage of the data and the tools used for the presentation and visualisation of results. Although there were many options available for the data analysis in this research, certain methods and tools were preferred over others. Some factors that required consideration were the large size of the dataset, the ease and speed of the data analysis, the time and costs associated with certain methods or tools and the end visualisation and presentation of the information. The methods and tools used by previous researchers dealing with network telescopes were investigated and tested before the final decision was taken of which to use. Section 3.5 summarises the chapter and creates the platform for the Results and Discussion which follows in Chapter 4.

3.1 Dataset

Data quality is an important aspect of academic research and will directly impact the value of the information produced in the results. It is therefor important to ensure that the data is from a reliable source, captured and stored in the correct format and has not been altered. The data for this research was collected using an internet facing passive network sensor, also known as a network telescope. As previously discussed in Section 2.1 and illustrated in Figure 3.1, IBR is essentially one-way traffic towards the network telescope with no responses or replies sent. The network telescope captured the traffic from a /24 IPv4 network segment, which contained no active hosts or services. All incoming network packets were stored in a packet capture (pcap) file, one for each month of the year. No outages of the network telescope, internet link or capturing of the data were experienced for the time period of the dataset, hence no data was lost.

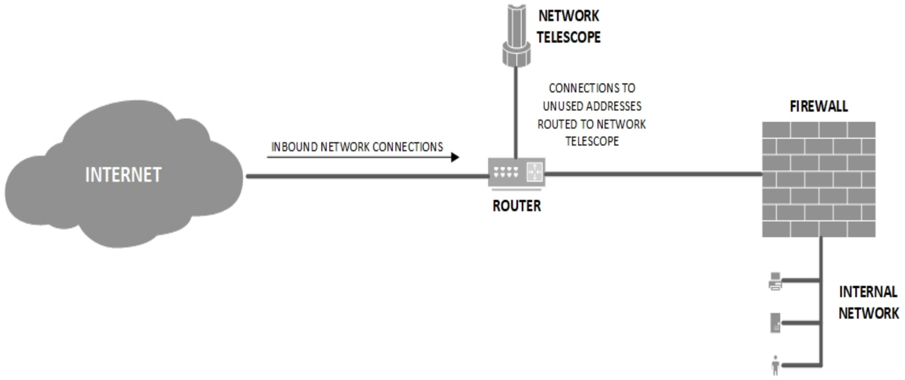

NETWORK TELESCOPE
CONNECTIONS TO UNUSED ADDRESSES ROUTED TO NETWORK TELESCOPE
FIREWALL
INBOUND NETWORK CONNECTIONS
INTERNET
ROUTER
INTERNAL NETWORK

Figure 3.1: Basic design for a network telescope setup

3.1.1 Source of the Dataset

The data for this research was provided by the Computer Science Department at Rhodes University, where they have been collecting network telescope data for many years. The internet services and IP address ranges are provided to Rhodes University by TENET[1]. Within the IP address ranges assigned to Rhodes, they operate five network telescopes, each monitoring its own contiguous /24 IPv4 subnet (256 addresses). Each of the five /24 address subnets are however non-contiguous to each other. The IP ranges are all internet routed and completely unused. Although these telescopes monitor a fairly small address space in comparison to much larger ones such as the /8 (2^{24} addresses) telescope

[1] TENET is the Tertiary Education and Research Network of South Africa. It provides internet connectivity services to South African universities and research facilities. https://www.tenet.ac.za/

run by CAIDA[2], they still provide a significant insight into IBR within the context of a South African IPv4 address space. The size of the network telescope, or the size of the IP address range being monitored was previously discussed in Section 2.2. The first sensor was established in August 2005, running on a 196/8 subnet. Four years later, a second sensor was implemented, capturing traffic on a 146/8 address range. These were the main Rhodes sources of data for the research done by Irwin (2011). In 2011, a further three sensors were implemented, namely the 155/8 and two within the 196/8 address ranges. Irwin (2013), Nkhumeleni (2014) and Irwin and Nkhumeleni (2015b) did a full correlation and comparative analysis of data from these five network telescopes.

3.1.2 Collection of Data

The network telescope sensors at Rhodes University are located in the primary data centre of the university and connected directly to the router located before the firewall, per Figure 3.1. The sensors are essentially computer hardware running FreeBSD 11, with various *tcpdump*[3] scripts used to capture incoming packets and write their contents to a file. These files are then moved to the internal campus network and stored on a redundant NAS infrastructure for archiving and further analysis.

3.1.3 Selection of Dataset

The data from the 155-x/24 sensor was chosen for this research, as no significant analysis work had been done on data from this sensor before. This network telescope was one of the last three to be setup at Rhodes, with it being the only one in the 155/8 range. The other two are in in the 196/8 ranges. The dataset was also one of the largest out of all the sensors and was expected to produce some interesting findings.

3.2 Data Overview

The analysis of the data was performed in two phases, firstly at a macro level and later, to get more depth and understanding of specific occurrences and events, at a micro level. The full year's data was firstly looked at to get a good overview of the traffic captured

[2] The Center for Applied Internet Data Analysis manages a large network telescope located at the University of California San Diego

[3] The defacto standard across platforms for the capture of network traffic. https://www.tcpdump.org/

in 2017. The analysis then moved on to a breakdown of monthly statistics, as compared to the full year's results. For the full duration of 2017, there was a total of 174 043 845 packets received, coming from 13 271 685 distinct IPv4 addresses and accounting for a total of 12.96 GB of traffic. Table 3.1 shows the comparative statistics of traffic for each month of 2017. January experienced the highest count for both packets and data in 2017, with the lowest count for packets occurring in September and the lowest volume for data occurring in June.

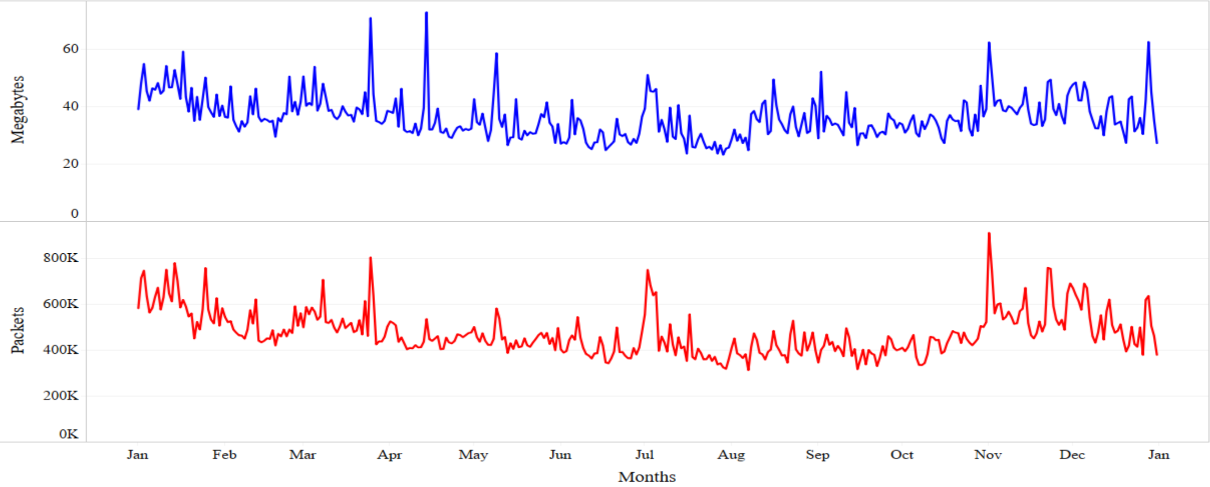


Figure 3.2: Traffic (pkts) and Data (MB) graphs for the year

Considering the total number of distinct source IP addresses recorded in 2017, the highest occurrence of 18.93% was in January (2 512 778), with the lowest of 5.93% seen in July (787 558). The graphs in Figure 3.2 show the differences in traffic patterns between packets received and data received over the time period. There is a fair consistency between the two graphs, with only few noticeable differences where total packet count versus total data transferred were higher or lower for the same time slice. Again, this is typically due to the TCP and UDP density in the data and how the protocols differ in their structure and packet size. The highest packet count per day is recorded in early November, with the lowest in early to mid August. On the contrary, the highest data count recorded for a day was in mid April. The network telescope received the highest recorded packet and data counts in January 2017, with the lowest recorded in June 2017. A more detailed analysis of the packet and data differences is performed in Section 4.5.

Statistics regarding the packet and data flow for 2017 and each individual month can be seen in Table 3.2. We see that the annual average data bit rate for incoming packets was

[4] Note that this is not the sum of the above rows as the IP's may be duplicated due to it appearing in multiple months

Table 3.1: Monthly comparison of traffic statistics for 2017 (155/8 sensor)

Month	Packets	% Packets	Data (GB)	% Data	Unique Source IP's
January	18 899 575	10.86	1.35	10.41	2 512 778
February	13 783 654	7.92	1.02	7.89	1 731 015
March	16 457 707	9.46	1.23	9.52	1 981 145
April	13 508 923	7.76	1.02	7.91	1 527 613
May	13 958 453	8.02	1.04	8.05	1 344 735
June	12 260 718	7.04	0.87	6.74	933 003
July	13 519 873	7.77	0.96	7.39	787 558
August	12 712 731	7.30	1.05	8.07	987 983
September	12 031 381	6.91	0.99	7.64	828 519
October	13 437 559	7.72	1.05	8.11	906 205
November	17 421 645	10.01	1.20	9.24	1 361 664
December	16 051 626	9.22	1.17	9.02	1 392 472
2017 Total	174 043 845	100	12.96	100	13 271 685[4]

3531 bits/sec, whilst the average packet size was 79.98 bytes and the average packet rate was 5 packets/sec. When comparing the average packet rates to previous research, Bailey *et al.* (2006) reports in their study that the sustained average packet rate for a /24 sensor was 9 packets/sec over a 2.5 year period, with spikes reaching as high as 290 packets/sec. The packet rate drastically increases when the network telescope size does. They reported that the average packet rate for a /16 telescope was approximately 75 packets/sec and a /8 increased to 5000 packets/sec. An interesting observation is that the top three months for highest data bit rate is also the top three for average packet rate. These are January, November and March respectively. However, the top three months experiencing the highest average packet size are September, August and October respectively. This is possibly due to the differences between TCP and UDP packets, which will be discussed in more detail in Section 4.5. As previously stated in Section 2.3, an analysis of internet traffic would usually include a comparison of TCP, UDP and ICMP traffic. Table 2.1 showed the comparison of these traffic types in previous research. The results of this research showed very similar results, with TCP, UDP and ICMP accounting for 89.70%, 9.52% and 0.77% respectively.

Table 3.2: Data and packet flow statistics by month

	Data Bit Rate (bits/sec)	**Average Packet Size (bytes)**	**Average Packet Rate (pkts/sec)**
January	4329	76.70	7
February	3632	79.69	5
March	3955	80.48	6
April	3396	81.47	5
May	3347	80.29	5
June	2895	76.52	4
July	3072	76.09	5
August	3357	88.41	4
September	3283	88.44	4
October	3373	84.05	5
November	3969	73.82	6
December	3749	78.21	5
2017 Average	3531	79.98	5

3.3 Tools and Analysis Methods

Many things needed to be considered before the tools and analysis methods were decided on. Although many commercial network analysis tools were available, many of them were quite expensive and a vast majority were geared towards analysing bidirectional network traffic flow. Open-source and free tools were therefor preferred, as many of them also gave the option of using customised scripts. After careful consideration, it was decided that the best option was to load all data into a relational database and then use an analysis or business intelligence tool to query the data. The methods and tools used are discussed further in the next sections.

3.3.1 System Software and Hardware

The data set consisted of over 174 millions packets. This meant that the relational database would also have over 174 million rows. This is a significant growth in data when compared to the 23 million packets collected over a period of five years, in the study conducted by Irwin (2011). The analysis or business intelligence tools would therefor require substantial resources to query and analyse the database. A system with high level hardware specifications was required that was able to support all the applications that were needed. A dual boot system was setup which ran Microsoft Windows 10 Enterprise 64 bit and Kali Linux 64 bit. The dual boot option was preferred over running a virtual machine as it allowed the operating system full access to all the resources available. The system was powered by an Intel Core i5-8600K hexa-core processor and had 32 gigabytes of memory. The data was stored on an internal Seagate four terabyte drive. Each operating system booted off its own Samsung 250 gigabyte solid state drive. The Windows operating system was used to run the relational database and BI tool, with the Linux operating system used to run Wireshark, tshark, shell scripts and other Linux tools.

3.3.2 Libpcap

The dataset was obtained in the form of 12 libpcap[5] files, one for each month of the year. The first packet captured in the dataset was at 00:00:00.028 UTC+2 on January 1st 2017 and the last one captured was at 23:59:59.883 UTC+2 on December 31st 2017. Details of the files can be seen in Table 3.3 and Appendix A.1.

Table 3.3: Statistics for libpcap data files

Month	**File Size (MB)**	**# Packets**	**First Packet Received**	**Last Packet Received**
January	1670	18 899 575	01-01-17 00:00:00.0282	31-01-17 23:59:59.8847
February	1257	13 783 654	01-02-17 00:00:00.1025	28-02-17 23:59:59.9409
March	1514	16 457 707	01-03-17 00:00:00.2443	31-03-17 23:59:59.7586
April	1255	13 508 923	01-04-17 00:00:00.0701	30-04-17 23:59:59.8893
May	1281	13 958 453	01-05-17 00:00:00.4649	31-05-17 23:59:59.9977
June	1081	12 260 718	01-06-17 00:00:00.3969	30-06-17 23:59:59.8844
July	1187	13 519 873	01-07-17 00:00:00.1042	31-07-17 23:59:59.2358
August	1265	12 712 731	01-08-17 00:00:00.2172	31-08-17 23:59:59.7010
September	1198	12 031 381	01-09-17 00:00:00.4976	30-09-17 23:59:59.7768
October	1282	13 437 559	01-10-17 00:00:00.3514	31-10-17 23:59:59.9546
November	1492	17 421 645	01-11-17 00:00:00.1727	30-11-17 23:59:59.8005
December	1442	16 051 626	01-12-17 00:00:00.0413	31-12-17 23:59:59.8837

Libpcap files are binary and requires an application to interpret the packet capture. Each libpcap file ranged between 1.1 GB and 1.7 GB. One of the most popular tools

[5] A Unix implementation of the packet capture library. http://http://www.tcpdump.org/

for analysing network traffic is Wireshark[6], which is available for Windows, Linux and Mac. Due to the size of the libpcap files, it was impossible to directly use any graphical tools to properly analyse any data, as this proved to be too resource intensive, particularly on RAM. The Wireshark package however provides many command line tools to do the same tasks that the graphical tool does. These tools are commonly installed with Wireshark. The command line tools used are listed below, along with a brief description of their functionality:

- **tshark** - A powerful command line equivalent of Wireshark
- **capinfos** - Reads the libpcap file and displays statistics about the captured data
- **mergecap** - Used to merge two or more libpcap files into one
- **rawshark** - Analyses the raw libpcap data

Capinfos was used to extract traffic statistics for each libpcap file, including the data flow rate, average packet size and the average packet rate. *Mergecap* was used to merge all libpcap files into a singular file and then capinfos was used to extract the statistics for the entire 2017.

It was important to firstly get an understanding of the data, its properties and all the fields that had been captured. Wireshark was used to get an overall view of the data and to do the initial searches. Wireshark also provides a very comprehensive graphical view of each packet, thereby allowing one to do an in-depth analysis of an individual packet. The various fields were closely studied to understand what information was required from them.

The monthly libpcap files were individually exported to CSV[7] files using a combination of *bash scripts* and *tshark*, details of which can be seen in Appendix A below. When extracting data with tshark, there are many options available for the format of the output file, including JSON[8] and CSV. As the intention was to eventually have all the data in a relational database, the simplest option was to export to CSV format and then import the CSV files into the database. Tshark allows the option of specifying the individual *filter fields* that are required when exporting from libpcap format. Only certain data fields were exported, as to not create too large a database with too many columns. The dataset already contained over 174 million entries, meaning the database would have that same amount of rows. Limiting the exported data to only the relevant fields would therefor

[6] An open source graphical network analyser. https://www.wireshark.org/

[7] A plain text file with each value separated by a comma

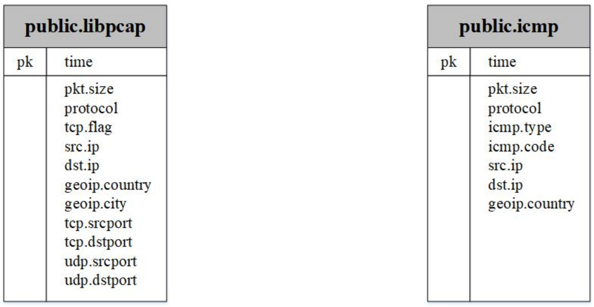
[8] A JavaScript Object Notation plan text file

cut down on unused data, thereby making the analysis much quicker and easier. The relevance of the various data fields were evaluated to see which impact they could have on the analysis and how they contributed to achieving the objectives of this research. As an example, each packet would have data fields that are not specifically required, such as the ethernet source and ethernet destination addresses. These ethernet source and destination addresses are essentially the hardware or MAC[9] addresses, which has no relevance to the research as each packet would have the same ones. That is because the ethernet source address would be the router's interface address and the ethernet destination address would be the network card of the telescope's address. There are many other "irrelevant" fields that were excluded from the export similarly. Two sets of CSV files were created. The first set contained the full complement of packets, all 174 million of them. The second set of files only contained ICMP packets. The reason that the second set of CSV files were created is because ICMP comprised the lowest volume of traffic, it had header fields found in only ICMP packets (icmp.type and icmp.code) and did not contain certain header fields that are found in TCP and UDP packets (tcp.srcport, tcp.dstport, udp.srcport and udp.dstport). Having a smaller data set allowed for faster analysis of the ICMP traffic. A list of all the relevant exported data fields and their descriptions can be seen in Table 3.4.

Table 3.4: Tshark filter fields and their corresponding descriptions

Tshark Field	**Description**
frame.time_epoch	Time and date that the packet arrive in *epoch* format. Epoch is a unix format used to display date and time. The timestamp refers the amount of seconds that have passed since midnight of 1st January 1970
ip.src	Source IP address
ip.dst	Destination IP address
ip.proto	Upper layer protocol
ip.geoip.src_country	Country of the source packet, based on geolocation data provided by MaxMind Lite, which is supported by Wireshark and tshark
tcp.srcport	TCP source port
tcp.dstport	TCP destination port
upd.srcport	UDP source port
upd.dstport	UDP destination port
icmp.type	ICMP Type field
icmp.code	ICMP Code field
frame.len	Size of the packet on the wire, including headers
tcp.flags	TCP flag field of TCP packet

[9] The MAC (Media Access Control) address is a unique hardware identifier address



public.libpcap	
pk	time
	pkt.size protocol tcp.flag src.ip dst.ip geoip.country geoip.city tcp.srcport tcp.dstport udp.srcport udp.dstport

(a) All packets

public.icmp	
pk	time
	pkt.size protocol icmp.type icmp.code src.ip dst.ip geoip.country

(b) ICMP packets only

Figure 3.3: Database table structures in PostgreSQL

3.3.3 Storage and Data Query

As previously mentioned in Section 3.3.2, the libpcap files were binary, but in order to do proper comparative graphical analysis of the data, it needed to be in a plain text format. The analysis tool, which is discussed in the next section, supported a large variety of data input methods, of which were various relational databases. PostgreSQL was used because of its native support for IPv4 addresses as a data type field, its ability to run on both Windows and Linux platforms and because it is open-source and free to use. The CSV files were imported into a PostgreSQL[10] database. pgAdmin 4[11] was used to do all the initial SQL query development, view the data and to create specific queries and views. Two tables were created in PostgreSQL, one which contained all the packets and the second which contained only the ICMP packets. Initially it was decided to create a single table to house all the data as there was no need to have a complicated structure with multiple tables with joins and duplicated data. Later, a second table was created to house all the ICMP packets. No joins were created between the tables and they operated independently of each other. The structure of the database tables can be seen in Figure 3.3, with the epoch time as the primary keys. The SQL creation scripts can be seen in Appendix A.1.

[10] An open source object-relational database. https://www.postgresql.org/

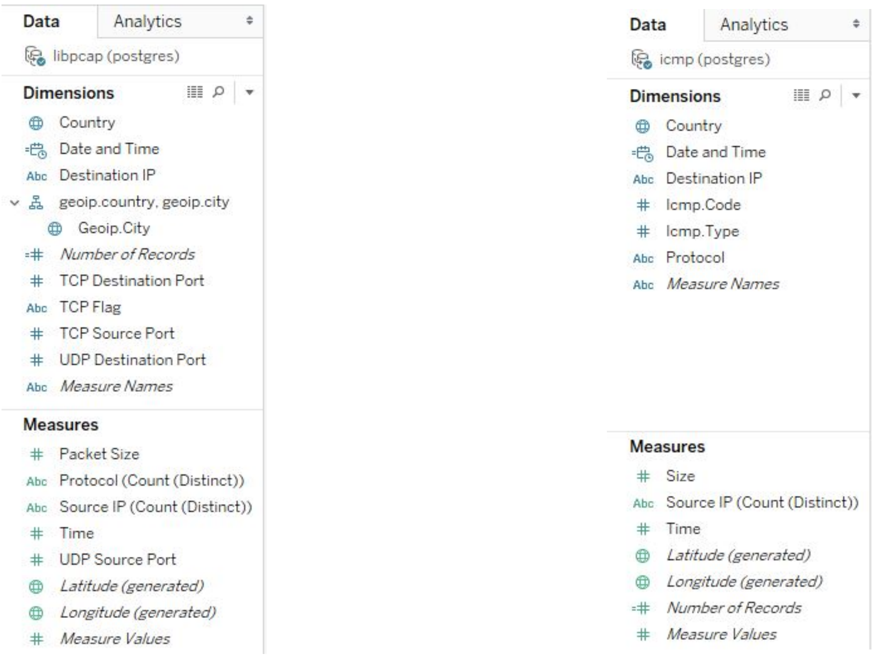
[11] pgAdmin is an open-source tool for the administration and development on PostgreSQL. https://www.pgadmin.org/

3.3.4 Analysis and Presentation of Results

Tableau Desktop[12] 10.5 was used to connect to the database and analyse the data. The data was cleaned up and aliases assigned to specific values and fields in Tableau. The epoch date was converted to a human readable date and time value using the following SQL statement within the Tableau environment:

```
DATEADD('hour',2,(Date("1/1/1970") + ([Time]/86400)))
```

Converting the epoch date to a proper date and time allowed for specific month, day, hour and minute instances to be defined for the analyses of events. All data analysis and visualisations were created with Tableau. Although Tableau is a commercial product, a 12 month full-access licence was available at the time of writing this, to registered students and researchers. Tableau interprets the input data and separates the values into either *dimensions* or *measures*. One is however able to interchange the values into measures and dimensions as required. Figure 3.4 shows the values within Tableau.



libpcap (postgres)

Dimensions
Country
Date and Time
Abc Destination IP
geoip.country, geoip.city
Geoip.City
Number of Records
TCP Destination Port
Abc TCP Flag
TCP Source Port
UDP Destination Port
Abc *Measure Names*

Measures
Packet Size
Abc Protocol (Count (Distinct))
Abc Source IP (Count (Distinct))
Time
UDP Source Port
Latitude (generated)
Longitude (generated)
Measure Values

(a) Data for all packets

icmp (postgres)

Dimensions
Country
Date and Time
Abc Destination IP
Icmp.Code
Icmp.Type
Abc Protocol
Abc *Measure Names*

Measures
Size
Abc Source IP (Count (Distinct))
Time
Latitude (generated)
Longitude (generated)
Number of Records
Measure Values

(b) Data for only ICMP packets

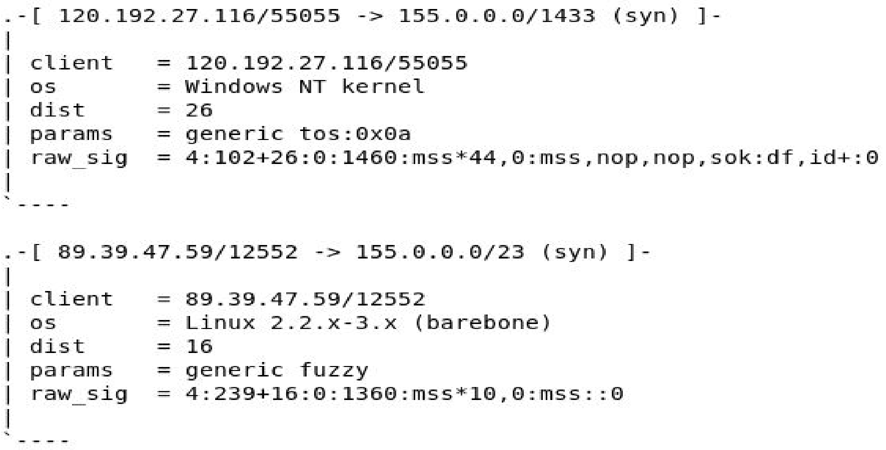
Figure 3.4: Tableau interpretation of the input data, showing dimensions and measures

[12] Tableau is a business intelligence tool used for the analysis and visualisation of data. https://www.tableau.com

3.3.5 Passive OS Fingerprinting

Passive operating system fingerprinting is a technique used to identify the operating system that sent a packet. This is done by looking at certain TCP/IP field values of the packet, which are unique to certain operating systems due to the way the TCP/IP stack is implemented by them. Some of the fields that are used to identify the operating system are the initial packet size, initial time to live (TTL) and the window size.

Two applications were used to perform the passive fingerprinting, namely *P0f*[13] and *NetworkMiner*[14]. P0f is an open source command line tool which reads in the libpcap file and outputs the findings based on the analysis. It passively identifies the operating systems based on a signature file provided with the software. It does not alter the pcap file at all. When running P0f, it will output the results to standard output or it can be sent to a text file for further viewing. P0f was run on Linux, as it is commonly installed as standard along with Kali Linux. NetworkMiner is a graphical open source tool used in network forensics. It can also nativelyz read pcap files and do an analysis on it. There is a free version which has limited capabilities, with passive OS fingerprinting being included in the free version. There is also a paid for professional version, which has much more network forensics capabilities. The free version of NetworkMiner 2.3.2 was used in this research. Figure 3.5 shows the typical output of p0f.



```
.-[ 120.192.27.116/55055 -> 155.0.0.0/1433 (syn) ]-
|
| client   = 120.192.27.116/55055
| os       = Windows NT kernel
| dist     = 26
| params   = generic tos:0x0a
| raw_sig  = 4:102+26:0:1460:mss*44,0:mss,nop,nop,sok:df,id+:0
|
`----

.-[ 89.39.47.59/12552 -> 155.0.0.0/23 (syn) ]-
|
| client   = 89.39.47.59/12552
| os       = Linux 2.2.x-3.x (barebone)
| dist     = 16
| params   = generic fuzzy
| raw_sig  = 4:239+16:0:1360:mss*10,0:mss::0
|
`----
```

Figure 3.5: Typical p0f output

Both applications were used to analyse traffic suspected of being malware related, specifically Mirai related. This is discussed in more detail in Section 5.3.

[13] http://lcamtuf.coredump.cx/p0f3/

[14] https://www.netresec.com/?page=NetworkMiner

3.4 IP Address Geolocation

The global IP addresses are co-ordinated by and allocated by IANA[15]. They in turn assign blocks to the various RIR's (Regional Internet Registries), who assign the IP addresses or blocks to ISP's and organisations. There are five RIR's that each manage a particular region or regions as follows:

- **ARIN** - Canada, United States and certain Caribbean Islands
- **LACNIC** - Latin America and certain Caribbean Islands
- **RIPE NCC** - Europe, Middles East and Central Asia
- **AFRINIC** - Africa
- **APNIC** - Asia and Pacific

As IP addresses are allocated, the country and city that they are allocated to are recorded. This makes it possible to track IP addresses to the country and regions where they originated from. There are many commercial software applications available that are able to provide these details. Wireshark and tshark are able to use the databases provided by Maxmind[16]. The free Lite version was used in this research, which allowed Wireshark and tshark to associate each IP address with its corresponding source country. Maxmind also produces a paid for version which is updated often and is much more accurate than the Lite version.

[15] IANA is the Internet Assigned Numbers Authority - https://www.iana.org

[16] MaxMind maintains a GeoLocation database, based on IP block information assigned to each country - https://www.maxmind.com

3.5 Summary

The chapter introduced the research project and provided the details of the data set. It covered the source of the data set, the structure of the raw data and how the information was extracted and analysed. It provided the an initial overview of the data, including the size of the data set, the amount of packets contained in the data and the count of unique IP addresses responsible for the traffic. A breakdown of the data per month is provided, including the usage in gigabytes.

The chapter continued by describing the tools that were used to store, query and analyse the data. Due to the size of the data set, it was important to consider the tools and analysis methods used. There needed to be a balance between the functionality of the tools used and the speed and accessibility of using it. The option of open source or free tools were always preferred, although knowledge of its usage and functionality was quite important as well. The data was loaded into a relational database once it was extracted from the raw data files. Loading data into a database has the accessibility advantage in that many applications are able to connect to it and access it. There is also the advantage that a database is able to export data to various formats if required. Tableau was the main tool used for both analysis and presentation of results. It is a really powerful tool allowing for SQL type queries of data and has a vast array of predefined graphs, bars and plots for the visualisation of results.

The analysis had a very repetitive approach in that when the first round of analysis found interesting results, a deeper or further analysis then ensued. This allowed for varied levels of interpretation of the findings. This was the very approach taken next. This chapter provided a basis and platform for the further analysis, which was then carried out in Chapters 4.

CHAPTER 4

Results and Discussion

Chapter 4 presents the overall results of the analysis and research and continues the discussion which was started in the previous chapter. Chapter 3 introduced some of the basic results of the analysis and described the methods used. In this chapter, the analysis methods are applied and the findings are discussed.

Section 4.1 discusses the statistics and metrics derived from the analysis. It covers a detailed discussion of the protocols and ports used, a comparison of traffic patterns based on TCP flags over time, a deeper analysis and geolocation plotting of the source IP addresses responsible for the traffic and a comparison between traffic and data usage patterns over time. Section 4.5 compares the traffic patterns of the packets and data per month in 2017. Notable variances in the traffic patterns were then further analysed and discussed. Section 4.2.1 performed an analysis on only the TCP traffic and categorised it based on the TCP flag set on each packet. The next two sections analysed the IP addresses in the traffic. Section 4.6 further analyses the reflected traffic and attempts to identify the sources of the traffic. Section 4.7 looks at the source IP addresses and Section 4.8 looks at the destination IP addresses. Lastly, the chapter is summarised and concluded in Section 4.9.

4.1 Introduction

The analysis commenced with a look at the traffic composition based on the main IP level protocols. TCP accounted for the largest portion of both packets and data, with UDP in at 2^{nd} and ICMP in at 3^{rd}. Figure 4.1 illustrates the traffic patterns for these protocols for packet count and data usage. Table 4.1 shows that TCP accounted for 89.70% of the total traffic in 2017, with UDP and ICMP accounting for 9.52% and 0.77% of the overall traffic respectively. These three figures combined to make up 99.99% of the total traffic. When comparing this to previous studies done (Pang *et al.*, 2004; Irwin, 2011; Yates, 2014), similar results were found, as depicted in Table 2.1. TCP always formed the largest portion of traffic, with UDP and ICMP coming in at second and third respectively. In a study conducted by Wustrow *et al.* (2010), similar results were observed. Their study looked at five year's worth of darknet traffic from 2006 to 2010. They noted a consistently gradual increase in percentage of TCP packets over the years, except for 2008 where the percentage of UDP packets was higher. They attributed this to the re-emergence of SQL Slammer in 2008, which scanned internet hosts on udp/1434, causing very large UDP traffic spikes Chindipha and Irwin (2017).

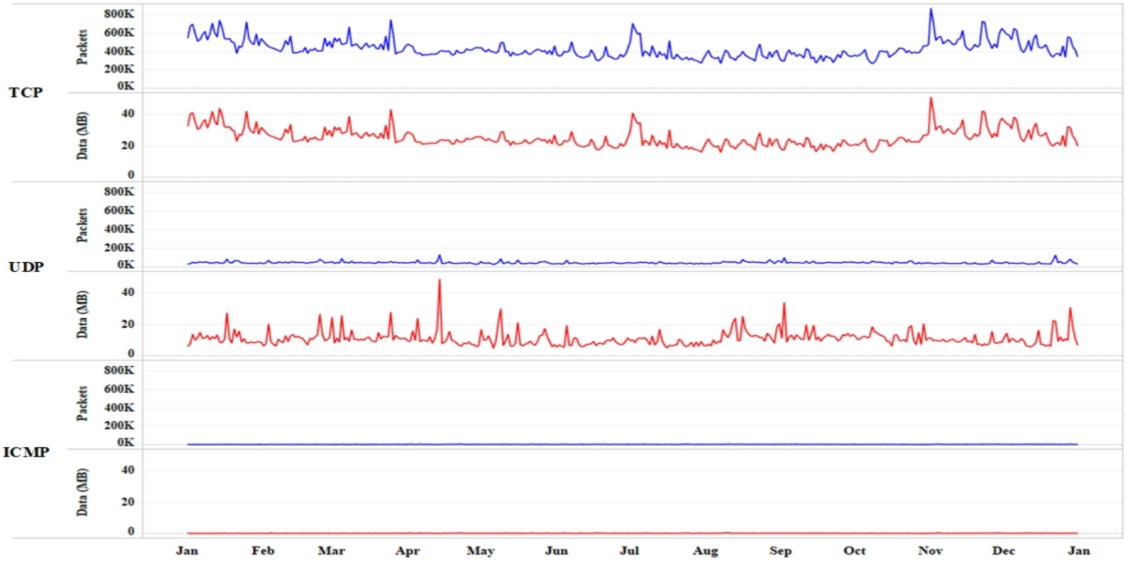


Figure 4.1: Packet and data graphs for TCP, UDP and ICMP

An analysis of the IP level protocols gives a directional indication as to what further analysis needs to be done and where to look. As an example, Figure 4.1 shows that UDP traffic is quite minuscule in comparison to TCP, however, the data graph for UDP is surprising high. This is an indication that the UDP packets are much larger than the TCP packets and warrants further investigation. As UDP is a stateless protocol and does

Table 4.1: Comparison between TCP, UDP and ICMP traffic

	All Packets	**TCP**		**UDP**		**ICMP**	
January	18 899 575	17 369 212	91.90 %	1 444 963	7.65 %	85 363	0.45 %
February	13 783 654	12 313 975	89.34 %	1 388 328	10.07 %	81 098	0.59 %
March	16 457 707	14 867 057	90.33 %	1 496 873	9.10 %	93 759	0.57 %
April	13 508 923	12 008 571	88.89 %	1 376 654	10.19 %	123 478	0.91 %
May	13 958 453	12 477 660	89.39 %	1 364 676	9.78 %	116 115	0.83 %
June	12 260 718	10 937 899	89.21 %	1 207 728	9.85 %	114 482	0.93 %
July	13 519 873	12 134 470	89.75 %	1 264 230	9.35 %	118 352	0.88 %
August	12 712 731	11 022 876	86.71 %	1 558 129	12.26 %	129 877	1.02 %
September	12 031 381	10 458 158	86.92 %	1 451 467	12.06 %	121 754	1.01 %
October	13 437 559	11 907 526	88.61 %	1 432 625	10.66 %	97 405	0.72 %
November	17 421 645	16 115 210	92.50 %	1 176 609	6.75 %	129 150	0.74 %
December	16 051 626	14 510 876	90.40 %	1 406 154	8.76 %	134 589	0.84 %
2017 Total	174 043 845	156 123 490	89.70 %	16 568 436	9.52 %	1 345 422	0.77 %

not require a complicated 3 way handshake to establish a connection, the initial UDP packets can be sent along with a payload. Over the following sections, each of the above mentioned protocols will be further analysed and discussed.

4.2 TCP

TCP packets accounted for the largest portion of traffic, accounting for nearly 90% of the total traffic for 2017. A total of 156 123 490 packets were received through the year, coming from 13 087 271 unique source IP addresses. January saw the highest volume of both total traffic and TCP traffic, with the lowest experienced in September.

Table 4.2 further analyses the TCP traffic and shows the top 10 most prominent TCP destination ports being targeted. The average packet size for traffic targeting the top 10 destination ports is 61.21 bytes. As previously discussed in Section 2.3.1, TCP requires a 3 way handshake before a connection is established. If no connection is made, no payload can be seen in the traffic, hence the small size of the packets. Telnet, a client-server protocol running on TCP port 23, makes up the largest portion of TCP traffic. It accounts for 39.56% of TCP traffic and 35.48% of the total data for 2017. Telnet is commonly used to remotely connect to servers, network devices and a large range of Smart IoT devices

Table 4.2: Top 10 TCP destination ports

Rank	Dest Port	Common Usage	# Packets	% Packets	Unique IP's	Data (MB)	Avg Pkt Size (bytes)
1	23	Telnet	61 764 103	39.56	9 017 289	3 591	60.97
2	22	Secure Shell	10 123 059	6.48	1 347 710	599	62.03
3	1433	MS SQL Server	9 316 775	5.97	72 803	536	60.31
4	2323	Telnet	4 822 170	3.09	1 383 786	276	60.21
5	5358	WSD	4 342 646	2.78	852 774	249	60.00
6	7547	Router Exploit	3 385 454	2.17	1 361 100	194	60.02
7	3389	MS RDP	2 233 794	1.43	54 363	132	61.88
8	445	SMB	2 167 062	1.39	388 850	130	63.07
9	80	HTTP	1 991 465	1.28	117 377	117	62.15
10	3128	Web Proxy	1 747 084	1.12	1 884	101	61.48
Σ			101 893 612	65.27	14 596 052	5925	61.21

via a command line interface. Telnet is unencrypted, with all communication being sent across the wire in plain text, including login usernames and passwords. As noted by Pa *et al.* (2015), many IoT devices have been targeted by telnet botnet attacks over the years. In 2012 the Carna botnet attack targeted more than 1.2 million IoT devices that had either no username or password set, or had a simple default username and password set. In 2017 the Mirai botnet also targeted IoT devices and broadband routers using telnet running on tcp/23 and tcp/2323 (Antonakakis *et al.*, 2017). Second from the top is SSH (Secure Shell), which generally operates on tcp/22. Though very similar to telnet in that it is used for command line communications to servers and network devices, SSH implements symmetrical encryption for all its communication. It accounts for 6.48% of TCP traffic and 5.81% of all data for 2017.

The remainder of the top 10 TCP ports are briefly discussed below:

- **TCP 1433** - This port is typically used by Microsoft SQL Server instances and the application listens for incoming connections on it. This a very common port to search for on the internet as many vulnerabilities have been found in MS SQL Server. According to CVE-Details (2018), there have been a total of 84 vulnerabilities published via CVE since its inception.

- **TCP 2323** - This is another common port for telnet servers to listen for connections on. It was found that the Mirai botnet searched for this open port on IoT devices (Antonakakis *et al.*, 2017). This port is further discussed in Section 5.1.

- **TCP 5358** - This port is used for Web Services for Devices. It advertises its services offered via an HTTP connection on port 5358 and is often used by WSD printers (de Bruijne *et al.*, 2017). This port was targeted by variants of Mirai due to its use in IoT devices.

- **TCP 7547** - This port is often used by networking and IoT devices for remote access management. Mirai its derivatives were found to target this port on IoT devices (Sutherland, 2016).
- **TCP 3389** - This port is used by Microsoft RDP (Remote Desktop Port) for the remote access administration of various Windows operating systems. It is often targeted, as weak or compromised credentials will allow an attacker full access to the system.
- **TCP 445** - This port is used by SMB (Server Message Block) file sharing and has been targeted by many trojans, worms and ransomware. The Conficker worm outbreak, which targeted this port, was covered in much detail by Irwin (2012). More recently, the WannaCry ransomware attack used this port to connect to vulnerable machines (Chen and Bridges, 2017).
- **TCP 80** - Typically used by HTTP web pages, this port is often targeted by attackers, looking for weaknesses in the web server or web page.
- **TCP 3128** - Most commonly used as a port for web proxy servers, attackers would look for badly configured or open proxy servers to compromise.

Table 4.3: Top 5 TCP source IP addresses based on packet count

Rank	IP Address	Reverse DNS Domain	# Packets	% Packets	Data (MB)	Country
1	94.102.49.7	towing.carsmemo.com	803 554	0.51	46	Netherlands
2	77.72.82.80	No reverse record	758 357	0.48	43	United Kingdom
3	163.172.135.224	$< IP >$.rev.cloud.scaleway.com	662 807	0.42	38	United Kingdom
4	74.125.206.197	wk-in-f197.1e100.net	591 265	0.37	39	United States
5	191.101.167.235	No reverse record	584 613	0.37	33	Netherlands
6	178.159.37.99	No reverse record	565 206	0.32	32	Russia
7	27.155.122.30	No reverse record	543 297	0.31	31	China
8	59.56.72.49	No reverse record	489 146	0.28	29	China
9	91.223.133.13	No reverse record	470 403	0.27	27	Ukraine
10	72.167.1.128	shr.prod.phx3.secureserver.net	469 028	0.27	27	United States
Σ			5 937 676	3.60	345	

Pang *et al.* (2004) list the ten most targeted TCP ports, with five of the top ten ports in this research also found in the top ten of their research. These ports are TCP 22, 23, 80, 445 and 1433. In a study conducted by Vichaidis *et al.* (2018) on darknet traffic, of their top nine TCP targeted ports, seven of them correspond to the results found in this research. These ports are TCP 23, 22, 2323, 5358, 7547, 3389 and 445. Their top two ports are also the top two ports in this research, namely tcp/23 and tcp/22. Irwin (2013) analysed the traffic from the same /24 sensor that this study used. The data set comprised a 15 month sample starting in February 2011 to May 2012. The listed top 10 TCP ports from that study contained six of the same listed ports in this study. Those ports are TCP 445, 3389, 1433, 80, 22 and 23. Figure 4.3 shows the top 10 TCP source IP addresses based on packet count. When compared to the overall top 10 IP addresses for all traffic types in Figure 4.13, six of these IP addresses are found in both tables. Source IP addresses are further discussed in Section 4.7.

4.2.1 TCP Flag

As previously discussed in Section 2.6 and described in Figure 2.4, TCP uses a three-way handshake to establish a connection. During the connection and communications, certain flags within the TCP header are set according to the whether it is a SYN, SYN-ACK, ACK, RST, etc packet. As darkweb traffic is one-way communication only, TCP flags will give an indication as to they type of traffic that is being received. Below is a brief overview of the the four most prominent TCP flags found within this traffic:

- **SYN** - These packets are the first ones sent to initiate a TCP connection and with darkweb traffic, would usually indicate network or port scanning activities as no active hosts exist within the IP address space.

- **ACK** - These packets are sent as an acknowledgement to a SYN packet and as no initial SYN packets are sent from the IP address range, these packets are a good indicator of IP address spoofing that has occurred and the result of a DDoS. This is referred to as backscatter.

- **SYN-ACK** - The SYN-ACK packet is the final step in the 3-way TCP handshake and this establishes the connection. This traffic is also backscatter and an good indicator of IP address spoofing due to a DDoS attack.

- **RST** - This is the packet that is sent in order to terminate a TCP connection. In this instance, it forms part of backscatter traffic.

Figure C.4 depicts the traffic patterns for the four most prominent traffic types over time. Table 4.4 further shows the statistics for these four traffic types. As can be seen from

Table 4.4: Packet and data volume based on TCP flag

TCP Flag	# Packets	% Packets	Data (MB)
SYN	142 286 942	81.75	8 252
ACK	249 336	0.14	78
SYN-ACK	12 178 682	6.99	724
RST	577 607	0.33	33
Σ	155 292 567	89.21	9087

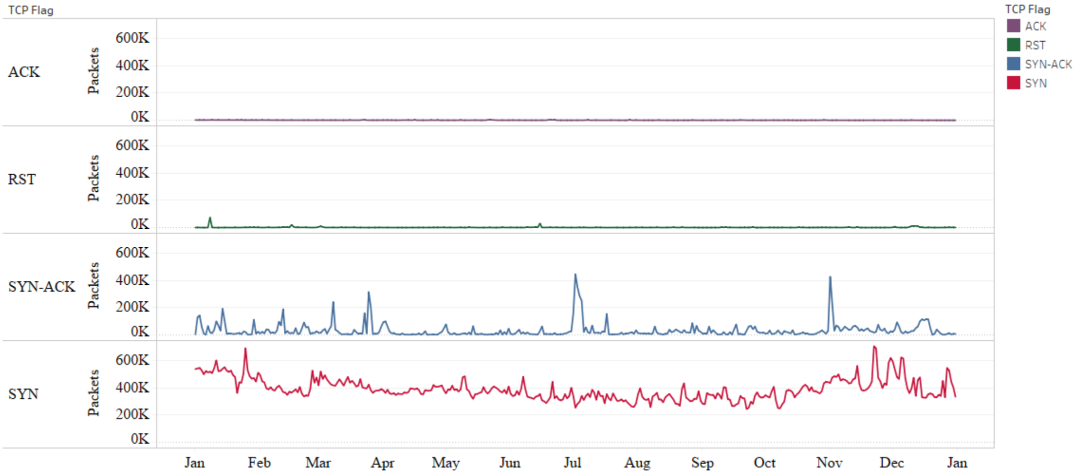


Figure 4.2: The four most prominent traffic, based on TCP Flag

the table, SYN traffic makes up the largest portion of the four types and accounts for 81.75% of the total traffic for 2017. The remainder of the packets from ACK, SYN-ACK and RST traffic combines to account for 7.46% of the total traffic for 2017. Pang *et al.* (2004) classifies SYN traffic as purely scanning, as packets are sent with the expectation of receiving a reply. This is also referred to as part of active traffic by Irwin (2011). On the other hand, SYN-ACK, ACK and RST are classified as reflected or backscatter traffic by Pang *et al.* (2004). Irwin (2011) does not specifically refer to SYN-ACK and ACK packets as passive traffic, but RST is. A more detailed analysis of scanning traffic is presented in Section 5.2, including a scrutiny of SYN traffic patterns as illustrated in Figure C.4. The reflected traffic will be examined as a specific case study in Section 4.6.

4.3 UDP

As shown in Table 4.5, the dominant UDP traffic is on port 5060. It accounts for a total of 27.67% of all UDP traffic and 2.63% of the total traffic for 2017. As previously mentioned in section 3.2, the total data for 2017 was 12.96 GB. Looking at Table 4.5, we see the total

Table 4.5: Top 10 UDP destination ports

Rank	Dest Port	Common Usage	# Packets	% Packets	Unique IP's	Data (MB)	Avg Pkt Size (bytes)
1	5060	SIP	4 585 939	27.67	2 436	1 981	79.92
2	1900	UPnP	2 422 888	14.62	12 889	313	69.83
3	123	NTP	1 163 070	7.01	3 926	104	94.01
4	53	DNS	719 371	4.34	2 992	56	81.65
5	53413	Router Exploit	688 720	4.15	1 723	61	93.23
6	161	SNMP	473 875	2.86	2 522	43	94.44
7	137	NetBIOS	301 063	1.81	2 334	26	92.02
8	19	Testing/Debugging	248 689	1.50	898	15	61.38
9	111	Portmapper	218 359	1.31	241	17	82.19
10	1434	MS SQL Server	213 276	1.28	676	30	147.60
Σ			11 035 250	66.55	30 637	2 646	89.62

data for udp/5060 is about 1.9 GB. UDP/5060 is a port typically used for SIP traffic. It is notable to mention that despite udp/5060 only accounting for 2.63% of the total packets, it accounts for 21.06% of the total UDP data. Irwin (2013) notes that udp/5060 was the top UDP port targeted in the findings of that study. In that research, the data from five /24 network telescopes were analysed. Each telescope's netblock was non-contiguous to each other. All telescopes had udp/5060 as the top UDP port, with each accounting for over 20% of the UDP traffic for the respective sensor. SIP is discussed at length as a case study in Section 5.3.

The rest of the top 10 UDP ports are briefly discussed below:

- **UDP 1900** - This port is used and has been registered by Microsoft for SSDP (Simple Service Delivery Protocol). By default this port was open on Windows XP and made it extremely vulnerable to remote attack (Majkowski, 2017). Bajpai *et al.* (2018) states that many IoT devices use this port to advertise their services. They further note that this port has been targeted to scan and map out IoT devices on a network.

- **UDP 123** - Used by the NTP (Network Time Protocol) for synchronisation of time across a network or internet (Wilkins, 2012).

- **UDP 53** - The standard port for DNS name resolution. DNS exploits are very popular, such as DNS poisoning which causes incorrect name server resolution (Shulman and Waidner, 2014).

- **UDP 53413** - Routers manufactured by Netcore have been found to have an exploitable backdoor running on this port (Yeh, 2014).

- **UDP 161** - This port is used by SNMP (Simple Network Monitoring Protocol), which is used by many network devices and servers to send logging information to. Insecure write access could lead to a fully compromised system (Wilkins, 2012).
- **UDP 137** - Used by the NETBIOS protocol which is used for file and print sharing on networks (Wilkins, 2012).
- **UDP 19** - Used by CGP (Character Generator Protocol, also known as chargen), which is used for debugging and testing. Services configured on this port accepts a stream of characters it receives (Postel, 1983b). Chargen is deprecated, but still often seen in reflected DDoS.
- **UDP 11** - This port is often used by a legacy Unix service used to dump active process information to, which is no longer used due to the inherent security risks associated with its implementation (Postel, 1983a).
- **UDP 1434** - This port is used for MS SQL Server management. Badly configured servers have had the application compromised through this port. An example of an attack which targeted this port is the SQL Slammer worm (Moore *et al.*, 2003).

Table 4.6: Top 10 UDP source IP addresses based on packet count

Rank	**IP Address**	**Reverse DNS Domain**	**# Packets**	**% Packets**	**Data (MB)**	**Country**
1	163.172.215.161	$< IP >$.rev.poneytelecom.eu	508 588	3.04	220	Netherlands
2	185.94.111.1	No reverse record	487 667	2.91	36	Russian
3	92.42.107.139	No reverse record	190 005	1.13	81	Switzerland
4	146.0.243.29	No reverse record	142 336	0.85	61	Germany
5	62.210.36.129	$< IP >$.rev.poneytelecom.eu	120 057	0.71	52	France
6	92.42.108.203	No reverse record	106 960	0.64	46	Switzerland
7	184.105.139.67	scan-01.shadowserver.org	89 149	0.54	11	United States
8	51.15.209.185	No reverse record	88 320	0.53	38	France
9	104.171.172.6	rev.cloud.scaleway.com	84 562	0.51	7	United States
10	209.126.136.2	No reverse record	83 817	0.50	9	United States
Σ			1 901 461	11.47	561	

Figure 4.6 depicts the top 10 source IP addresses based on UDP packet count. The packets from IP addresses ranked 1^{st}, 3^{rd}, 4^{th}, 5^{th} and 8^{th} were found to only contain SIP traffic. These are listed in Table 5.5 as the top four IP addresses for SIP traffic in Section 5.3, which deals with specific case studies that were found. The 2^{nd} ranked IP address communicated on UDP ports 17, 19, 53, 111, 123, 137, 161, 520 and 1900. The packet distribution across these ports were fairly consistent throughout the entire year. The

reverse DNS record for the 7th ranked IP address resolved to scan-01.shadowserver.org. The Shadowserver Foundation[1] is an organisation that does scanning and intelligence gathering of the internet as a way to help prevent cyber crime. This IP address sent packets on udp/161 and udp/53, which are the ports used for SNMP and DNS respectively. Source IP addresses are discussed further in Section 4.7.

4.4 ICMP

ICMP traffic constitutes the smallest portion of the low levels protocols. A total of 1 345 422 ICMP packets were received for the duration of 2017, which accounted for only 0.77% of 2017's total traffic. Table 4.7 shows the distribution of ICMP packets according to their type and code. The largest portion of packets are Echo Requests, which is a total of 1 051 536 packets and making up 78.15% of the ICMP total . Echo requests are sent to test if a host is up and responsive and this is typically done using an application such as ping, as mentioned previously in Section 2.3.3.

Table 4.7: Composition of all ICMP traffic, categorised by ICMP type and code

ICMP Type	ICMP Code	# Packets	% of Total
0 - Echo Reply	0	71 708	5.32
	5	1	< 1
	15	2	< 1
3 - Destination Unreachable	0 - Net Unreachable	2 961	< 1
	1 - Host Unreachable	1 958	< 1
	2 - Protocol Unreachable	19 027	1.41
	3 - Port Unreachable	136 812	19.16
	4 - Fragmentation Needed	753	< 1
	9 - Net Administratively Prohibited	2	< 1
	10 - Host Administratively Prohibited	19 990	1.48
	13 - Comms Administratively Prohibited	392	< 1
4 - Source Quench	0 *deprecated*	15	< 1
5 - Redirect	1 - Redirect for Host	155	< 1
8 - Echo	0 - Echo Request	1 051 536	78.15
	9 - *deprecated*	183	< 1
11 - Time Exceeded	0 - TTL Expired in Transit	36 458	2.70
	1 - Fragment Reassembly Time Exceeded	4	< 1
12 - Parameter Problem	0 - Pointer Indicates Error	7	< 1
13 - Timestamp	0 - Timestamp	3 454	< 1
17 - Address Mask Request	0 - *deprecated*	1	< 1
69 - Unassigned	0 - Unassigned	3	< 1

[1] https://www.shadowserver.org/wiki/

As can be seen in Table 4.7, the only significant packet count other than the Echo Requests (Type 8) are the Destination Port Unreachable (Type 3) and Echo Reply (Type 0). These account for 19.16% (136 812 packets) and 5.32% (71 708 packets) of the total ICMP packets respectively. There were 81 154 unique IP addresses responsible for the ICMP traffic. Table 4.8 shows the top 10 ICMP source IP addresses based on packet count.

Table 4.8: Top 10 ICMP source IP addresses

Rank	IP Address	Reverse DNS Domain	# Packets	% Packets	Country
1	46.234.125.89	prague-ping-1.cdn77.com	107 991	8.02	Czech Republic
2	185.94.111.1	No reverse record	53 439	3.97	Russia
3	146.64.28.14	dolphin.meraka.csir.co.za	38 607	2.86	South Africa
4	46.166.148.176	No reverse record	17 338	1.28	Netherlands
5	117.131.215.170	No reverse record	14 740	1.09	China
6	207.226.141.42	$< IP >$.static.pccwglobal.net	14 188	10.5	United States
7	111.161.35.146	dns146.online.tj.cn	13 748	1.02	China
8	42.81.86.90	No reverse record	13 480	1.00	China
9	107.161.88.35	No reverse record	12 808	0.95	United States
10	222.34.18.27	No reverse record	12 390	0.92	China
Σ			298 729	22.00	

A further analysis of these top five IP addresses indicated that the addresses ranked 1^{st}, 2^{nd}, 3^{rd} and 5^{th}, only sent echo request packets. All these IP addresses sent packets to all 256 IP addresses in the telescope's range. The address ranked 4^{th} only sent echo reply packets, but once again also to all IP addresses in the telescope's range. As the network telescope is a passive sensor and would not have sent the initial echo request, this means that the IP address range of the network telescope is most likely being used as spoofed addresses. All the IP addresses listed in Table 4.8, other than the ones ranked 2^{nd} and 4^{th}, only sent ICMP packets. The 2^{nd} ranked source IP address also sent packets to 15 different TCP and UDP ports. The 4^{th} ranked IP sent 38 packets to tcp/22 and only a single packet each to 2053 various other TCP ports.

4.5 Monthly Traffic vs. Data

The network telescope received over 174 million packets in 2017, with the total data count in at just under 13GB. The number of packets and amount of data received by the network telescope for each month of 2017 were previously illustrated in Figures 4.3 and 4.4.

Table 4.9: Top 10 packet sizes based on packet count

Rank	**Packet Size (bytes)**	**# Packets**	**% of Total**
1	60	138 941 161	79.83
2	62	8 398 469	4.82
3	74	7 021 604	4.03
4	66	3 846 185	2.20
5	70	379 842	< 1
6	71	308 268	< 1
7	578	148 679	< 1
8	566	130 913	< 1
9	72	119 098	< 1
10	106	68 460	< 1

The sizes of all the received packets were not uniform and ranged between 60 and 1514 bytes, the same as reported by Balkanli *et al.* (2014) in their research on a CAIDA data set. All packets (TCP, UDP and ICMP) reported packet sizes in that range. Table 4.9 shows the distribution of packets according to top the 10 most common packet sizes in this data set. As can be noted from the table, nearly 80% of the packets were 60 bytes in size and over 90% of the total packets were found to be between 60 and 74 bytes in size.

Figures 4.3 and 4.4 displays the monthly traffic patterns for data (measured in megabytes) and number of packets over time. The traffic patterns for the data and packets display quite similar trends overall. There are however a few notable exceptions where the data or packet traffic graphs at particular points are very different to each other. At these points it is noted that one of the two spikes while the other dips. Some of these are highlighted as “A”, “B” and “C” in Figure 4.3 and as “D” and “E” in Figure 4.4. These annotated exceptions are analysed next.

A

This occurred on 17 January 2017, where a total of 620 150 packets were received on that day, accounting for a total of 59.26 MB (62 135 948 bytes) of traffic. The point annotated in Figure 4.3 shows a distinct increase in data over packet count. Taking into account the number of packets and total data, if all packets were equal in size, the each packet size would be 100.19 bytes. However, the data showed that there was a total 484 237 packets that were 60 bytes in size, accounting for 78.08% of the data for that day. There was also 68 056 packets that are above 100 bytes in size, caused mainly by a large amount of SIP and SSDP[2] traffic detected on that day. The sizes of these packets ranged in sizes from 103 bytes to 930 bytes, which accounted for the spike in data on that day.

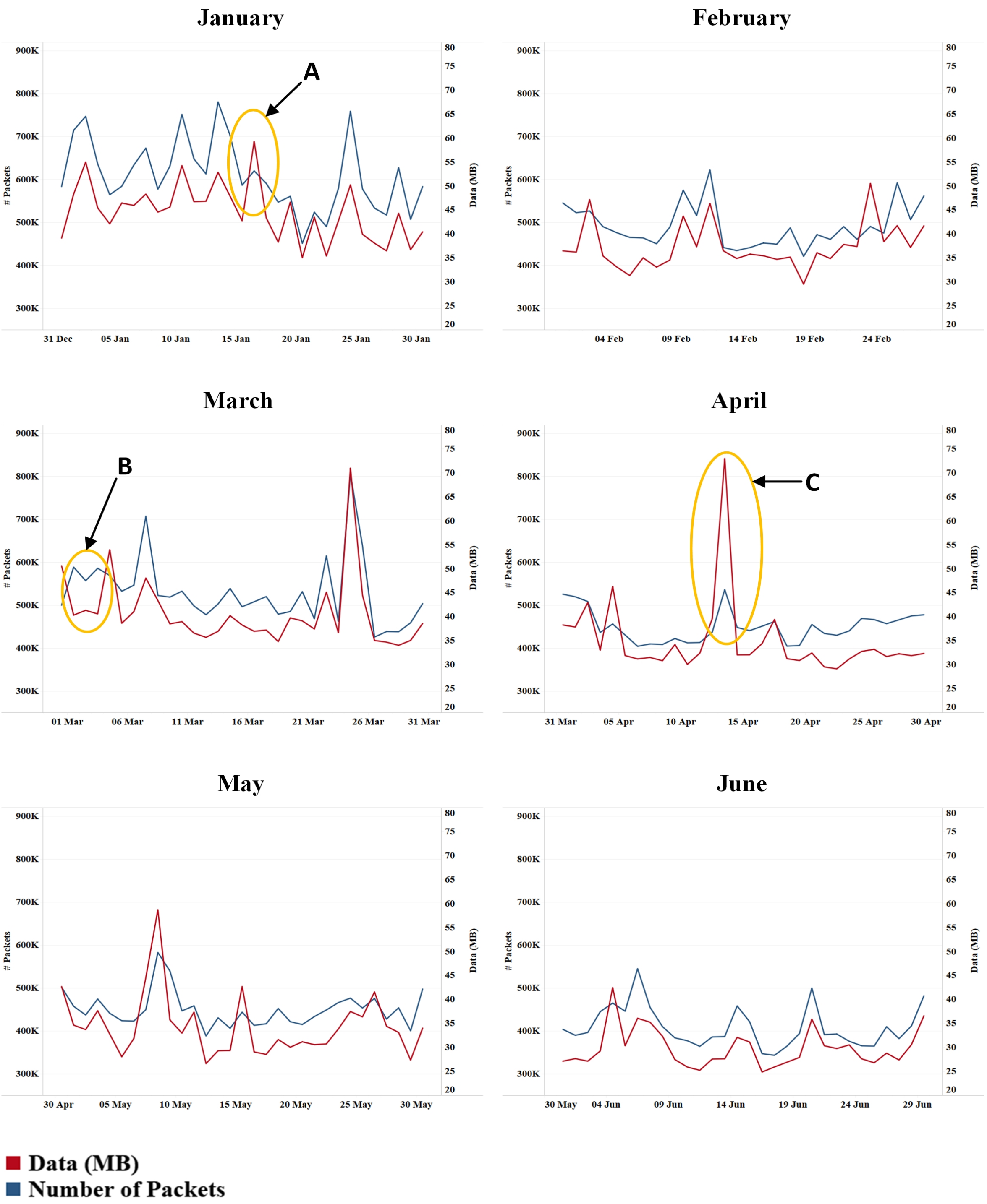


Figure 4.3: Daily packet and data graphs for January to June

[2] Simple Service Discovery Protocol - Used to discover Universal Plug and Play devices on a network https://en.wikipedia.org/wiki/Simple_Service_Discovery_Protocol

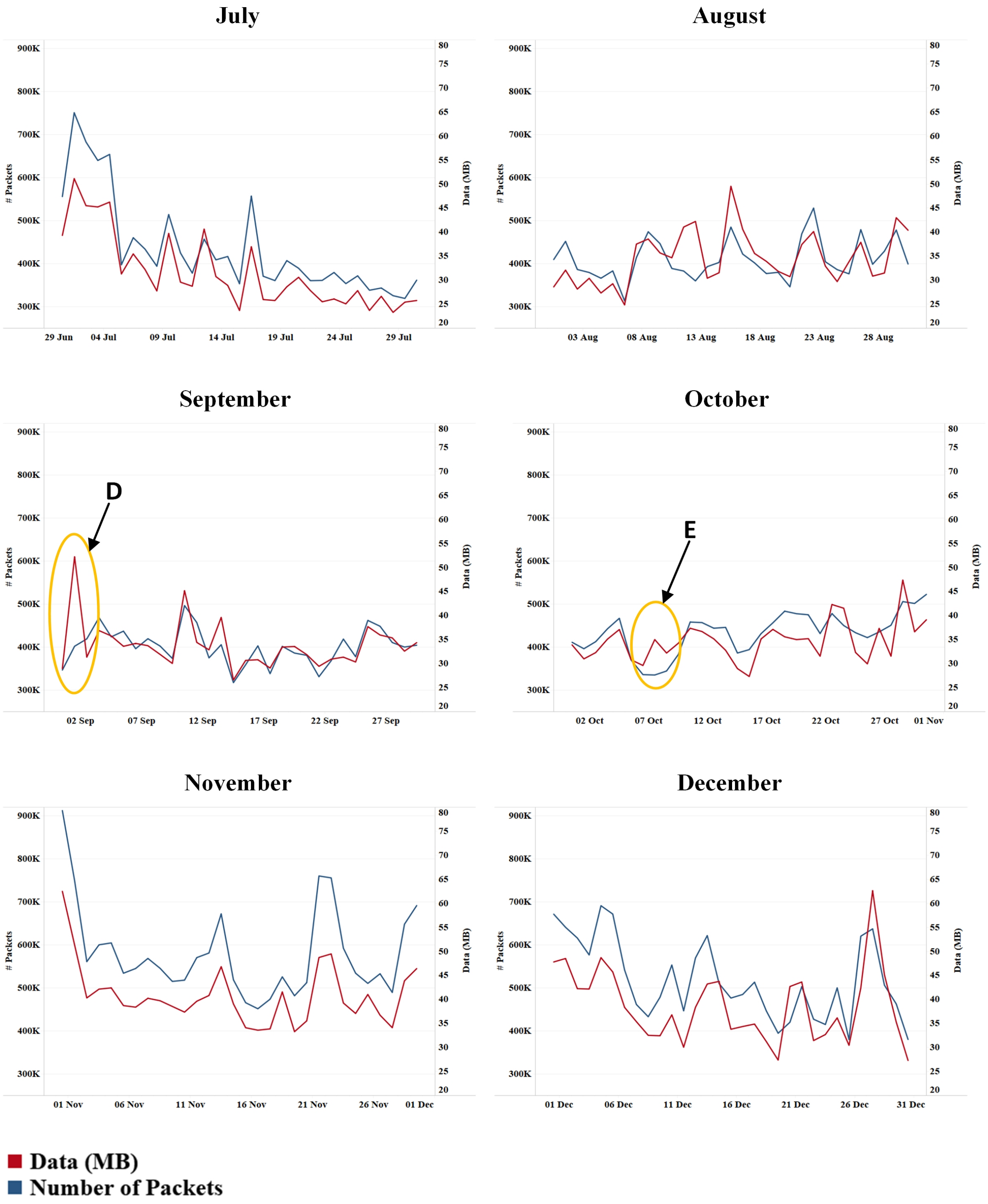


Figure 4.4: Daily packet and data graphs for July to December

B

On the 2^{nd} and 4^{th} March 2017, there are distinct patterns in the graphs where the data and packets moved in the opposite direction to each other. On these two days, traffic spiked and data dipped. When looking at the data for the individual days, there was a total of 588 686 packets received on the 2^{nd}, accounting for a total of 42 292 029 bytes. For the traffic received on this day, 93% of it comprised of TCP packets. This TCP traffic was found to consist of mostly scanning or SYN packets, accounting for 95.01% of it. The packet sizes ranged between 60 and 78 bytes only, with an overwhelming 92.02% of the packets being 60 bytes in size. Very similar traffic patterns can be seen on the graph for the 4^{th} March. A closer inspection of the traffic showed a resemblance in traffic make up between the 2^{nd} and 4^{th}. This high count of small packets is what caused the differences in the packet and data graphs. For both days, the major contributor to the traffic was tcp/23 and to a lesser amount tcp/2323, which is a possible indicator of Mirai like activity.

C

This exception is similar to what was seen in "A", where the data graph spiked much higher than the traffic graph. This spike occurred on the 14^{th} April and was coincidentally the highest recorded data for a single day throughout the entire 2017 period. A total of 76 510 453 bytes of traffic were received on this day, coming from 536 396 packets. There was a big increase in SIP (udp/5060) traffic on this day. These packets contributed to 61.41% of the total data for the day, coming from only 19.14% of the total packets.

D

A total of 401 988 packets were received on the 2^{nd} September 2017, making up 54 770 715 bytes of traffic. TCP comprised the largest portion of packet count with 296 787 packets. It did however only account for 18 283 457 bytes of traffic. SIP (udp/5060) accounted for the second highest count of packets with 71 898 in total. Although this was only accounted for 17.88% of the total packets for the day, it was responsible for 32 320 947 bytes of traffic, which is 50.01% of the total data for the day. The spike in the data graph was therefor largely caused by this udp/5060 traffic.

E

This exception occurred on the 8^{th} October 2017, where a total of of 335 182 packets were received. The largest portion of the traffic comprised of TCP, accounting for 80.02% of the total packets on that day. SIP was the second highest, contributing 10.49% towards the total packets. Despite the big difference in packet count between TCP and SIP, they both contributed a very similar amount to the overall data on this day. TCP packets added up to 16 599 448 bytes of traffic and SIP to 16 136 707 bytes of traffic. The SIP traffic was not only on udp/5060, but detected across 37 different ports. Most of the ports had exactly 256 packets to it, indicating that this traffic was some form of SIP scanning.

4.6 Backscatter

Backscatter traffic is essentially reflected packets and when found on a network telescope, is usually a sign of a DDoS happening somewhere. Reflected TCP packets can be identified when the TCP flags are set to either ACK, SYN-ACK or RST. UDP, being a stateless protocol, does not have flags that are set during a communication session. It is nearly impossible to determine if UDP traffic is backscatter without doing a deep protocol inspection of the packets. The fact that "response" packets are received in darknet traffic is a clear indication that IP address spoofing has occurred. When attackers perform a denial of service attack on a victim, the IP address of the source packets are spoofed and the victim sees it coming from from other third party address. In the case of this data, the attacker received packets with a spoofed source IP address belonging in the darknet range. The various TCP flags provides an insight into the type of traffic that was used in the DDoS attack. As noted by Moore *et al.* (2006), specific flags set in TCP responses are linked to specific TCP requests. The most common responses are listed below.

- Attacker sends a SYN packet to a port that is open – Victim responds with a SYN-ACK packet
- Attacker sends a SYN packet to a port which is closed – Victim responds with a RST packet
- Attacker sends an ACK packet with no previous SYN or SYN-ACK having being sent – Victim responds with a RST packet
- Attacker sends a RST packet – No response is sent by victim

Figure 4.5 illustrates the comparative traffic patterns of ACK, RST and SYN-ACK packets over time, using the same axis scale. Having the graphs depicted on a standard scale makes it quite easy to compare the traffic to each other, but causes a severe loss in detail for the traffic with lower packet count. Figures 4.6, 4.7 and 4.8 therefore shows the individual traffic patterns for each of the traffic types, with each having an individual axis scale, thereby providing a greater detail.

Backscatter makes up a very small portion of the total traffic for 2017, with a combined total of only 13 005 625 packets, accounting for only 7.47% of the total traffic. From the total backscatter traffic, SYN-ACK makes up the largest portion with 12 178 682 packets. ACK and RST comprises the remainder with 249 336 and 577 607 packets respectively. Table 4.10 shows the packets count per month for all the backscatter traffic. The highest levels for each traffic type is marked in blue and the lowest is marked in red.

Table 4.10: Packet count per month for all backscatter traffic

Month	ACK	% of Total	RST	% of Total	SYN-ACK	% of Total
January	52 932	21.22	101 760	17.61	1 271 134	10.43
February	28 591	11.46	68 554	11.86	1 033 553	8.48
March	29 294	11.74	49 493	8.56	1 473 470	12.09
April	26 849	10.76	23 284	4.03	515 405	4.23
May	27 525	11.03	27 388	4.74	440 889	3.62
June	22 789	9.13	72 074	12.47	401 463	3.29
July	18 432	7.39	24 834	4.29	2 096 071	17.21
August	8 964	3.59	30 638	5.30	716 161	5.88
September	11 109	4.45	32 564	5.63	712 085	5.84
October	13 200	5.29	29 342	5.07	484 853	3.98
November	4 978	1.99	30 506	5.28	1 808 688	14.85
December	4 637	1.85	87 170	15.09	1 224 910	10.05

For the duration of 2017 the highest packet count for ACK traffic occurred in January, with the lowest count occurring in December. RST traffic had the highest count in January as well, with the lowest count recorded in April. With SYN-ACK traffic, the highest packet count occurred in July, with the lowest recorded in June.

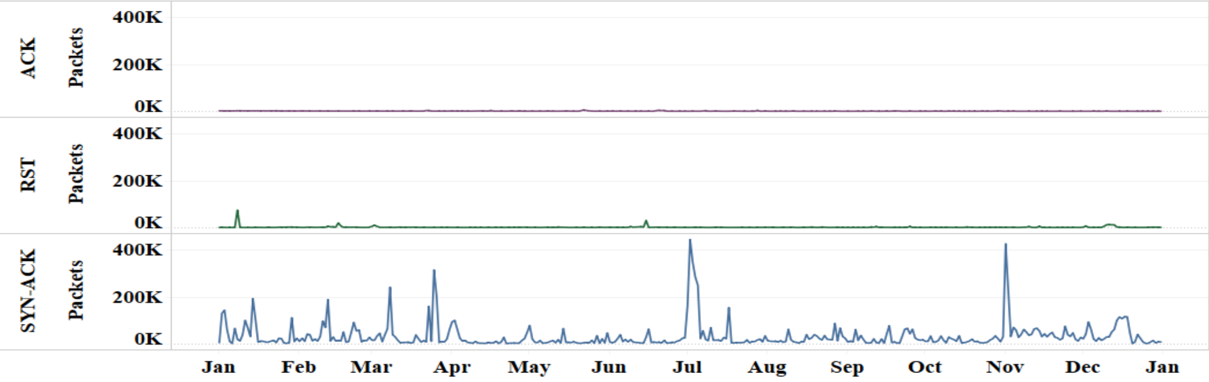


Figure 4.5: ACK, RST and SYN-ACK traffic patterns over time

The individual traffic types, along with their respective peaks are discussed further below.

The ACK traffic makes up the smallest portion of the total backscatter. In Figure 4.6 we see that there is a general downwards trend in average traffic from the beginning of

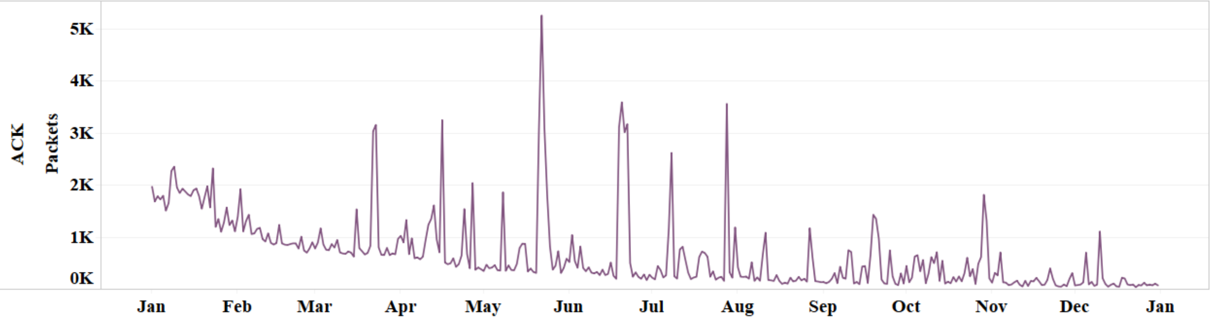


Figure 4.6: ACK traffic pattern over time

the year until the end. There are five noticeable peaks occurring in the traffic on 23 March, 16 April, 22 May, 20 June and 28 July. The peak on 22nd May is the largest of the five peaks. For the traffic on that day, there was one particular IP address that accounted for the majority of traffic on that day. 4571 Packets were received from IP address 210.48.154.99, accounting for 88.92% of the total ACK traffic for that day. There was a total of 256 destination IP addresses associated with these packets, which means that the full range of IP addresses in the network telescope was spoofed. The IP address resolves to *quid.centralmalaysia.com*, a web hosting company in Malaysia. All traffic received from this IP address had a source port of 80, which means that this was most likely a denial of service type of attack on their website or one hosted by them.

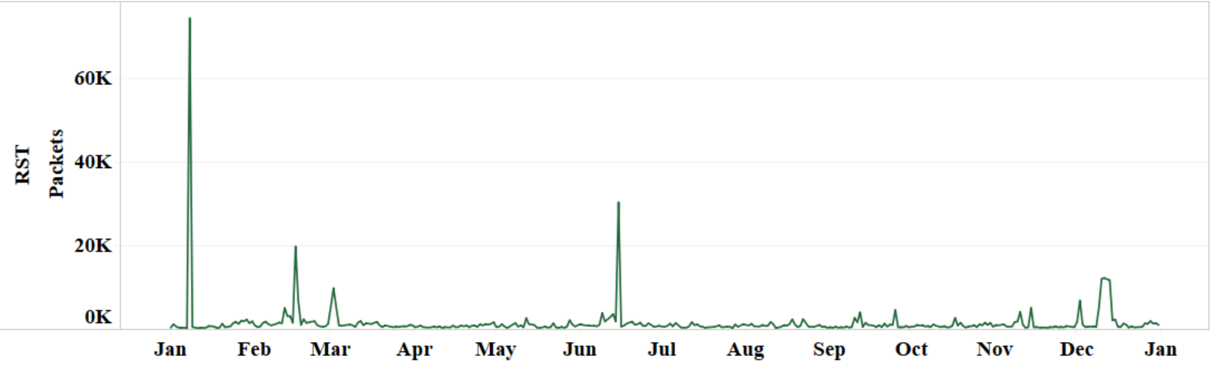


Figure 4.7: RST traffic pattern over time

Figure 4.7 looks at the traffic patterns for all RST traffic for the year. There are five large peaks that can be seen on the graph, with the first one being the biggest. The highest three peaks in traffic occurred on 8 January, 16 February and 15 June. When further analysing the highest traffic peak, there was a total of 73 851 packets received from a single IP address on that day, accounting for 99.25% of the day's traffic. All traffic had a single destination IP address and source port. The destination IP address was A.B.C.100 (Only the actual last octet is shown in order to keep the network telescope IP range private) and the destination port was 15004. The source IP address of 137.74.206.93 is located in France and has a reverse DNS entry of ns3057763.ip-137-74-206.eu. According to the WhoIs[3] record, the company who is associated to this IP address is ovh.com, a web and cloud hosting organisation in Europe. According to Paganini (2016), they were previously hit by a DDoS attack in September of 2016. This attack saw more than 1Tbps of traffic, the largest attack in recorded history, hitting their servers.

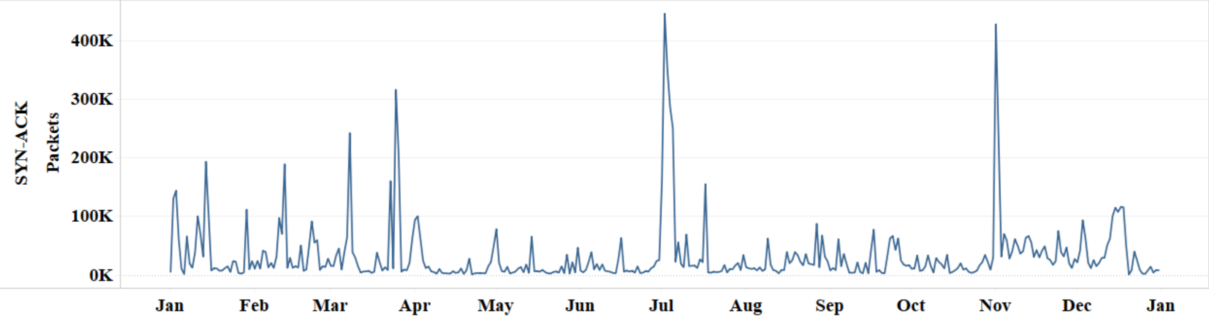


Figure 4.8: SYN-ACK traffic pattern over time

The largest of all the backscatter traffic was the SYN-ACK traffic. Figure 4.8 shows a constant average traffic flow, with many noticeable spikes observed throughout the year. The three largest spikes occurred on 25 March, 2 July and 1 November, with the July spike being the highest. When further analysing the traffic spike in July, it was found that the top two IP addresses accounted for 51.90% of the total traffic for the day. The following three in the list of the top five IP addresses make up 38.88% of the total traffic. Table 4.11 shows the details of the top five IP addresses for the day. Both IP addresses were located in the United States and all traffic from these IP addresses had a source port of tcp/80.

Table 4.11: Statistics of top 5 SYN-ACK IP addresses for July peak

Rank	IP Address	Packets	% of Total	Country	Source Port
1	69.195.124.205	114 388	26.05	United States	tcp/80
2	72.167.1.128	113 470	25.84	United States	tcp/80
3	59.56.97.105	76 346	17.39	China	tcp/80
4	144.76.237.113	66 287	15.09	Germany	tcp/80
5	37.182.9.32	28 057	6.39	Italy	tcp/80

The top five IP addresses account for 90.76% of the total traffic for the day, all of which

[3] WhoIs is a querying protocol used to query various domain registry databases who store information related to internet resources such as DNS and IP information

have the same source port of tcp/80. Once again, this is an indication that these backscatter packets are possibly the result of some form of denial of service attack on a website. DNS details according to WhoIS for the top five IP addresses are listed below.

- **69.195.124.205** - The reverse DNS for this IP is `box1005.bluehost.com`. Bluehost is a website hosting company.
- **72.167.1.128** - The reverse DNS for this IP is `p3nlhg114c1114.shr.prod.phx3.secureserver.net`. Secureserver is a domain registration and website and email hosting company.
- **59.56.97.105** - There is no reverse DNS record for this IP address, but the IP is assigned to a Chinese telecommunications company called Chinanet.
- **144.76.237.113** - The reverse DNS record for this IP address is `static.113.237.76.144.clients.your-server.de`. The IP address is assigned to Hetzner Online GmbH, which is a website and email hosting company.
- **37.182.9.32** - The reverse DNS record for this IP address is `net-37-182-9-32.cust.vodafonedsl.it`. The IP address is assigned to Vodafone Omnitel BV, the Italian subsidiary of the Vodafone Group.

Although we have found that all the IP addresses within the network telescope's range had received packet, there are some that had received substantially more traffic to that others. Table 4.12 shows the top five destination IP addresses according to packet count for the entire year. Only the last octet of the destination IP address is listed in the table..

Table 4.12: Top 5 destination IP addresses for backscatter traffic

Rank	**IP Address**	**Packets**	**% of Total**
1	**.202**	597 037	4.59
2	**.84**	520 295	4.00
3	**.154**	232 437	1.78
4	**.42**	181 498	1.39
5	**.245**	180 266	1.38

4.7 Source IP Addresses

As mentioned in Section 3.4, Wireshark used the databases provided by Maxmind Lite, which allowed each IP address to be mapped to its source country. Because geolocation data gets updated often due to IP addresses or blocks being assigned to different organisations, geolocation information for below IP addresses may have changed from the time this research was carried out.

Table 4.13: Top 10 source IP addresses based on packet count

Rank	IP Address	# Packets	% Packets	Data (MB)	Country	Ports Accessed
1	94.102.49.7	803 554	0.46	46	Netherlands	800 various TCP ports
2	77.72.82.80	758 357	0.44	43	United Kingdom	2615 various TCP ports
3	163.172.135.224	662 807	0.38	38	United Kingdom	tcp/3128
4	185.94.111.1	625 962	0.36	44	Russia	15 various TCP & UDP ports
5	74.125.206.197	591 265	0.34	39	United States	20 536 various TCP ports
6	191.101.167.235	584 613	0.33	33	Netherlands	6 various TCP ports
7	178.159.37.99	565 206	0.32	32	Russia	tcp/3128
8	27.155.122.30	543 297	0.31	31	China	udp/5060 & udp/5080
9	163.172.215.161	508 588	0.29	220	Netherlands	udp/5060
10	59.56.72.49	489 146	0.28	29	China	tcp/63507,tcp/63544 & tcp/63572
Σ		6 159 795	3.51	555		

Table 4.13 presents the top five IP addresses based on packet count. The differences in both packet count and data between the top 10 are rather marginal. With 803 554 packets and accounting for only 0.13% of total traffic, the top IP only surpasses the fifth IP by 212 289 packets. This marginal decrease can be seen throughout the list of top 20 source IP addresses, full details of which can be seen in Appendix A.1. Below are some DNS details related to the top five IP addresses:

- **94.102.49.7** - Quasi Networks LTD (towing.carsmemo.com)
- **77.72.82.80** - United Protection (UK) Security LTD (hostby.ups-gb.co.uk)
- **163.172.135.224** - scaleway.com
- **185.94.111.1** - Qrator Labs (qrator.net)
- **74.125.206.197** - Google LLC (wk-in-f197.1e100.net)

For the entire 2017 time period, packets from 237 countries were captured by the network telescope. Table 4.14 shows the statistics for the top 10 countries based on packet

Table 4.14: Top Countries based on packet count

Rank	Country	# Packets	% Packets	Unique IP's	Data (MB)	Top 5 Ports in Order
1	China	37 123 833	21.33	2 057 423	2 228	TCP 23, 1433, 22, 2323, 3306
2	United States	23 513 819	13.51	245 084	1 768	TCP 23, 22, 80, 443, 1433
3	Russia	11 315 663	6.50	846 655	854	TCP 23, 3128, 22, 7547, 2323
4	Netherlands	8 932 781	5.13	14 078	1 015	TCP 8545, 23, 445, 22, 80
5	Brazil	7 567 119	4.35	2 108 782	453	TCP 23, 5358, 22, 2323, 6789
6	Korea	6 342 037	3.64	115 504	506	TCP 23, 5358, 22, 7547, 1433
7	India	6 301 886	3.62	960 340	377	TCP 23, 2323, 22, 21, 5358
8	United Kingdom	5 797 870	3.33	153 746	626	TCP, 3128, 23, 3389, 22, 1433
9	Vietnam	4 932 397	2.83	676 664	293	TCP 23, 5358, 2323, 7547, 22
10	France	4 787 156	2.75	61 642	882	TCP 22, 23, 80, 46307, 44418
Σ		116 614 561	67.00	7 239 918	9003	
38	South Africa	490 563	0.28	41 203	36	TCP 23, 3389, 7547, 1433, 2323

count. IP addresses attributable to China were responsible for 37 123 833 packets, which accounted for 21.33% of the total traffic for 2017.

The combined packets from the top 10 countries were responsible for 67% of all traffic and 67.81% of the total data. They also accounted for a total of 54.55% of all distinct IP addresses. A full global geolocation heatmap, based on packet count, can be seen in Figure 4.9.

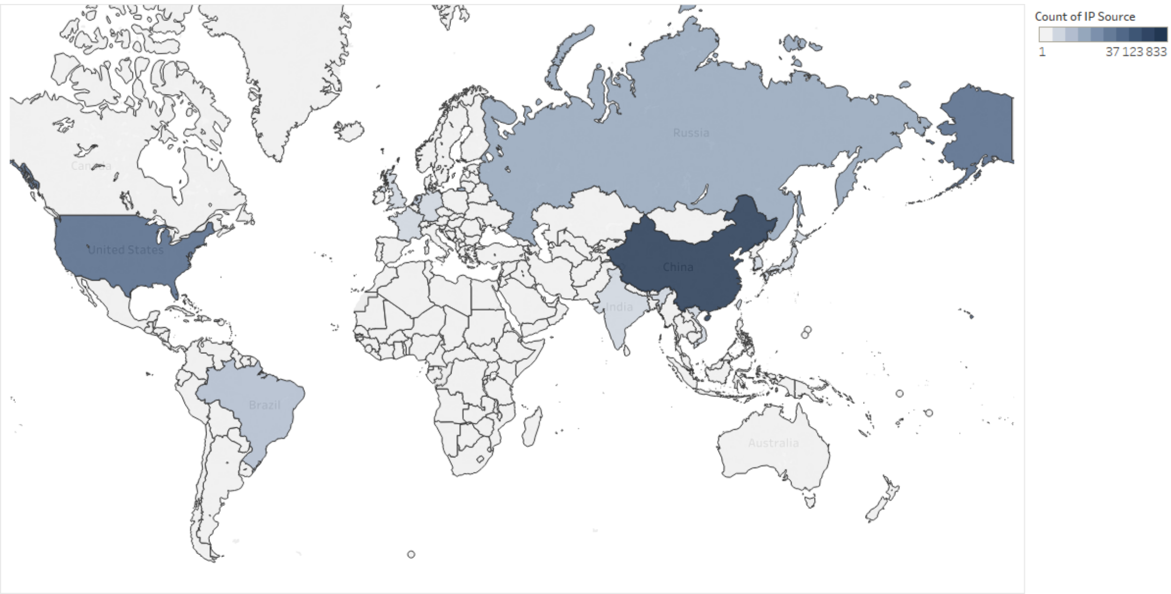


Figure 4.9: Global heatmap of source IP addresses based on packet count

As was previously mentioned in Section 3.2, there was a total of 13 271 685 unique source

IP addresses that sent packets. Figure 4.10 shows the traffic pattern for the count of unique source IP addresses over time. The highest count of unique source IP addresses occurred on 2nd January, with a total of 248 588. Of this total unique IP addresses, 98.50% of them were TCP packets as well as 98.03% of them being SYN packets. The five most prominent destination ports were tcp/23, tcp/23231, tcp/6789, tcp/2323 and tcp/22, all of which are associated with Mirai or variants of Mirai (Van der Elzen and van Heugten, 2017). Traffic on these ports are discussed in more detail in Sections 4.1 and 5.1. A further noticeable spike in unique source IP addresses occurred on 29 November 2017, as annotated in Figure 4.10. A total of 648 101 packets were received on this day, coming from 193 129 unique source IP addresses. TCP traffic accounted for 99.23% of the total unique source IP addresses, with 599 057 packets. The top two destination ports were found to be tcp/23 and tcp/2323, accounting for 54.56% and 7.79% of packets respectively. The next three ports of the top five were tcp/22, tcp/1433 and tcp/445. Once again, the high presence of tcp/23 and tcp/2323 is a good indicator of Mirai like traffic activity. The lowest occurrence was on 27 July, with only 50 298 unique IP addresses.

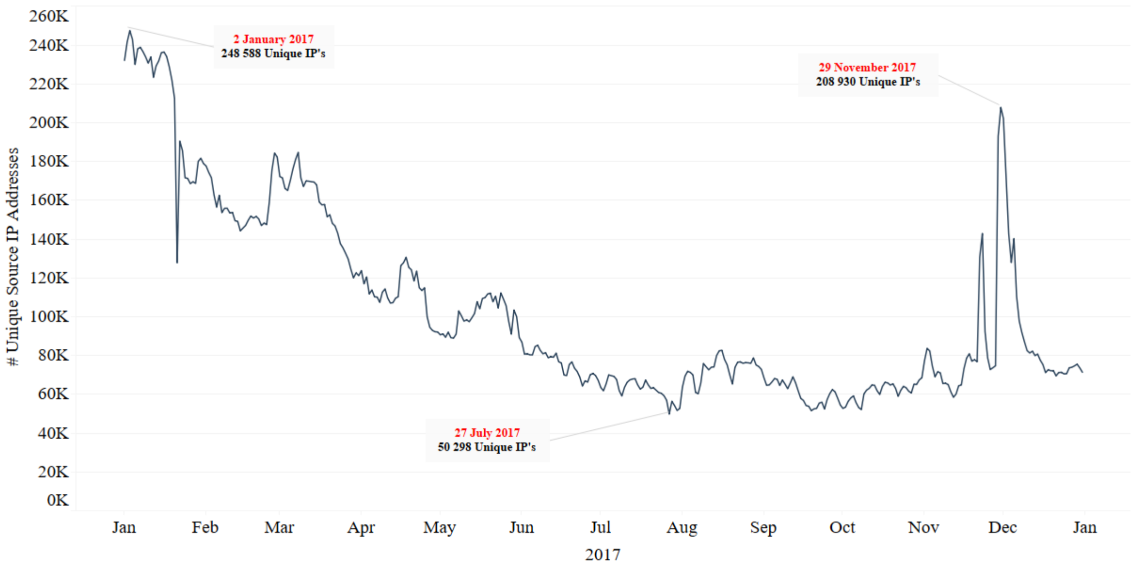


Figure 4.10: Traffic pattern for the count of unique IP addresses over time

4.8 Destination IP Addresses

The network telescope used in this research monitored a /24 IPv4 netblock located within the 155/8 range. All 256 IP addresses in the netblock received traffic to it, with some being targeted more often than others. Table 4.15 lists the top 10 destination IP addresses ranked by volume of packets received. Note that only the last octet of the destination IP address is shown.

Table 4.15: Top 10 destination IP addresses based on packet count

Rank	Last Octet	# Packets	TCP	UDP	ICMP	Data (MB)	Distinct IP's
1	202	1 227 698	1 160 986	63 298	3 940	83	283 622
2	84	1 141 390	1 072 310	63 618	6 346	79	282 235
3	1	902 832	600 355	83 708	219 748	65	286 145
4	154	852 864	787 628	61 946	3 799	62	281 535
5	245	849 161	754 337	91 412	3 926	63	282 880
6	42	805 201	738 048	63 726	4 222	59	281 703
7	142	789 363	723 426	62 007	4 464	58	281 889
8	100	778 803	710 984	64 187	4 506	57	282 847
9	102	778 313	711 545	63 260	4 672	57	282 392
10	222	769 798	696 591	69 622	4 125	63	285 138
Σ		8 895 423	7 956 210	68 6784	259748	64.6	283 038

Figure 4.11 displays the traffic patterns for the top two ranked IP addresses from Table 4.15. As can be seen, the traffic to these IP addresses are fairly consistent throughout the year, with the exception of the three large spikes, as annotated in the figure. The first two spikes are from the top ranked destination IP address. The analysis of the first spike showed that out of the 311 652 packets received on that day, 309 022 packets came from a single IP address located in China. These were all SYN-ACK packets with a TCP destination port of 29526 and a source port of 80. It can therefore be deduced that this is likely to be the reflected traffic from some form of DDoS web attack. The smaller spike occurring in May was further analysed as well. Again a single IP address from China was responsible for largest portion of the traffic on that day. There was a total of 42 221 SYN-ACK packets received this IP address with a destination port of TCP/63690 and a source port of TCP/30000.

The third spike in Figure 4.11 is from the second ranked destination IP address. As with the top ranked IP, the traffic is quite low and consistent until the huge spike at the end of October. Analysis of this traffic spike showed that of the total of 305 837 packets received on that day, 301 010 of them originated from a single IP address located in China as well.

This too was SYN-ACK traffic with a destination port of 63572 and a source port of 4220.

In the cases of the second and third spikes in traffic, the source ports for the SYN-ACK traffic are most likely the destination ports for the initial SYN packets to the targeted servers in China. These are uncommon ports with no official services allocated to them, hence it was possibly a very targeted attack or an open port on a system that was being compromised.

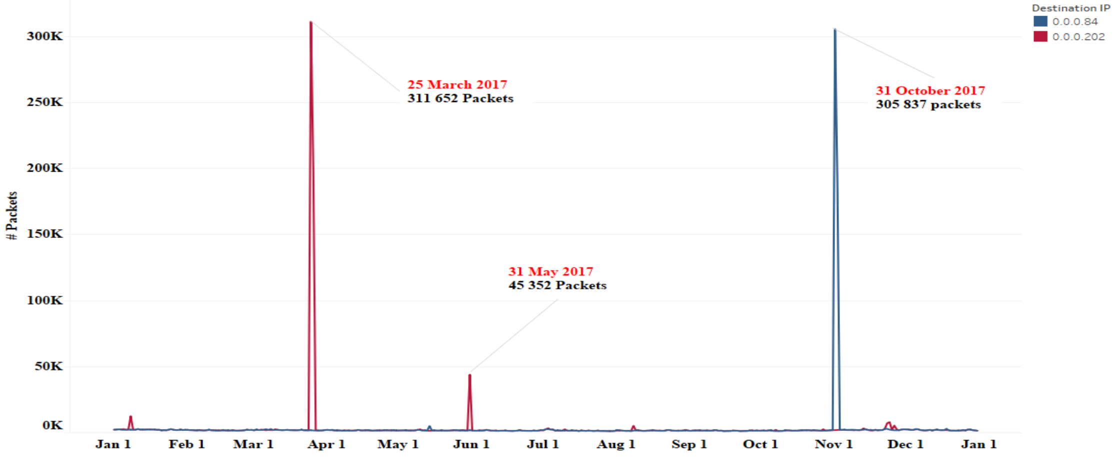


Figure 4.11: Traffic pattern for the count of unique IP addresses over time

4.9 Summary

This chapter applied the analysis methods described in Chapter 3 and presented the analysis results of this research project. The research performed an in-depth analysis of darknet traffic captured by a passive network sensor located within a South African IPv4 netblock.

The overall theme of the chapter was to discuss and present the statistics and metrics of the results and was divided into five separate sections. Section 4.1 introduced the chapter and some of the high level results. It started by discussing the three most common IP level protocols, namely TCP, UDP and ICMP. It presented the detailed analyses of lower level protocols and the individual contributions made by TCP, UDP and ICMP packets. Monthly details for traffic analysis were shown and the top contributing destination ports for both TCP and UDP were discussed. The ICMP traffic was further classified according to the type, i.e. Echo Requests, Echo Reply, Time Exceeded, etc. For all traffic types, the top contributing source IP addresses were discussed and details illustrated. Sections 4.2, 4.3 and 4.4 performed a detailed analysis of TCP, UDP and ICMP traffic respectively. Section 4.2.1 focused on the TCP flag that had been set on each packet. The TCP flag provided an indication into what traffic was purely scanning and what was deflected or backscatter traffic. Section 4.5 continued by looking at the packets and associated data received for each month of 2017. Interesting anomalies in the graphs were highlighted and discussed in this section. As TCP comprised an overwhelming majority of the traffic, a further analysis and classification of the traffic was carried out. Section 4.6 used the discussions from Section 4.2.1 as a base to build on and discuss the backscatter traffic. The final two sections, Section 4.7 and Section 4.8, looked and the source and destination IP addresses respectively. Geolocation information provided by Maxmind Lite databases within Wireshark matched each source IP address to the country that it was assigned to. This allowed classification of traffic according to country, with the top contributing countries based on packet count illustrated. The destination IP addresses were ranked according to the ones most targeted and the details of this traffic was illustrated.

Interesting and specific findings that were uncovered in this chapter will be handled as individual case studies in the next chapter.

CHAPTER **5**

Case Studies

Chapter 5 discusses a few notable case studies that were found in the results. Section 5.1 looks at possible malware activity within the traffic patterns and flows with regards to the behaviour of certain source IP addresses and ports targeted. Continuing on from the initial discussions of TCP flags in Section 4.2.1, Section 5.2 investigates the network and port scanning activities detected. The final case study, Section 5.3, discusses the high occurrence of SIP traffic in the data set. Lastly, the chapter is summarised by concluding the findings and discussions in Section 5.4.

5.1 Malware Activity

Finding traces of malware in such a large dataset is quite challenging, especially when it is only one way traffic. As previously mentioned in Section 2.3.1, TCP goes through a 3 way handshake to establish a connection and to eventually send data, hence, the handshake is never created and we therefor do not see the payload. One is however able to match certain traffic patterns to the way specific malware operates, thereby allowing one to deduce the presence of the malware based on the observations. One such example is that of the *Mirai* malware.

Mirai essentially operated as a DDoS attack on embedded IoT devices running Linux on ARM CPU architecture. Many broadband routers and IoT webcams were affected by this (Kolias *et al.*, 2017). The malware firstly scanned what appeared to be pseudo-random internet IP addresses on ports 23 and 2323. Once a vulnerable device had been discovered, a brute force password attack was launched in order to gain access to the shell via a telnet connection. Mirai had a preconfigured list of default usernames and passwords that it would use to perform the brute force attack with. When a device became compromised, the malware was executed on it, essentially turning it into a bot which continued to scan the internet for new targets (Antonakakis *et al.*, 2017).

The first cases of Mirai were detected in the beginning of August 2016, with a prolific rise in attacks following over the next three months. The source code for Mirai was publicly released by the group responsible for it, which led to many further Mirai variants spreading over the internet well into 2017 (Kolias *et al.*, 2017).

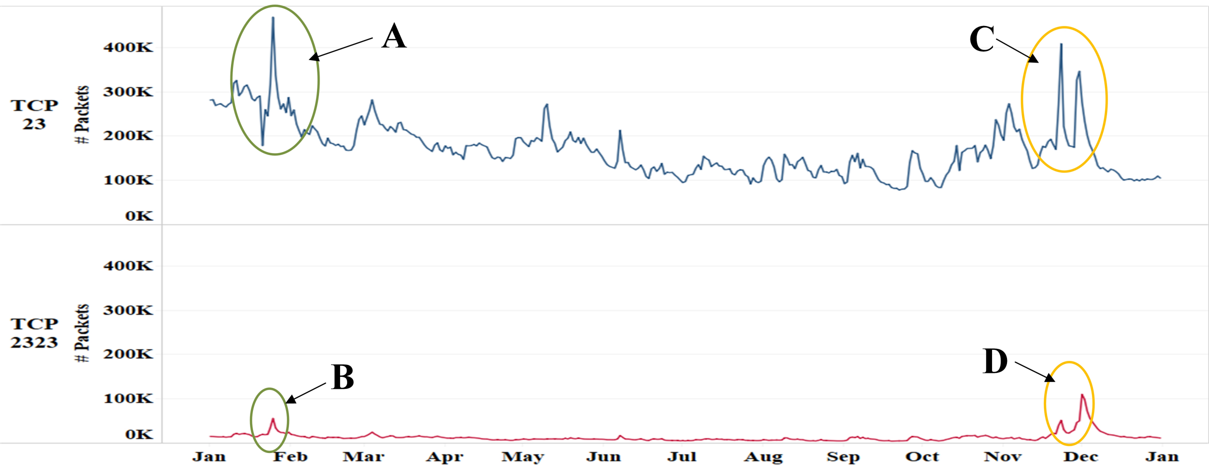


Figure 5.1: Traffic patterns over time for TCP/23 and TCP/2323

Looking at the list of top TCP ports (Table 4.2), we saw that the highest occurrence was that of tcp/23, with tcp/2323 following soon after at fourth highest. Figure 5.1 shows the traffic patterns for tcp/23 and tcp/2323, with specific points highlighted as “A”, “B”, “C” and “D”. Point “A” represents a peak in tcp/23 traffic towards the end of January, with a coinciding peak in tcp/2323 traffic marked as point “C”. A similar trend can be seen towards the end of November / beginning of December, marked as “B” and “D” respectively. Table 5.1 shows the top ten IP addresses for traffic having made made connections on both tcp/23 and tcp/2323. All IP addresses listed only sent SYN packets, meaning the connections were all initiated from these IP addresses and were not backscatter or reply packets.

Table 5.1: Top 5 IP addresses based on packet count (TCP/23 and TCP/2323)

Rank	IP Address	TCP/23	TCP/2323	Country	Owner according to WhoIs
1	185.188.207.26	203 520	22 411	Germany	Proact Deutschland GmbH
2	185.188.207.28	159 002	17 865	Germany	Proact Deutschland GmbH
3	202.98.59.37	104 925	11 726	China	Chinanet CQ
4	101.251.213.198	70 254	7 589	China	Beijing Capitalonline Data Service
5	111.62.44.99	60 571	6 939	China	China Mobile Communications Corporation

A further analysis of these IP addresses were conducted, including a deeper inspection of the individual tcp/2323 packets. All packets from the top two ranked IP addresses in Table 5.1 were extracted from the data set. Passive OS fingerprinting, as outlined in Section 3.3.5, was used to fingerprint them. All packets returned as "unknown". As tcp/23 traffic formed a much larger volume of the overall traffic and was also a more commonly used port, there was a greater chance that the traffic could contain a higher percentage of non-Mirai traffic as well. The tcp/2323 traffic was then instead taken, as it was a smaller volume of traffic and because the port is not commonly used for services on the internet. This traffic was analysed using both p0f and NetworkMiner, who both returned very similar results. Table 5.2 shows the results from p0f for all tcp/2323 traffic.

Table 5.2: Results of p0f analysis of TCP/2323 traffic

	# Packets	% Packets
All tcp/2323 packets	4 427 821	100
Unknown	3 749 034	84.66
Linux Kernel	677 863	15.30
Windows	923	0.02
Apple Macintosh	1	<0.01

The largest volume of tcp/2323 traffic could not be positively identified and returned as "unknown". This unkown packets accounted for 84.66% of the total tcp/2323 traffic. Packets identified as Linux Kernel accounted for 15.30%, with Windows packets only accounting for 0.02% of the total traffic.

It was not possible to conclusively confirm that this traffic was responsible by a Mirai variant and quite difficult to attribute it 100%, although it is quite probable due to connectivity only on tcp/23 and tcp/2323 and the fact that the operating system could either not be matched, or the ones that did match were largely from a Linux based OS. The traffic patterns in the dataset also fits in with the timeline of the Mirai variants.

One of the biggest cyber security incidents of 2017 was the WannaCry[1] Ransomware attack. This malware took advantage of a security vulnerability in the Microsoft SMB protocol (Mohurle and Patil, 2017). This was a self-propagating worm which would infect a machine and then spread across the network infecting other machines. Infected machines would have their data encrypted and message demanding a ransom would be displayed on the screen (Chen and Bridges, 2017).

The WannaCry traffic would spread by scanning the network on tcp/445, looking for a machine with the SMB vulnerability. The first recorded infection of WannaCry was on 12 May 2017 (Jones, 2017). Figure 5.2 shows the traffic patterns for tcp/445 and the traffic patterns for number of unique IP addresses over time. The graph shows fairly constant activity of between January and April, with a continuous sharp rise in traffic from May onwards. There is also a sharp rise in unique source IP addresses from May 2017 until the end of the year.

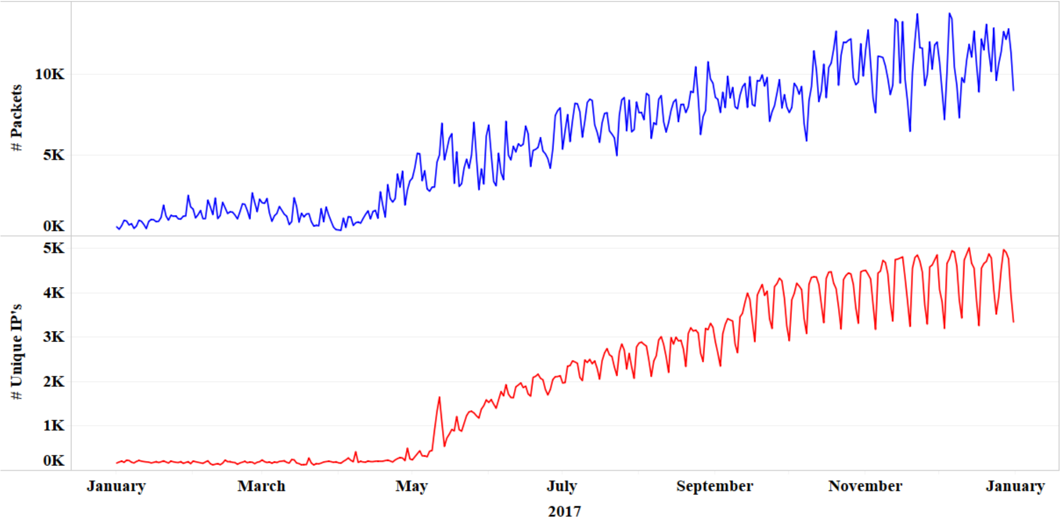


Figure 5.2: Traffic patterns for TCP/445 and unique source IP's over time

More than 99% of the tcp/445 traffic was found to only be SYN packets. Because TCP requires a 3 way handshake to establish the connection, as previously described in Section 2.3.1, only the TCP SYN packet header fields were recorded. Without the payload information, this traffic cannot actually be confirmed as WannaCry scanning. However, the sharp increase in tcp/445 traffic from May 2017 onwards, coincides with the timeline of the WannaCry attack (Jones, 2017).

Table 5.3 displays the top 10 countries, based on packet count, where tcp/445 connections originated from. According to Ghosh (2017), systems in approximately 150 countries were

[1] https://en.wikipedia.org/wiki/WannaCry_ransomware_attack

Table 5.3: Top 10 countries for TCP/445 traffic based on packet count

Rank	Country	# Packets	% Packets
1	United States	377 372	17.41
2	China	224 446	10.36
3	Hong Kong	206 599	9.53
4	Netherlands	180 785	8.34
5	Russia	115 579	5.33
6	India	113 437	5.23
7	Indonesia	105 940	4.89
8	Vietnam	78 200	3.61
9	Seychelles	69 856	3.22
10	France	40 133	1.85
Σ		1 512 347	69.78

affected by WannaCry, with Russia and India being two of the worst affected. All the countries listed in Table 5.3 were also listed by Ghosh (2017) as countries affected by WannaCry.

5.2 Network and Port Scanning

The differences between network and port scanning has previously been discussed in section 2.7.1. It is essentially a series of continuous connections to IP addresses on various ports with the intention of checking which services are active. It can be seen as a type of reconnaissance, usually done over short periods of time.

The top five source IP addresses for all traffic, as briefly discussed previously in 4.7, were further analysed to get a fuller understanding of it. Figure 5.3 displays the traffic patterns for the top five IP addresses over time. Table 5.4 displays some extended statistics about the top five.

Looking at the traffic patterns in Figure 5.3 and statistics from Table 5.4, they can be divided into roughly two groups. IP addresses 94.102.49.7, 77.72.82.80 and 74.125.206.197 connected to multiple destination ports which spanned the entire destination IP range of the network telescope. IP addresses 163.172.135.224 and 185.94.111.1 also connected to the full range of destination IP addresses of the network telescope as well, but only

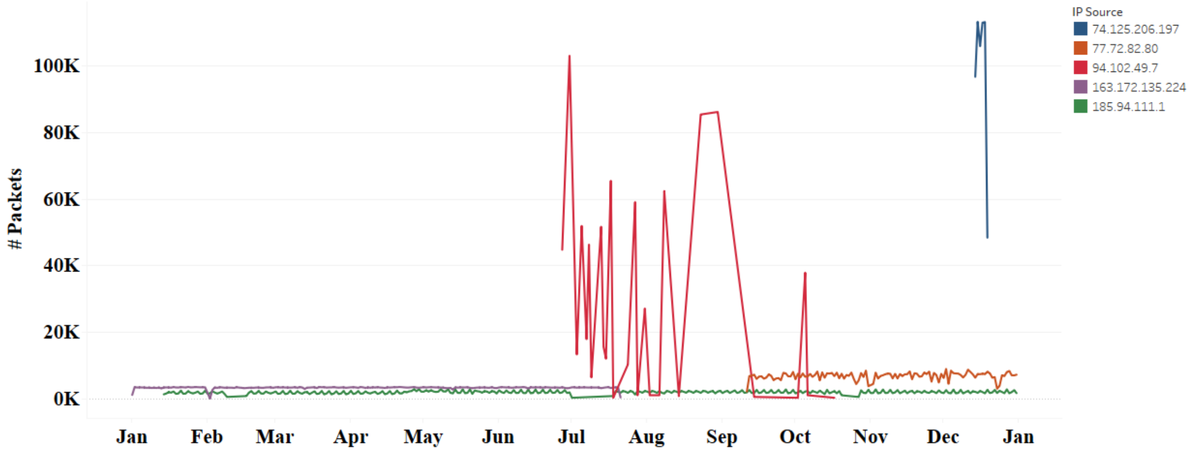


Figure 5.3: Traffic patterns of top 5 source IP's over time

connected on a handful of destination ports.

Table 5.4: Extended statistics of top 5 source IP addresses

Rank	IP Address	# Unique TCP Ports	# Unique UDP Ports	# Distinct Dest IP's	Sum Dest IP's
1	94.102.49.7	800	None	256	591 265
2	77.72.82.80	2 615	None	256	758 357
3	163.172.135.224	1	None	256	803 554
4	185.94.111.1	6	9	256	662 807
5	74.125.206.197	20 536	None	256	625 962

The first group of IP addresses performed a very typical type of port scan in that a large array of ports were checked for activity. These scans were much more aggressive and done for a short period of time. In analysing the destination ports being targeted by this group of IP addresses, the following interesting details were picked up.

- IP 94.102.49.7 only appeared to scan the lower and middle end ports, with certain exceptions found in the higher range, i.e. scanned ports were mainly in the range of ports 1 - 3388 and 6500 - 45000
- IP 77.72.82.80 only appeared to scan ports within the middle range with certain exceptions found in the lower and higher ranges, i.e. scanned ports were mainly in the range of ports 3388 - 6500
- IP address 74.125.206.197 only scanned ports in the high end, i.e. scanned ports were mainly in the range of ports 45000 - 65535
- Very seldom did any of these IP addresses scan the same ports
- No UDP ports were scanned by these IP addresses

In the second group, a total of 1 288 768 scans were performed across a total of 16 destination ports. When referring to Figure 5.3, we notice that there was a very low intensity scan done by IP address 163.172.135.224 from January up until August. The only port scanned was TCP 3128, which is generally a port used by a web proxy server such as squid. Initially this could be mistaken for a misconfigured server or device due to the length of the scans and the limitation of the single destination port. However, when looking at the actual packets in Wireshark, the packets arrive in bunches of the same source ports trying to connect to random destination IP addresses. A few seconds later the same is repeated, but with different source ports and different random destination IP addresses. This is typical of an automated scan trying to find an open web proxy server. With IP address 185.94.111.1 we see that the there is also a very low intensity scan performed for nearly the entire 2017 period. Figure 5.4 shows the packet distribution of IP 185.94.111.1 for all the destination ports over time.

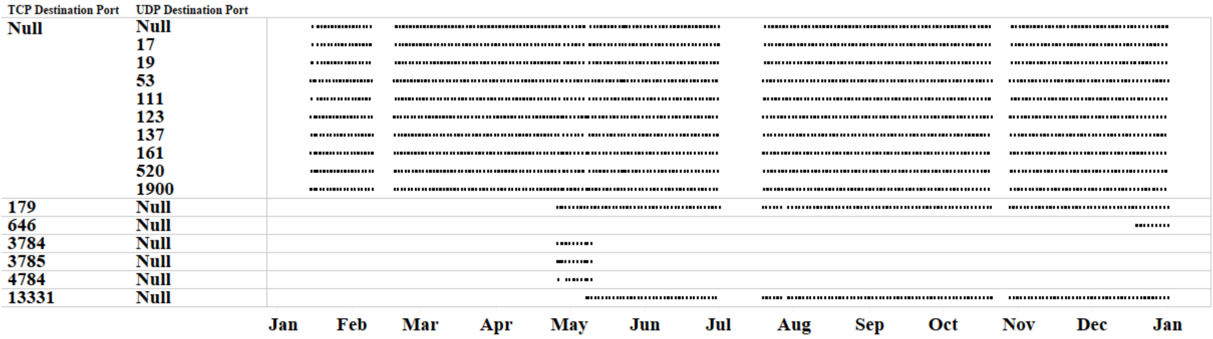


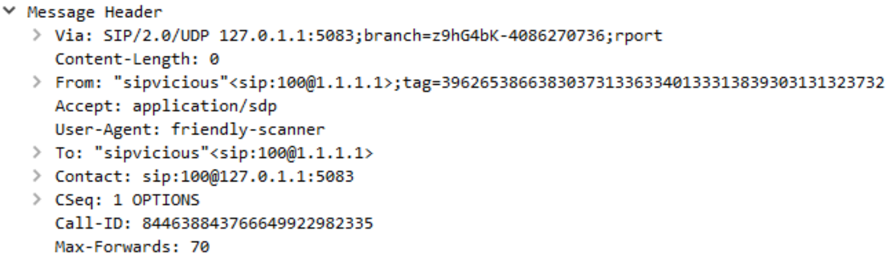
Figure 5.4: Traffic patterns of top 5 source IP's over time

The UDP ports were scanned continuously throughout the entire year, with intermittent breaks across all destination ports for short time periods only. The TCP ports on the other hand were each scanned for only specific time periods. The list of TCP and UDP ports can be seen in Figure 5.4.

5.3 Session Initiation Protocol

As introduced in Section 2.5, SIP traffic operates independently of the underlying protocol and can be encapsulated in either TCP or UDP. All the SIP traffic recorded in this data was found to be encapsulated in UDP. The total SIP traffic only accounted for 2.63% of the total packets received in 2017, but it was responsible for a massive 71.14% of the total data for 2017. Because the SIP traffic was encapsulated in UDP, it allowed the packet sizes to be much larger. All TCP packets which were received were no larger that 60 bytes, whereas the average UDP encapsulated SIP packet was about 450 bytes. Raftopoulos *et al.* (2015) states that the packet sizes for their UDP encapsulated sipscan traffic ranged between 382 and 451 bytes. The vast majority of SIP traffic was found to be directed towards udp/5060, which received 70.81% of all SIP traffic. However, there were many other ports which were targeted as well. In total, 722 different UDP ports received SIP traffic.

Figure 5.5 shows a screenshot of a portion of a single SIP packet displayed in Wireshark. This screenshot was taken of the actual research data. The header fields that were described in Section 2.5 can be seen in the figure.



```
⌄ Message Header
  > Via: SIP/2.0/UDP 127.0.1.1:5083;branch=z9hG4bK-4086270736;rport
    Content-Length: 0
  > From: "sipvicious"<sip:100@1.1.1.1>;tag=3962653866383037313363340133313839303131323732
    Accept: application/sdp
    User-Agent: friendly-scanner
  > To: "sipvicious"<sip:100@1.1.1.1>
  > Contact: sip:100@127.0.1.1:5083
  > CSeq: 1 OPTIONS
    Call-ID: 844638843766649922982335
    Max-Forwards: 70
```

Figure 5.5: Screenshot of SIP header, as seen in Wireshark

Figure 5.6 illustrates the traffic patterns for all SIP traffic in 2017. There is a fairly average and steady traffic flow throughout the year, with only a slight increase from January through to December, as indicated by the trend line.

Apart from the slight increase in overall traffic through the year, there are distinct spikes in traffic observable on particular days. These spikes occurred throughout the year. The top five traffic spikes through the year, which are highlighted in Figure 5.6, were further analysed and are listed below in order of greatest to smallest.

- **14 April 2017** - The largest spike of traffic occurred on this day, with a total of 102 674 SIP packets that were received, accounting for 1.58% of the total traffic for 2017. This traffic was responsible for 45MB of data. There were 39 unique IP addresses

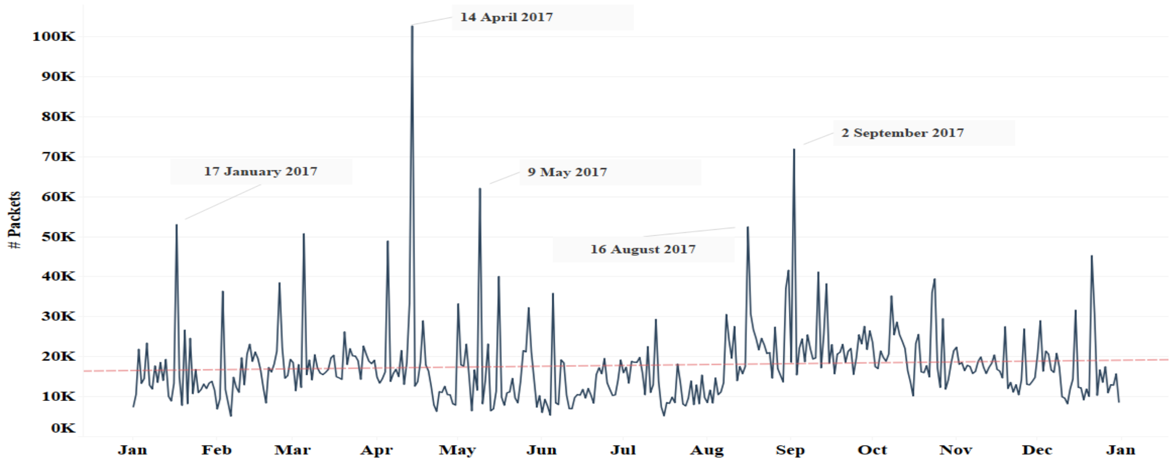


Figure 5.6: SIP traffic pattern for 2017

responsible for the SIP traffic on this day. Of these IP addresses, the top three were responsible for 84.27% of the total traffic for the day. The top three IP addresses in order from top to bottom are 213.202.254.134 (49 151 packets), 5.189.136.244 (27 133 packets) and 155.94.65.2 (10 240 packets).

- **2 September 2017** - The second largest traffic spike occurred on this date, with a total of 71 638 SIP packets received from 57 unique IP addresses. Of the source IP addresses, the top two accounted for 82.54% of the total traffic for this day. These two IP addresses, in order of top to bottom, are 51.15.143.144 (29 952 packets) and 92.4.108.203 (29 103 packets). The first source IP address originated from France and the second IP address originated from Switzerland.

- **9 May 2017** - There were a total of 61 828 SIP packets received coming from 53 unique IP addresses on this day. These unique IP addresses originated from only eight countries, with the United States taking the lead with a total of 27 IP addresses. The top IP address (155.94.88.18) according to packet count, originated in the United States and was responsible for 54.24% (33 536 packets) of the SIP traffic for that day.

- **17 January 2017** - There was a total of 53 015 packets received on this day, coming from 32 unique IP addresses. The top IP address (104.193.11.107) originated from the United States and accounted for 33 280 packets, which is 62.77% of the total SIP traffic for that day.

- **16 August 2017** - There was a total of 52 196 SIP packets received from 67 unique IP addresses on this day. The top IP address according to packet count, originated from Switzerland and was the only IP address from this country on this day. This

IP address, which is 92.42.108.203, accounted for a total of 27 091 packets, which is 51.90% of the total SIP traffic for the day.

As very briefly touched on previously, the SIP traffic had much larger packet sizes than the other packets. For the full period of 2017, there were a total of 6 471 918 SIP packets received, accounting for approximately 9 GB of data. When compared to the total packets and data received for the full 2017 period, the SIP traffic accounted for only 3.71% of the total packets, yet it accounted for 21.06% of the total data. Table 5.5 displays the top 10 IP addresses based on SIP packet count. Figure 5.7 further shows the traffic patterns for the top five of these IP addresses.

Table 5.5: Top 10 IP addresses for SIP traffic

Rank	IP Address	Packets	% Packets	Data (MB)	Country
1	163.172.215.161	508 588	7.85	220	Netherlands
2	92.42.107.139	190 005	2.93	81	Switzerland
3	146.0.243.29	142 336	2.19	61	Germany
4	62.210.36.129	120 057	1.85	52	France
5	92.42.108.203	102 097	1.57	44	Switzerland
6	51.15.209.185	88 320	1.36	38	France
7	155.94.89.42	67 071	1.03	29	United States
8	51.15.12.233	63 221	0.97	27	Netherlands
9	51.15.87.3	60 928	0.94	26	France
10	155.94.65.2	50 176	0.77	22	United States

All the top five source IP addresses had multiple connections to every destination IP address in the network telescope range. This is very typical network scanning that is taking place. Looking at the traffic patterns of the top five IP addresses, we see that there are two distinct scanning methods occurring. The first method sees a very low number of connections over a very long period of time. The second method sees a much more erratic scan, with higher number of connections over a shorter period of time. When viewing the packets in Wireshark, further details such as the SIP Message Header can be seen, which gives more insight into the source of the traffic. An example of this message header can be seen in Figure 5.5. Over 99% of the SIP traffic had an indication that the traffic was generated by a program called "SIPVicious"[2]. SIPVicious is an open source collection of software tools used to scan SIP enabled Voice over IP systems. It is often used in penetration testing. This indication can be seen in the "From:" field in the mes-

[2] http://sipvicious.org

sage header where it states the sender's name as "sipvicious", as can be seen in Figure 5.5.

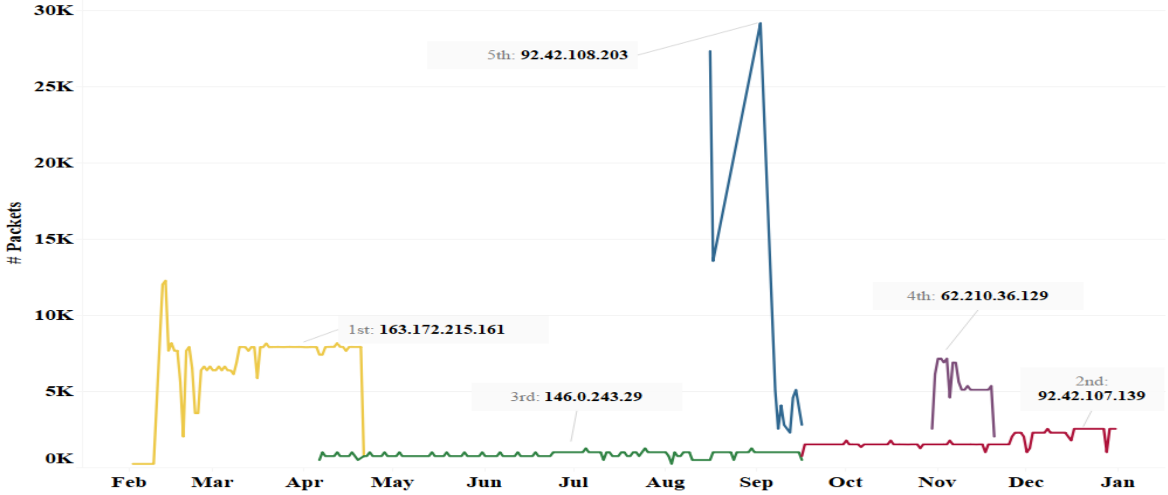


Figure 5.7: Top 5 source IP addresses traffic patterns

A macro analysis of the geolocation of source IP addresses for SIP traffic was conducted. SIP packets were received from IP addresses coming from 60 distinct source countries. Table 5.6 shows the statistical data for SIP traffic, based on the geolocation information of the source IP addresses for the top 10 countries. As can be seen, the majority of SIP traffic originated from IP addresses in France. It accounted for nearly a quarter of all SIP packets as well as nearly a quarter of all SIP data in 2017. The table also shows that the top five countries combined, accounted for roughly 84.8% of all SIP traffic and data in 2017. Furthermore, despite receiving packets from 60 different countries, IP addresses coming from the top ten countries were responsible for 97% of all SIP traffic and data in 2017.

Table 5.6: Top 10 countries for SIP traffic based on packet count

Rank	Country	# Packets	% Packets	Data (MB)	% Data	# Unique IP's
1	France	1 581 059	24.42	683	24.42	496
2	Netherlands	1 263 771	19.52	544	19.45	263
3	Germany	1 083 011	16.73	471	16.84	418
4	United States	819 029	12.65	352	12.58	735
5	United Kingdom	743 323	11.48	322	11.51	184
6	Switzerland	351 002	5.42	151	5.40	17
7	Russian Federation	188 820	2.91	81	2.89	117
8	Canada	156 888	2.42	69	2.46	91
9	Lithuania	74 751	1.15	33	1.18	37
10	India	24 021	0.37	10	0.35	23

In the research conducted by Raftopoulos *et al.* (2015), they analysed telescope data from three different sources. The SIP scanning traffic was categorised according to country based on packet count. In their top 10 countries, six of the top 10 countries in this research list also appeared in theirs. The countries are India, Russian Federation, Switzerland, Germany, United States and Canada. Figure 5.8 shows a stacked bar graph of the top five country's packets for each month of 2017.

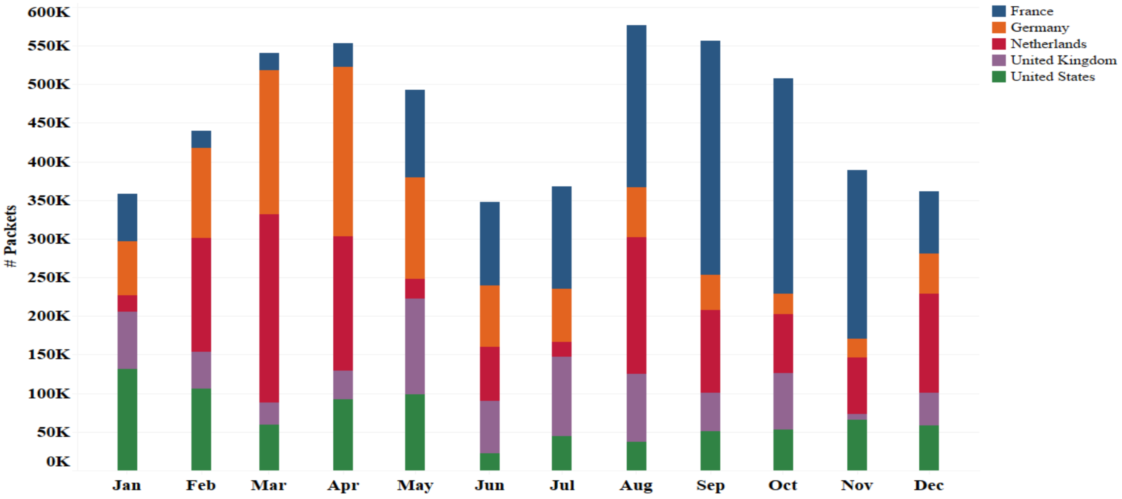


Figure 5.8: SIP traffic by top 5 countries per month

5.4 Summary

Chapter 5 dealt with notable case studies found in the results. This section built on the results in the previous section, but did a more granular analysis of it. The chapter was divided into four specific sections, each dealing with a specific case study.

Section 5.1 started off the chapter by discussing the presence of malware activity discovered in the data, specifically instances of Mirai like traffic were identified. Section 5.2 performed a detailed investigation of the network and port scanning traffic. The top five source IP addresses were looked at and traffic patterns illustrated. The monthly high and low occurrences were highlighted as well as the top contributing source IP addresses. Lastly, Section 5.3 analysed the source of the SIP traffic, with specific reference to source IP address and geolocation of source traffic.

CHAPTER 6

Comparison with Previous Work

This chapter will compare the results of this study against the results of previous research done in this field of study, specifically research done at Rhodes University on all the different data sets collected in the past 12 years.. A detailed analysis of this specific data set and the manner in which it was analysed had never been done before, which is one of the reasons why it was chosen. There have in the past however been various other studies done on the broader data sets collected by Rhodes University, albeit with regards to different aspects of the data. The comparison will focus on the following studies and their results, as listed in Table 6.1.

Table 6.1: Previous research used in the comparison

Research Subject	**Authors and Year**
Towards a Taxonomy of Network Scanning Techniques	Barnett and Irwin (2008)
A Framework for the Application of Network Telescope Sensors in a Global IP Network	Irwin (2011)
A Network Telescope Perspective of the Conficker Outbreak	Irwin (2012)
A Baseline Study of Potentially Malicious Activity across Five Network Telescopes	Irwin (2013)
A System for Characterising Internet Background Radiation	Yates (2014)
Observed Correlations of Unsolicited Network Traffic over Five Distinct IPv4 Netblocks	Irwin and Nkhumeleni (2015a)
Effectiveness of Sampling a Small Sized Network Telescope in IBR Data Collection	Chindipha *et al.* (2018)

The previous research, as listed in Table 6.1, dealt with many different aspects of IBR and would not necessarily have analysed the data as was done in this research project. The comparison will therefore be limited to the following aspects:

- IBR traffic volumes
- Traffic composition
- Targeted destination ports
- Geographic location of source IP address

6.1 IBR Traffic Volumes

As previously mentioned in Section 3.3.1, the data used by Irwin (2011) was collected over a five year period and across five /24 IPv4 telescopes. This was the first detailed analysis done on the data collected by the Rhodes telescopes. The total volume of traffic for the duration was approximately 23 million packets. Over a very similar time period, Irwin (2012) recorded 16 million packets over a four year period from a single /24 sensor. Yates (2014) notes that the volume of packets reaching the five sensors differ quite substantially. The 'older' 196/8 sensors received almost double the amount of packets to the newer 146/8 and 155/8 sensors. The 155/8 sensor, which is the same one used in this study, received approximately 9 million packets over a six month period. The data for Chindipha *et al.* (2018) was collected on the 155/8 sensor in February 2018. A total of 27.5 million packets were collected over that single month. The data for this research project consisted of approximately 174 million packets collected from a single /24 sensor over a 12 month period. The earliest network telescope was commissioned in 2005, which means that it has been operating for approximately 14 years. In that time, IBR has increased from under 1 million packets per month to currently almost 28 million packets per month. The arrival rates of the packets have increased tremendously too. Irwin (2011) recorded the packet rate to be at around 18 packets per minute in 2009. The average packet rate in the data set in this study (2017) had grown to around 300 packets per minute.

6.2 Traffic Composition

A comparison of IBR composition at the IP layer is a good indicator of how internet traffic patterns have changed over the years. In this section the three most common lower level protocols will be compared, i.e. TCP, UDP and ICMP. Results for this study was previously mentioned in Section 4.2 and found the composition as follows: TCP accounted for the largest portion of overall traffic, accounting for 89.70% of the total traffic. UDP traffic accounted for 9.52% and ICMP accounted for 0.77%. Table 6.2 shows the protocol comparison as found in the previous studies. All statistics are for the 155/8 telescope except for the Irwin (2011). It is being included as it was the earliest work done on the Rhodes telescopes' data over the period 2007 - 2010.

Table 6.2: Comparison of IP protocols

Author and Year	**Data Period**	**% TCP**	**% UDP**	**% ICMP**
This Research (2017)	2017	89.70	9.52	0.77
Irwin (2011)	2005 - 2009	81.57	12.62	5.89
Irwin (2013)	2011 - 2012	77.18	14.45	8.36
Yates (2014)	2013 - 2014	57.84	37.21	4.94
Nkhumeleni (2014)	2011 - 2012	76.34	14.79	8.86

Table 6.2 shows that TCP has always been the most dominant protocol, with UDP and ICMP in at 2^{nd} and 3^{rd} respectively. The decrease in TCP and increase in UDP reported by Yates (2014) has been attributed to the presence of SQL Slammer worm during that time period. The worm operated on udp/1434, hence the increase in overall UDP traffic. 2017 saw a significant decrease in overall ICMP traffic when compared to previous years' findings. The full year's protocol analysis is shown in Table 4.1. The highest percentage of ICMP traffic was recorded in August 2017, with only 1.02% in total for the month. This may be an indication that ICMP echo is no longer preferred for simple network host detection scan, but rather applications such as ZMap[1] which is able to scan using TCP SYN packets at quite high rates.

6.3 Targeted Destination Ports

The top TCP and UDP destination ports offer a good insight into what applications are most targeted on the internet. During certain periods of worm or virus activity, spikes in particular destination ports are noticed immediately. Tables 4.2 and 4.5 have previously shown the top TCP and UDP identified in this research respectively. Tables 6.3 and 6.4 compares the top TCP and UDP destination ports with previous research. The second column in each of these tables contains the port statistics from this research.

[1] An open source network scanner - https://zmap.io/

Table 6.3: Comparison of top 10 TCP destination ports

Rank	This Research	Irwin (2011)	Irwin (2013)	Yates (2014)	Nkhumeleni (2014)
1	23	445	3389	22	3389
2	22	135	1433	3389	1433
3	1433	139	445	80	80
4	2323	22	80	23	445
5	5358	1433	57471	8080	57471
6	7547	2967	22	1433	22
7	3389	5900	8080	445	8080
8	445	23	23	443	23
9	80	80	3072	5900	1234
10	3128	50272	135	1234	1024

For the top 10 TCP ports identified in this research, seven of them appear in the top 10 lists of previous work. The ones who appear in previous research are TCP ports 23, 22, 1433, 3389, 445, 80 and 3128. These are all very common ports and it is understandable that they would appear in a top 10 list. The ports who do not appear in previous research are 2323, 5358 and 7547. As previously mentioned in Section 5.1, tcp/2323 was identified as being associated with variants of the Mirai malware. The timeline for the tcp/2323 traffic fits in well with that of the activity of the Mirai variants. As Mirai also scanned on tcp/23, a distinct correlation in traffic patterns have also been observed between tcp/23 and tcp/2323, as illustrated in Figure 5.1. The remaining two ports that do not appear in previous results are tcp/5358 and tcp/7547. de Bruijne *et al.* (2017) state that there has been an increase in activity on these ports over the past few years due to the fact that these ports are associated with various IoT devices. They are used the for remote connectivity or management of devices. IoT has only started appearing over the last few years, which would explain why these ports were never seen in top 10 lists before now. The most targeted TCP ports, which appear in the top 10 lists of all the comparative research, are 22, 23, 80, 445 and 1433.

For the UDP traffic listed in Table 6.4, only a single port has consistently appeared in the top 10 list of this research and all previous research. This port is udp/1434, which is used by Microsoft SQL Server. Another interesting port, which appears in the top two of all research after Irwin (2011), is udp/5060. As previously discussed in Sections 4.3 and 5.3, this port is commonly used by SIP traffic. SIP scanning has become quite prevalent on the internet and occupies a large portion of IBR traffic. DNS typically runs on udp/53 and is also found in the top 10 of all research other than Irwin (2011). DNS

Table 6.4: Comparison of top 10 UDP destination ports

Rank	This Research	Irwin (2011)	Irwin (2013)	Yates (2014)	Nkhumeleni (2014)
1	5060	1434	5060	53	5060
2	1900	137	1434	5060	1434
3	123	1026	137	19	6257
4	53	1027	6257	3544	137
5	53413	38293	32737	1434	53
6	161	19932	53	6588	6568
7	137	135	6568	161	60505
8	19	1028	60505	12	43815
9	111	1029	43815	39455	32737
10	1434	5158	39455	19222	39455

cache poisoning has been an attack vector for many years and continues to be targeted still. This method gets a DNS server to store incorrect or malicious DNS details about a webpage or host. Subsequently when valid requests are sent to the DNS server for this particular webpage or host, the the malicious details are returned to the victim, who then inadvertently accesses the malicious link (Trostle *et al.*, 2010).

6.4 Geolocation of Source IP Address

The only research to fully report the geolocation findings of source IP address analysis is Irwin (2011). Table 6.5 displays the comparison of top 10 countries based on geolocation of the source IP addresses. The statistics on the left side are from the results of this research and those on the right are from Irwin (2011). Looking at the list of top 10 countries, five of the countries from this research can also be found in the previous research. These countries are China, United States, Russia, Brazil and Korea. South Africa is ranked at 3^{rd}, but if it were to be removed from the list, then the top three ranked countries would be the same in both sets of results.

It's interesting to note that the percentage volume of traffic for China, United States and Russia remain very similar between the two sets of results, despite there being approximately a 10 year gap between when the packets were recorded. Currently, South Africa does not even appear in the top 10 list and in actual fact, is ranked as 38^{th} in the results of this research. It only accounts for 0.28% of the traffic compared to 9.51% previously. Egypt, which is ranked 10^{th} in the previous research, has also dropped quite a bit now

Table 6.5: Comparison of top 10 countries based on geolocation of the source IP addresses

	This Research		Irwin (2011)	
Rank	**Country**	**% Packets**	**Country**	**% Packets**
1	China	21.33	China	19.73
2	United States	13.51	United States	10.94
3	Russia	6.50	South Africa	9.51
4	Netherlands	5.13	Russia	5.55
5	Brazil	4.35	Taiwan	3.65
6	Korea	3.64	Brazil	3.53
7	India	3.62	Germany	3.11
8	United Kingdom	3.33	Korea	2.66
9	Vietnam	2.83	Italy	2.58
10	France	2.75	Egypt	2.42
Σ		66.99		63.68

and is listed in 32^{nd} place currently. Previously it contributed to 2.42% of the traffic, but currently it only represents 0.44% of the total traffic. When summing up the contributions of the top 10 countries, it is interesting to see that the volumes for both sets of data are very close, with an approximate difference of about 3%.

6.5 Independent Research Outside of Rhodes

Much of the seminal work in IBR research can be attributed to Moore *et al.* (2004); Pang *et al.* (2004); Barford *et al.* (2006); Wustrow *et al.* (2010), where many of the terms used today were coined. This work laid the foundation for the future research done using network telescopes.

Moore *et al.* (2004) discussed what the actual network telescope was and how it could be used as a resource to study internet events. Pang *et al.* (2004) did an analysis of IBR traffic and produced many results similar to this research. Their results were used as a comparison throughout the results of this research. Pang *et al.* (2004) looked at the sources of the traffic, the protocols and ports used and pointed out the differences between scanning traffic and reflected backscatter traffic. They provided a characterisation of the traffic and was able to demonstrate the significance of the background radiation. Barford *et al.* (2006) built on this work by identifying and classifying malicious source IP addresses.

The aim of the research was to understand the distribution of these source IP addresses to see where the traffic originates.

The subsequent work done in this area has assisted in identifying and studying many of the global internet events including the SQL Slammer worm (Moore *et al.*, 2003), Code Red worm (Moore *et al.*, 2002) and Conficker (Hick *et al.*, 2009; Irwin, 2012).

6.6 Summary

The comparisons discussed in this chapter were the most basic statistics from the results being compared. The findings of this research was compared to previous, with specific focus on the work done at Rhodes. It did however briefly look at and compare the results to other work in the larger research fields. The discussions covered in this chapter are a good indicator of how IBR has evolved through the years.

CHAPTER 7

Conclusion

This chapter summarises the thesis and concludes the discussion on the findings of this research. The chapter starts with Section 7.1 providing a summary of each chapter of this thesis. Section 7.2 addresses the initial project goals and objectives and evaluates the findings. Section 7.3 then looks forward to possible future research in this field of study. Lastly, Section 7.4 provides the final remarks on the thesis and the significance of research in this field.

7.1 Research Project Overview

This thesis paper consists of seven chapters, with this concluding chapter being the seventh. Each chapter contains multiple sections, each discussing and presenting a different aspect of the subject matter as it relates to the objectives of the project.

Chapter 1 started by introducing the subject matter of this research project. The problem statement and research objectives and goals were clearly defined in this chapter. Building on from that, the scope of the research was outlined and the methodology used was further explained.

Chapter 2 conducted a review of previous literature and research dealing with IBR and network telescopes. The chapter started by introducing the subject and discussed a classification scheme used for network telescopes, based on the netblock being monitored and the active hosts found within the netblock. The size of the network telescope was investigated to determine the impact it had on observing random packets on the internet. Because network telescope traffic is only TCP/IP, the analysis of the most dominant lower level protocols (TCP, UDP and ICMP) were looked at in great detail. Along with the lower level protocols, the application level ones were also discussed as part of the traffic analysis and classification. Key research focus areas of previous work were investigated, which created a platform for the work that was done in this study. IBR has been able to provide valuable insight for the research of global network events. The main events being internet worm and malware activity, network and port scanning activities and distributed denial of service attacks. This chapter looked at some of the seminal work done at CAIDA.

Chapter 3 was divided into two sections. The first section presented the the data set and expanded on the source of it, how it was collected and provided a brief overview of the initial analysis. This included the basic statistics regarding the packet count, the number of unique source IP addresses, data count and the rate of data flow. The second section took a look at the raw data files and discussed the tools that were used to extract the relevant data, the storage of the outputted data, the analyse it and eventually the visual presentation of the findings. This section also discussed any other tools that were used in the analysis of the data.

Chapter 4 presented the findings of the analysis. A large portion of this chapter was dedicated to the statistics and metrics which were derived from a very detailed analysis of the data. This included a comparison of TCP, UDP and ICMP traffic. Within these lower level protocols, the application level protocols were also categorised and traffic patterns compared. The top TCP and UDP destination ports were listed, along with the common usage of the ports and the security risks associated with them. The largest portion of traffic was TCP, which was then subsequently further categorised according the the flag that had been set on each packet. The categorisation based on TCP flag allowed for the traffic to be identified as either pure scanning traffic, or reflected traffic from internet events such as DDoS attacks or malware activity.

Chapter 5 looked at the findings of the previous chapter and highlighted four specific case studies to be further investigated and analysed. The first case study looked at possible Mirai like malware activity discovered in the data set. The second case study specifically looked at the network and port scanning traffic from the top source IP addresses based on packet count. This traffic was identified by the SYN TCP flag that had been set. The third case study performed a deeper analysis of the backscatter traffic and attempted to identify the sources of the packets. The final case study examined the high occurrence of SIP scanning traffic.

Chapter 6 looks at some of the key results of this research and compares it to the findings of previous ones. The commonalities and differences in findings are highlighted and illustrated in this chapter.

7.2 Project Goals and Objectives Revisited

This research project stated in Section 1.2 the goals and objectives set out. There are two primary objectives and four supporting or secondary objectives. In order to evaluate the effectiveness of this research project, the goals and objectives are revisited to see if they have been achieved.

7.2.1 Primary Objective 1

The first primary objective was to compare the results of the IBR analysis from this research with the results of previous work, specifically with regards to research done in collaboration with Rhodes University. This objective was fully met, as was demonstrated in Chapter 6. Besides the comparison presented in Chapter 6, within Chapters 4 and 5 certain findings were compared with research done in the greater global research communities.

7.2.2 Primary Objective 2

The second primary objective was to see if any significant anomalies could be detected in the traffic that pertained to any global malware attacks. Chapter 5 discussed different case studies, one of which dealt with the presence of Mirai like scanning traffic which had been detected. This objective was therefor fully met as well.

7.2.3 Secondary Objectives

The secondary objectives, albeit seemingly secondary in nature, played an integral role in providing the evidence for and supporting the primary objectives. The secondary objectives required a granular level analysis of the data in order to support and provide evidences for the primary objectives. The secondary objectives also required that observations, trends and conclusions be derived from the analysis of the traffic. Chapter 4 provided a detailed analysis of the traffic which included statistics and metrics of the results. These objectives were therefor fully met as well.

7.3 Future Work

This research focused on the analysis of IBR from a single /24 IPv4 network telescope. As mentioned before, Rhodes University operates this telescope along with four other /24 IPv4 telescopes. The netblocks being monitored by all these five telescopes are non-contiguous to each other. As stated by Nkhumeleni (2014), the logical distances between the netblocks being monitored are quite substantial. All traffic reaching these telescopes are currently being captured and are being archived. A detailed analysis of the captured traffic, as was performed in this research, has not been carried out on the data from the other telescopes in the past few years. It would be interesting to analyse the traffic from the remaining four telescopes in order to determine if the same traffic patterns are observable across them. Moore *et al.* (2004) states that the probability of observing random packets on the internet is greater with larger netblocks. Much of the previous research done at other institutions used large /8 darknets, which even if it was split into smaller netblocks, would be contiguous to each other. Analysing the traffic from five random non-contiguous netblocks should surely provide some valuable insight.

The study of darknets and IBR has been ongoing for almost two decades, with much progress made in the identification and analysis of the traffic. Nearly all this work has been done on IPv4 networks. This is understandable as IPv6 has not yet been adopted and very limited ranges are currently routed on parts of the internet. It was previously mentioned that when IPv6 does eventually get implemented globally, there will be a large increase in internet traffic as it will be implemented alongside IPv4 at that time. As stated by Irwin (2011), no IPv6 packets were detected in the data set from that study, which ran over a five year period. The data from that research was however collected 10 years ago. Implementing an IPv6 telescope to detect the implementation of IPv6 over the years would provide valuable insight as it changes and the movement from IPv4 to IPv6.

7.4 Final Remarks

The work done on IBR identification and detection has had a significant impact on global internet pollution research. This has led to strategies and mechanisms to limit the impact that it has. It is unfortunately a “game” where the poles keep shifting, which means that continuous research is required to stay ahead of new methods. This research has also played a crucial role in the post analysis of various malware activity. Certain malware attacks were able to be studied from the start of the infection till it eventually died down. This research now needs to be used to implement an autonomous IBR and malware detection system that can be used to lower the impact that it has on the internet.

References

Adrian, D., Durumeric, Z., Singh, G., and Halderman, J. A. Zippier ZMap: Internet-Wide Scanning at 10 Gbps. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*. USENIX Association, San Diego, CA, 2014.

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D. *et al.* Understanding the Mirai Botnet. In *Proceedings of the 26th USENIX Security Symposium (Security)*. 2017.

Bailey, M., Cooke, E., Jahanian, F., Myrick, A., and Sinha, S. Practical Darknet Measurement. In *Information Sciences and Systems, 2006 40th Annual Conference on*, pages 1496–1501. IEEE, 2006. DOI:10.1109/CISS.2006.286376.

Bajpai, P., Sood, A. K., and Enbody, R. J. The Art of Mapping IoT Devices in Networks. *Network Security*, 2018(4):8–15, 2018. DOI:10.1016/S1353-4858(18)30033-3.

Balkanli, E., Alves, J., and Zincir-Heywood, A. N. Supervised Learning to Detect DDoS Attacks. In *Computational Intelligence in Cyber Security (CICS), 2014 IEEE Symposium on*, pages 1–8. IEEE, 2014. DOI:10.1109/CICYBS.2014.7013367.

Barford, P., Nowak, R., Willett, R., and Yegneswaran, V. Toward a Model for Source Addresses of Internet Background Radiation. In *Proc. of the Passive and Active Measurement Conference*. 2006.

Barnett, R. J. and Irwin, B. Towards a Taxonomy of Network Scanning Techniques. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, pages 1–7. ACM, 2008. DOI:10.1145/1456659.1456660.

Caceres, R., Duffield, N., Feldmann, A., Friedmann, J. D., Greenberg, A., Greer, R., Johnson, T., Kalmanek, C. R., Krishnamurthy, B., Lavelle, D. *et al.* Measurement and Analysis of IP Network Usage and Behavior. *IEEE Communications Magazine*, 2000. DOI:10.1109/35.841839.

Chen, Q. and Bridges, R. A. Automated Behavioral Analysis of Malware A Case Study of WannaCry Ransomware. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2017. DOI:10.1109/ICMLA.2017.0-119.

Chindipha, S. D., Irwin, B., and Herbert, A. Effectiveness of Sampling a Small Sized Network Telescope in Internet Background Radiation Data Collection. In *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*. 2018. DOI:10.1145/1879141.1879149.

Chindipha, S. D. and Irwin, B. V. An Analysis on the Re-Emergence of SQL Slammer Worm using Network Telescope Data. SATNAC, 2017.

Cowie, B. and Irwin, B. Data Classification for Artificial Intelligence Construct Training to Aid in Network Incident Identification using network Telescope Data. In *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists*, pages 356–360. ACM, 2010. DOI:10.1145/1899503.1899544.

CVE-Details. Microsoft SQL Server Security Vulnerabilities. https://www.cvedetails.com/vulnerability-list/vendor_id-26/product_id-251/Microsoft-Sql-Server.html, 2018. [Online; accessed 03-September-2018].

Czyz, J., Lady, K., Miller, S. G., Bailey, M., Kallitsis, M., and Karir, M. Understanding IPv6 Internet Background Radiation. In *Proceedings of the 2013 conference on Internet Measurement*, pages 105–118. ACM, 2013. DOI:10.1145/2504730.2504732.

Dainotti, A., King, A., and Claffy, K. Analysis of Internet-wide Probing using Darknets. In *Proceedings of the 2012 ACM Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, pages 13–14. ACM, 2012. DOI:10.1145/2382416.2382423.

Dainotti, A., Squarcella, C., Aben, E., Claffy, K. C., Chiesa, M., Russo, M., and Pescapé, A. Analysis of Country-wide Internet Outages caused by Censorship. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 1–18. ACM, 2011. DOI:10.1109/TNET.2013.2291244.

de Bruijne, M., van Eeten, M., Gañán, C. H., and Pieters, W. Towards a New Cyber Threat Actor Typology. *Delft University of Technology*, 2017.

Durumeric, Z., Wustrow, E., and Halderman, J. A. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *USENIX Security Symposium*, volume 8, pages 47–53. 2013.

Fachkha, C., Bou-Harb, E., Boukhtouta, A., Dinh, S., Iqbal, F., and Debbabi, M. Investigating the Dark Cyberspace: Profiling, Threat-based Analysis and Correlation. In *Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on*, pages 1–8. IEEE, 2012. DOI:10.1109/CRISIS.2012.6378947.

Fachkha, C. and Debbabi, M. Darknet as a Source of Cyber Intelligence: Survey, Taxonomy, and Characterization. *IEEE Communications Surveys & Tutorials*, 18(2):1197–1227, 2016. DOI:10.1109/COMST.2015.2497690.

Fomenkov, M., Keys, K., Moore, D., and Claffy, K. Longitudinal Study of Internet Traffic in 1998-2003. In *Proceedings of the Winter International Synposium on Information and Communication Technologies*, WISICT '04, pages 1–6. Trinity College Dublin, 2004.

Ford, M., Stevens, J., and Ronan, J. Initial results from an IPv6 Darknet13. In *Internet Surveillance and Protection, 2006. ICISP'06. International Conference on*, pages 13–13. IEEE, 2006. DOI:10.1109/ICISP.2006.14.

Forouzan, B. A. and Fegan, S. C. TCP/IP Protocol Suite. McGraw-Hill Higher Education, 2002. ISBN 978-0073376042.

Fuller, V. and Li, T. Classless Inter-Domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan, RFC 4632. Technical report, 2006. DOI:10.17487/RFC4632.

Ghosh, A. WannaCry: List of Major Companies and Networks hit by Ransomware around the Globe. *International Business Times,[Online]. Available: https://www.ibtimes.co.uk/wannacry-list-major-companies-networks-hit-by-deadly-ransomware-around-globe-1621587*, 2017. [Online; accessed 03-September-2018].

Hick, P., Aben, E., Andersen, D., and Claffy, K. The CAIDA UCSD Network Telescope “Three Days of Conficker”. *Online, CAIDA Network Telescope Project-Backscatter*, 2009.

Irwin, B. A Framework for the Application of Network Telescope Sensors in a Global IP Network. Ph.D. thesis, Rhodes University, 2011.

Irwin, B. A Network Telescope Perspective of the Conficker Outbreak. In *Information Security for South Africa (ISSA), 2012*, pages 1–8. IEEE, 2012. DOI:10.1109/ISSA .2012.6320455.

Irwin, B. A Baseline Study of Potentially Malicious Activity across Five Network Telescopes. In *Cyber Conflict (CyCon), 2013 5th International Conference on*, pages 1–17. IEEE, 2013.

Irwin, B. and Nkhumeleni, T. Observed Correlations of Unsolicited Network Traffic over Five Distinct IPv4 Netblocks. In *Iccws 2015-The Proceedings of the 10th International Conference on Cyber Warfare and Security: ICCWS2015*, page 135. Academic Conferences Limited, 2015a.

Irwin, B. and Nkhumeleni, T. M. Observed Correlations of Unsolicited IP Traffic across Five Distinct Network Telescopes. *Journal of Information Warfare*, 14(3):1–14, 2015b.

Johnston, A. B. SIP: Understanding the Session Initiation Protocol. Artech House, 2015.

Jones, S. Timeline: How the WannaCry Cyber Attack Spread. *Financial Times,[Online]. Available: https://www.ft.com/content/82b01aca-38b7-11e7-821a-6027b8a20f23*, 2017. [Online; accessed 29-December-2018].

Jonker, M., King, A., Krupp, J., Rossow, C., Sperotto, A., and Dainotti, A. Millions of Targets Under Attack: A Macroscopic Characterization of the DoS Ecosystem. In *Proceedings of the 2017 Internet Measurement Conference*. 2017. DOI: 10.1145/3131365.3131383.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. DDoS in the IoT: Mirai and other Botnets. *Computer*, 50(7):80–84, 2017. DOI:10.1109/MC.2017.201.

Li, Q., Qin, T., Guan, X., and Zheng, Q. Exploring Flow Characteristics in IPv6: A Comparative Measurement Study with IPv4 for Traffic Monitoring. *KSII Transactions on Internet & Information Systems*, 8(4), 2014. DOI:10.3837/tiis.2014.04.009.

Liu, J. and Fukuda, K. Towards a Taxonomy of Darknet Traffic. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2014 International*, pages 37–43. IEEE, 2014. DOI:10.1109/IWCMC.2014.6906329.

Liu, J. and Fukuda, K. An Evaluation of Darknet Traffic Taxonomy. *Journal of Information Processing*, 26:148–157, 2018. DOI:10.2197/ipsjjip.26.148.

Majkowski, M. Stupidly Simple DDoS Protocol (SSDP) generates 100 Gbps DDoS. *Cloudflare,[Online]. Available: https://blog.cloudflare.com/ssdp-100gbps/*, 2017. [Online; accessed 11-October-2018].

Mirkovic, J. and Reiher, P. A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004. DOI:10.1145/997150.997156.

Mohurle, S. and Patil, M. A Brief Study of WannaCry Threat: Ransomware Attack 2017. *International Journal*, 8(5), 2017.

Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., and Weaver, N. Inside the Slammer Worm. *IEEE Security & Privacy*, 99(4):33–39, 2003. DOI: 10.1109/MSECP.2003.1219056.

Moore, D., Shannon, C., Brown, D. J., Voelker, G. M., and Savage, S. Inferring Internet Denial-of-Service Activity. *ACM Transactions on Computer Systems (TOCS)*, 24(2):115–139, 2006. DOI:10.1145/1132026.1132027.

Moore, D., Shannon, C., Voelker, G. M., and Savage, S. Network Telescopes: Technical Report. Department of Computer Science and Engineering, University of California, San Diego, 2004.

Moore, D., Shannon, C. *et al.* Code-Red: A Case Study on the Spread and Victims of an Internet Worm. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, pages 273–284. ACM, 2002. DOI:10.1145/637201.637244.

Nakashima, E. and Timberg, C. NSA officials worried about the day its potent hacking tool would get loose. Then it did. *Washington Post,[Online]. Available: https://www.washingtonpost.com/business/technology/nsa-officials-worried-about-the-day-its-potent-hacking-tool-would-get-loosethen-it-did/2017/05/16/50670b16-3978-11e7-a058-ddbb23c75d82_story. html*, 2017. [Online; accessed 17-October-2018].

Nkhumeleni, T. M. Correlation and Comparative Analysis of Traffic Across Five Network Telescopes. Master's thesis, Rhodes University, 2014.

Orendorff, A. Global Ecommerce Statistics. https://www.shopify.com/enterprise/global-ecommerce-statistics, 2017. [Online; accessed 01-March-2018].

Pa, Y. M. P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., and Rossow, C. IoTPOT: Analysing the Rise of IoT Compromises. *EMU*, 9:1, 2015.

Paganini, P. OVH hosting hit by 1Tbps DDoS attack, the largest one ever seen. https://securityaffairs.co/wordpress/51640/cyber-crime/tbps-ddos-attack.html, 2016. [Online; accessed 15-September-2018].

Pang, R., Yegneswaran, V., Barford, P., Paxson, V., and Peterson, L. Characteristics of Internet Background Radiation. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, pages 27–40. ACM, 2004. DOI:10.1145/1028788.1028794.

Pemberton, D., Komisarczuk, P., and Welch, I. Internet Background Radiation Arrival Density and Network Telescope Sampling Strategies. In *Telecommunication Networks and Applications Conference, 2007. ATNAC 2007. Australasian*, pages 246–252. IEEE, 2007. DOI:10.1109/ATNAC.2007.4665254.

Postel, J. User Datagram Protocol, RFC 768. Technical report, 1980. DOI:10.17487/RFC0768.

Postel, J. Active Users, RFC 866. Technical report, 1983a. DOI:10.17487/RFC0866.

Postel, J. Character Generator Protocol, RFC 864. Technical report, 1983b. DOI:10.17487/RFC0864.

Postel, J. *et al.* Internet Control Message Protocol, RFC 792. Technical report, 1981a. DOI:10.17487/RFC0792.

Postel, J. *et al.* Transmission Control Protocol, RFC 793. Technical report, 1981b. DOI:10.17487/RFC0793.

Raftopoulos, E., Glatz, E., Dimitropoulos, X., and Dainotti, A. How Dangerous Is Internet Scanning? A Measurement Study of the Aftermath of an Internet-Wide Scan. In *International Workshop on Traffic Monitoring and Analysis*, pages 158–172. Springer, 2015. DOI:10.1007/978-3-319-17172-2_11.

Richardson, R. and North, M. Ransomware: Evolution, Mitigation and Prevention. *International Management Review*, 13(1):10, 2017.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E. SIP: Session Initiation Protocol RFC 3261. Technical report, 2002.

Sachdeva, M., Singh, G., Kumar, K., and Singh, K. DDoS Incidents and their Impact: A Review. *International Arab Journal of Information Technology*, 7(1):14–20, 2010.

Shannon, C. and Moore, D. The Spread of the Witty Worm. *IEEE Security & Privacy*, 2(4):46–50, 2004. DOI:10.1109/MSP.2004.59.

Shulman, H. and Waidner, M. Fragmentation Considered Leaking: Port Inference for DNS Poisoning. In *International Conference on Applied Cryptography and Network Security*, pages 531–548. 2014. DOI:10.1007/978-3-319-07536-5_31.

Stallings, W. IPv6: The New Internet Protocol. *IEEE Communications Magazine*, 34(7):96–108, 1996. DOI:10.1109/35.526895.

Sutherland, L. Mirai Evolving: New Attack Reveals Use of Port 7547. https://securityintelligence.com/mirai-evolving-new-attack-reveals-use-of-port-7547/, 2016. [Online; accessed 05-September-2018].

Trostle, J., Besien, B. V., and Pujari, A. Protecting against DNS Cache Poisoning Attacks. In *2010 6th IEEE Workshop on Secure Network Protocols*, pages 25–30. Oct 2010. DOI:10.1109/NPSEC.2010.5634454.

Van der Elzen, I. and van Heugten, J. Techniques for Detecting Compromised IoT Devices. *University of Amsterdam*, 2017.

Vichaidis, N., Tsunoda, H., and Keeni, G. M. Analyzing Darknet TCP Traffic Stability at Different Timescales. In *2018 International Conference on Information Networking (ICOIN)*, pages 128–133. IEEE, 2018. DOI:10.1109/ICOIN.2018.8343098.

Wang, S., Xu, D., and Yan, S. Analysis and Application of Wireshark in TCP/IP Protocol Teaching. In *E-Health Networking, Digital Ecosystems and Technologies (EDT), 2010 International Conference on*, volume 2, pages 269–272. IEEE, 2010. DOI:10.1109/EDT.2010.5496372.

Weaver, N., Hamadeh, I., Kesidis, G., and Paxson, V. Preliminary Results using Scale-down to Explore Worm Dynamics. In *Proceedings of the 2004 ACM workshop on Rapid malcode*, pages 65–72. ACM, 2004. DOI:10.1145/1029618.1029628.

Wetteroth, D. OSI Reference Model for Telecommunications, volume 396. McGraw-Hill New York, 2002. ISBN 0071380418.

Wilkins, S. TCP/IP Ports and Protocols. *Pearson IT Certification,[Online]. Available: http://www.pearsonitcertification.com/articles/article.aspx?p=1868080*, 2012.

Woodhead, S. Monitoring Bad Traffic with Darknets. *Network Security*, 2012(1):10–14, 2012. DOI:10.1016/S1353-4858(12)70006-5.

Wustrow, E., Karir, M., Bailey, M., Jahanian, F., and Huston, G. Internet Background Radiation Revisited. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 62–74. ACM, 2010. DOI:10.1145/1879141.1879149.

Yates, D. A System for Characterising Internet Background Radiation. Technical report, Rhodes University, Department of Computer Science, 2014.

Yegneswaran, V., Barford, P., and Ullrich, J. Internet Intrusions: Global Characteristics and Prevalence. *ACM SIGMETRICS Performance Evaluation Review*, 31(1):138–147, 2003. DOI:10.1145/781027.781045.

Yeh, T. Netis Routers Leave Wide Open Backdoor. *Trend Micro Security Intelligence Blog,[Online]. Available: https://blog.trendmicro.com/trendlabs-security-intelligence/netis-routers-leave-wide-open-backdoor/*, 2014. [Online; accessed 03-September-2018].

Zou, C. C., Gong, W., Towsley, D., and Gao, L. The Monitoring and Early Detection of Internet worms. *IEEE/ACM Transactions on Networking (TON)*, 13(5):961–974, 2005. DOI:10.1109/TNET.2005.857113.

APPENDIX **A**

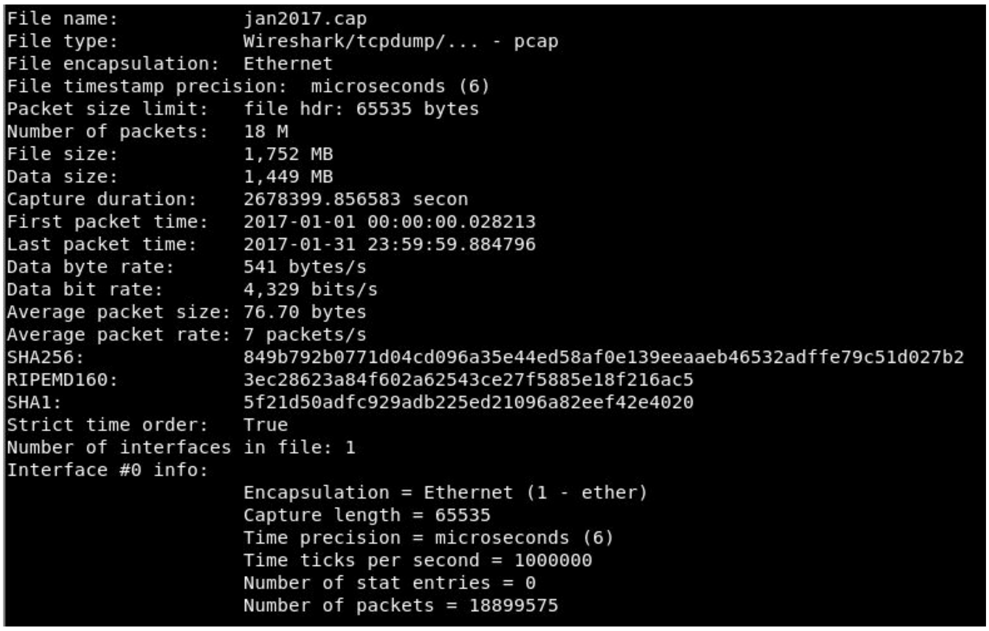
Overview of libpcap raw files

In this appendix, details about the libpcap raw data files are provided. The screenshots were taken of the outputs of running the following command in Linux:

```
capinfos -A <filename>
```

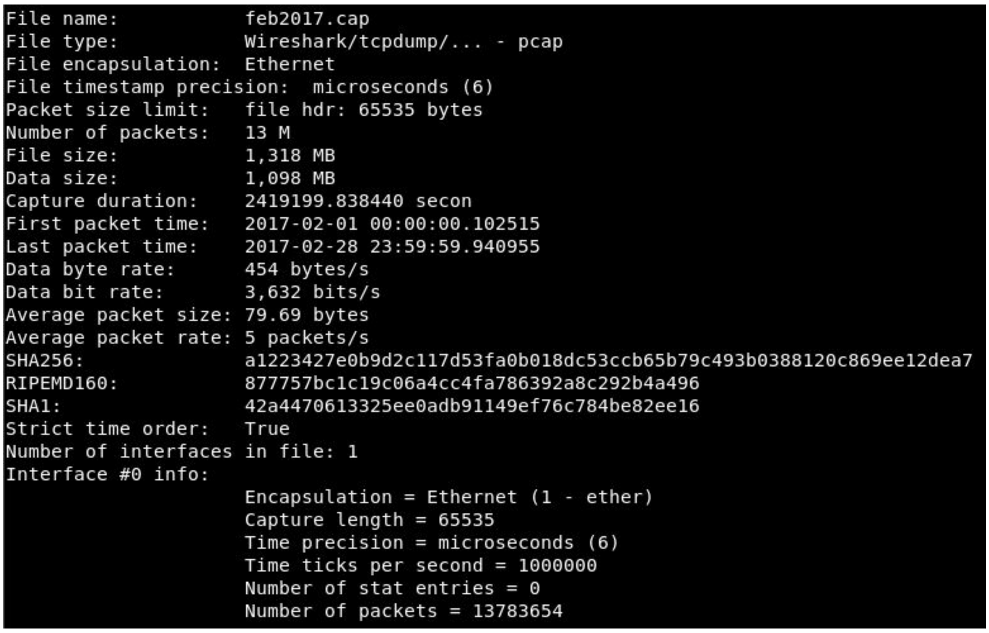
The raw files were labelled as follows:
January - jan2017.cap
February - feb2017.cap
March - mar2017.cap
April - apr2017.cap
May - may2017.cap
June - june2017.cap
July - july2017.cap
August - aug2017.cap
September - sep2017.cap
October - oct2017.cap
November - nov2017
December - dec2017.cap

A.1



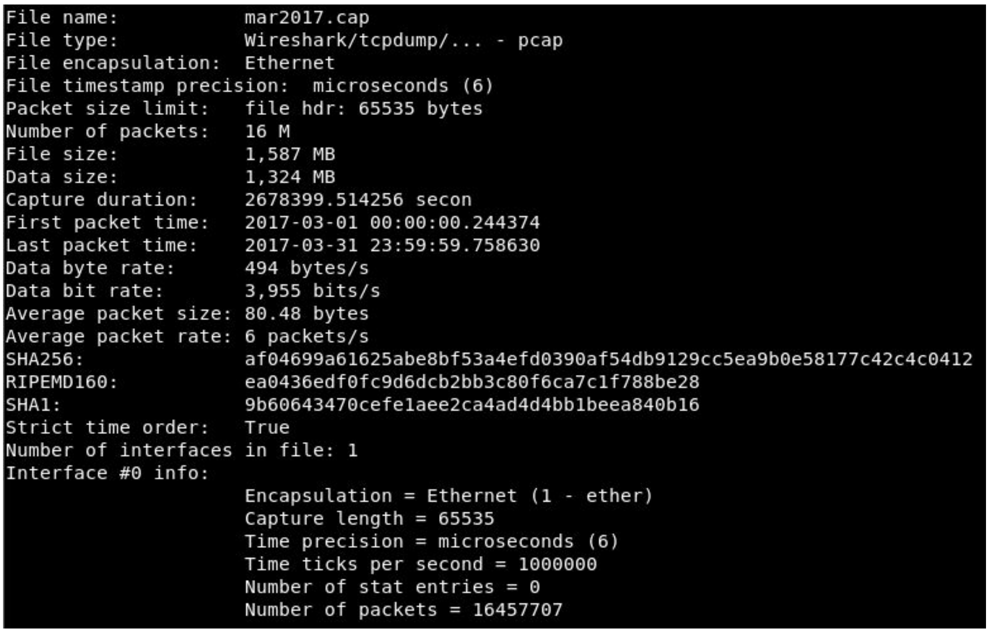
```
File name:           jan2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   18 M
File size:           1,752 MB
Data size:           1,449 MB
Capture duration:    2678399.856583 secon
First packet time:   2017-01-01 00:00:00.028213
Last packet time:    2017-01-31 23:59:59.884796
Data byte rate:      541 bytes/s
Data bit rate:       4,329 bits/s
Average packet size: 76.70 bytes
Average packet rate: 7 packets/s
SHA256:              849b792b0771d04cd096a35e44ed58af0e139eeaaeb46532adffe79c51d027b2
RIPEMD160:           3ec28623a84f602a62543ce27f5885e18f216ac5
SHA1:                5f21d50adfc929adb225ed21096a82eef42e4020
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 18899575
```

Figure A.1: Capinfos output for jan2017.cap



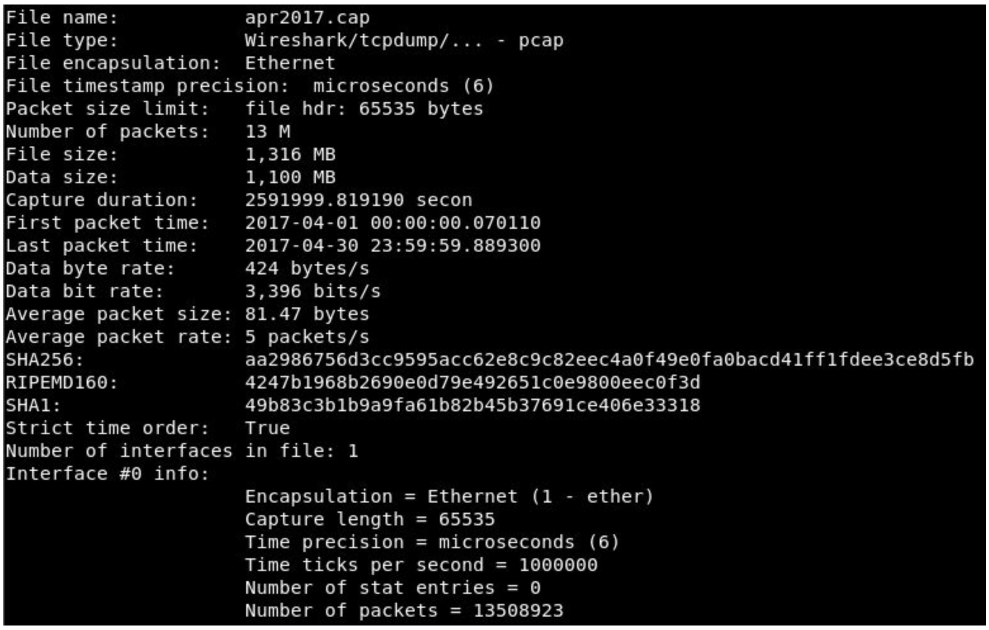
```
File name:           feb2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   13 M
File size:           1,318 MB
Data size:           1,098 MB
Capture duration:    2419199.838440 secon
First packet time:   2017-02-01 00:00:00.102515
Last packet time:    2017-02-28 23:59:59.940955
Data byte rate:      454 bytes/s
Data bit rate:       3,632 bits/s
Average packet size: 79.69 bytes
Average packet rate: 5 packets/s
SHA256:              a1223427e0b9d2c117d53fa0b018dc53ccb65b79c493b0388120c869ee12dea7
RIPEMD160:           877757bc1c19c06a4cc4fa786392a8c292b4a496
SHA1:                42a4470613325ee0adb91149ef76c784be82ee16
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 13783654
```

Figure A.2: Capinfos output for feb2017.cap



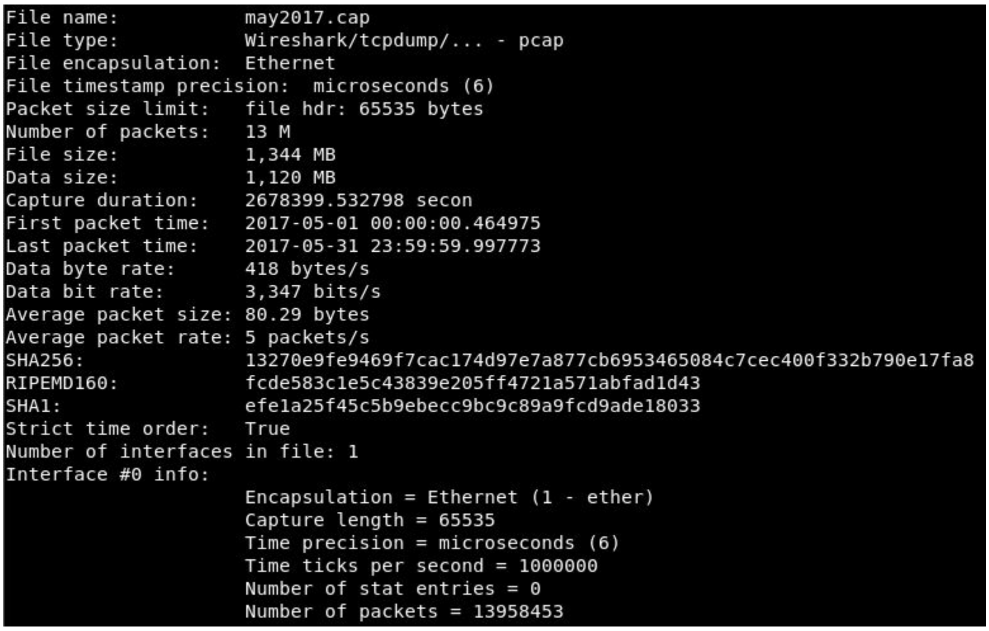
```
File name:           mar2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   16 M
File size:           1,587 MB
Data size:           1,324 MB
Capture duration:    2678399.514256 secon
First packet time:   2017-03-01 00:00:00.244374
Last packet time:    2017-03-31 23:59:59.758630
Data byte rate:      494 bytes/s
Data bit rate:       3,955 bits/s
Average packet size: 80.48 bytes
Average packet rate: 6 packets/s
SHA256:              af04699a61625abe8bf53a4efd0390af54db9129cc5ea9b0e58177c42c4c0412
RIPEMD160:           ea0436edf0fc9d6dcb2bb3c80f6ca7c1f788be28
SHA1:                9b60643470cefe1aee2ca4ad4d4bb1beea840b16
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 16457707
```

Figure A.3: Capinfos output for mar2017.cap



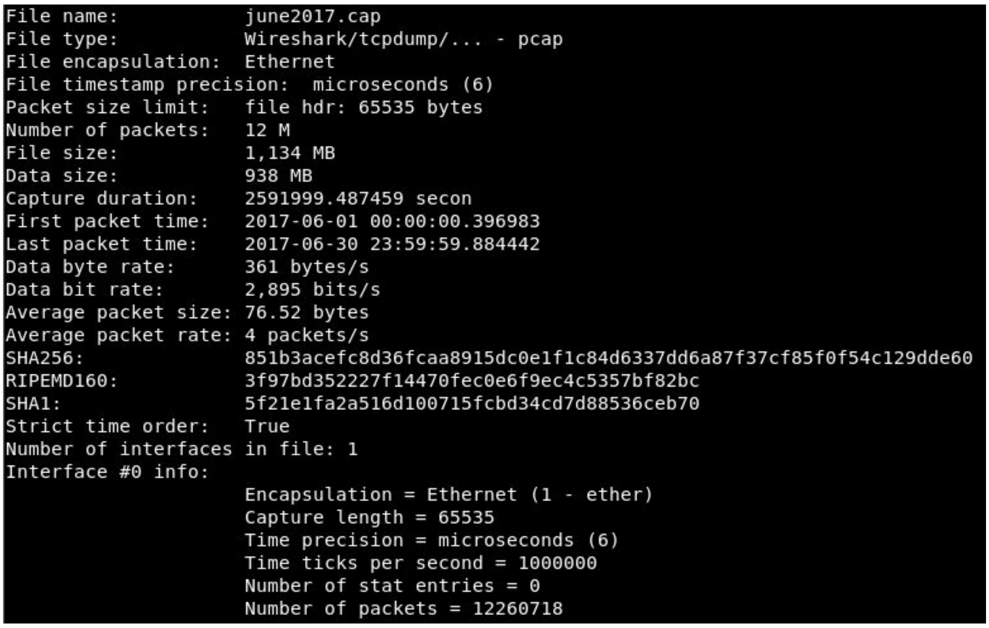
```
File name:           apr2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   13 M
File size:           1,316 MB
Data size:           1,100 MB
Capture duration:    2591999.819190 secon
First packet time:   2017-04-01 00:00:00.070110
Last packet time:    2017-04-30 23:59:59.889300
Data byte rate:      424 bytes/s
Data bit rate:       3,396 bits/s
Average packet size: 81.47 bytes
Average packet rate: 5 packets/s
SHA256:              aa2986756d3cc9595acc62e8c9c82eec4a0f49e0fa0bacd41ff1fdee3ce8d5fb
RIPEMD160:           4247b1968b2690e0d79e492651c0e9800eec0f3d
SHA1:                49b83c3b1b9a9fa61b82b45b37691ce406e33318
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 13508923
```

Figure A.4: Capinfos output for apr2017.cap



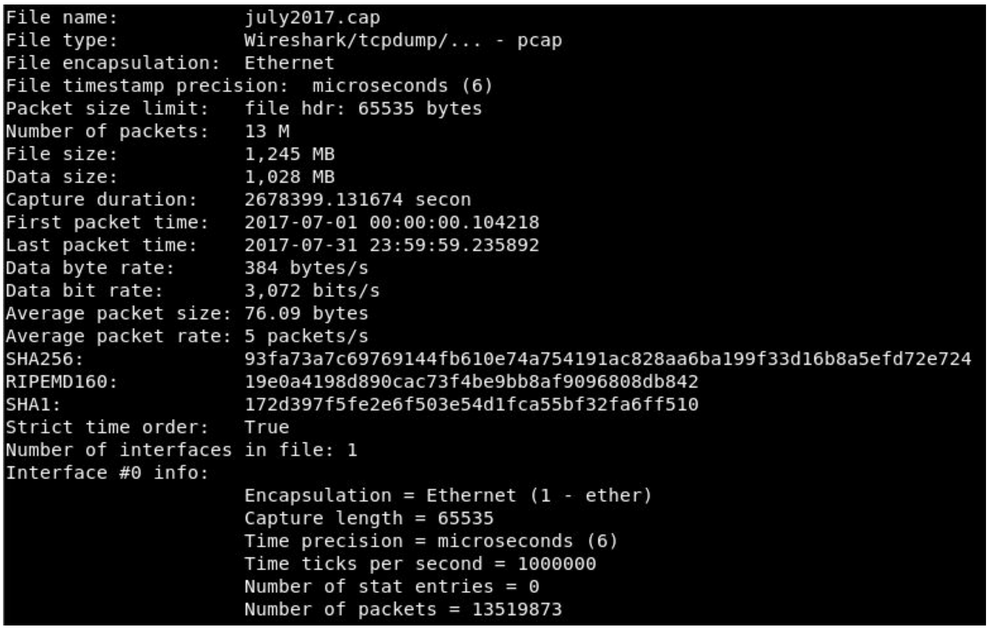
```
File name:           may2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   13 M
File size:           1,344 MB
Data size:           1,120 MB
Capture duration:    2678399.532798 secon
First packet time:   2017-05-01 00:00:00.464975
Last packet time:    2017-05-31 23:59:59.997773
Data byte rate:      418 bytes/s
Data bit rate:       3,347 bits/s
Average packet size: 80.29 bytes
Average packet rate: 5 packets/s
SHA256:              13270e9fe9469f7cac174d97e7a877cb6953465084c7cec400f332b790e17fa8
RIPEMD160:           fcde583c1e5c43839e205ff4721a571abfad1d43
SHA1:                efe1a25f45c5b9ebecc9bc9c89a9fcd9ade18033
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 13958453
```

Figure A.5: Capinfos output for may2017.cap



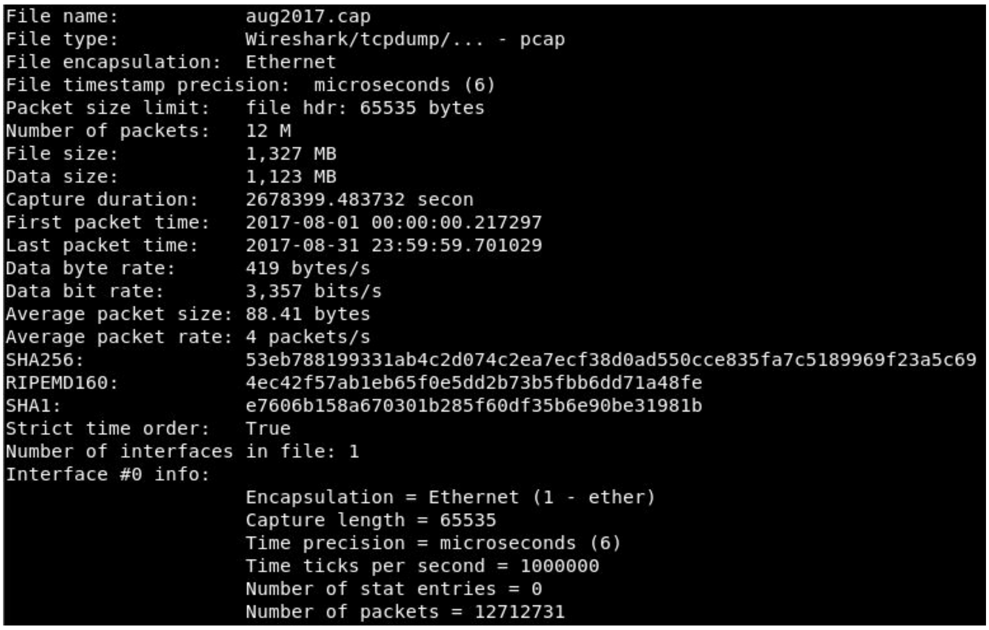
```
File name:           june2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   12 M
File size:           1,134 MB
Data size:           938 MB
Capture duration:    2591999.487459 secon
First packet time:   2017-06-01 00:00:00.396983
Last packet time:    2017-06-30 23:59:59.884442
Data byte rate:      361 bytes/s
Data bit rate:       2,895 bits/s
Average packet size: 76.52 bytes
Average packet rate: 4 packets/s
SHA256:              851b3acefc8d36fcaa8915dc0e1f1c84d6337dd6a87f37cf85f0f54c129dde60
RIPEMD160:           3f97bd352227f14470fec0e6f9ec4c5357bf82bc
SHA1:                5f21e1fa2a516d100715fcbd34cd7d88536ceb70
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 12260718
```

Figure A.6: Capinfos output for june2017.cap



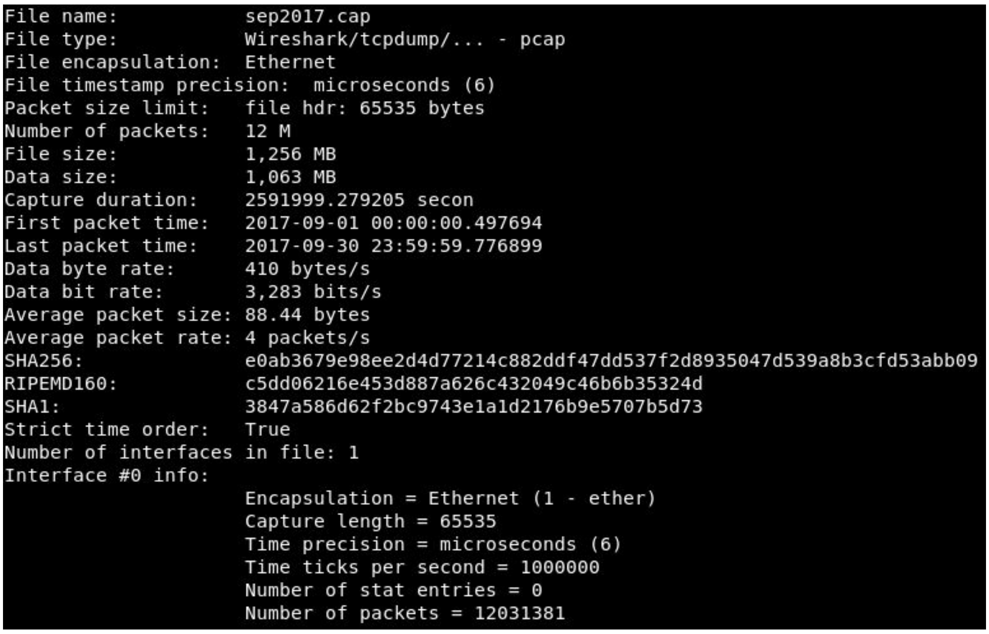
```
File name:           july2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   13 M
File size:           1,245 MB
Data size:           1,028 MB
Capture duration:    2678399.131674 secon
First packet time:   2017-07-01 00:00:00.104218
Last packet time:    2017-07-31 23:59:59.235892
Data byte rate:      384 bytes/s
Data bit rate:       3,072 bits/s
Average packet size: 76.09 bytes
Average packet rate: 5 packets/s
SHA256:              93fa73a7c69769144fb610e74a754191ac828aa6ba199f33d16b8a5efd72e724
RIPEMD160:           19e0a4198d890cac73f4be9bb8af9096808db842
SHA1:                172d397f5fe2e6f503e54d1fca55bf32fa6ff510
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 13519873
```

Figure A.7: Capinfos output for july2017.cap



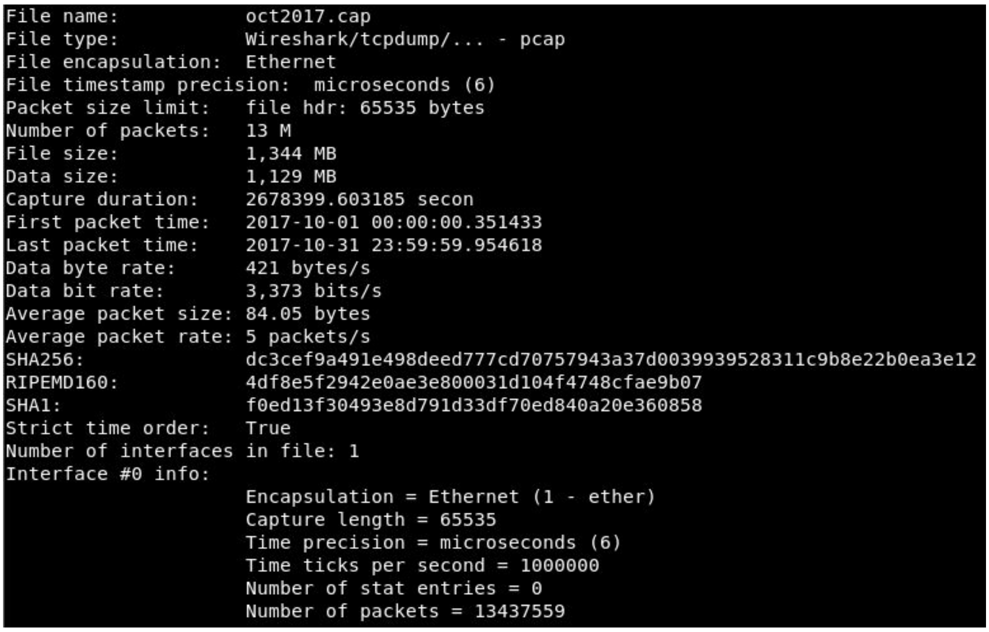
```
File name:           aug2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   12 M
File size:           1,327 MB
Data size:           1,123 MB
Capture duration:    2678399.483732 secon
First packet time:   2017-08-01 00:00:00.217297
Last packet time:    2017-08-31 23:59:59.701029
Data byte rate:      419 bytes/s
Data bit rate:       3,357 bits/s
Average packet size: 88.41 bytes
Average packet rate: 4 packets/s
SHA256:              53eb788199331ab4c2d074c2ea7ecf38d0ad550cce835fa7c5189969f23a5c69
RIPEMD160:           4ec42f57ab1eb65f0e5dd2b73b5fbb6dd71a48fe
SHA1:                e7606b158a670301b285f60df35b6e90be31981b
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 12712731
```

Figure A.8: Capinfos output for aug2017.cap



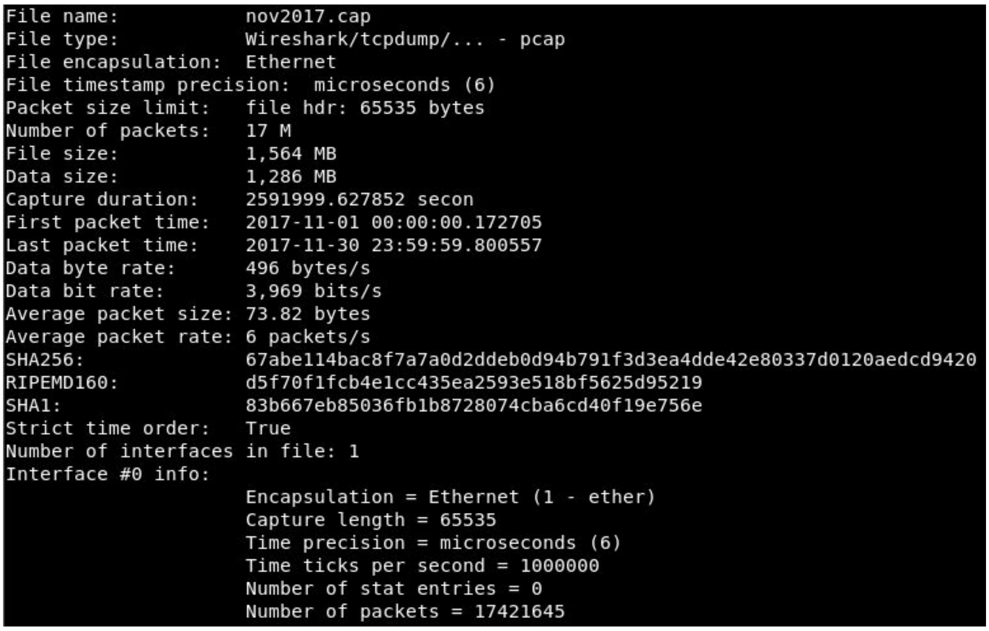
```
File name:           sep2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   12 M
File size:           1,256 MB
Data size:           1,063 MB
Capture duration:    2591999.279205 secon
First packet time:   2017-09-01 00:00:00.497694
Last packet time:    2017-09-30 23:59:59.776899
Data byte rate:      410 bytes/s
Data bit rate:       3,283 bits/s
Average packet size: 88.44 bytes
Average packet rate: 4 packets/s
SHA256:              e0ab3679e98ee2d4d77214c882ddf47dd537f2d8935047d539a8b3cfd53abb09
RIPEMD160:           c5dd06216e453d887a626c432049c46b6b35324d
SHA1:                3847a586d62f2bc9743e1a1d2176b9e5707b5d73
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 12031381
```

Figure A.9: Capinfos output for sep2017.cap



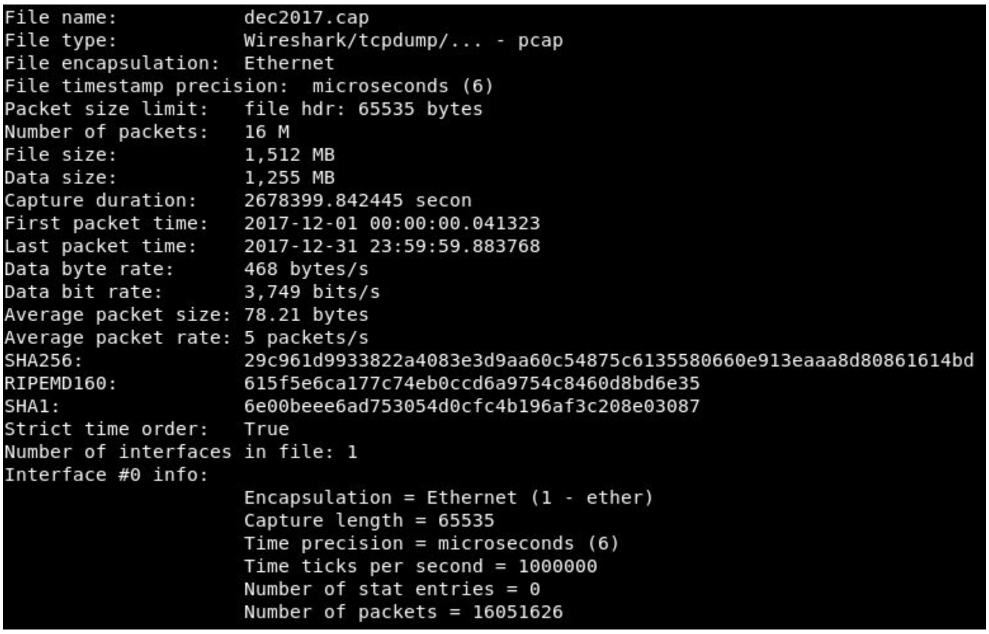
```
File name:           oct2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   13 M
File size:           1,344 MB
Data size:           1,129 MB
Capture duration:    2678399.603185 secon
First packet time:   2017-10-01 00:00:00.351433
Last packet time:    2017-10-31 23:59:59.954618
Data byte rate:      421 bytes/s
Data bit rate:       3,373 bits/s
Average packet size: 84.05 bytes
Average packet rate: 5 packets/s
SHA256:              dc3cef9a491e498deed777cd70757943a37d0039939528311c9b8e22b0ea3e12
RIPEMD160:           4df8e5f2942e0ae3e800031d104f4748cfae9b07
SHA1:                f0ed13f30493e8d791d33df70ed840a20e360858
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 13437559
```

Figure A.10: Capinfos output for oct2017.cap



```
File name:           nov2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   17 M
File size:           1,564 MB
Data size:           1,286 MB
Capture duration:    2591999.627852 secon
First packet time:   2017-11-01 00:00:00.172705
Last packet time:    2017-11-30 23:59:59.800557
Data byte rate:      496 bytes/s
Data bit rate:       3,969 bits/s
Average packet size: 73.82 bytes
Average packet rate: 6 packets/s
SHA256:              67abe114bac8f7a7a0d2ddeb0d94b791f3d3ea4dde42e80337d0120aedcd9420
RIPEMD160:           d5f70f1fcb4e1cc435ea2593e518bf5625d95219
SHA1:                83b667eb85036fb1b8728074cba6cd40f19e756e
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 17421645
```

Figure A.11: Capinfos output for nov2017.cap



```
File name:           dec2017.cap
File type:           Wireshark/tcpdump/... - pcap
File encapsulation:  Ethernet
File timestamp precision:  microseconds (6)
Packet size limit:   file hdr: 65535 bytes
Number of packets:   16 M
File size:           1,512 MB
Data size:           1,255 MB
Capture duration:    2678399.842445 secon
First packet time:   2017-12-01 00:00:00.041323
Last packet time:    2017-12-31 23:59:59.883768
Data byte rate:      468 bytes/s
Data bit rate:       3,749 bits/s
Average packet size: 78.21 bytes
Average packet rate: 5 packets/s
SHA256:              29c961d9933822a4083e3d9aa60c54875c6135580660e913eaaa8d80861614bd
RIPEMD160:           615f5e6ca177c74eb0ccd6a9754c8460d8bd6e35
SHA1:                6e00beee6ad753054d0cfc4b196af3c208e03087
Strict time order:   True
Number of interfaces in file: 1
Interface #0 info:
                     Encapsulation = Ethernet (1 - ether)
                     Capture length = 65535
                     Time precision = microseconds (6)
                     Time ticks per second = 1000000
                     Number of stat entries = 0
                     Number of packets = 16051626
```

Figure A.12: Capinfos output for dec2017.cap

APPENDIX B

Creation of CSV files from libpcap

In this appendix, some details about the csv file creation is provided. The relevant fields and information were extracted from the libpcap and exported to CSV format. The commands used to extract the data is as follows:

For all packets:

```
tshark -r libcap_file -T fields -e frame.time_epoch -e _ws.col.Protocol -e tcp.flags -e ip.src -e ip.dst -e ip.geoip.src_country -e tcp.srcport -e tcp.dstport -E header=y -E separator=/t/ -E quote=d -E occurrence=f > csv_output
```

For only ICMP packets:

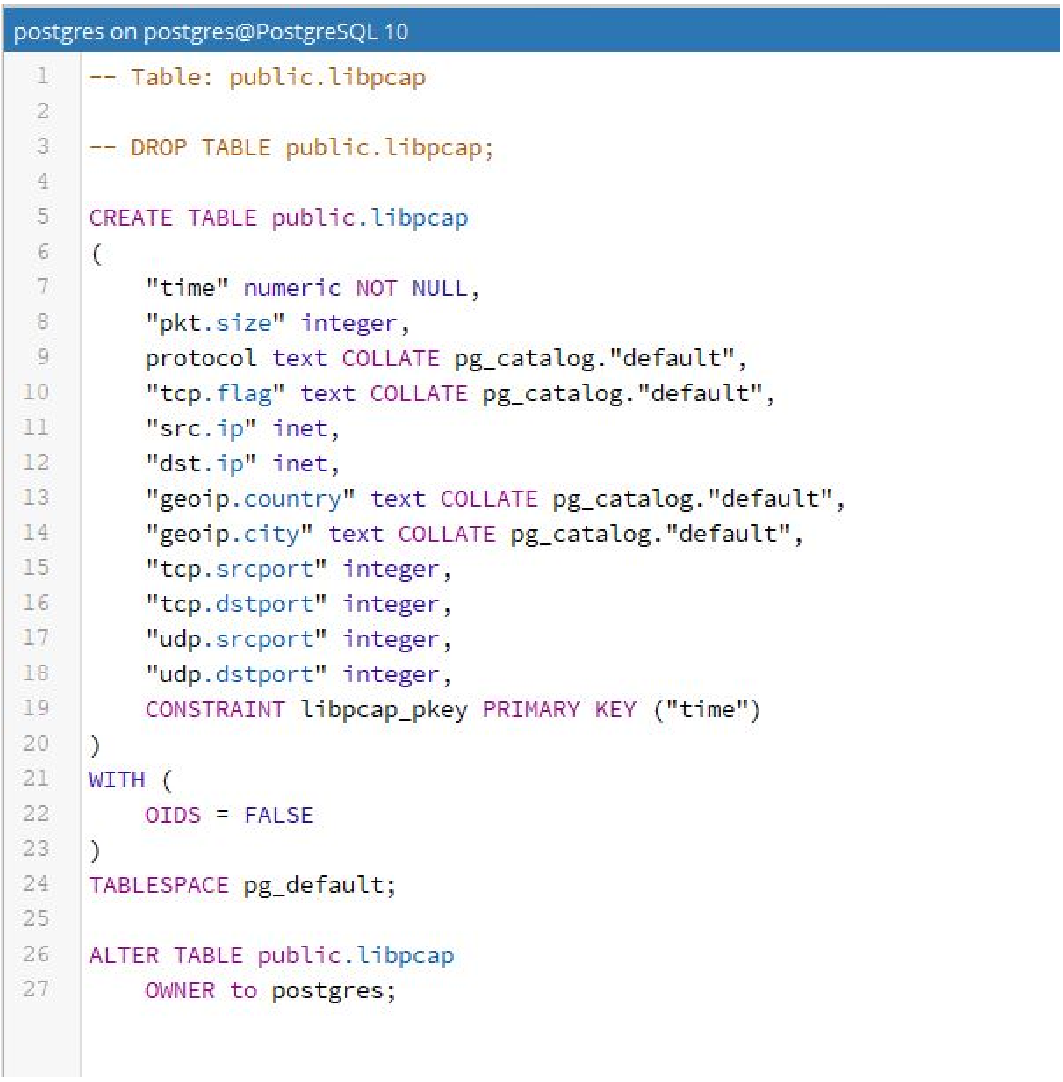
```
tshark -r libcap_file -Y icmp -T fields -e frame.time_epoch -e _ws.col.Protocol -e icmp.type -e icmp.code -e ip.src -e ip.dst -e ip.geoip.src_country -E header=y -E separator=/t/ -E quote=d -E occurrence=f > csv_output
```

APPENDIX **C**

Overview of database tables

This appendix gives a brief overview of the database tables that were created for this research project. It provides screenshots of the table creation scripts as well as screenshots of the table sctructure and the field types.

C.1



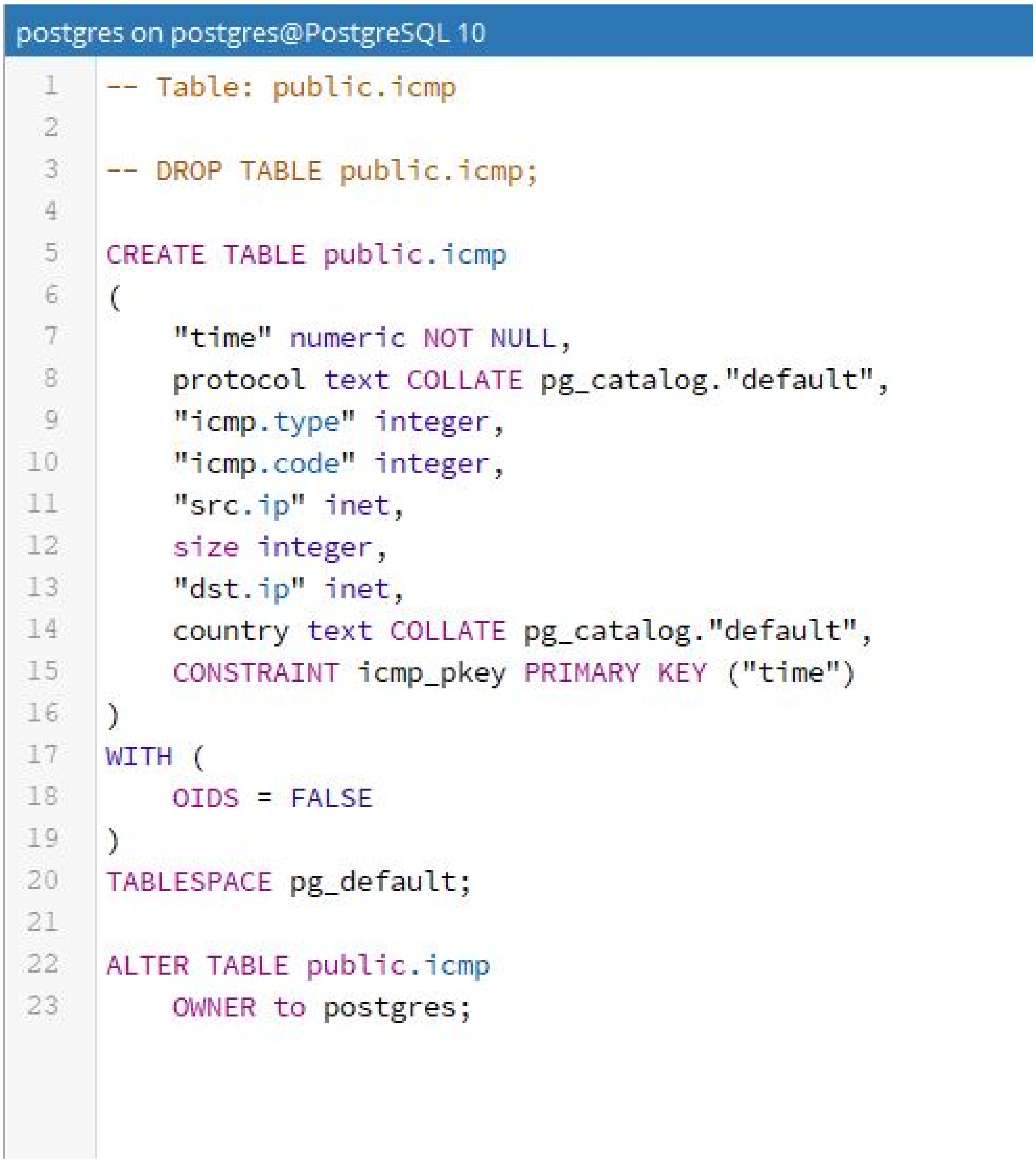
```
-- Table: public.libpcap

-- DROP TABLE public.libpcap;

CREATE TABLE public.libpcap
(
    "time" numeric NOT NULL,
    "pkt.size" integer,
    protocol text COLLATE pg_catalog."default",
    "tcp.flag" text COLLATE pg_catalog."default",
    "src.ip" inet,
    "dst.ip" inet,
    "geoip.country" text COLLATE pg_catalog."default",
    "geoip.city" text COLLATE pg_catalog."default",
    "tcp.srcport" integer,
    "tcp.dstport" integer,
    "udp.srcport" integer,
    "udp.dstport" integer,
    CONSTRAINT libpcap_pkey PRIMARY KEY ("time")
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.libpcap
    OWNER to postgres;
```

Figure C.1: SQL create script for libpcap table



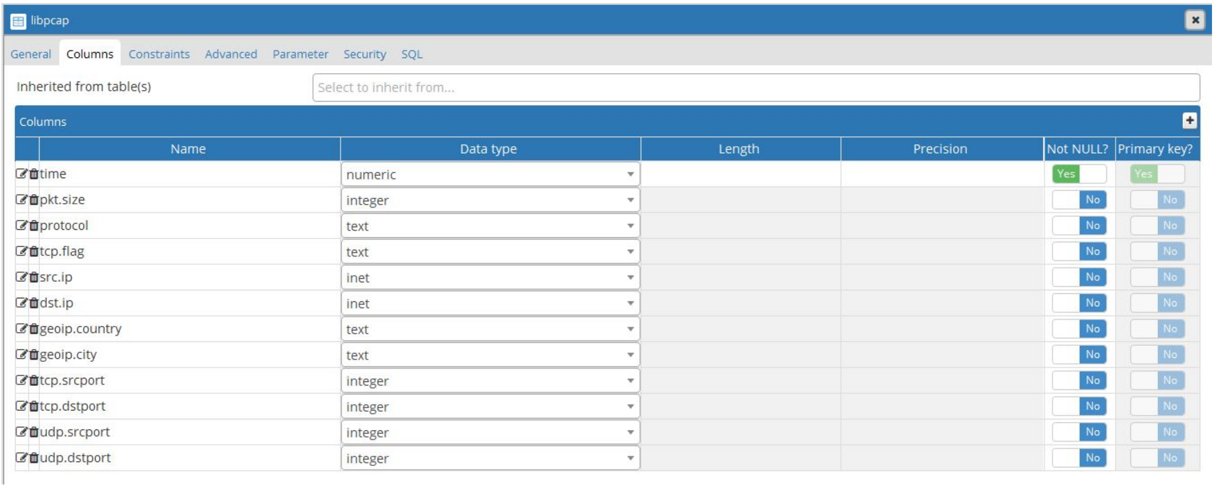
```
-- Table: public.icmp

-- DROP TABLE public.icmp;

CREATE TABLE public.icmp
(
    "time" numeric NOT NULL,
    protocol text COLLATE pg_catalog."default",
    "icmp.type" integer,
    "icmp.code" integer,
    "src.ip" inet,
    size integer,
    "dst.ip" inet,
    country text COLLATE pg_catalog."default",
    CONSTRAINT icmp_pkey PRIMARY KEY ("time")
)
WITH (
    OIDS = FALSE
)
TABLESPACE pg_default;

ALTER TABLE public.icmp
    OWNER to postgres;
```

Figure C.2: SQL create script for icmp table

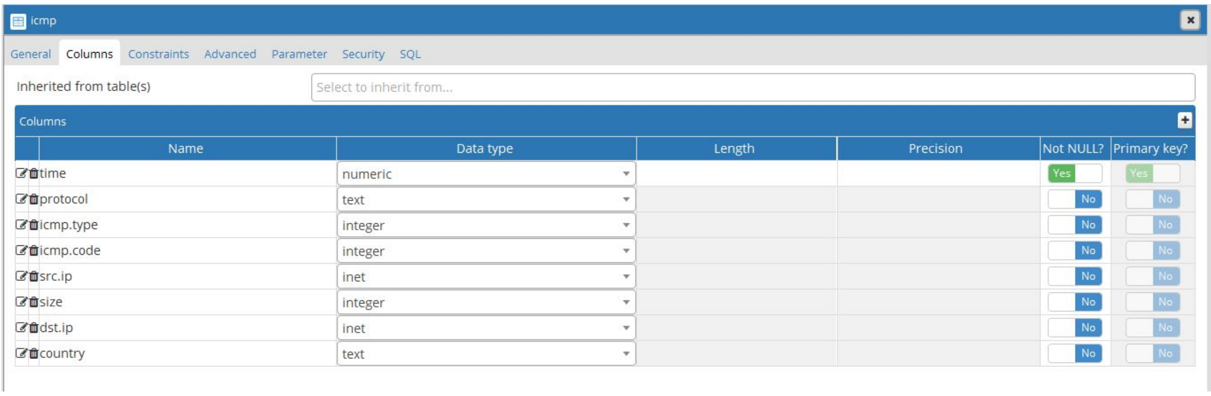


Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length	Precision	Not NULL?	Primary key?
time	numeric			Yes	Yes
pkt.size	integer			No	No
protocol	text			No	No
tcp.flag	text			No	No
src.ip	inet			No	No
dst.ip	inet			No	No
geoip.country	text			No	No
geoip.city	text			No	No
tcp.srcport	integer			No	No
tcp.dstport	integer			No	No
udp.srcport	integer			No	No
udp.dstport	integer			No	No

Figure C.3: Table and column structure for libpcap table



Inherited from table(s): Select to inherit from...

Columns

Name	Data type	Length	Precision	Not NULL?	Primary key?
time	numeric			Yes	Yes
protocol	text			No	No
icmp.type	integer			No	No
icmp.code	integer			No	No
src.ip	inet			No	No
size	integer			No	No
dst.ip	inet			No	No
country	text			No	No

Figure C.4: Table and column structure for icmp table