CATEGORISING NETWORK TELESCOPE DATA USING BIG DATA ENRICHMENT TECHNIQUES

Submitted in partial fulfillment
of the requirements of the degree of

Master of Science

of Rhodes University

Michael Reginald Davis

Grahamstown, South Africa
January 2019

Abstract

Network Telescopes, Internet backbone sampling, IDS and other forms of network-sourced Threat Intelligence provide researchers with insight into the methods and intent of remote entities by capturing network traffic and analysing the resulting data. This analysis and determination of intent is made difficult by the large amounts of potentially malicious traffic, coupled with limited amount of knowledge that can be attributed to the source of the incoming data, as the source is known only by its IP address.

Due to the lack of commonly available tooling, many researchers start this analysis from the beginning and so repeat and re-iterate previous research as the bulk of their work. As a result new insight into methods and approaches of analysis is gained at a high cost. Our research approaches this problem by using additional knowledge about the source IP address such as open ports, reverse and forward DNS, BGP routing tables and more, to enhance the researcher's ability to understand the traffic source.

The research is a BigData experiment, where large (hundreds of GB) datasets are merged with a two month section of Network Telescope data using a set of Python scripts. The result are written to a Google BigQuery database table. Analysis of the network data is greatly simplified, with questions about the nature of the source, such as its device class (home routing device or server), potential vulnerabilities (open telnet ports or databases) and location becoming relatively easy to answer. Using this approach, researchers can focus on the questions that need answering and efficiently address them.

This research could be taken further by using additional data sources such as Geo-location, WHOIS lookups, Threat Intelligence feeds and many others. Other potential areas of research include real-time categorisation of incoming packets, in order to better inform alerting and reporting systems' configuration.

In conclusion, categorising Network Telescope data in this way provides insight into the intent of the (apparent) originator and as such is a valuable tool for those seeking to understand the purpose and intent of arriving packets. In particular, the ability to remove packets categorised as non-malicious (e.g. those in the Research category) from the data eliminates a known source of 'noise' from the data. This allows the researcher to focus their efforts in a more productive manner.

ACM Computing Classification System Classification

Thesis classification under the ACM Computing Classification System[1] (2012 version, valid through 2018):

• **Security and privacy** → **Network security**; • **Networks** → **Packet classification**; **Network security**; *Transport protocols*; *Control path algorithms*; *Network structure*; • **Computing methodologies** → **Distributed algorithms**; **MapReduce algorithms**;

[1] http://www.acm.org/about/class/2012/

Acknowledgements

I would first like to thank my thesis advisor Prof. Barry Irwin of the Computer Science Department at Rhodes University: firstly for the endless patience during the many reviews, and secondly for the carefully considered use of red ink. His guidance during this time was invaluable, and through this helped me to extract myself from many of the rabbit holes I enthusiastically climbed into.

I would also like to thank Jon Hart and the research team at OpenData/Rapid7 for granting access to their research and their data. Their quick response to questions about the data and its structure were invaluable during the initial exploration. Their work in making these data-sets available at no cost is an invaluable resource for researchers in this field.

Additionally I would like to thank my classmates for their help with LaTeX, reviews and encouragement. Special thanks goes to the LaTeX wizard Ivan Burke who made writing this document far easier.

Leon Jacobs from Sensepost gets kudos for producing the Open-Source tool called `gowitness`, which allowed for the automation of screen-shot creation for 820 web-pages. This saved many hours of manual labour.

To my wife Penny I owe a debt of gratitude that can never be repaid, for all the events missed in favour of writing this document, and especially for the two missed days in Venice. I would like to thank her for all of the encouragement, patience, love, advice and reviews she provided during this process.

My children also deserve thanks for supporting me and keeping me sane by insisting on playing Minecraft and other games as often as possible.

The cats 'Trouble' and 'Nala' allowed me to use my keyboard on the occasions when they were not stretched out across it. For this I owe them many thanks.

And finally, without my dear friend Coffee I would not have been able to produce this document. Thank you!

Contents

List of Figures

List of Tables

List of Listings

Glossary

ACF Auto Correlation Function

ACK Acknowledge

Alexa Alexa

ANSI American National Standards Institute

API Application Programming Interface

AS Autonomous System

ASN Autonomous System Number

bash Bash

BGP Border Gateway routing Protocol

BigQuery Google BigQuery

Botnet Botnet

BYOD Bring Your Own Device

C2C Malware Command and Control

CAIDA Center for Applied Internet Data Analysis

CDN Content Distribution Network

CPU Central Processing Unit

CSV Comma Separated Variable

Darknet an announced, routable block of IP addresses, with no active hosts

DDoS Distributed Denial of Service

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

DNS PTR DNS Pointer

DOS Denial of Service

DWT Discrete Wavelet Transform

ECT Act South African Electronic Communications and Transaction Act, 2002

EFF Electronic Frontier Foundation

epoch Unix Epoch time

ETL Extract-Transform-Load

FDNS Forward DNS

GB Gigabyte

GBs Gigabytes

GeoIP Geographical IP

GeoLocation Geographical location lookup

Greynet an announced, routable block of IP addresses, with some active hosts

Honeypot an instrumented and monitored system whose purpose is to allow itself to be compromised for research or alerting purposes

HTTP Hyper-Text Transfer Protocol

HTTPS Secure HTTP

IANA Internet Assigned Numbers Authority

ICMP Internet Control Message Protocol

ICS Industrial Control System

IDS Intrusion Detection System

IGMP Internet Group Management Protocol

IKE Internet Key Exchange

IOC Indicator Of Compromise

IoT Internet of Things

IP Internet Protocol

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

IRC Internet Relay Chat

ISP Internet Service Provider

ISP Internet Service Providers

jq JSON Query Tool

JSON JavaScript Object Notation

LDAP Lightweight Directory Access Protocol

LSB Least Significant Bit

MBs Megabytes

MITM Man-in-The-Middle

MRT Multi-Threaded Routing Toolkit

MSB Most Significant Bit

NetFlow Network Flow

NSP Network Service Provider

OpenData Rapid7 OpenData Project

OpenSSH OpenSSH server

OS Operating System

OSINT Open-Source Intelligence

PCAP Packet Capture

PKI Public Key Infrastructure

PostgreSQL PostgreSQL Database

ppd packets per day

PTR Pointer

Python Python Programming Language

RADB the Routing Assets Database

RDBMS Relational Database Management System

RDNS Reverse DNS

RIR Regional Internet Registry

RIR Regional Internet Registry

RSA RSA encryption algorithm

RST Reset Flag

RU Rhodes University

SANReN South African National Research Network

SCANN UNKNOWN

SCTP Stream Control Transmission Protocol

SIP Session Initiation Protocol

SMTP Simple Message Transport Protocol

SQL Structured Query Language

SSH Secure Shell

SSL Secure Socket Layer

SYN Connection Initiation

TB Terabyte

TBs Terabytes

TCP Transmission Control Protocol

Telescope Network Telescope

ThreatIntel Threat Intelligence

Timestamp Timestamp

TLD Top-Level Domain

TLS Transport Level Security

UDP User Datagram Protocol

UN United Nations

Unicast Unicast network address

USD US Dollars

VM Virtual Machine

VMs Virtual Machines

VoIP Voice over IP

Chapter 1

Introduction

1.1 Context of Research

The Internet is a widely distributed, loosely coupled set of systems and services designed to cheaply and quickly transport data around the world. The majority of the billions of users of these services use the Internet for a variety of purposes including entertainment, banking, commerce, work, and increasingly as part of home automation through the use of Internet of Things (IoT) (Shirer and Torchia, 2017). The Internet has become a critical utility in the lives of those who have access to it, and as such the United Nations (UN) has declared that "that rights that people have offline must also be protected online" (UN Human Rights Council, 2016).

Unfortunately the Internet also provides many opportunities for activities that may be viewed as unethical, malicious or simply illegal. In order to defend against these activities, one must first understand what they are, and how they operate within the Internet as a whole. This research focuses on the Internet on the network level, in terms of potentially malicious network traffic that may be directed at any Internet-facing system.

The modern Internet has a significant portion of its resources dedicated to unsolicited network-level interactions including Distributed Denial of Service (DDoS) attacks (Cloudflare, 2018), port scanning and hacking attempts. As part of a pro-active Threat Intelligence (ThreatIntel) strategy, researchers will attempt to capture and analyse related network traffic, in order to gain insight into the reasons and methods behind this kind of activity. Tools that assist with capturing this traffic include Honeypots, Honeynets and Network Telescope (Telescope).

1.2 Research Question

After network traffic has been captured, the packets making up that traffic need to be analysed in a meaningful way, so as to inform decisions around defense against potential threats. Given the current state of the Internet, over a typical month, a Telescope may collect millions of packets originating from hundreds of thousands of Internet Protocol (IP) addresses - a potentially overwhelming amount of data. In this case it is possible to efficiently and effectively categorise the data.

In order to address this problem, a set of techniques were created that allow a security researcher to enrich the captured data with publicly available (or collectable) data sets, and to efficiently and cost-effectively query this data for analysis purposes. Patterns in this data are visualised and explored in detail in Chapters 4, 5, 6 and 7.

The Rhodes Computer Science department maintains a reasonably sized Telescope and was willing to provide full access to the resulting data. For this reason, the research focused exclusively on this data. The creation and management of a Telescope was not addressed as part of this research.

1.3 Objectives

The primary objective of this research is to explore techniques that allow a researcher to understand the origins and nature of the incoming network data. The primary technique is to categorise incoming data according to source, and then use that categorisation as a basis upon which to perform further analysis.

In order to achieve the primary objective, large amounts of enrichment data had to be acquired, processed, imported and matched to each captured packet. The use of Big Data platforms was evaluated and categorisation techniques explored.

1.4 Approach

In late 2014 and early 2015, scans.io[1] was established by the University of Michigan in order to publicly share raw network-based ThreatIntel data. It is possible that this

[1] https://scans.io/

was inspired by the release of an Internet "census" performed in early 2013 by unknown researchers, using the Carna botnet (Krenc *et al.*, 2014).

Scans.io includes full scans of the Internet Protocol version 4 (IPv4) address space from several different perspectives, such as Hyper-Text Transfer Protocol (HTTP), and Secure HTTP (HTTPS) (with full handshakes), as well as forward and reverse Domain Name System (DNS) lookups (Rapid7, 2017a,b,c). Other data such as Cowrie honeypot logs and banners for various protocols are also available (Heisenberg, 2017). These scans are performed on a regular basis (monthly/weekly/daily) and made available on the scans.io website.

Censys[2] was created by Durumeric *et al.* (2015) in 2015 as a searchable platform for researchers to query this data with a complex and powerful query language. Regular scans of the IPv4 space are performed and the database is updated with these results. Access to historical data is possible on request.

While Censys is a powerful tool, as of January 2019, it is not possible to perform large-scale matching against historical Telescope data. As such it was decided to build tools and techniques for this specific purpose. This required access to fairly regular scans of the Internet's IPv4 address space. An up-to-date, well formatted selection of this type of data can be acquired from OpenData (Rapid7, 2019). The majority of the available data is in JavaScript Object Notation (JSON) format.

A sensible place to begin would be to create a query-able database containing the Telescope data. This data can then be enhanced with other data such as that found in the OpenData data sets, such as DNS query dumps and Cowrie honeypot logs. It may also be possible to add indicators based on anonymised packet captures from Internet backbones.

In order to build a system that can store and query data for every IP address in the IPv4 address space, one would require a datastore with a theoretical maximum size of 2 to the power of 32 rows (roughly four billion). Luckily this is not the case for the Telescope data - it contained 83 million rows for March 2017, and 71 million for April 2017. This reduces the complexity of the problem substantially, as only the required data must be matched against the OpenData data-sets.

While a traditional ACID-compliant Relational Database Management System (RDBMS) could be set up to deal with data of this magnitude, better tools exist that are more able to ingest and manage data on this scale. Apache Hadoop and Spark are good examples.

[2]https://censys.io/

Commercial offerings such as Amazon EMR (Amazon, 2017) or Google's BigTable are also options to be considered.

Using data provided by Opendata is particularly advantageous in the South African context, given the expense and difficulty of acquiring the necessary Internet bandwidth and storage resources. These data-sets are an acceptable work-around in the local context of the legal uncertainty of public host scanning, which could be deemed to be unauthorised access, according to the South African Electronic Communications and Transaction Act, 2002 (ECT Act) (Swart *et al.*, 2014).

1.5 Document Structure

In Chapter 2, relevant research is discussed and examined. Chapter 3 explains the concepts, data and technologies required in order to categorise the Telescope data. Initial analysis and an overview of the Telescope data is performed in Chapter 4. This is followed by an in-depth DNS based analysis of *Active Traffic* in Chapter 5 and *Passive Traffic* in 6, followed by a re-examination of the data using Border Gateway routing Protocol (BGP) data in Chapter 7. Finally, the document is concluded in Chapter 8.

Chapter 2

Literature Review

2.1 Overview

In this Chapter, research relating to the ethics, tools, processing and analysis of Telescope data is reviewed. Section 2.2 discusses the ethics of generating and using data acquired through active scanning of hosts on the Internet. In Section 2.3, a set of modern, effective large-scale Internet scanners are compared. This is followed by Section 2.4 which explores passive data collection tools such as Telescopes. Finally, Section 2.5 examines research into various methods of processing captured network data.

2.2 Scanning Ethics and Behaviour

An in-depth examination of the methods, ethics and reliability of the data produced by the Carna botnet is performed in Krenc *et al.* (2014). This botnet was constructed by unknown researcher(s) for the purposes of conducting a publicly accessible Internet Census. The participating devices were home routers and similar devices. In late 2016 these devices were compromised using default credentials, in a manner similar to the Mirai botnet. In this research, the authenticity and accuracy of the resulting scan data is explored and found to be authentic. The required meta-data for a full validation was found to be absent, however some of the the data was reproduced and confirmed. Additionally other research institutions such as Center for Applied Internet Data Analysis (CAIDA) confirmed that the scan data matched their internal data. The authors dig deep into the

quality of the data, and find that the data are not of the quality required for academic analysis. It may however be useful if used in combination with other data-sets. Some thought is given to the ethics of other researchers using the results of the scans, and the conclusion is that this would be unethical according to academic standards.

Further analysis of the ethics of using the Internet Censys data in academic research is performed in Dittrich *et al.* (2014). Parallels are drawn to the medical industry, in which informed consent of the target of an ethical study is required before data can be collected. Using this argument, and the assumption that the owners of the devices used in the Carna botnet did not give any form of consent, it can again be said that the use of this data is unethical. The actions of the Internet Census researchers are compared against the principles of the Menlo Report (Dittrich *et al.*, 2012), and found to be unethical.

2.3 Scanning Tools

There are many tools which can be used to probe Internet-connected devices. Some are standard debugging tools such as `ping` and `traceroute` (based on Postel (1981)), as well as `telnet`, while others are purpose-built for examining a host, for example `Nmap`[1] (Fyodor, 1997), `ZMap` (Durumeric *et al.*, 2013) or `masscan` (Graham, 2014; Graham *et al.*, 2015). In this section, research relating to the creation or comparative analysis of such tools is discussed.

2.3.1 Nmap

`Nmap` was one of the first freely available purpose-built network scanning tools. It was introduced by Fyodor in an article in Phrack Magazine in September 1997 (Fyodor, 1997). The careful, parallel yet synchronous approach used by `Nmap` highlights its focus on accuracy. Unfortunately this approach results in long scan times when attempting to scan large segments of a network. For example, if one assumes an ideal scenario in which a single address scan has a duration of 1ms and `Nmap` runs with 1024 non-overlapping threads, a scan of `0.0.0.0/0` (the entire IPv4 address space) will complete in a *minimum* time of $\frac{2^{32} \cdot 1\text{ms}}{1024} = 4194304\text{ms} = 4194.304\text{s}$. On the other hand, the maximum scan time can be measured in decades if the tool is constrained in terms of processing, storage performance, connectivity or indeed physics. The performance of `Nmap` is discussed in Fyodor (2019).

[1] https://nmap.org/

2.3.2 ZMap

`ZMap` is described in detail in Durumeric *et al.* (2013). In this paper, it is claimed that `ZMap` is significantly faster than `Nmap` (1300 times faster) because of its improved architecture and approach. `ZMap` is stateless and asynchronous, and implements its own TCP/IP stack. It also assumes that its Internet connection is sufficient for the task, and attempts to randomise the order in which hosts are scanned. All of these optimisations result in a single-port scan of the Internet being completed within 45 minutes.

Performance measurements are provided and validated for both the tool standalone, and as compared against `Nmap`. `ZMap` is a clear winner for large-scale scanning, but as others have found (Markowsky and Markowsky, 2015; Jicha *et al.*, 2016), it can be inaccurate as a result of its approach. Justification in the form of positive/White-hat use-cases of such a tool are discussed, as well as recommendations for good behaviour in terms of the use of the tool. Overall, it would appear that `ZMap` represents a significant step forward in producing quick, large-scale scans of the Internet.

When `ZMap` was introduced, the creators claimed that the tool could saturate a 1Gbps network card. Adrian *et al.* (2014) provide `ZMap` optimisations to enable scanning of the Internet at a rate of 10 Gigabits/second. Target address generation is parallelised, black-list processing is optimised, and packets are more efficiently moved to the network interface through *zero-copy packet transmission*. These improvements allow `ZMap` to saturate a 10 Gigabit Ethernet card, and perform a single-port scan of the Internet in 4 minutes and 29 seconds. The short timespan improves the temporal accuracy of results, in the face of dynamically allocated addresses and other changes in the Internet hosting landscape. The researchers unfortunately also found that this dramatic increase in scanning rate produced a far lower 'hit-rate' - 61% of a scan performed at 1Gbps. This again highlights the need to understand the source and scan rate of consumed scan data-sets.

2.3.3 Masscan

At Defcon 22, Graham *et al.* (2015) announced `masscan`[2]: an asynchronous network scanner that is able to scan the entire IPv4 address space within a short amount of time, limited chiefly by the available Internet bandwidth and the networking equipment along the path to the Internet. It is shown to be similar to `ZMap` in terms of performance, accuracy and resource usage.

[2]https://www.youtube.com/watch?v=nX9JXI4l3-E

2.3.4 Shodan

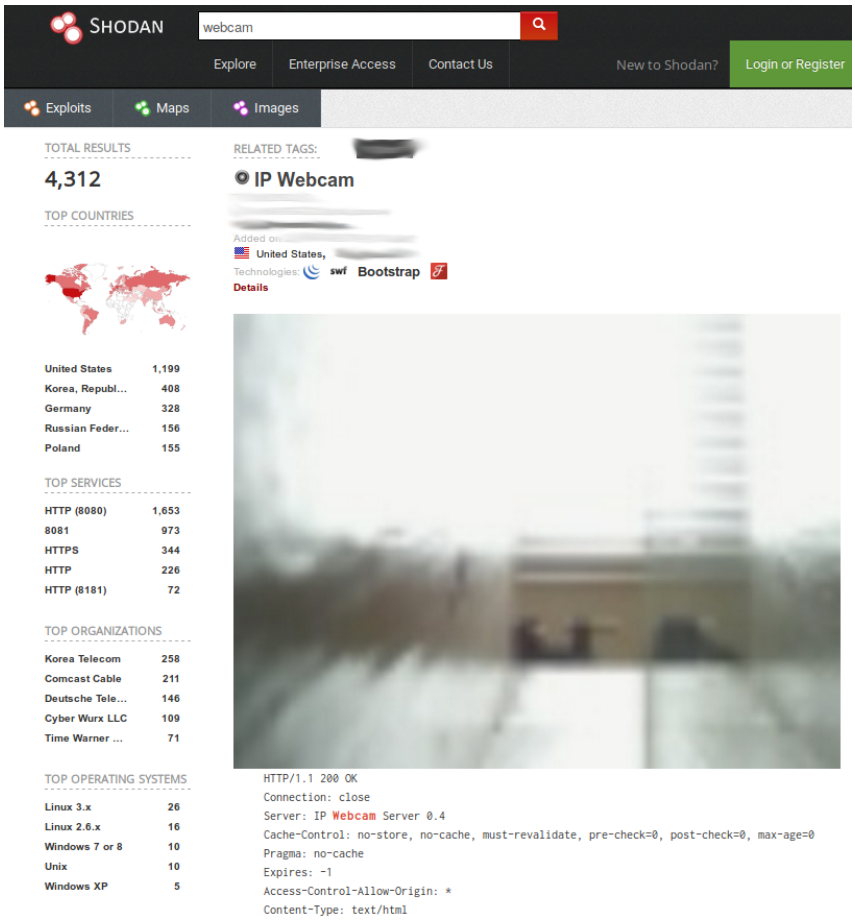


Figure 2.1: Example shodan.io search result (webcam)

A short examination of the Shodan search engine[3] from a historical service point of view is performed in Genge and Enăchescu (2015) and Shodan (2017). Shodan regularly performs selective scans the IPv4 range, searching for hosts with Internet-exposed services. The results of these scans are analyzed, enhanced and loaded into a database. This data is then exposed via both a web interface and as an Application Programming Interface (API). Users can search for devices matching any of the fields present in the database. An example of searching for "webcam" is shown in Figure 2.1.

Markowsky and Markowsky (2015) discuss the Internet Census 2013 (Dittrich *et al.*, 2014; Krenc *et al.*, 2014), along with the botnet used to generate the data. The researchers explore the use of Shodan as a reconnaissance tool for discovering potentially vulnerable hosts. A short qualitative comparison of `masscan` and `Nmap` is performed, in which `masscan` is initially used to discover hosts, due to its vastly superior speed. `Nmap` is then

[3]https://www.shodan.io/

used to perform detailed host vulnerability analysis. It is asserted that `masscan` is less accurate due to its stateless nature, and as such cannot detect packet loss or other errors during scanning.

2.3.5 WiScan

François *et al.* (2016) announce `WiScan`: a scanning method optimised for Industrial Control System (ICS). The researchers explain that the normal approach to scanning risks overloading these systems due to limited resources within their processing units. This could cause failures, and as these are typically used to control industrial processes, including large-scale sites such as power stations, the consequences of such failures could be severe. The researchers then propose an IP generation algorithm with four specific properties: full IPv4 address space coverage, speed, address randomisation and an unpredictable address generation sequence. These properties are desirable when used with ICS.

The algorithm attempts to ensure that adjacent IP addresses are not scanned consecutively within a single scan, by first modifying the higher-order bits during IP generation. This should theoretically increase the time between scans on IP addresses within the same subnet, and so limit the likelihood of overwhelming individual networks.

`ZMap` and `masscan`, respectively, iterate over a multiplicative group and use an encryption-based IP generation algorithm in an attempt to reduce the impact of fast scanning (or perhaps the likelihood of detection!). The improved IP generation algorithm is incorporated into a modified version of `ZMap` and used to perform a full IPv4 scan, and the results compared against both Shodan and Censys.

2.4 Passive Data Collection Tools

Some passive data collection approaches produce valuable intelligence about new and existing Internet-based threats. Darknets, Greynets and Honeypots are amongst the more common varieties of passive network sensors.

2.4.1 Internet Background Radiation/Network Telescopes

Pang *et al.* (2004) discussed the concept of *background radiation* in the context of public Internet traffic. This is *nonproductive*, unsolicited traffic, produced through some form of malicious activity, or misconfiguration. The authors' goals include attempts to classify such traffic, and to produce filters to remove it. Another approach is to actively respond to the incoming traffic and so interrogate the sender in order to understand the intention behind the traffic. In an attempt to avoid unintentional interception or interference of legitimate traffic, the authors propose that unused IP blocks are used as passive collectors of traffic. As these IP addresses have no legitimate traffic travelling inbound or outbound, any received packets can be assumed to be malicious or from misconfigured hosts.

A stated benefit is that there should be little to no harm in responding to this traffic. This approach is typically called a *Network Telescope*, or *Darknet*. A large amount of filtering is performed, so as to limit the processing required for the large volumes of traffic. The remaining traffic is passed to *Application-level responders*, which are protocol-specific responders whose purpose is to extract as much information as possible from the source of the incoming packets. A large amount of port and IP address-based analysis is performed, along with a limited amount of temporal analysis.

2.4.2 Temporal Analysis of IBR

Wustrow *et al.* (2010) examine changes within the Internet by analysing long-term Darknet data collected between 2006 to 2010 on the `35.0.0.0/8` subnet, against a set of newly deployed `/8` subnets. These subnets (`1.0.0.0/8`, `50.0.0.0/8` and `107.0.0.0/8`) were publically announced via BGP through a willing Internet Service Provider (ISP) for a period of one week. A large amount of effort was put into ensuring the new `/8` announcements were successful, and not discarded by upstream ISP filters. Information relating to the IP blocks was also made available to those that might view the appearance as malicious, by publishing information in the the Routing Assets Database (RADB). The authors then categorised network traffic that arrived on these unused blocks into one of three types:

- Scanning: infected hosts or research
- Backscatter: the result of DDOS attacks or similar
- Misconfiguration: some form of hardware or software error

Temporal analysis of the data shows substantial increases in received packet volume over the examined period, growing at approximately double the growth rate of the overall Internet's traffic. The authors then perform a more detailed analysis and attribution of received traffic to the various worms or targeted attacks occurring at particular times during the period.

A full spatial analysis is given, showing traffic rates for various aspects such as source and destination addresses, along with source and destination ports. The interesting concept of *pollution* within the darknet subnets is defined as certain blocks of IP addresses receiving disproportionate amounts of traffic on a particular set of ports. The given potential reasons for this pollution are mostly misconfiguration or bugs, or some form of localised vulnerability. An interesting example of this is the use of the `1.2.3.4` IP address as what is most likely a default or sample IP. Another example is a byte order reversal of the common router gateway address `192.168.0.1`, which becomes `1.0.168.192` when run on a router platform with a different byte-ordering (eg Most Significant Bit (MSB) vs. Least Significant Bit (LSB)) than the manufacturer intended. Once a subnet can be positively identified as *polluted*, it can be filtered from results. Additionally, the responsible Regional Internet Registry (RIR) can mark that subnet as unallocatable.

In RFC6018, an interesting modification of the typical "Darknet" approach is introduced by Harrop and Armitage (2005) and expanded on in Baker *et al.* (2010). The proposal is to capture traffic sent to unoccupied public IP addresses within an active subnet. As these IP addresses are not actively used, no traffic should be received. According to the researchers, this is not the case, and this approach is highly effective. This form of intelligence is possible for any organisation with available public IP addresses.

2.4.3 Denial of Service Attacks

During certain types of DDoS attacks, the attacker's source IP addresses are spoofed. If these source addresses are generated randomly, a darknet network may be included in the source address, and as such will receive packets related to the attack. Based on these packets, an analysis technique called *backscatter analysis* can be used to estimate the number and size of the attacks.

Moore *et al.* (2006) describe *backscatter analysis* as applied to purely non-reflective attacks, wherein the systems under the attackers' control send traffic directly to the victim, with randomised source addresses. The systems under attack, or routers in the path

of the attack, will then reply with either a legitimate response or some form of ICMP packet. These packets can be analysed in terms of their frequency and spread across the Darknet IP range. From three CAIDA network traces, the researchers are able to successfully describe a large number of DDoS attacks within the data. This is achieved by using flow-based analysis, in which short-lived, low packet-count flows are removed from consideration. The remaining flows containing packets destined for more than one glsip addresses within the CAIDA Darknet ranges are considered as attack flows.

2.4.4 Network Telescope Detection Countermeasures

Oberheide *et al.* (2007) show the need for Darknet operators to properly emulate an active network, specifically in terms of its DNS configuration. As may be expected, attackers perform perform reconnaissance on their targets through large-scale scans, which can be detected and examined through the use of a Darknet. The researchers show that in some cases these scans are preceded by reverse-DNS queries for the destination IP addresses. If these so-called *dark DNS* requests produce no results, a potential Darknet has been found, and as such could be filtered from the scanning list. This is not desirable if the goal of a Darknet is to examine scanning and attack techniques, or to create attack rate and size estimates for the larger Internet. To examine this phenomenon, a DNS server was created and configured to be authoritative for a Darknet IP block. The three different potential responses to a reverse-DNS query (none, not present and valid) were configured independently and left to run for seven days each.

A set of tools for automatically responding to DNS queries for large IP ranges is then introduced. This work shows the importance of both configuring some form of DNS response for IP addresses within a Darknet, and of processing the resulting request data.

2.5 Scan Data Processing

Many papers discuss methods for processing the large amount of information provided by large-scale scanning. A large number of use-cases for scan data have been published by researchers. Some of the researchers may have benefited from a view of changes within their datasets over time.

2.5.1 Analysis Methods

In the Real World Crypto conference talk, Holz *et al.* (2011) discuss methods of analysing SSL and TLS datasets, based on a set of open-source tools and frameworks. Concerto (ANSSI, 2018) is a tool-set for Secure Socket Layer (SSL) and Transport Level Security (TLS) dataset analysis, focused primarily on X.509 certificates. Parsifal (ANSSI, 2018) provides parsing tools for (inter alia) X.509 certificate parsing, along with other cryptographic containers and messages. The approach is based on work detailed in the PhD thesis of Levillain (2016). The thesis also provides an analysis of available public datasets, including the EFF SSL Observatory (2010), the Internet Censys (2012) and scans.io data.

Prudhomme *et al.* (2013) examine the IPsec protocol using `ZMap` with a custom payload. The returned certificates and security properties contained within the responses were processed and analysed in detail. Further scanning with protocol-specific tools were performed using a list of IPSec candidates generated from the `ZMap` scan. Low and high security Internet Key Exchange (IKE) scans were performed, respectively taking five and seven days. Several sets of scans were performed in order to examine temporal changes within the data. As with other research, the ethics of scanning and generating unsolicited traffic is discussed. The results were stored in a standard relational database (MySQL). A large amount of statistical analysis was then performed on this data, along with a *12th-order Hilbert curve* visualisation of the results in the IPv4 space.

Visualisation of the IPv4 address space as a Hilbert curve can also be seen in Munroe (2006); Irwin and Pilkington (2008); King *et al.* (2014); Rudis (2015). Visualisation of events over time using this type of view may provide a more intuitive understanding of the changes occuring.

2.5.2 Scanning methods

In 2014, the OpenSSL Heartbleed vulnerability was discovered (US-CERT, 2014; Codenomincon, 2017). It is a high severity vulnerability due to the potential exposure of sensitive key material by the affected systems. In Durumeric *et al.* (2014), the researchers performed scans (using `ZMap`) of the IPv4 address space at regular intervals within 48 hours of the vulnerability disclosure, and analysed the results in terms of the number of vulnerable hosts. The results showed that a large number of sites were patched within 48 hours, yet some hosts were not patched several months later. Some effort was made by

the researchers to inform the owners of these unpatched systems. This work highlights the need for tools such as `ZMap` and `masscan` that can regularly map the Internet in a short space of time. In this way researchers can examine the response of system maintainers over the short timeframe of an event such as Heartbleed.

Antonakakis *et al.* (2017) examine Mirai from many angles using Network Telescopes, honeypots, active Internet scans, the Mirai source code, captured Malware Command and Control (C2C) traffic and others. Gathering this data across time allows the researchers to provide a timeline of infection rates and study the activities of the botnet. The researchers used Censys to analyse scanning data. The Censys data collection period (24 hours or more) limited the usefulness of the data. This is largely due to the infection context (home routers and IoT devices), which typically operate behind a dynamically changing IP address. Other more active measures such as Honeypots were employed in order to gather further intelligence about infected devices. By using these data sources, the researchers were able to analyse the Mirai botnet in depth, and so gain a clear understanding of its size, geographic distribution and behaviour.

Sargent *et al.* (2017) performed scans of the Internet, looking for routers and other devices that respond to Internet Group Management Protocol (IGMP) packets, as a potential amplification attack vector. This task was performed using `ZMap` and a custom module. As is usual with this class of tool, some time is spent attempting to quantify packet-loss through their network. A reasonable number of vulnerable devices are found using this approach.

Hastings *et al.* (2016) used a variety of available scan data-sets[4] in order to examine the strength of the RSA encryption algorithm (RSA) keys used by devices on a variety of public-facing protocols on the Internet. They were then able to factor keys generated using systems with a known weakness in entropy collection. Examining this data over time enabled the researchers to examine both vendor and user patching behaviour in the face of a serious flaw in these products.

Mirian *et al.* (2016) scanned for Internet-exposed ICS devices, using an extended version of `ZMap` that implements the common ICS protocols such as Modbus, BACnet, DNP3, and Siemens S7. The devices found in this manner are then examined and categorised. The authors then analyse data provided by a Network Telescope with "nearly one million addresses". After some analysis it was found that approximately 80% of the received

[4]EFF SSL Observatory (July-December 2010), Heninger, Durumeric, Wustrow, and Halderman (October 2011), Rapid7's Project Sonar (October 2013 onwards) and the Censys scans

ICS scans came from a known set of security research institutions or universities. The researchers then deploy a set of ICS honeypots, in order to gain a better understanding of the intentions of the attackers. The history and usage of various ICS communication protocols are discussed in depth.

Over a period of 1.5 years, Holz *et al.* (2011) performed scans on port 443 for the Alexa top 1 million websites, in order to better understand the state of the public Public Key Infrastructure (PKI). The researchers used `Nmap` to discover valid SSL hosts, and then downloaded the presented certificate chain using OpenSSL. The Electronic Frontier Foundation (EFF) SSL observatory scan of the full IPv4 address space was used as a baseline to validate these results. In addition, the relatively large Internet uplink (10Gbps) of the Munich Scientific Research Network was monitored. An initial sampling effort was performed in April 2011 in which the first 400kB of each outgoing connection was extracted in order to fully capture the initial TLS/SSL handshake and extract the certificate chain. A second sampling run improved the performance by using the Bro Intrusion Detection System (IDS)[5] protocol decoding stack. Further analysis was done on the effects of geo-location on the returned certificate chain, and found no clear evidence of Man-in-The-Middle (MITM) attempts. The analysis shows that the public PKI system had clear issues in terms of certificate validity, configuration and issuance. The results of their research were released in 2011 [6].

Dainotti *et al.* (2012) used a UCSD[7] Telescope with 16 777 216 addresses (a `/8`) to examine the origins and behaviour of a twelve day Sality botnet scan during February 2011. This botnet scanned the entire IPv4 address space searching for Session Initiation Protocol (SIP) based Voice over IP (VoIP) systems. The scans originated from approximately three million unique IP addresses, and employed novel techniques both for coordination of scanning activities, and in the scanning methods used.

Several methods for detecting source IP address spoofing were employed in order to ensure that the packets received by the Telescope were indeed scan packets. This included successfully using the country-wide Internet outage in Egypt (RIPE NCC, 2019) during the period between 27 January 2011 and 2 February 2011 in order to test for decreases in botnet scanning traffic from that country. Additionally, testing for random IP generation and analysing port selection behaviour further confirmed that these were not spoofed IP addresses. Evidence was then found to support the assertion that these scans targeted

[5] https://bro.org/
[6] http://pki.net.in.tum.de/
[7] https://ucsd.edu/

the entire `ipv4` address space, including analysing packet captures from other datasets, and by examining the distribution of packets over the Telescope IP address range.

Lastly, analysis of source port selection for a limited number of persistent hosts allowed the researchers to infer the infected system's Operating System (OS). With this knowledge it was possible to successfully model the expected behaviour of the host's network stack, and use this to interpolate the number of outgoing connections from these hosts, again confirming the `/0` hypothesis. Some interesting visual representations of the traffic, and of the botnet communication channels were performed in this research, notably the use of a space-filling Hilbert curve to represent IP address ranges.

2.5.3 Scan Processing

When announcing the creation of Censys, the authors wrote a paper (Durumeric *et al.*, 2015) discussing their motivation, reasoning and approach. The goal of Censys is to provide security researchers with the ability to easily search the Censys dataset using a powerful query language. The Censys database includes the results of active ports scans, enhanced with Autonomous System Number (ASN), DNS, Geographical IP (GeoIP) and other lookups. Censys has similar goals to the author, although without the addition of passively captured datasets, and time as a factor.

Won *et al.* (2013) use the daily (15 minute) packet traces found in the MAWI repository[8] and perform a *longitudinal study*[9] on traffic patterns on the WIDE[10] Trans-pacific backbone. A piece of software called `MAWILab` Fontugne *et al.* (2010) is able to apply *anomaly detectors* to the data in order to extract anomalous streams from the data in the repository. These streams are classified in terms of either a root cause (Blaster worm etc), known offending ports or unknown cause. From this point the researchers examine the number of unique IP addresses as well as the anomaly count. Various analytical techniques are applied in order to extract meaningful information from the data. These techniques include Auto Correlation Function (ACF), power-law fitting, Gamma, Hough, UNKNOWN (SCANN), Discrete Wavelet Transform (DWT), some of which may be useful when analyzing Network Telescope, scans (Transmission Control Protocol (TCP)/User Datagram Protocol (UDP)) and Packet Capture (PCAP) data.

[8] http://mawi.wide.ad.jp/mawi/

[9] "a correlation research that involved repeated observations over long periods, e.g., decades"

[10] http://www.wide.ad.jp/

2.6 Post Enrichment Work

Jicha *et al.* (2016) present an explanation and method of utilising the output of a connectionless scanner such as `ZMap` or `masscan` to inform more detailed scanning with `Nmap`. The researchers show a higher rate of accuracy with this approach.

These results highlight the requirement for data relating to the origin of publicly available data, in order to have a clear understanding of the validity of that data.

2.7 Summary

There is a large body of work related to capturing, analysing and visualising Telescope data. The data are usually processed by some form of explicitly coded utility, or loaded into a traditional RDBMS. Additionally there appear to be very few (if any) attempts to use DNS attributes to group Telescope data.

This research attempts to provide methods to assist with all of the above through extensive use of Google Google BigQuery (BigQuery) as a database, and by categorising packets based on enrichment data.

In Chapter 3, the processing of enriching the Telescope data with DNS, BGP and port scan data is discussed.

Chapter 3

Experiment Setup

3.1 Overview

The primary goal of this experiment is to construct a queryable database containing all of the PCAP entries captured by a Telescope (Moore *et al.*, 2004). Each individual entry represents a packet received by the Telescope. After enrichment, these entries will contain additional fields relating to the relevant matching enrichment datasets.

In this chapter the experiment is discussed in terms of the PCAP (source) data (section 3.2), the enrichment data (section 3.3), overall data quality (section 3.5) and the process of enriching the Telescope data (section 3.10). Each enrichment dataset is described and examined in terms of size, content, quality and purpose. In addition, the methods required to clean, validate, match and load the enrichment datasets are described for each dataset. Once the enrichment is complete, the resulting data is loaded into a BigQuery database for further analysis. Analysis of the data is described in Chapter 4.

The bulk of the source data was acquired from a commercial company[1] that performs regular scans and lookups of the entire publicly routable IPv4 space. The resulting data is stored and made available to researchers through the Rapid7 OpenData Project (OpenData) project.

The dataset under analysis is a Telescope run by Rhodes University (RU) and made available for the purposes of this research (Irwin, 2011, 2013). The data are analysed over the period covering March 2017 to April 2017. Enrichment attributes PCAP were limited

[1]Rapid7: https://www.rapid7.com/

to those listed in Table 3.2. Further enrichment, such as comprehensive port scans, GeoIP and whois lookups, are possible given additional datasets such as Maxmind[2], HostIP[3] and those in the CAIDA project[4].

3.2 Packet Capture Data

The Telescope run by RU captures all IPv4-based traffic sent to five unoccupied, unpublished IP ranges that are fully accessible by the Internet (Irwin, 2013). Each range is a `/24` subnet[5] containing 256 IP addresses, for a total of 1280 addresses (Fall and Stevens, 2011; Microsoft, 2018). In theory, no packets should ever be addressed to these IP ranges, but in reality a large number of packets are received. As an example, one of the monitored subnets received just over 16 million unsolicited packets in December 2017. The goal of this Telescope is to better understand the origin and intentions of the system generating the packet(s) (Bailey *et al.*, 2006; Moore *et al.*, 2006; Dainotti *et al.*, 2012).

All of the incoming packets are captured in a binary format (PCAP) that stores at least all of the relevant details for the IP and TCP or UDP layers of the network stack (Tcpdump, 2018). In this binary format, individual capture files for all subnets during 2017 range between 1 Gigabyte (GB) and 2 GBs in size. It is important to note that as this is a passive sensor which does not respond to incoming packets, the bulk of the packets can reasonably be expected to be either connection initiation packets, or response packets generated by a victim of a source IP DDoS attack. This limits analysis based on the packet payload, such as may be performed in signature or protocol analysis.

Before the PCAP files can be processed by Python Programming Language (Python) and BigQuery, they must first be converted into a text-based format containing only the fields relevant to this research, such as Protocol, Source/Destination IP address, and Source/Destination Port (where relevant). The JSON format was chosen as it is supported by both platforms, and is also the format used by the structured enrichment data. The conversion process inflates the files by approximately a factor of 17. An example of this inflation is shown by the 7.9 GB March 2017 PCAP binary files, which inflate to 123 Gigabytes (GBs) after conversion to JSON, as can be seen in Table 3.1.

A typical entry from a converted PCAP file can be seen in Listing 1.

[2]https://www.maxmind.com/en/home
[3]http://www.hostip.info/
[4]http://www.caida.org/home/
[5]IPv4 subnet: a subset of the IPv4 network address space

```
{
        "frame":{
                "time":"Nov 30, 2017 23:00:01.473299000 CET",
                ...
        },
        "ip":{
                "dst":"146.x.x.143",
                "proto":"6",
                "src":"61.203.232.191",
                "version":"4",
                ...
        },
        "tcp":{
                "tcp_dstport":"23",
                "tcp_flags":{
                        "tcp_flags_ack":"0",
                        "tcp_flags_reset":"0",
                        "tcp_flags_syn":"1",
                        "tcp_flags_fin":"0",
                        ...
                },
                "tcp_port":"23",
                "tcp_srcport":"38622",
                ...
                },
                ...
        }
}
```

Listing 1: Example PCAP JSON entry

Table 3.1: Network Telescope data

Date range	Packet count	PCAP-format size	JSON size
March 2017	83 million	7.9 GBs	123 GBs
April 2017	71 million	7.3 GBs	106 GBs

3.3 Enrichment Data

There are many sources of data relating to Internet-facing hosts that are available to researchers. Some of these data sources require membership of a particular organisation or group (e.g. VirusTotal/malware groups/Threat Intelligence groups), some (e.g. MaxMind) are commercial offerings and as such are not freely available (but do include a free version), while others are either publicly available or freely available to researchers. For the purposes of this experiment the datasets are limited to three that are publicly available from Opendata[6] and one BGP routing table dataset from South African National

[6]https://opendata.rapid7.com/

Research Network (SANReN). More details about these datasets are shown in Table 3.2.

Table 3.2: Enrichment data

Enrichment type	Source	Capture Frequency	Count	Total Size	Total line count
Reverse DNS (RDNS)	Opendata	Once per week	8	1 Terabyte (TB)	10.16 billion
Forward DNS (FDNS)	Opendata	Once per week	7	1.4 TB	14 billion
Port scans (TCP/UDP)	Opendata	Multiple per week	62	8.5 GB	632 million
BGP routing paths	SANReN	Daily	61	2.5 GB	41 million

3.4 DNS Lookups

The DNS system allows systems (and users of those systems) to use human-readable names in order to discover the underlying host IP address(es). An example of a commonly used DNS name is *www.google.com*. From a machine in Europe this name maps to the IP address `172.217.22.4` (as at 23rd October 2018). These mappings can be used in both directions, in order to gain insight into the intended purpose of a particular IP address.

3.4.1 DNS ANY Lookups

If one performs a reverse DNS lookup on this IP address, two DNS names are returned: `fra15s24-in-f228.1e100.net.` and `fra15s24-in-f4.1e100.net.`. It is important to note that due to the global scale of the company behind this domain (Google), DNS lookups for this domain could return different results in other locations due to geographic load-balancing techniques (Abley and Lindqvist, 2006; Martin, 2018).

Discovery of DNS names is a difficult task due to the unrestricted nature of DNS domain names. Simple enumeration techniques involve using indices such as Alexa (Alexa) or Cisco Umbrella top million sites (Hubbard, 2016; Amazon Alexa, 2019), and performing lookups on those domains with common prefixes such as www, ftp, mail, and the like. This is especially difficult when attempting to enumerate services hosted within the Internet Protocol version 6 (IPv6) space, as discussed in Czyz *et al.* (2014). Once these domains have been mapped to an IP address, performing a reverse DNS lookup will often result in a new domain name upon which a forward lookup can be performed. This process can be repeated until no new domains have been found. Gasser *et al.* (2016) discuss other more complex schemes which result in a higher number of discovered domains.

For the purposes of this research the output of other researcher's efforts in this area was used. The OpenData project makes this data available to researchers[7]. The data contain the results of DNS *ANY* queries on a variety of hostnames.

Each file contains JSON entries, and each entry contains the Unix Epoch time (epoch), the DNS query name, a response type and a response value. Examples of the entries in the file *20170225-fdns.json_any-cleaned.json* can be seen in Listing 2. This file contains just over 2 billion entries and has an uncompressed size of approximately 215 GBs. When compressed this file is just over 22 GBs.

```
{"timestamp":"...943","name":"*.718apts.com","type":"a","value":"66.96.147.87"}
{"timestamp":"...809","name":"*.777040.com","type":"a","value":"199.59.242.150"}
{"timestamp":"...813","name":"*.777142.com","type":"a","value":"199.59.242.150"}
{"timestamp":"...880","name":"*.777145.com","type":"a","value":"199.59.242.150"}
```

Listing 2: Example DNS ANY entry

Once loaded into a database, the data can be queried in order to find hostnames associated with a particular IP address, and as for the RDNS data, used to establish the ownership and related hosts. Given a source IP address of an incoming packet, the researcher can use the FDNS entries to better understand the purpose of the originating system. As an example, a system with DNS entries typical of an web hosting system should typically not be sending packets to an essentially random, unoccupied IP address within a Telescope IP range. This is a potential Indicator Of Compromise (IOC). One can also examine other domains hosted on a single IP address, in an attempt to understand the general purpose of this IP address.

3.4.2 Reverse DNS (RDNS) Lookups

The administrator of an IP netblock can create a DNS Pointer (DNS PTR) entry for the IP addresses under their control. This entry maps an IP address to a user-defined text string, which is typically (Barr, 1996) but not always (Jung *et al.*, 2002) a resolvable DNS entry Lee and Spring (2017). These mappings are used and logged by a variety of systems during normal operation, for example the OpenSSH server (OpenSSH) as described in (Barrett *et al.*, 2005, p. 158), and by systems/network operators performing configuration or operational debugging with a tool such as `mtr`[8]. These DNS records are called *Pointer (PTR)* records (Gulbrandsen *et al.*, 2000).

[7] https://opendata.rapid7.com/sonar.fdns_v2/
[8] mtr: https://github.com/traviscross/mtr

If one iterates over the entire IPv4 address space performing DNS PTR requests for each IP address, a large number of DNS entries can be captured: on the order of approximately 1.2 billion records. There are a total of 4 billion valid IP addresses, once the approximately 324 million reserved addressed are accounted for(IANA, 2018). Surveys of this kind are performed on a regular basis by research institutions such as CAIDA, as well as commercial ThreatIntel providers such as *Censys* and *Rapid7*. In some cases the historical raw data are made available to researchers (CAIDA, 2019; Rapid7, 2019). Other institutions provide the data as an online searchable resource Durumeric *et al.* (2015). For our purposes only the raw data are useful.

Up until March 2018, files containing RDNS lookups for all IP addresses in the IPv4 space could be downloaded directly from the scans.io website[9]. The data is currently (as of October 2018) only available on the OpenData[10] website, and the scans.io links redirect to OpenData. The files are in compressed JSON format with a capture timestamp, IP address and DNS PTR lookup result. The uncompressed file is approximately 136 GBs in size, with approximately 1.2 billion lines. When compressed the file is reduced to 19 GBs in size.

A sample of the file `20170503-rdns.json` can be found in Listing 3. The `timestamp` field is a epoch timestamp, which contains the number of seconds since the Unix Epoch ie. 1st January 1970 (Laboratories *et al.*, 1979). The `name` field is the IP address that formed part of the DNS query, and the `value` field contains the result of the query. The `type` field is always 'ptr' and so can safely be ignored.

```
{
        "timestamp":"1490202186",
        "name":"137.56.53.34",
        "type":"ptr",
        "value":"ww060290.uvt.nl"
}
```

Listing 3: Example reverse DNS JSON entry

Given the entry in Listing 3, looking up an IP address of `137.56.53.34` would produce an entry with an RDNS value of `ww060290.uvt.nl`.

The RDNS name provides insight into the original intention for the host (Eidnes *et al.*, 1998). It can also be used to repeatedly map between RDNS and FDNS, in order to

[9] https://scans.io
[10] https://opendata.rapid7.com/sonar.rdns_v2/

discover related hosts on the same IP address. It is important to note that a malicious or incompetent administrator can cause this mapping to be incorrect or misleading (Zhang *et al.*, 2014). Such entries are interesting in that one can detect and examine these entries in an attempt to understand the intent of this action. Despite this, the data are useful in that the IP addresses of home routers, hosted servers and other potentially compromised sources typically have fixed DNS PTR records set by the ISP. Those that have been modified can be detected by examining IP address group by ASN number. In-depth analysis of this topic can be found in section 3.5.2.

3.4.3 IP Address ASN and BGP Paths

The Internet is a large-scale routed network consisting of thousands of inter-connected systems, as analysed in Tozal (2016). Each system is connected by one or more links, and each system advertises the systems that it is aware of using a protocol called BGP (Rekhter *et al.*, 2006). In a correctly configured system this allows traffic to flow along optimal paths between two systems within the network. For this research full BGP route lists were used. These were sourced from BGP routing tables provided by SANReN, in order to map each incoming packet's source IP address to a BGP routing entry. This then enabled source IP addresses to be grouped by BGP source ASN. Another source of historical BGP data is the University of Oregon Route Views Project[11], which contains historical archives[12] for 21 different sensors going back to 2001.

Each system connected to the BGP network requires a globally unique ASN that is allocated by the local Regional Internet Registry (RIR) (e.g. AfriNIC[13]). This number identifies the Autonomous System (AS), and is used by routers to map between a destination IP address and BGP route entries. The resulting BGP entries are then mapped by the router to a physical link (aka next hop) through which to send packets destined for IP subnets advertised by that AS.

The tables used by the router(s) routing the provided Telescope IP blocks are exported every hour to a binary formatted Multi-Threaded Routing Toolkit (MRT) file (Blunk *et al.*, 2011). Each file is roughly 130 Megabytes (MBs) in size and, after conversion to a Comma Separated Variable (CSV) format, it is reduced to 42 MBs. A sample of the file `routes-mrt.20170320.0800.csv` can be seen in Listing 4. Each line consists of a epoch

[11] University of Oregon Route Views Project: http://www.routeviews.org/routeviews/
[12] http://archive.routeviews.org/bgpdata/
[13] AfriNIC: https://www.afrinic.net/

timestamp, the router ASN, the IP subnet for the entry, and the BGP path to the IP subnet. To reduce processing overhead, a single file per day (08:00 SAST) was selected for the period under study.

```
1 1489989600,37520,100.12.110.0/24,37520 36937 6453 701
2 1489989600,37520,100.12.114.0/24,37520 36937 6453 701
3 1489989600,37520,100.12.131.0/24,37520 36937 6453 701
4 1489989600,37520,100.12.22.0/24,37520 36937 6453 701
```

Listing 4: Example BGP entries

Each entry in Listing 4 contains a Timestamp, router ASN, IP range, and routing path (as a list of ASNs). The IP range entry uses a compact format called IP subnet masksMogul (1984); Mogul and Postel (1985), which specify a base IP address and the number of matching bits (`24` in these entries). For example, line 1 will match 256 IP addresses between `100.12.110.0` and `100.12.110.255`. This is due to the specified `24` bitmask, which matches `32 - 24 = 8` bits i.e. 256 in decimal.

Mapping between IP address and BGP/ASN allows the researcher to cluster IP addresses into related groups. This grouping gives additional weight to an assertion that a particular packet originated from within a particular ISP or network. Additionally, the full BGP path mapped over time allows for the detection of maliciously (or accidentally) injected routes. Recent techniques for anomaly detection are examined in Ćosović *et al.* (2015) and a general survey performed in Al-Musawi *et al.* (2017). This form of IP hijacking allows an attacker to masquerade as another entity or source, and also hides the origin of attacks (Vervier *et al.*, 2015). It is important to note that the path from one router ('A') to another ('B') is not necessarily the same as the return path ('B' to 'A'). This especially true where the originator has injected packets with an incorrect/modified IP address. In this case the BGP lookup will return an ASN that is not that of the packet originator, but for the owner of the modified IP address.

3.4.4 Active Scan Data

The TCP (Postel *et al.*, 1981) and UDP (Postel, 1980) protocols provide both a source and destination port. Typically the source port is randomly selected from a high rangeLarsen and Gont (2011), and the destination port is selected according to a destination service, such as HTTP (port 80/tcp or 443/tcp), Simple Message Transport Protocol (SMTP) (port 25/tcp), Internet Relay Chat (IRC) (port 6667/tcp) and many others (IANA, 2019).

In the case of TCP one can send a single packet to a host's port with the Connection Initiation (SYN) flag set, and if the host has a service running on that port it will respond with a packet with both the SYN and Acknowledge (ACK) flags set. This is part of the standard TCP 3-way handshake (Postel *et al.*, 1981). Many tools such as Nmap[14], ZMap[15] and masscan[16] use some portion of the handshake to *scan* a host in order to determine which ports are open and accessible from an external perspective. A variety of scanning techniques are used by these tools, some of which attempt to conceal scanning activity by exploiting error conditions within the hosts' network stacks (stealth scans) (De Vivo *et al.*, 1999), or to avoid detection by modifying the scan rate (slow scanning) (Dabbagh *et al.*, 2011).

Port scanning is often performed during the reconnaissance phase of an attack. In our case, knowing the active and open services for every host on the Internet is a valuable tool in terms of ThreatIntel. As such, many research institutions scan the entire Internet, looking for hosts with open ports. Some institutions, such as CAIDA[17] and OpenData[18], make these scans available to researchers, and it is these scans from OpenData that were used for this research.

Files containing port scan results (for a limited set of ports) are available from the OpenData[19] [20]. The scans are performed roughly twice a week and published immediately. Historical scans starting from early 2016 are also available for download.

Each file is named according to date, timestamp and port scan type. An example filename is `2017-03-12-1489342874-tcp_syn_3306.csv.gz`, which represents a file containing a list of IP addresses that respond to a TCP connection attempt on port 3306/tcp (*mysql*). This file contains 8.3 million IP addresses. Examples of other file names are shown in Listing 5.

Given a list of open ports on an Internet host, one can begin to make assumptions about the OS, original purpose and current purpose of that host. A host with the Secure Shell (SSH) port open usually indicates some form of Unix, BSD or derivative host. Combinations of ports such as telnet (port 23), ssh (port 22) and http (port 80) are often associated with historically low-cost home routers. Routers that are performing

[14]Nmap: https://nmap.org/
[15]ZMap: https://zmap.io/
[16]MASSCAN: https://github.com/robertdavidgraham/masscan
[17]https://www.caida.org/data/
[18]https://opendata.rapid7.com/
[19]https://opendata.rapid7.com/sonar.tcp/
[20]https://opendata.rapid7.com/sonar.udp/

```
2017-03-01-1488344461-tcp_smb_445.csv.gz
2017-03-01-1488344521-tcp_ssh_22.csv.gz
2017-03-01-1488344821-tcp_clamav_3310.csv.gz
2017-03-01-1488344941-tcp_ldap_389.csv.gz
```

Listing 5: Example port scan filenames

port scans have a high probability of being part of some form of a Botnet (Botnet) such as Conficker, Mirai or similarly managed attack system (Maier *et al.*, 2011; Antonakakis *et al.*, 2017). Large numbers of mongodb (27017/tcp or 27018/tcp) were compromised in late 2016/early 2017 (Kadlec, 2017) due to insecure-by-default configurations.

3.5 Analysing Data Quality

As all of the data used by this research is provided by external parties, it is important to have an understanding of the quality of the data. Problems such as bias, missing or duplicate data can cause the enrichment and analysis process to produce flawed or incorrect results. In order to better understand these issues, the data is first analysed and any flaws eliminated, or the data discarded.

An initial analysis of the data was performed in order to find corrupt, duplicate entries, which were then discarded. Any missing records were noted and taken into account during the analysis phase. Once this was complete, a more detailed data-specific analysis was performed, for instance attempting to understand the number of PCAP entries without corresponding entries within the enrichment data, and vice-versa.

Overall the data were complete, with small numbers of duplicate entries, but a large number of unmatched PCAP entries.

3.5.1 DNS ANY Lookups

Once the DNS ANY lookup data has been processed by JSON Query Tool (jq), it contains only well-formed JSON entries. It was noted that most files contain a low number (less than 100) of entries with missing fields, which is possibly due to corruption during the data collection process.

Due to the *many-to-many* mapping between DNS names and IP addresses, and the effectively arbitrary nature of the data, data quality analysis cannot be definitive. Initially each file was compared against the others, to ensure that at the very least the files were roughly consistent in terms of number of records, unique domain names, and unique IP address count. A summary of the results can be found in table 3.3.

Table 3.3: OpenData FDNS record counts for February/March/April 2017

Filename	FDNS records	Unique FDNS
20170225-fdns.json	2041398012	1205116548

3.5.2 Reverse DNS Lookups

A reasonable (but naive) assumption would be that every valid IPv4 Unicast network address (Unicast) address should be present within the data. This is not the case, as the files contain approximately 1.2 billion records each (more detail in table 3.4), and there are 4 billion valid, routable IP addresses, as discussed in section 3.4.2. This represents a lookup success rate of 0.3. It is difficult to assess the reasons for the missing data, but factors such as failures due to high query rates, exclusion lists and non-routed networks are a possibility.

The CAIDA DNS data[21] for the same period has a higher overall success rate (approximately 0.57), as can be seen in table 3.5, but is biased by the fact that lookups were only performed on IP addresses that were present in active sensor data at the time. The fact that the CAIDA success rate is close to our RDNS enrichment rate is interesting and worthy of further analysis.

Table 3.4: OpenData RDNS record counts for February/March/April 2017

Filename	RDNS records
20170208-rdns.json.gz	1268274655
20170222-rdns.json.gz	1265300959
20170301-rdns.json.gz	1266181574
20170315-rdns.json.gz	1270352864
20170322-rdns.json.gz	1269122458
20170329-rdns.json.gz	1233773475
20170410-rdns.json.gz	1240224642
20170412-rdns.json.gz	1266979223
20170419-rdns.json.gz	1263440219
20170426-rdns.json.gz	1274798784

[21] https://www.caida.org/data/active/ipv4_dnsnames_dataset.xml

Table 3.5: CAIDA RDNS statistics for the first two weeks of March 2017

Filename	Success	Failure	Success rate
dns-names.l7.20170301.txt	2645455	4601863	0.574
dns-names.l7.20170302.txt	2727521	4743622	0.574
dns-names.l7.20170303.txt	2833151	4926050	0.575
dns-names.l7.20170304.txt	2510127	4436143	0.565
dns-names.l7.20170305.txt	2661465	4655401	0.571
dns-names.l7.20170306.txt	2787175	4840018	0.575
dns-names.l7.20170307.txt	2606282	4508434	0.578
dns-names.l7.20170308.txt	2758740	4781959	0.576
dns-names.l7.20170309.txt	2826258	4848894	0.582
dns-names.l7.20170311.txt	2481560	4310130	0.575
dns-names.l7.20170312.txt	3012689	5131231	0.587
dns-names.l7.20170313.txt	2896557	4975587	0.582
dns-names.l7.20170314.txt	2754246	4754909	0.579

For this type of enumeration data, one would reasonably expect a single IP address to appear exactly once in a particular file. Unfortunately all of the files examined had a small number of duplicates, due to an error in the system used to perform the scanning. This was reported to *Rapid7* and fixed as of June 2018. Any duplicates were excluded during processing by selecting the earliest record observed.

The bulk of the data contains entries that conform to a reasonable approximation of a 'standard' DNS pattern, and map to a valid Top-Level Domain (TLD). Some notable exceptions include IP addresses mapped to 'localhost' or other single-word entries.

3.5.3 IP Address ASN and BGP Paths

All of the BGP MRT files for the period under study were successfully transformed with no errors. The files produced by the conversion could be read successfully by the enrichment code with no errors.

During the initial enrichment process, it was noted that a significant number (up to 20%) of source IP addresses were not found within the tables. After some analysis it was found that the data contained overlapping IP ranges: usually a larger range such as a `/8` range and then one or more smaller `/24` ranges. The overlap caused the lookup tree code to completely discard the larger range in favour of the smaller ranges, causing gaps in the tree. The code was corrected such that the larger range is returned instead.

Further enrichment rounds found missing entries for approximately 0.15% of the packets. Given that not every IP address is globally routable due to unused, private or unallocated

IP ranges (IANA, 2018), it is not surprising that the BGP routing tables have gaps. It was, however, surprising to find a small number of entries with source IP addresses that fall within the unroutable (martian) IP ranges (Cotton *et al.*, 2013) such as `10.0.0.0/8`.

3.5.4 Port Scans

The data are presented as a simple list of IP addresses, and no invalid or corrupted entries were found. A basic check for duplicate IP addresses was performed after the data were loaded into BigQuery, using Structured Query Language (SQL) as per Listing 6. This check found no duplicates.

```
select
  count(ip_address)
  ,scan_date
  ,ip_address
from
  censys.tcp_scans_with_timestamp
group by
  scan_date
  ,ip_address
having count(ip_address) > 1
```

Listing 6: Port scan duplicate check SQL

3.6 Experiment Overview

This experiment aims to provide researchers with a method of enriching PCAP data, typically captured as part of either a Telescope system or a honeypot. To that end an attempt is made to give the researcher tools and potential sources of data with which to give an advantage over only using standard methods such as those discussed in Fachkha and Debbabi (2016).

The experiment initially attempted to use only Big Data tools such as BigQuery, but some portions were moved over to a combination of Python and PostgreSQL databases. It would appear that at the time of experimentation, BigQuery did not support queries involving a range of values. Unfortunately the enrichment process involves heavy usage of Timestamp (Timestamp) and IP address lookups to map between the datasets and as such the use of BigQuery is inappropriate for the enrichment portion of the experiment. Analysis of

the resulting enriched dataset is where the Big Data approach is most appropriate and useful. The entire dataset can be quite efficiently queried at a low cost, both in terms of time and money.

Each entry in the PCAP represents a single packet received by the Telescope, and each of these will have fields relating to the four enrichment datasets added by the enrichment process. In every case the matching process uses the IP address of the packet's originating host, along with the arrival time of the packet.

3.6.1 Scope

This research was limited to Telescope packets arriving during the two months of March and April 2017. This period was chosen due to the relatively large scale of the both the enrichment data and the Telescope data, and the costs associated with pre-processing and storing the data within an appropriately sized server, as well as the cost of storing and analysing the data using BigQuery. This approach could be used for continuous enrichment of incoming Telescope data by using a similar time-window and carefully managing the enrichment data sets in use.

3.7 Experiment Prerequisites

As previously mentioned the experiment required a significant amount of resources, in terms of transmission, processing and storage of the Telescope data and the associated enrichment data. Careful planning and allocation of these resources was required in order to proceed with the experiment in a timely and cost-efficient manner. Additionally, in cases where data was not freely available, it was important to maintain a good working relationship with the owner of that data.

3.7.1 Data Acquisition

Despite the name, the OpenData project strongly limits access to the data. A vetting process must be completed, after which the data is provided through either a web site or an API. Due to the size of the files (tens of GBs), an appropriately sized Internet connection is required. The BGP datasets and PCAP files were provided by Rhodes University and transferred to a dedicated server for processing.

3.7.2 Storage and Processing

This experiment requires a large amount of readily accessible, fast, storage: on the order of tens of Terabytes (TBs). The data are provided as compressed JSON files, one per scan, covering multiple time periods. This data can be processed in multiple independent stages and as such a concurrent pipeline approach is appropriate. A modern, multiple core processing system is thus essential if the experiment is to be completed within a reasonable amount of time. In this case, a capable server was rented from a large German hosting provider[22].

3.7.3 BigQuery

As of August 2018, BigQuery provides a new user with $300 of credit. If the researcher is careful, this is sufficient for a limited experiment such as the one described here. Further experimentation would require a larger amount of funding. It is also quite possible to quickly deplete this initial credit through careless automation, as was discovered during this experiment.

3.7.4 Privacy and Operational Security Concerns

PCAP data sourced from a Telescope contains sensitive information such as destination IP addresses. If these IP addresses are made publicly available, they may be used to assist attackers in avoiding detection by those running the Telescope. With this in mind it is appropriate to avoid storing or processing the data in locations where it may be externally accessible, or to distribute the information without careful consideration. During analysis all destination IP addresses were replaced by mapping the five Network Telescope ranges to the Internet Assigned Numbers Authority (IANA) test network ranges.

3.8 Working with BigQuery or Equivalent Databases

Generally speaking, it is advantageous to take advantage of the distributed nature of BigQuery, and structure data such that it is both easy to query and to generate intermediate

[22] https://hetzner.de/

tables for further analysis. "Wide", denormalised tables are advisable when dealing with data of this type. This approach was used to quickly query the data, extract smaller subsets, perform more expensive transforms on the results, and iterate. This technique is more generally known as Extract-Transform-Load (ETL).

In the case of Telescope data, the two primary queries would be source and destination IP addresses and ports, along with the time of the Network Flow (NetFlow) or PCAP event. Given these parameters, a very limited time-slice of all the data-sets can be extracted, and many queries or transforms performed on the resulting data.

3.8.1 Costing Concerns

Platforms such as BigQuery perform billing by calculating the number of GBs processed. It is thus very important to be aware of the amount of data involved in a particular query. If one uses a wildcard such as '*' on a *wide* table, the platform will add all of the available fields to the billing amount. Any unused fields become an unnecessary cost to the researcher, and so limiting the number of fields included in a query is essential.

Other costing concerns include the total amount of data stored on the platform. A relatively large amount of temporary data is generated as part of the enrichment process. This data is stored in intermediate tables, temporary files for ETL, and source files, and will usually be a large multiple of the size of the final tables. It is thus crucial to automatically remove this data either during the process or as soon as possible after completion. As all data on the platform attracts a storage charge, this increases the overall cost of using the system.

3.9 Initial Clean-up and Processing

Before the data can be processed or loaded it is appropriate to ensure it is well-formed and contains no errors that could cause the process to halt. In this section we examine each set of files and create a set of processes that will convert the input files into a format suitable for further processing.

With the exception of the port scan files, the data used in this research is originally provided as files containing lines of JSON. A tool called *jq*[23] can be used to efficiently manipulate JSON entries using a simple but powerful formatting language.

[23] https://stedolan.github.io/jq/

Simple `Python` scripts are used where appropriate to further manipulate some of the data into a format that can be used as part of the enrichment process.

3.9.1 Reverse DNS Lookups

While the DNS PTR files processed in this research had no errors or otherwise corrupted entries as determined by the Python parser, it was required that they were first processed by *jq* in pass-through mode (Listing 7), as the entries could not be successfully parsed by the strict BigQuery JSON parser. At this point it is possible to load the file into BigQuery as-is using the automatic schema processing.

```
jq . 201701.pcap.json > 201701_clean.pcap.json
```

Listing 7: jq pass-through for JSON clean-up

3.9.2 IP Address ASN and BGP Paths

The routing table extracts are in a binary MRT format (Blunk *et al.*, 2011). The binary files are approximately 130MBs in size, and require processing before they can be used. A tool called *bgpdump*[24] was used to convert each file into a CSV file, as per Listing 8.

```
for i in routes-mrt.20170*.0800
do
        echo $i
        bgpdump -m $i |\
         sed 's/|/,/g' |\
         cut -f2,5,6,7 -d',' |\
         sort |\
         uniq > $i.csv
done
```

Listing 8: BGP MRT to CSV conversion

Each line of the CSV files contains a epoch timestamp, source ASN, IP subnet and the path to the destination ASN. Both the timestamp and the source ASN are discarded, as they are redundant for the purposes of this research.

[24]https://bitbucket.org/ripencc/bgpdump/wiki/Home

At this stage it is appropriate to convert provided the IP subnet into an IP range represented by a starting integer and ending integer. In this way, mapping between source IP and BGP entry becomes a simple integer range lookup.

3.9.3 Port Scans

The files retrieved from OpenData are simple lists of IP addresses, one per line. Each file's name is comprised of the date, timestamp, protocol, service and port as shown in Listing 9. Processing the files in this format is very inefficient, so a combination of Bash (bash) and Python is used to transform the filenames, along with IP addresses, into a file with a single entry for every entry in the source files. An example of this can be found in Listing 10.

```
2017-04-05-1491369121-tcp_ldap_3269.csv.gz
```

Listing 9: Port scan file name

```
20170301,1488344461,tcp_smb_445,133.65.20.252
```

Listing 10: Port scan entry example

3.9.4 DNS ANY Lookups

The source data were captured at 3-10 day intervals. Each dataset contains billions of entries, an example of which can been seen in Listing 11. Each file is cleaned by passing it through the jq utility, which accepts malformed JSON and converts it to a well-formed version that is parsable by both BigQuery and Python's JSON parser. An example of this process can be seen in Listing 12. At this stage the data can either be loaded into BigQuery tables using automatic schema detection, or processed by Python scripts.

3.10 Enriching the Network Telescope Data

Once the data has been cleaned, processed and the data quality assessed, the process of enriching the Telescope data can begin. The processing is performed in stages, with the

```
{
        "timestamp":"1522598072",
        "name":"*.5thlegdata.com",
        "type":"a",
        "value":"199.34.228.100"
}
```

Listing 11: DNS lookup example

```
zcat 2018-03-31-1522483201-fdns_any.json.gz |\
 jq -c '{ timestamp:.timestamp, name:.name, type:.type, value:.value}' \
 > 2018-03-31-1522483201-fdns_any-cleaned.json
```

Listing 12: Example DNS ANY file clean-up processing

PCAP file and enrichment data as input to the stage. The output is an enriched Telescope file in JSON format, with the additional data (if available) added to each entry in the file. An overview of this process is shown in Figure 3.1.

3.10.1 RDNS (Only BigQuery)

Once both the Telescope and DNS PTR data has been loaded into BigQuery, the Telescope rows can be enriched with the corresponding DNS PTR entries, by matching the PCAP Timestamp to the DNS PTR time-range, and the source IP to the lookup IP.

After some experimentation it was discovered that joining the Telescope table to the DNS PTR table using text-based IP addresses was so inefficient as to cause the process to fail. Converting the IP addresses to a `long` format resulted in a significant increase in speed. At this stage it also became apparent that BigQuery cannot perform joins of two tables where one of the join fields is a range. This is the case with the DNS PTR lookup, as the scans are performed roughly every 3-6 days and so the available data covers PCAP entries up until the following DNS PTR file's scan date. This approach was abandoned in favour of a code-based solution.

3.10.2 RDNS (ETL/Python/PostgreSQL)

Based on the above discovery, an ETL approach was used to enrich the PCAP. The DNS PTR data is first loaded into a PostgreSQL Database (PostgreSQL) database. A python

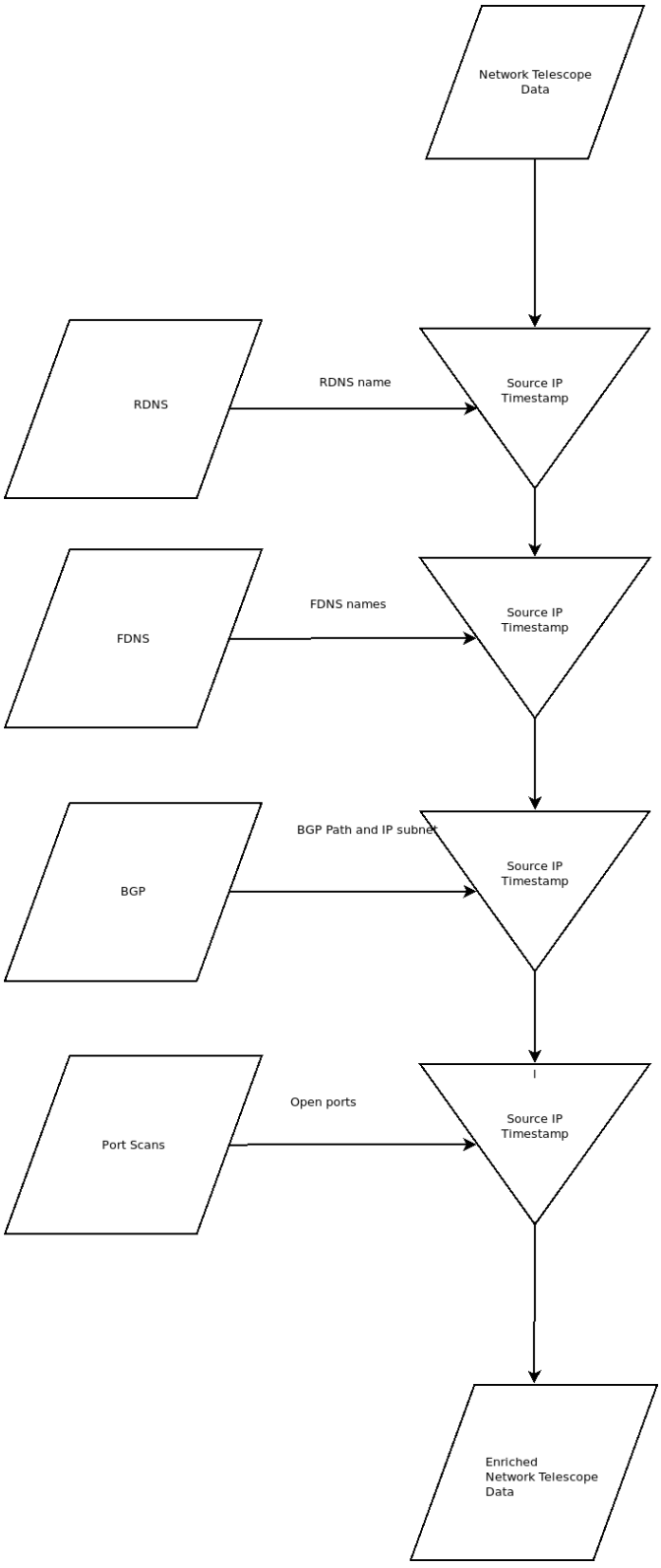


Figure 3.1: Enrichment process overview

script parses the PCAP JSON data, and performs a lookup of the `integer` representation of the source IP address for every row in the PCAP data. The result of this lookup is added to the row and converted back to a JSON entry in a new file. This process is substantially slower than BigQuery should be, taking into account the speed of other queries. Concurrently loading the DNS PTR data into PostgreSQL takes approximately 9 hours, and mapping the data takes a further 10-15 hours. The resulting PCAP data is passed onto the following stage.

3.10.3 BGP (Only BigQuery)

Initially the data were loaded into BigQuery and attempts were made to enrich the PCAP data with the BGP tables. The source IP address was used to look up the related BGP entry in the table, using the start/end IP range. Unfortunately these attempts were not successful due to BigQuery's lack of efficiency when a range query is a significant part of processing. A significant amount of time was spent attempting to work around this issue, and eventually this approach was abandoned.

3.10.4 BGP (Pure Python)

An IPv4 subnet is effectively a bitmask over a 32-bit number . As such a bit-level binary tree is a highly effective lookup mechanism for our purposes, with the desired BGP table entry as the leaf node. This was implemented in Python and resulted in approximately 1.4GB of memory required per daily lookup table.

Each entry in each PCAP file was parsed, the source IP address looked up in the BGP table, the resulting IP subnet plus BGP path added to the entry and written to a new file as JSON. The resulting files were then provided to the following stage for further enrichment.

3.10.5 Port Scans

An sensible approach may be to load the port scans into memory, and enrich each PCAP line using a port scan lookup based on the ip address and timestamp. Unfortunately when using Python loading the port scans uses a large (>64 GB) amount of RAM, exhausting the available resources.

A useful optimisation is to perform multiple passes of processing over the PCAP file. The initial process extracts a list of unique source IP addresses from the PCAP. This IP address list is used to filter and load only port scan entries with an IP address in the list. At this point memory usage has been substantially reduced to less than 2 GBs, and it is possible to proceed with PCAP enrichment. Each line of the PCAP is parsed, the timestamp and IP address is extracted, and a lookup is performed. The resulting list of open ports, if present, is added to the line and output to a new file. This file is passed onto the following stage.

3.10.6 DNS ANY

Due to the size of the DNS ANY files, these entries cannot be stored in a memory-based lookup. As before, we can optimise this process by performing multiple passes over the data, and limiting lookups to only those IP addresses present within the PCAP files. Another alternative is to load the lookup files into a PostgreSQL database and perform the lookups this way.

Each line of the PCAP is parsed within a Python script, the IP address extracted and a lookup performed against the FDNS entries. As there may be multiple entries for each IP address, these are stored as a JSON array in the entry and output to a new file. While further enrichment of this data is possible, for the purposes of this research, the enrichment process can now be considered complete.

3.11 Transforming and Cleaning the PCAP Data

The original source PCAP data is transformed from a binary PCAP format into a single JSON line per packet using *tshark* as per Listing 13. This process is very resource-intensive both in terms of memory and Central Processing Unit (CPU). It consumes multiple GBs of RAM, and takes a large amount of time to complete.

```
tshark -r 201701.cap -T json |\
  dd status=progress |\
  jq -c . |\
  gzip >201701.cap.json.gz
```

Listing 13: Convert binary PCAP to JSON using Tshark

BigQuery requires that input files contain a single line of JSON per row, and so the *tshark* file requires some post-processing to remove the first and last few lines (Listing 14).

```
tail -n +2 jan2017.cap.json > jan2017.cap.json.no_starting_array
head -n -1 jan2017.cap.json.no_starting_array \
  > jan2017.cap.json.no_starting_array_no_ending_array
```

Listing 14: Remove first and last lines from PCAP JSON file

We then remove any commas between entries as per Listing 15.

```
sed 's/^  ,\$//' jan2017.cap.json.no_starting_array_no_ending_array |\
  jq -c . > jan2017.cap.json.no_starting_array_no_ending_array_clean
```

Listing 15: PCAP JSON remove commas between lines

A limited yet large number of fields are extracted from the resulting file using jq and saved to the file which will be imported.

At this stage the input files are ready for import into BigQuery. A schema for the resulting table is required, as the schema detection system appears to sample only a limited number of initial rows from the input file. If subsequent rows contain extra fields, or the datatype differs from what was detected from, say, an empty field, the import will fail.

3.12 Summary

In this Chapter the process of enriching the data captured by a Telescope was described, with the result being a BigQuery database. This database contains the original packets, with added date-specific attributes related to the source IP's RDNS, BGP ASN and the results of periodic port scans.

In the following Chapter 4, this database is used to analyse the Telescope data with respect to these enrichment attributes. Efforts are made to categorise the source IP address at the time of each packet's arrival, and conclusions drawn from this categorisation, as it relates to the intent behind the packet's generation.

Chapter 4

Analysis

4.1 Overview

In Chapter 3, the process of enriching Telescope data using third party data and libraries was discussed. The output of this process was a set of enriched Telescope files that were loaded into a BigQuery database for analysis purposes.

In this chapter, Section 4.2 discusses the analysis process in terms of technologies used, which is followed by an in-depth examination of the enriched Telescope data in Section 4.3. Section 4.4 gives a brief overview of the enriched Telescope data, in terms of the enrichment attributes. Section 4.5 provides a summary.

4.2 Google BigQuery

As its name suggests, Google's BigQuery is a large-scale database, capable of quickly processing and querying TB scale (and up) databases. The analysis in this Chapter was performed using BigQuery for data manipulation and extraction. In this section the reasoning behind this decision is discussed.

4.2.1 Processing

At this stage all of the enriched Telescope data was loaded into a queryable BigQuery database. All analysis was performed either through direct SQL queries or by extracting

subsets of the data and processing it in Python.

A typical approach for processing this volume of data is to load Telescope data into an RDBMS such as PostgreSQL, and rely on optimisations such as fast disk arrays to improve performance. On data-sets of the size required for this research, this approach leads to long-running queries (on the order of minutes to hours).

Examples of BigQuery query times for queries used within this research can be seen in Table 4.1, which shows the real-time duration, 'slot time', maximum worker count and the amount of data processed. These queries performed operations over selected columns in the enriched Telescope, and returned within a few seconds. BigQuery partitions the work-load over many workers, with the total worker time spent returned as the 'slot time'. This time is far higher than the query duration, and is roughly equivalent to the query time of a single database instance.

Table 4.1: Example BigQuery full table query duration

Description	Duration	Slot time	Max. Worker count	Data processed
IP by day	5 seconds	255 seconds	202	3.37 GB
IP by Root Domain Name	13 seconds	735 seconds	200	12.2 GB

The raw, enriched, JSON formatted Telescope data as described in Chapter 3 is approximately 250GB in size before ingestion by BigQuery. Once the data has been imported, the table size reduces to just over 17GBs per Telescope IP range.

At this point it is possible to perform complex queries over the data through the use of a SQL variant tailored towards Big Data scenarios[1]. Data processing is distributed amongst a large number of nodes, aggregated and the results returned.

```
1 select
2   count(*) as packet_count,
3   ip.proto
4 from
5   `network_telescope.split_domains_pcap_all`
6 group by
7   ip.proto
8 order by
9   packet_count desc
```

Listing 16: Example BigQuery SQL statement

The example BigQuery SQL statement shown in Listing 16 appears similar to American National Standards Institute (ANSI) standard SQL (American National Standards In-

[1]https://cloud.google.com/bigquery/docs/reference/standard-sql/

stitute, 2016). There are limitations to this similarity, particularly when concepts such as first-class, queryable, data-structures within columns allows for row-level storage of data-types such as an array of BGP ASN path entries, or a struct containing TCP flags. Additionally, syntactically correct SQL, commonly used with standard SQL engines, can cause BigQuery's partitioning system to produce a sub-optimal processing strategy. An example of this is an outer join between two tables, with ranges. This form of query will cause the entire pipeline to be processed by a single worker node, which results in extremely long processing times or queries that do not complete. In other words, care must be taken to test and understand where the differences lie.

Using BigQuery greatly improves the researcher's ability to query the entire data-set with a relatively small wait time. This allows for quick iterative exploration of the data at relatively low cost both in terms of time and money, as discussed in Section 4.2.2.

Additionally, with knowledge of a programming language such as Python, the researcher is easily able to automate graph and table generation through the use of a combination of the BigQuery API and libraries such as `matplotlib`[2].

4.2.2 Cost

Careful management of cost is essential when using a service such as BigQuery. Data at rest (both files and databases) attract an hourly charge, and each database query is charged according to the amount of data processed during the query. Unoptimised automatic generation of graphs and tables may execute a large number of expensive queries totalling hundreds of US Dollars (USD).

To counter this a cost-limiting strategy is required. Input files, intermediate files and tables should be automatically deleted and the results of queries should be cached. Automated budget alerts should be enabled, such that the research team is notified in the case where excessive costs are being incurred.

Having said this, once all of the queries have been optimised through the use of summary tables, limited usage of columns, and aggressive caching, this approach uses a fraction of the cost of a self-hosted database of equivalent capability - as of January 2019, the Telescope database costs less than $20 per month, including queries and storage.

[2] https://matplotlib.org/

4.2.3 Summary

A distributed database such as BigQuery is a powerful tool for processing and analysing the large data-sets generated by enriching Telescope data. As long as care is taken to manage costs, one can avoid the resource constraints of a typical self-managed standalone database, and see significant gains in terms of reduced processing time. This will result in higher quality research due to the ability to more rapidly change and adapt the research approach to suit changes in results discovered during analysis.

4.3 Network Telescope Data Overview

As introduced in Chapter 3, a Telescope is a set of IP addresses that can be reached from the Internet, but are unoccupied and as such have no legitimate reason to receive any incoming traffic. Any traffic arriving at the Telescope is captured and data written to a PCAP file. In this section some of the defining attributes of IP-based packets such as IP protocol, TCP/UDP ports and source IP address are examined. The goal is to understand the data's complexity and to establish some potential areas where categorisation could assist with analysis of the Telescope data. These attributes are *intrinsic*, in that they are required by the relevant protocol in order to function correctly.

The Telescope data used in this research consists of five `/24` subnets, each containing 256 IP addresses, for a total of 1280 addresses. It is important to note that in the interests of maintaining the integrity of the Telescope, the specific IP ranges used will not be directly mentioned in this document.

For the purposes of this research, all of the Telescope traffic arriving for all of the above IP addresses was treated as if it was destined for a single IP address. Treating the traffic this way has the potential to introduce a bias towards traffic sources that send packets to the entire Telescope, but has the benefit of dramatically simplifying analysis.

To test this bias, the data were aggregated by destination IP address and the number of received packets counted. From this it can be seen that 1271 of the destination IP addresses received between 104 981 and 186 593 packets over the period, while the remaining 9 destination IP addresses received between 210 014 and 616 955 packets. While these 9 received a substantially greater amount of packets, they represent less than 1% of the IP addresses, and the extra packets represent far less than 0.1% of the total packets

received. Given the above, it would appear that packets are reasonably well spread across the destination IP addresses.

4.3.1 IP Protocols

IP packets are intended to facilitate the transport of data between two (or more) IP addresses. These packets must specify a protocol number for the encapsulated data, which is further structured according to the protocol[3]. The packets were broken down by protocol and examined further.

As shown in Table 4.2, all but 1 117 481 of the 153 790 240 packets received by the Telescope consist of TCP and UDP packets. The remaining packets consist of 1 116 418 Internet Control Message Protocol (ICMP), 1 056 Stream Control Transmission Protocol (SCTP) packets, 3 packets for IP protocol 7 (CBT) and a single packet for each of IP protocols 0 (HOPOPT), 41 (IPv6 encapsulation), 95 (MICP), and 216 (Unassigned). These protocols are not included in any analysis due to the relatively low number of associated packets.

Table 4.2: Network Telescope IP protocol breakdown

IP Protocol	Packet count	% all packets
TCP (6)	134 367 171	87.4
UDP (17)	18 305 588	11.9
ICMP (1)	1 116 418	0.7
SCTP ()	1 056	0
7	3	0
0	1	0
41	1	0
95	1	0
216	1	0
Total	153 790 240	100%

Figure 4.1 shows the daily count of packets the Telescope receives for the TCP, UDP and ICMP protocols. TCP packets clearly dominates in this respect, with UDP following closely behind. There are relatively few ICMP packets received by the Telescope, and for this reason ICMP is not included in the analysis.

Certain features of Figure 4.1 are potentially interesting, such as the many spikes in the daily packet counts for TCP, and the two large spikes for UDP. At this level of detail, these appear to be independent of IP protocol, and so are most likely the result of independent activity on protocol-specific ports, as opposed to general scanning activity. This is explored further in Chapters 5 and 6.

[3]See https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml for a list of protocols

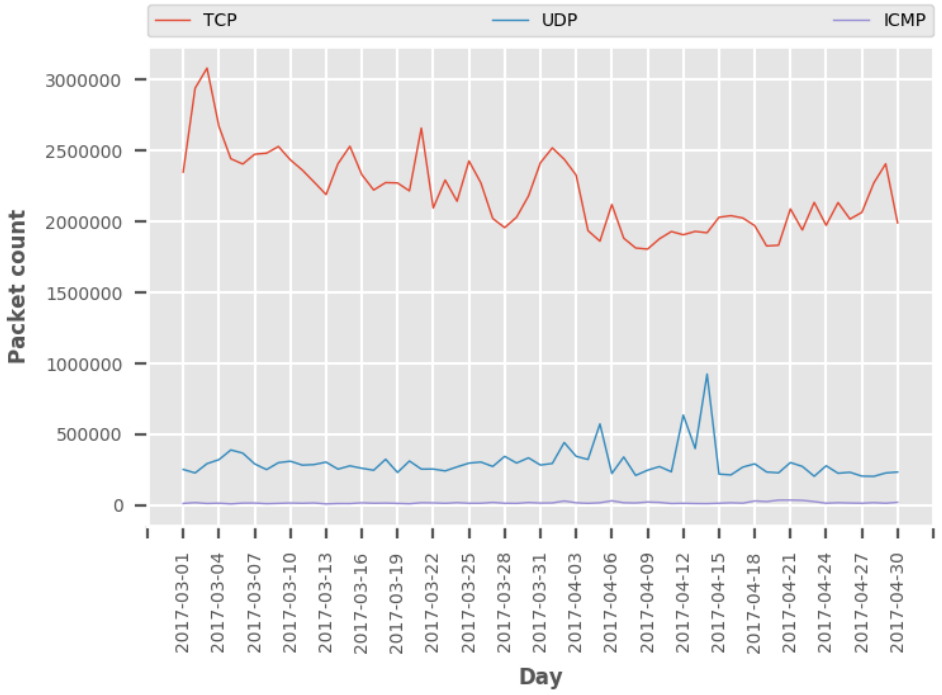


Figure 4.1: IP protocols by day

4.3.2 TCP and UDP Destination Ports

As both TCP (Postel *et al.*, 1981) and UDP (Postel, 1980) have the concept of source and destination *ports*, the packets for these protocols can be broken down by port. Table 4.3 shows the top 10 destination ports per protocol. Figure 4.2 shows the same ports, broken down by day. For clarity's sake, the ports are split into those below 10000 and those above.

For both protocols the top destination ports are mostly those of popular or vulnerable services at the time of packet arrival. Packets destined for port 23/tcp represent a significant portion (37%) of all packets received by the Telescope. This is quite likely due to Mirai or a similar replicating botnet (Antonakakis *et al.*, 2017) scanning for vulnerable routers. This is discussed further in Section 4.3.4. A continuous set of UDP packets with destination ports in the upper ranges are possibly *reflected* packets, as discussed in Section 6.2.

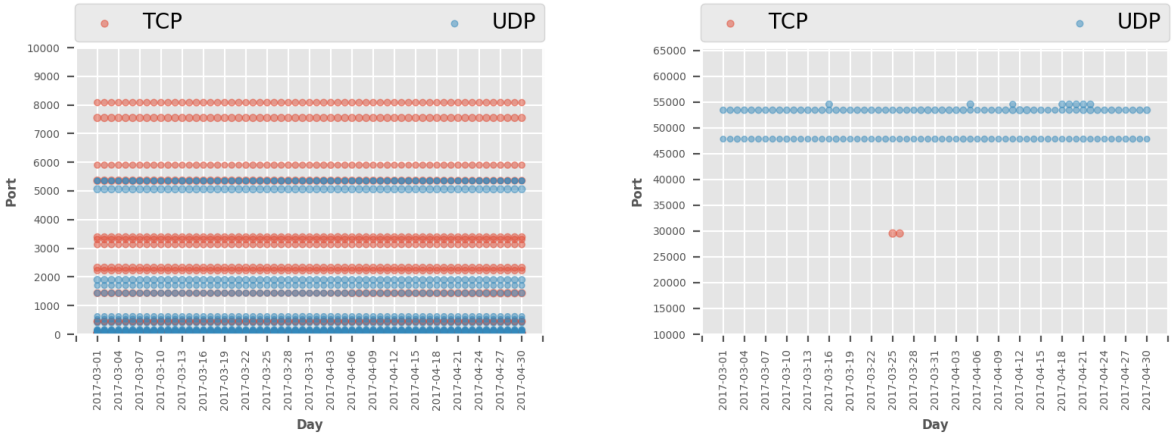
TCP Port	TCP Packets	% Total
23	56 965 111	37.0
22	10 473 768	6.8
5 358	9 349 799	6.1
7 547	7 176 048	4.7
1 433	4 218 327	2.7
2 323	3 834 052	2.5
445	2 687 791	1.7
81	2 137 565	1.4
80	2 112 426	1.4
3 389	1 806 977	1.2
8 080	1 138 733	0.7
2 222	966 252	0.6
443	905 856	0.6
3 306	850 587	0.6
21	669 069	0.4
29 526	506 444	0.3
25	409 785	0.3
5 900	387 588	0.3
3 128	381 739	0.2
88	293 987	0.2
Total	**107 271 904**	**69.8**

(a) Top TCP ports

UDP Port	UDP Packets	% Total
5 060	5 711 716	3.7
1 900	1 870 830	1.2
123	887 942	0.6
53 413	775 373	0.5
53	610 129	0.4
161	399 681	0.3
137	391 260	0.3
1 434	231 778	0.2
19	224 741	0.1
111	163 338	0.1
54 545	155 540	0.1
17	134 388	0.1
1 701	125 576	0.1
47 808	122 499	0.1
5 353	119 008	0.1
389	119 002	0.1
69	115 381	0.1
523	109 484	0.1
5 351	95 689	0.1
623	94 773	0.1
Total	**12 458 128**	**8.1**

(b) Top UDP ports

Table 4.3: Top 20 TCP and UDP destination ports



(a) Low Ports

(b) High Ports

Figure 4.2: Top 20 TCP and UDP destination ports

4.3.3 Source IP Addresses

Figure 4.3 shows the top five sources of Telescope packets by source IP address. The IP addresses exhibit substantial differences in packet generation rates. IP address `185.94.111.1`

appears to follow a pattern of continuous packet generation between 6000 and 8000 packets per day (ppd), with a regular spike of up to 12000 ppd, every 4 days (on average). The second largest packet source (IP `183.60.48.25`) generates a more consistent number of ppd centred around 10500 , but varies by up to 1500 ppd. The third highest source of packets has a packet rate with very little variance over the period. The remaining two packet sources generate packets at a fairly consistent rate, with some small amount of daily variance.

Without further investigation, these differences in behaviour between packet sources are difficult to understand, and hint at underlying complexity which is better understood after categorisation techniques are applied, as discussed in Chapters 5 and 6.

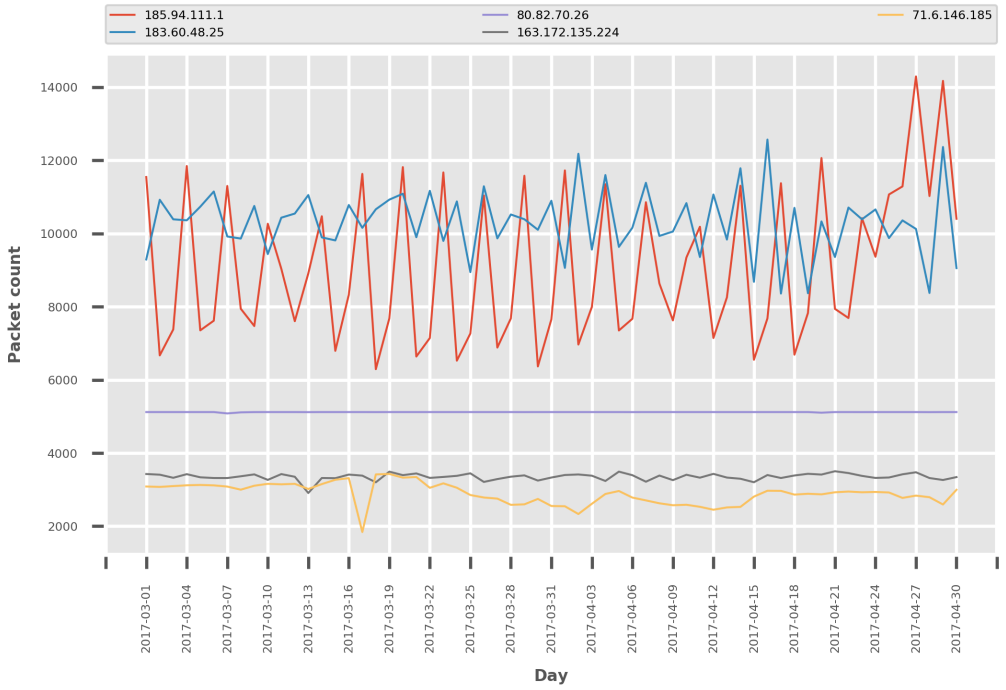


Figure 4.3: Top 5 source IP addresses by protocol

4.3.4 Unique IP Addresses

During March and April 2017, the number of unique source IP addresses across the data drops by approximately 50%: from a maximum of just over 400 000 to a minimum of

200 000 at the end of the period. It could be hypothesised that an offensive campaign started prior to the start of the period, and had not been completed by the end. It is equally likely that a botnet consisting of large numbers of systems had been shut down, resulting in a lower number of systems available for scanning purposes. Figure 4.4 shows the steady reduction in unique IP addresses over the two month period under study.

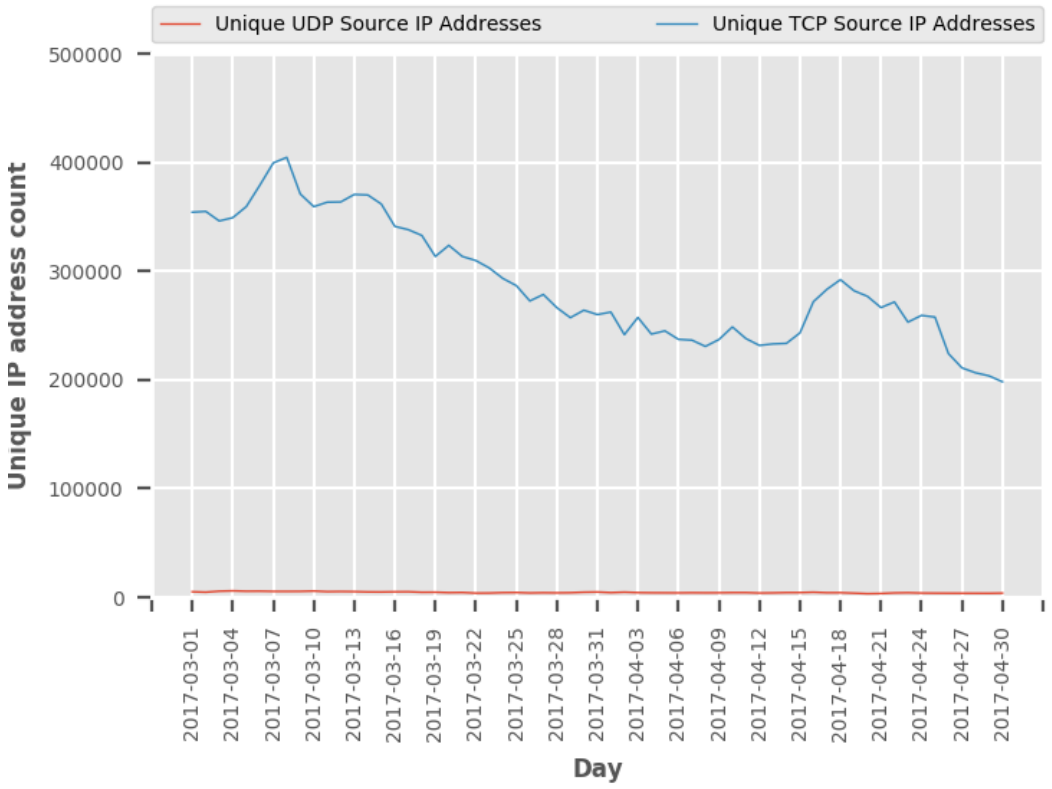


Figure 4.4: Unique source IP addresses

It should be noted that the number of unique IP addresses for TCP packets far outweigh those for UDP packets: Tables 4.4 and 4.5 show 9 406 501 unique IP addresses for TCP vs. 28 516 for UDP. In order to better understand this, source IP addresses were examined in terms of how they relate to other attributes such as destination port.

Table 4.4 also lists the top 10 TCP and Table 4.5 the top 10 UDP destination ports, ordered by the number of unique source IP addresses. The data shows that a large number of source IP addresses generate packets for a small number of destination ports, most notably the ports associated with vulnerable home routers. The top 10 ports account for 66% of all packets received by the Telescope and for 94.8% of the source IP addresses. Packets destined for the ports 23/tcp, 7547/tcp and 2323/tcp account for 44.2% of all packets and for 76.6% of the source IP addresses.

Table 4.4: Unique IP addresses by TCP port

TCP dst port	Packet Count	% Total	Unique IP Count
23	56 965 111	37.0	4 515 396
7 547	7 176 048	4.7	1 545 266
22	10 473 768	6.8	889 470
5 358	9 349 799	6.1	699 916
2 323	3 834 052	2.5	674 259
2 222	966 252	0.6	298 381
81	2 137 565	1.4	295 301
80	2 112 426	1.4	187 493
445	2 687 791	1.7	185 396
88	293 987	0.2	115 623
Sub-total	**95 996 799**	**62.4**	

Table 4.5: Unique IP addresses by UDP port

UDP dst port	Packet Count	% Total	Unique IP Count
3 544	42 765	0.0	14 569
1 900	1 870 830	1.2	3 709
123	887 942	0.6	1 665
9 999	19 201	0.0	1 658
53	610 129	0.4	1 542
137	391 260	0.3	1 531
161	399 681	0.3	1 396
53 413	775 373	0.5	1 137
5 060	5 711 716	3.7	756
19 221	1 212	0.0	553
Sub-total	**10 710 109**	**7.0**	

Given the above, a large portion of the these packets could be attributed to home routers compromised by Mirai, which attempt to propagate through a set of ports common to certain home routers. These ports include 23/tcp (telnet), 7547/tcp (a router management service) and 2323/tcp (a secondary telnet port) that are known to be targeted by Mirai, as discussed in Antonakakis *et al.* (2017); Franceschi-Bicchierai (2016); MalwareTech (2016); Zorz (2016).

The source categorisation efforts described later in this research attempt to confirm this hypothesis by examining the behaviour of packets generated by IP addresses categorised as originating from broadband networks or other home providers.

4.3.5 Summary

Analysis of the Telescope's intrinsic attributes show that the packet sources have varying behaviours and methods of packet generation. Some of these, such as reflected packets

(Paxson, 2001), have well understood intentions such as Denial of Service (DOS) attacks, while others appear to be part of malware propagation attempts as examined in Chen and Ji (2009); Gu *et al.* (2007). Further investigation may yield more potential explanations for the packets source's behaviour. The following

4.4 Enriched Network Telescope Data

In this section the Telescope data is briefly examined in light of the enrichment described in Section 3.3.

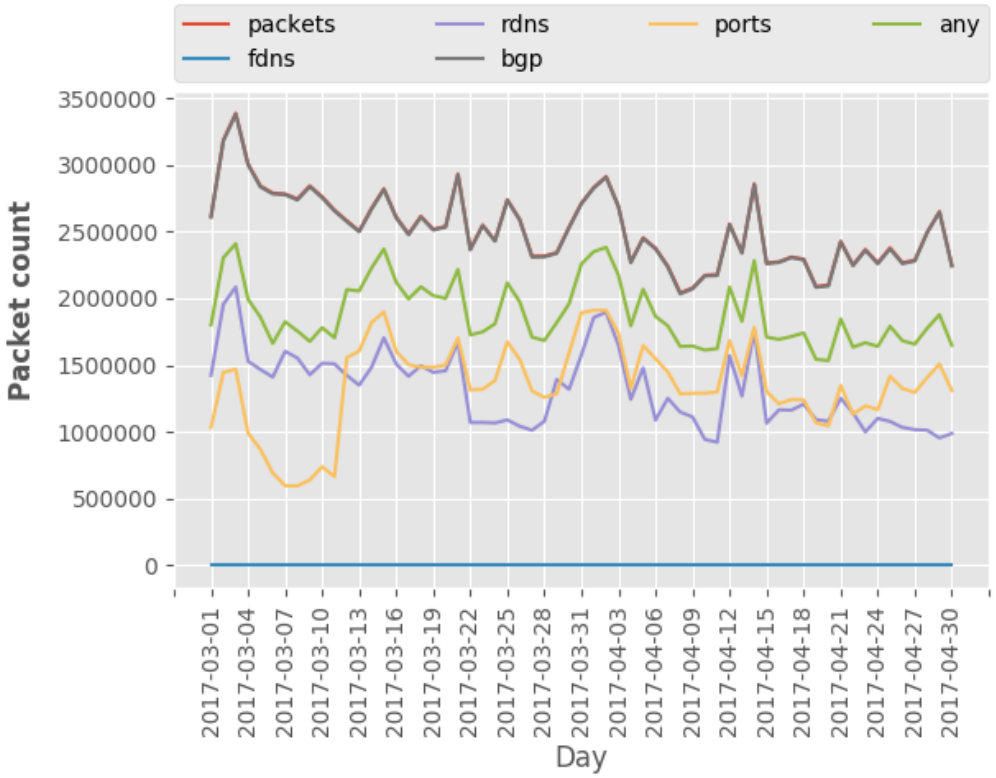


Figure 4.5: Enrichment statistics

Figure 4.5 shows overall success rates for enrichment of the Telescope data. The BGP data (Section 3.4.3) can be matched to close to 100% of the incoming packets, whereas FDNS (Section 3.4.1), RDNS (Section 3.4.2) and port-scan data (Section 3.5.4) can (on average) be matched to 50% of the incoming packets. While the enrichment data-sets have significant overlap, roughly 20% of the packets have only one matching enrichment attribute. This was successfully used these attributes to categorise a large number of source IP addresses.

A portion of the enrichment data is from a reputable source[4], but can be shown as having been corrupted or filtered in a variety of different ways. The most important acknowledged change to the original data is intentional removal of data for a set of undisclosed IP addresses. This is in response to a direct request by the owner of the IP address, or an abuse notice from the owner of the IP block. The purpose of this list is to prohibit scanning of these IP addresses. Unfortunately the list of blocked IP addresses is not made available by the source of the enrichment data.

4.5 Summary

In the following Chapters 5, 6 and 7, the data is categorised using various techniques, and then progressively categorised in terms of the Telescope packet sources, until it can be reasonably stated that the top 500 packet sources, which account for the bulk of the received packets, have been successfully categorised.

[4]Rapid7: https://www.rapid7.com

Chapter 5

Active Traffic Categorisation

5.1 Overview

Active traffic consists of packets where the intention was to interact with a service on the destination system. For the TCP, this would be packets with a destination IP in the Telescope range and with the TCP SYN flag set. This represents the start of a TCP connection handshake (aka 3-way handshake) (Postel *et al.*, 1981). While UDP does not have the concept of flags in its protocol specification, packets with standard UDP service ports have a high likelihood of being connection establishment packets. In this Chapter we attempted to analyse only the traffic which can be considered *active*. Passive traffic was removed from consideration, and is discussed in Chapter 6.

Section 5.5 addresses packets that originated within research organisations. Mobile networks are examined in Section 5.6, hosting and service providers in Section 5.7 and residential broadband in Section 5.8.

Unsuccessful categorisation, either as a result of a problem during the categorisation process or due to no RDNS data, are respectively addressed in Section 5.9 and 5.10. The Chapter is summarised in Section 5.11.

5.2 Network Telescope Data Categorisation

The goal of this research is to effectively categorise captured Telescope data. This is achieved by enriching the data using publicly available Open-Source Intelligence (OSINT)

data-sets. Using this additional information, it was possible to infer details relating to the originator of incoming packets. Examples of these details include the originating organisation or hosting company, network location and path, open ports on the originating host and DNS forward/reverse mappings. For some hosts it may also be possible to infer knowledge about the state of the system (compromised/not compromised) and its original purpose.

With this goal in mind, a series of categorisation efforts was performed. Initially every source was broadly categorised into one of the following:

- Research institution (commercial or otherwise): organisations whose purpose is understood to include Internet-based research
- Hosting or service provider: providers of Internet hosting services such as web-hosting or dedicated servers
- Residential: residential broadband ISP
- Uncategorised, with enrichment data: category could not be determined, either due to no web presence, or language difficulties
- Absent or unreliable, where enrichment data is not present, or unreliable
- Reflected: traffic from known-good networks, with ephemeral destination ports, and service source ports (Discussed in Passive Traffic Section 6.1)

5.3 Root Domain Aggregation

Initial categorisation was performed using DNS enrichment attributes. For every incoming packet within the Telescope data, the associated RDNS attribute is reduced to the first part after the TLD, which was named the *root domain name*. Packets are then grouped by the root domain name and sorted according to the number of packets that arrived.

An example is `em1-114-17-2.pool.e-mobile.ne.jp`, where the domain name is reduced to `e-mobile.ne.jp`. This reduction is performed by searching for a list of typical first or second level structural domain name parts such as `com/net/org` for first level, or `co/ac/org` for second level. In the case of Japanese domain names, `ne` is the equivalent to `co`. The domain name part before this is prefixed to the structural parts, and this becomes the root domain name.

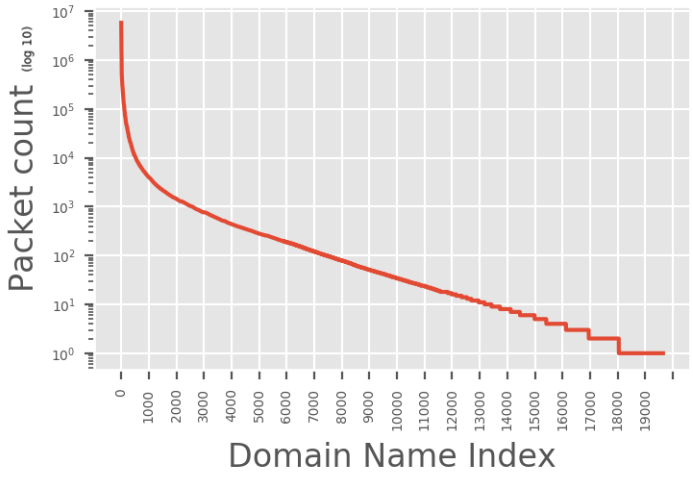


Figure 5.1: Packet sources by category

This list of root domains was the basis for further investigation. Figure 5.1 shows the distribution of packets across all domain names, with a small number of domains accounting for a large portion of the traffic. Table 5.1 shows that the top 20 root domains produce 24% of the packets with DNS attributes.

Table 5.1: Top root domains, by packet count

Rank	**Domain Name**	**Packet count**	**% Total**
1	hinet.net	5 804 412	3.8
2	poneytelecom.eu	4 571 598	3.0
3	shodan.io	2 713 214	1.8
4	fastwebserver.de	2 684 037	1.7
5	ttnet.com.tr	2 683 356	1.7
6	speedy.com.ar	2 561 841	1.7
7	shadowserver.org	2 304 662	1.5
8	1e100.net	1 786 883	1.2
9	stretchoid.com	1 786 068	1.2
10	gvt.net.br	1 557 185	1.0
11	vnpt.vn	1 454 584	0.9
12	163data.com.cn	1 373 006	0.9
13	virtua.com.br	1 229 498	0.8
14	linode.com	1 153 341	0.7
15	prod-infinitum.com.mx	818 472	0.5
16	telecomitalia.it	815 023	0.5
17	dreamhost.com	759 059	0.5
18	servdiscount-customer.com	757 517	0.5
19	ngprobd.com	547 565	0.4
20	vultr.com	516 913	0.3
	Sub-total	**37 878 234**	**24.6**

5.4 Domain Categorisation

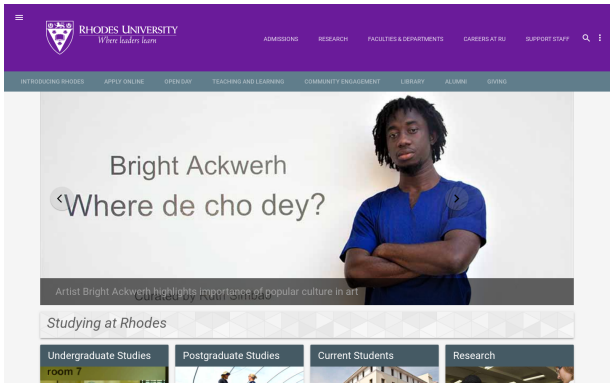


Figure 5.2: Screen capture of the URL https://ru.ac.za/

The top 500 unique root domains (2.6% of all root domains) account for just under 90% of the DNS-enriched packets and so it was effective to limit the scope to these domains. Classifying each domain is a manual process requiring a fair amount of investigation. A tool called `gowitness`[1] was used to create screen captures of the HTTP and HTTPS versions of the domain names, as well as a version prefixed with `www.`. Each of the 820 images was manually examined and used to classify the domain. Figure 5.2 is an example of a screen capture for the domain `https://ru.ac.za/`.

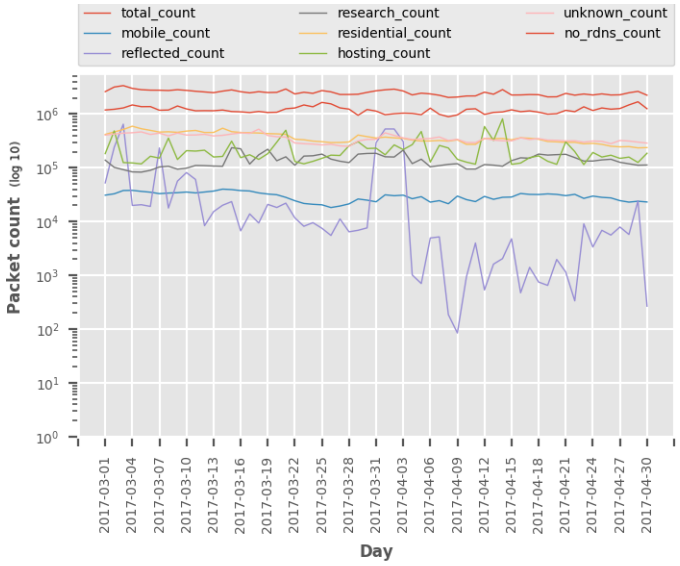


Figure 5.3: Packet sources by classification

Classification using DNS attributes yielded a success rate of 52.4%. Table 5.2 and Figure 5.3 show that a large portion (14.7%) of the packets originated from within residential

[1]https://github.com/sensepost/gowitness

Internet Service Providers (ISP) networks (Section 5.8), at a consistent rate. Hosting networks (Section 5.7) produced packets at a rate that varied on a daily basis, with an average of 8.6% of the total. This was closely followed by research networks (Section 5.5) at 5.5%. Mobile networks (Section 5.6) produced a significant yet relatively small number of packets, at 1.2%. Uncategorised packets (Section 5.9) with RDNS attributes, but without a category, total 14.2%.

Table 5.2: Packet sources by classification

frame day	total	mobile	% total	reflected	% total	research	% total	residential	% total	hosting	% total	unknown	% total	no rdns	% total
2017-03-01	2 615 609	31 097	1.2	52 561	2.0	139 262	5.3	419 306	16.0	184 161	7.0	406 292	15.5	1 192 474	45.6
2017-03-02	3 186 470	32 971	1.0	242 208	7.6	102 246	3.2	465 411	14.6	486 996	15.3	426 558	13.4	1 230 987	38.6
2017-03-03	3 387 701	37 941	1.1	648 698	19.1	92 709	2.7	517 089	15.3	125 458	3.7	442 318	13.1	1 302 267	38.4
2017-03-04	3 007 789	38 201	1.3	20 029	0.7	84 215	2.8	595 821	19.8	123 642	4.1	450 408	15.0	1 479 850	49.2
2017-03-05	2 842 547	36 570	1.3	20 661	0.7	83 372	2.9	534 624	18.8	118 603	4.2	466 538	16.4	1 375 359	48.4
2017-03-06	2 789 110	35 294	1.3	19 198	0.7	90 160	3.2	498 478	17.9	164 431	5.9	418 530	15.0	1 379 251	49.5
2017-03-07	2 782 508	33 164	1.2	234 688	8.4	104 079	3.7	462 941	16.6	152 023	5.5	442 253	15.9	1 179 593	42.4
2017-03-08	2 745 110	34 037	1.2	17 926	0.7	107 237	3.9	469 276	17.1	357 672	13.0	393 731	14.3	1 192 649	43.4
2017-03-09	2 843 201	34 532	1.2	56 705	2.0	94 664	3.3	457 798	16.1	143 168	5.0	444 392	15.6	1 416 163	49.8
2017-03-10	2 762 021	35 567	1.3	81 825	3.0	98 718	3.6	483 138	17.5	209 096	7.6	407 861	14.8	1 248 341	45.2
2017-03-11	2 661 143	34 225	1.3	60 952	2.3	110 924	4.2	496 791	18.7	205 249	7.7	408 821	15.4	1 152 206	43.3
2017-03-12	2 581 837	35 573	1.4	8 421	0.3	110 076	4.3	454 877	17.6	213 065	8.3	420 947	16.3	1 159 577	44.9
2017-03-13	2 504 550	37 032	1.5	15 355	0.6	108 214	4.3	455 701	18.2	159 256	6.4	393 012	15.7	1 155 101	46.1
2017-03-14	2 675 234	40 245	1.5	20 177	0.8	107 049	4.0	546 706	20.4	163 426	6.1	403 263	15.1	1 191 397	44.5
2017-03-15	2 821 146	39 463	1.4	23 597	0.8	237 001	8.4	471 371	16.7	314 051	11.1	425 550	15.1	1 116 812	39.6
2017-03-16	2 609 716	37 816	1.4	6 745	0.3	226 704	8.7	455 421	17.5	154 875	5.9	448 182	17.2	1 102 263	42.2
2017-03-17	2 484 217	37 180	1.5	13 927	0.6	117 714	4.7	450 072	18.1	175 463	7.1	437 701	17.6	1 068 498	43.0
2017-03-18	2 615 062	34 244	1.3	9 440	0.4	174 708	6.7	444 768	17.0	143 845	5.5	524 535	20.1	1 120 055	42.8
2017-03-19	2 517 501	32 661	1.3	20 956	0.8	222 268	8.8	433 914	17.2	183 632	7.3	399 454	15.9	1 071 991	42.6
2017-03-20	2 539 516	31 651	1.2	18 345	0.7	135 740	5.3	429 161	16.9	300 868	11.8	379 542	14.9	1 082 388	42.6
2017-03-21	2 932 275	28 308	1.0	22 094	0.8	160 633	5.5	420 030	14.3	501 748	17.1	374 488	12.8	1 258 260	42.9
2017-03-22	2 368 730	24 500	1.0	11 744	0.5	111 951	4.7	341 049	14.4	131 361	5.5	290 724	12.3	1 299 106	54.8
2017-03-23	2 549 570	21 799	0.9	8 206	0.3	165 510	6.5	333 201	13.1	118 998	4.7	281 450	11.0	1 479 488	58.0
2017-03-24	2 433 223	21 019	0.9	9 578	0.4	167 950	6.9	314 968	12.9	132 417	5.4	277 208	11.4	1 367 038	56.2
2017-03-25	2 739 492	20 567	0.8	7 529	0.3	180 336	6.6	306 737	11.2	150 601	5.5	267 234	9.8	1 650 805	60.3
2017-03-26	2 592 556	18 302	0.7	5 540	0.2	145 570	5.6	295 486	11.4	172 414	6.7	275 005	10.6	1 551 047	59.8
2017-03-27	2 316 101	19 524	0.8	11 228	0.5	132 963	5.7	296 565	12.8	169 866	7.3	257 151	11.1	1 304 593	56.3
2017-03-28	2 317 431	21 414	0.9	6 472	0.3	125 334	5.4	303 480	13.1	258 164	11.1	251 337	10.8	1 238 581	53.4
2017-03-29	2 342 702	26 396	1.1	6 907	0.3	180 319	7.7	407 851	17.4	306 875	13.1	316 258	13.5	950 690	40.6
2017-03-30	2 535 038	25 121	1.0	7 660	0.3	185 411	7.3	381 131	15.0	229 250	9.0	337 943	13.3	1 216 825	48.0
2017-03-31	2 714 060	23 410	0.9	295 380	10.9	186 590	6.9	358 689	13.2	232 358	8.6	338 146	12.5	1 137 600	41.9
2017-04-01	2 831 283	31 549	1.1	529 348	18.7	160 870	5.7	373 760	13.2	175 244	6.2	448 223	15.8	974 531	34.4
2017-04-02	2 913 365	30 344	1.0	529 018	18.2	159 211	5.5	361 615	12.4	267 665	9.2	401 562	13.8	1 017 302	34.9
2017-04-03	2 690 833	31 095	1.2	308 534	11.5	213 970	8.0	353 714	13.1	218 962	8.1	361 810	13.4	1 043 217	38.8
2017-04-04	2 273 225	26 703	1.2	1 022	0.0	120 006	5.3	328 117	14.4	270 559	11.9	337 553	14.8	1 030 599	45.3
2017-04-05	2 453 938	29 158	1.2	706	0.0	144 972	5.9	316 122	12.9	474 027	19.3	345 781	14.1	975 265	39.7
2017-04-06	2 376 599	23 066	1.0	4 967	0.2	104 721	4.4	316 676	13.3	127 947	5.4	352 253	14.8	1 289 397	54.3
2017-04-07	2 241 792	24 597	1.1	5 211	0.2	110 185	4.9	323 380	14.4	261 666	11.7	375 489	16.7	990 646	44.2
2017-04-08	2 041 402	21 536	1.1	187	0.0	116 297	5.7	314 313	15.4	234 345	11.5	322 589	15.8	893 795	43.8
2017-04-09	2 077 158	29 919	1.4	85	0.0	119 647	5.8	332 643	16.0	144 133	6.9	339 442	16.3	966 020	46.5
2017-04-10	2 172 832	25 700	1.2	937	0.0	95 251	4.4	273 840	12.6	127 909	5.9	296 324	13.6	1 230 237	56.6
2017-04-11	2 178 980	23 615	1.1	4 008	0.2	95 175	4.4	274 084	12.6	117 197	5.4	294 516	13.5	1 257 435	57.7
2017-04-12	2 557 677	29 483	1.2	537	0.0	115 208	4.5	351 042	13.7	587 698	23.0	355 214	13.9	986 397	38.6
2017-04-13	2 343 429	26 045	1.1	1 622	0.1	112 404	4.8	349 643	14.9	322 663	13.8	328 952	14.0	1 075 808	45.9
2017-04-14	2 857 710	28 375	1.0	2 055	0.1	107 427	3.8	353 306	12.4	816 475	28.6	319 152	11.2	1 097 329	38.4
2017-04-15	2 265 394	28 682	1.3	4 807	0.2	136 317	6.0	332 819	14.7	116 202	5.1	317 158	14.0	1 201 263	53.0
2017-04-16	2 274 038	33 379	1.5	475	0.0	153 867	6.8	362 693	15.9	123 028	5.4	362 530	15.9	1 109 347	48.8
2017-04-17	2 309 967	32 162	1.4	1 413	0.1	151 180	6.5	354 809	15.4	155 054	6.7	336 623	14.6	1 148 187	49.7
2017-04-18	2 294 792	31 965	1.4	752	0.0	180 625	7.9	340 925	14.9	167 264	7.3	350 098	15.3	1 087 364	47.4
2017-04-19	2 089 526	32 861	1.6	653	0.0	172 879	8.3	309 700	14.8	132 861	6.4	327 883	15.7	998 614	47.8
2017-04-20	2 098 873	32 143	1.5	1 981	0.1	176 294	8.4	304 875	14.5	116 409	5.5	325 470	15.5	1 019 802	48.6
2017-04-21	2 427 833	30 435	1.3	1 156	0.0	178 780	7.4	297 867	12.3	304 873	12.6	316 736	13.0	1 176 532	48.5
2017-04-22	2 250 011	32 219	1.4	337	0.0	155 393	6.9	298 713	13.3	201 036	8.9	319 045	14.2	1 109 108	49.3
2017-04-23	2 366 597	27 082	1.1	9 102	0.4	133 868	5.7	280 805	11.9	115 097	4.9	300 805	12.7	1 367 531	57.8
2017-04-24	2 267 064	29 901	1.3	3 379	0.1	134 037	5.9	285 907	12.6	193 709	8.5	315 499	13.9	1 166 802	51.5
2017-04-25	2 378 428	28 340	1.2	6 904	0.3	140 747	5.9	274 105	11.5	159 136	6.7	311 573	13.1	1 300 973	54.7
2017-04-26	2 267 007	27 658	1.2	5 630	0.2	144 605	6.4	252 904	11.2	171 985	7.6	281 878	12.4	1 234 690	54.5
2017-04-27	2 285 629	24 590	1.1	7 957	0.3	126 981	5.6	245 953	10.8	150 280	6.6	323 052	14.1	1 269 746	55.6
2017-04-28	2 494 608	23 081	0.9	5 776	0.2	119 131	4.8	248 799	10.0	158 671	6.4	315 798	12.7	1 482 682	59.4
2017-04-29	2 649 679	23 995	0.9	24 379	0.9	112 117	4.2	235 560	8.9	126 019	4.8	301 424	11.4	1 695 908	64.0
2017-04-30	2 247 405	23 228	1.0	268	0.0	113 114	5.0	238 333	10.6	187 153	8.3	289 102	12.9	1 260 215	56.1
Total	**153 790 240**	**1 802 730**	**1.2**	**3 447 958**	**2.2**	**8 394 906**	**5.5**	**22 588 873**	**14.7**	**13 208 299**	**8.6**	**21 780 366**	**14.2**	**73 127 997**	**47.6**

5.5 Research Institutions

For a variety of reasons, many research institutions perform scans of the Internet's IPv4 address space. The port scan data-set used with this research was sourced from one of those institutions, and the resulting packets can be found within the Telescope data. Table 5.3 shows a list of root domains classified as organisations that fall into this category.

Table 5.3: Packets from the top 10 research institutions

Rank	**Root Domain**	**Packet Count**	**% Total**
1	shodan.io	2 713 214	1.8
2	shadowserver.org	2 304 662	1.5
3	stretchoid.com	1 786 068	1.2
4	umich.edu	480 095	0.3
5	binaryedge.ninja	349 666	0.2
6	rapid7.com	221 850	0.1
7	ru.ac.za	166 668	0.1
8	ipip.net	86 526	0.1
9	ruhr-uni-bochum.de	67 323	0.0
10	ucl.ac.uk	63 139	0.0
	Sub-total	**8 239 211**	**5.4**

Guidelines for good scanning behaviour are discussed in Section 5 of Durumeric *et al.* (2013). Included in the list is a request to use DNS entries to clearly indicate the source of the packets. This type of recommendation increases the chances of categorising research institutions using RDNS.

5.5.1 Packet Distribution

Figure 5.4 show the number of packets per day identified as originating within the research institutions identified during the period under study. While most of the institutions are 'well behaved', Figure 5.5 shows that others such as Shodan generate scans that target a wider range of ports, resulting in larger amount of scanning packets relative to the others in this category. Comprehensive scans of this nature are quite likely to trigger alerting systems and could possibly be mis-attributed as as attack.

Another scanning guideline is to "spread traffic over time or source addresses when feasible". Over the period under study, Rapid7 and Stretchoid performed what appear to be short-lived campaigns, whilst the others produce packets at a close to constant rate.

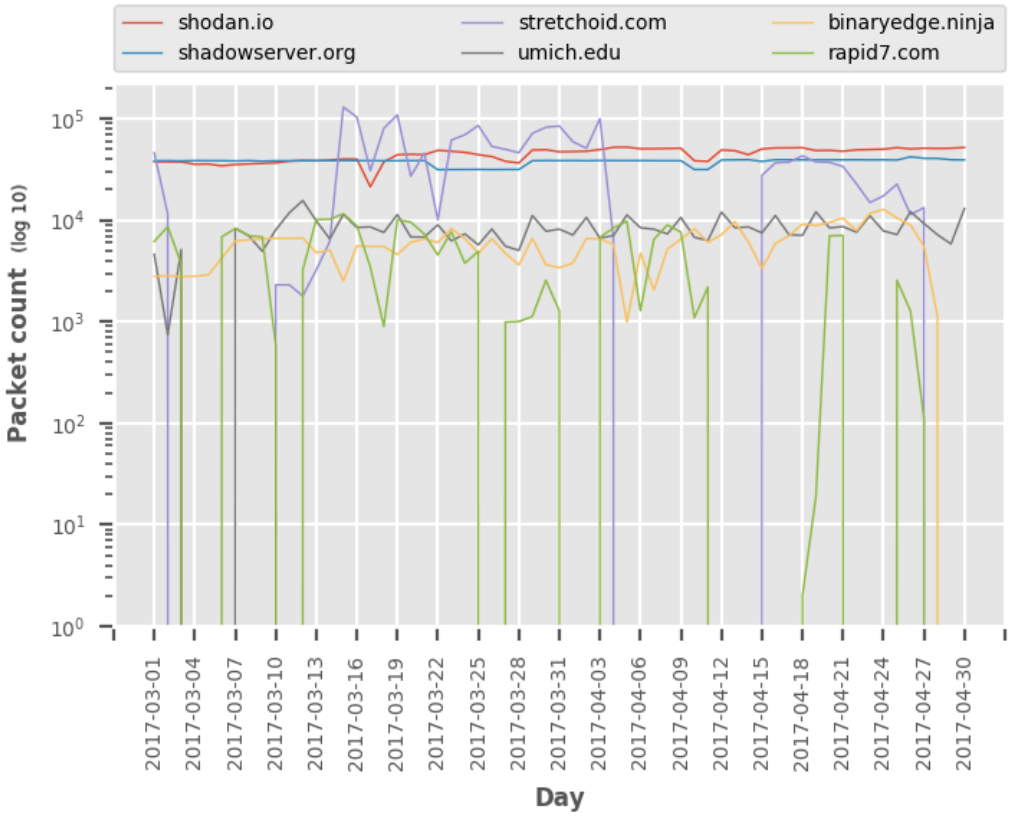


Figure 5.4: Daily traffic for the top 6 in the Research category

5.5.2 Destination Ports

Figure 5.5 shows the overall distribution of packets originating from research institutions across different destination ports. As with the packet count, Shodan performs scans across a wider range of ports. Other institutions have a narrower focus on a limited number of ports. The difference in behaviour could be attributed to the potential goals of scanning, such as:

- Understanding the prevalence of known vulnerabilities
- Examining changes in host behaviour over time
- Simple cataloguing of host attributes, presented via search engine
- Specialised research

The intent of the researchers is by itself a useful source of knowledge. By examining the scanning behaviour of these research institutions, one can become aware of new threats

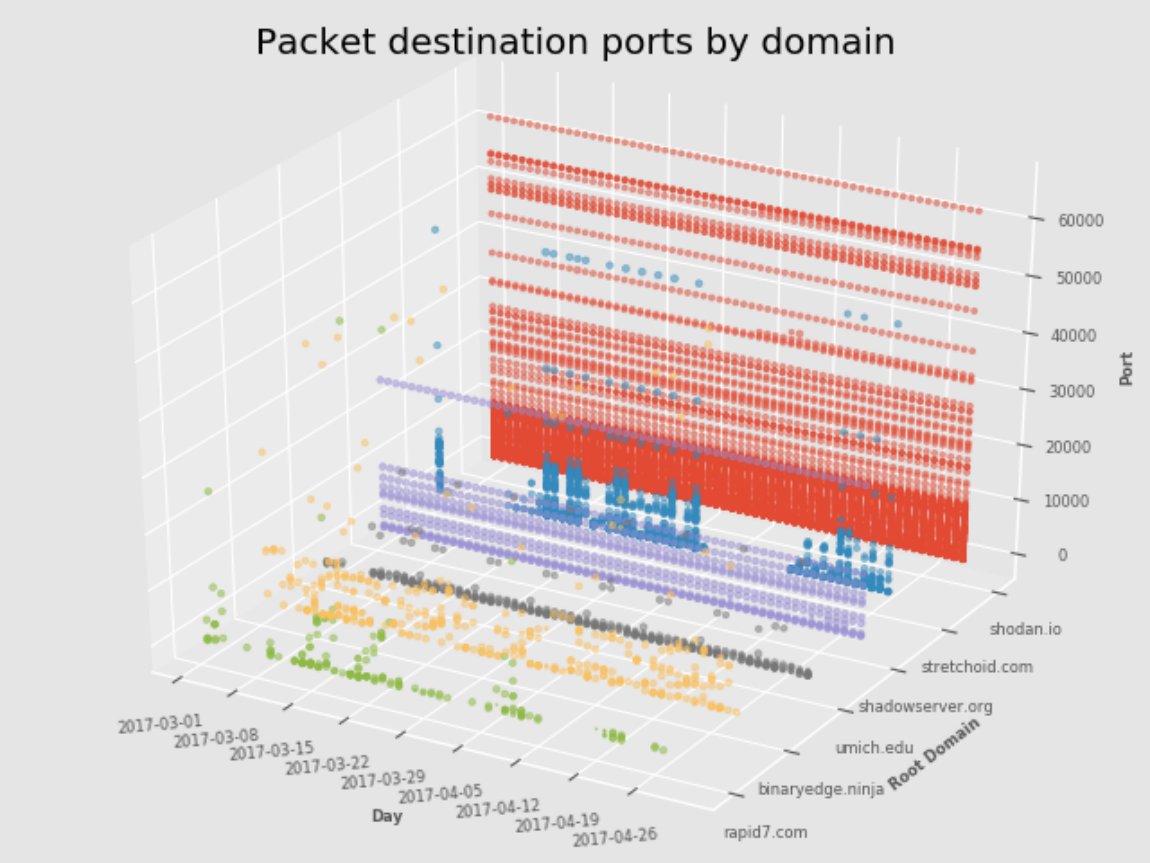


Figure 5.5: Daily traffic for the top 6 in the Research category, by port

or vulnerabilities available to those researchers. An example is scans of unusual destination ports (2323/tcp is a good example). Other examples include unusual header flags, destination ports or new scanning patterns.

5.6 Mobile-based Networks

Root domains classified as 'mobile' were selected due to their focus on providing Internet connectivity via a cellular network to mobile devices. When compared against other categories, Table 5.4 shows that the Mobile category domains produced substantially lower amounts of packets. Given that the GSMA[2] estimates 5.1 billion unique mobile subscribers (GSMA, 2019), it could be expected that these networks would generate an amount of packets that is similar or more than those generated by other categories.

While some modern smartphones have the processing power equivalent of a low-end desk-

[2]https://www.gsma.com/

Table 5.4: Packets from the top 10 known mobile networks

Rank	**Root Domain**	**Packet Count**	**% Total**
1	tpnet.pl	290 061	0.2
2	iol.cz	247 919	0.2
3	une.net.co	207 696	0.1
4	ucom.am	158 690	0.1
5	movistar.cl	130 088	0.1
6	virginm.net	115 511	0.1
7	orange.es	94 395	0.1
8	telekom.hu	68 934	0.0
9	ctm.net	55 299	0.0
10	wind.it	51 161	0.0
	Sub-total	**1 419 754**	**0.9**

top computer, and so could possibly participate in large-scale scanning activities, they are constrained in terms of power, storage and connectivity. Power management strategies in mobile devices are surveyed in Ahmad *et al.* (2015), highlighting the limits of battery technology. Additionally, the changing environment would influence the result during situations such as poor connectivity or low-power. This makes a mobile device an impractical platform for reliable long-term scanning activities.

Equally, the bulk of mobile malware may be more focused on financial gain using mechanisms such as Premium SMS fraud, banking fraud, or personal information capture (Jiang and Zhou, 2012).

And finally, given that mobile devices are able to use home broadband WiFi, coupled with the use of corporate WiFi (and associated Bring Your Own Device (BYOD) policies (Horton, 2013)), mobile device network usage may be mixed into the Residential and possibly Hosting/Service Provider categories.

5.6.1 Packet Distribution

These networks produce a fairly consistent amount of packets throughout the period under study, as can be seen in Figure 5.6). A notable exception is that of `iol.cz`, which rapidly increased on the 31st March, had an equally rapid decrease on the 8th of April, followed by another increase. These changes were approximately one order of magnitude in terms of packet count, which hints at a new type of malware, a DDoS attack or possibly one or more entities using `iol.cz` to perform scans.

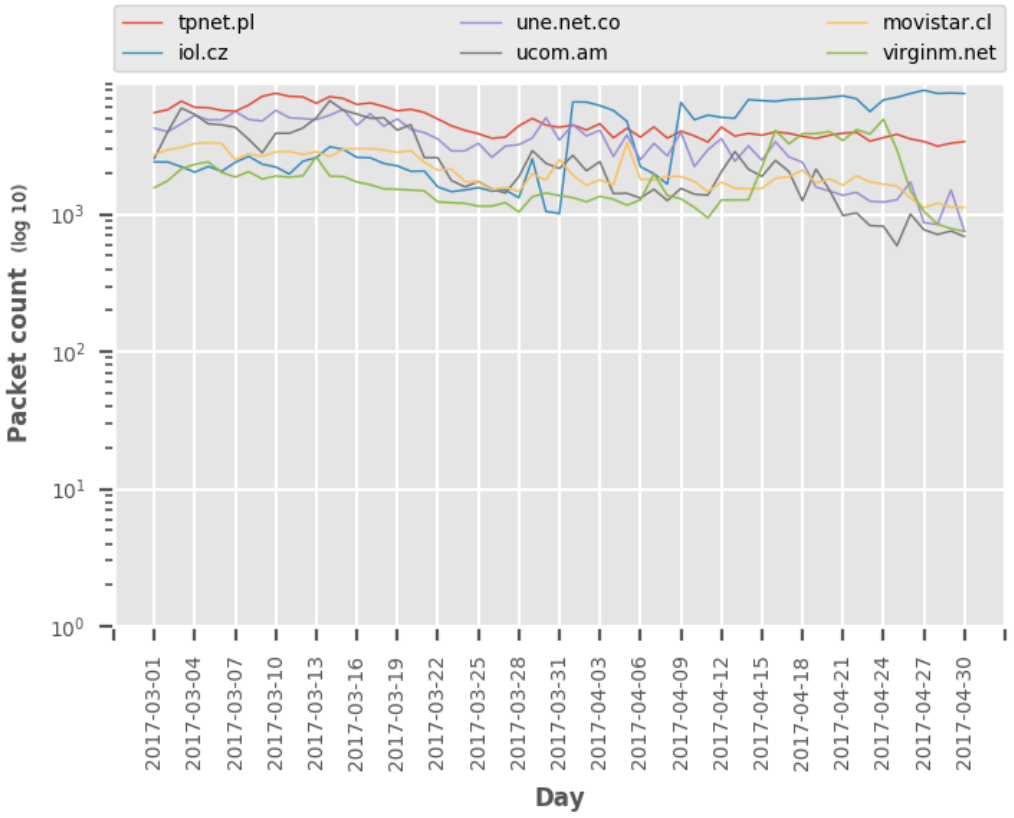


Figure 5.6: Daily traffic for the top 6 in the Mobile category

5.6.2 Destination Ports

An examination of the packets' destination ports shows a far more consistent set of ports, when compared to other categories as seen in Figure 5.7. If one assumes malicious intent, and some form of mobile-based malware, this could be attributed to resource constraints within the mobile-phone ecosystem. Different types of malware may impact mobile device resources such as the battery, processor, storage or mobile data in a variety of ways, as discussed in Zheng *et al.* (2018). In this respect Mobile users would be more sensitive to changes in device behaviour and connectivity cost, and as such it is likely that mobile malware-based scanners will attempt to minimise this impact by limiting packet generation rates, and use only common/plausible destination ports.

It should be noted that while cellular devices are usually mobile, some portion of this traffic could be generated by home or office networks using a residential broadband cellular gateway. Users behind these gateways would be vulnerable to the standard range of mobile and PC-based malware.

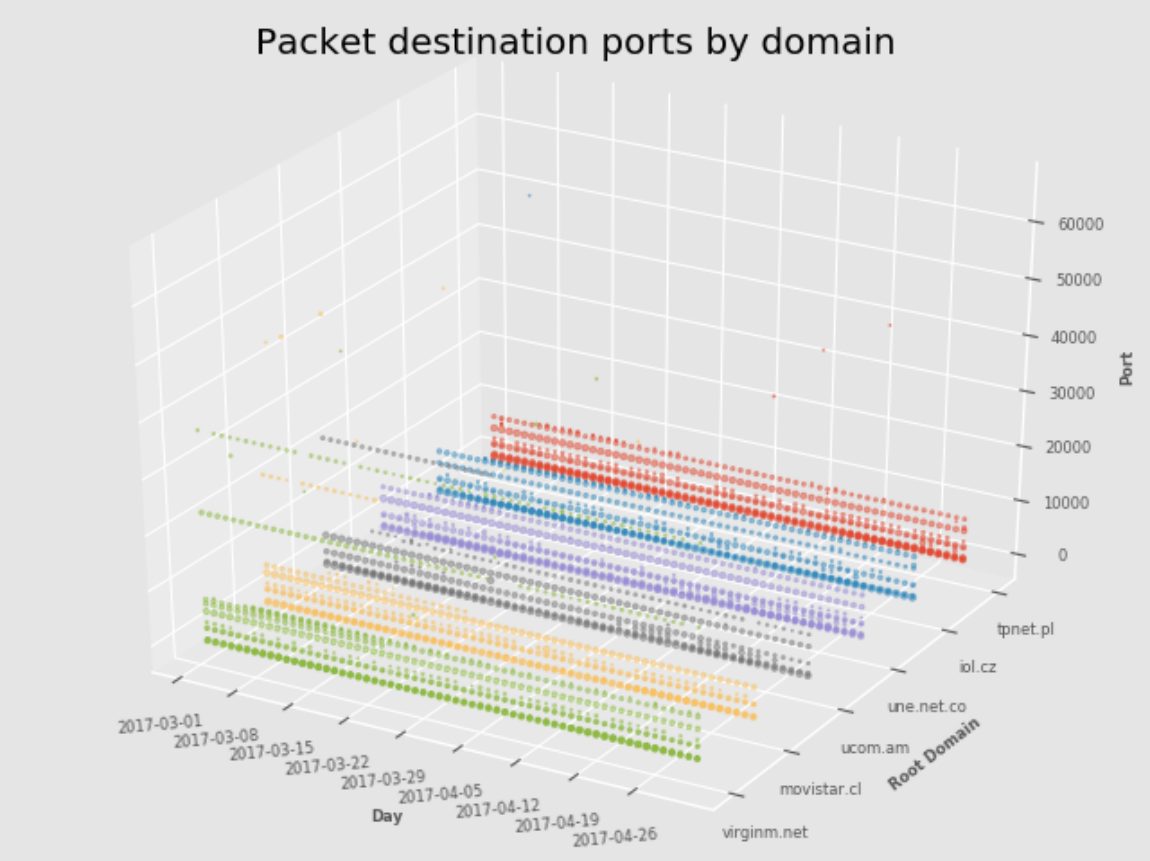


Figure 5.7: Daily traffic for the top 6 in the Mobile category, by port

5.7 Hosting or Service Provider

Packets originating from companies offering dedicated bare metal servers, Virtual Machines (VMs) or other Internet-based services were grouped into the Hosting category. Hosting services such as these provide service guarantees such as scalability, network availability, processing power, storage, DDoS protection, or even tolerance of criminal behaviour by so-called bulletproof hosting providers such as those described in (Alrwais *et al.*, 2017).

Clients of these providers are given access to an OS of their choice, and as such are able to run whichever service is required for their needs. These services may be compromised in some way and allow an attacker to use the host as a platform for further operations. In the case of bulletproof hosting, attackers may perform attacks directly from the host, or use it as part of reconnaissance, which a Telescope may see as a set of packets.

Table 5.5 shows that the top 3 hosting providers generate the bulk (74.1%) of the packets for this category, and that the top 10 account for 7.4% of all packets.

Table 5.5: Packets from hosting organisations

Rank	**Root Domain**	**Packet Count**	**% Total**
1	poneytelecom.eu	4 571 598	3.0
2	fastwebserver.de	2 684 037	1.7
3	linode.com	1 153 341	0.7
4	servdiscount-customer.com	757 517	0.5
5	ngprobd.com	547 565	0.4
6	vultr.com	516 913	0.3
7	scaleway.com	366 881	0.2
8	amazonaws.com	349 385	0.2
9	contabo.host	219 059	0.1
10	ip-192-99-55.net	180 833	0.1
	Sub-total	**11 347 129**	**7.4**

Amazon AWS[3] is a large hosting provider that, amongst others, provides Virtual Machine (VM)s, serverless and Content Distribution Network (CDN) services. These services can be abused in the same was as the others, but one would not expect a reputable hosting provider to tolerate abuse for a long period of time. Linode and Poneytelecom are examined further in Sections 5.7.3 and 5.7.4.

5.7.1 Packet Distribution

This category is significantly different from other categories in terms of the packet rate variance over the period. This is possibly due to the variety of services, OS and software present in this ecosystem, which is essentially where the majority of Internet services are hosted.

As discussed in (Alrwais *et al.*, 2017), bulletproof hosting providers abuse smaller, legitimate hosting providers by signing up for reseller accounts, and reselling services to those wishing to abuse them. This may account for some of the more interesting sets of behaviour discovered within the data.

For the majority of the root domains, daily incoming packet rates varied significantly: in some cases by up to several orders of magnitude, as shown in Figure 5.8.

The root domain `servdiscount-customer.com` shows several sets of traffic that are similar in volume and duration to those such as stretchoid and Rapid7 in the Research Institution category. It is possible this similarity is due to large-scale scanning behaviour, or possibly participation in some form of DDoS activities.

[3]https://aws.amazon.com/

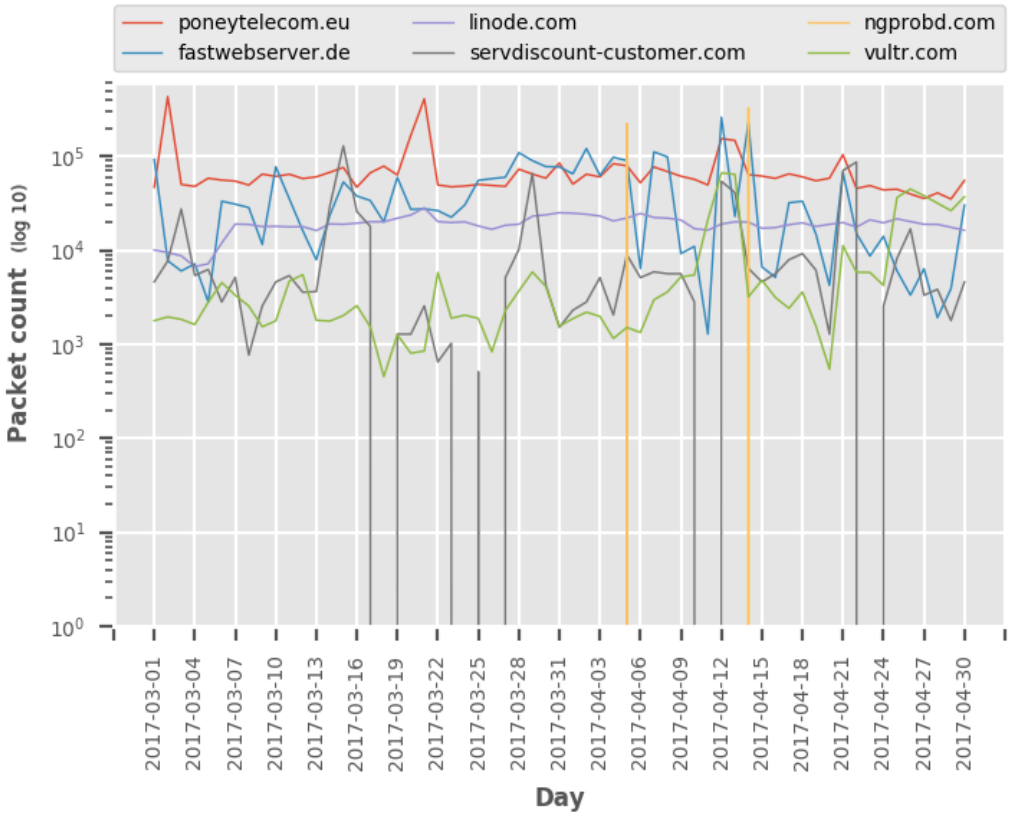


Figure 5.8: Daily traffic for the top 6 in the Hosting category

5.7.2 Destination Ports

The vertical lines in Figure 5.9 highlight the irregular scanning behaviour of three of the top six root domains in the hosting category (poneytelecom.eu, vultr.com and scaleway.com). These hosts appear to be performing full port-scans of the Telescope domain. This behaviour may be the result of reflected packets.

Reflected packets are those packets generated as a result of an attacker spoofing a victim's IP address as the source of an IP packet and sending to a server hosting a vulnerable service such as a recursive DNS server. This results in the vulnerable service sending a response to the *victim*, possibly causing some form of DOS.

A simple test for reflected packets has two criteria. The first is that such packets would be sent in response to an initial SYN packet addressed to a common TCP service port, and as such, the packet received by the Telescope will have that service port as its source port, and an ephemeral port as the destination. The second criteria is that the service port is below port 32768/tcp, and the ephemeral port is above. These criteria form the

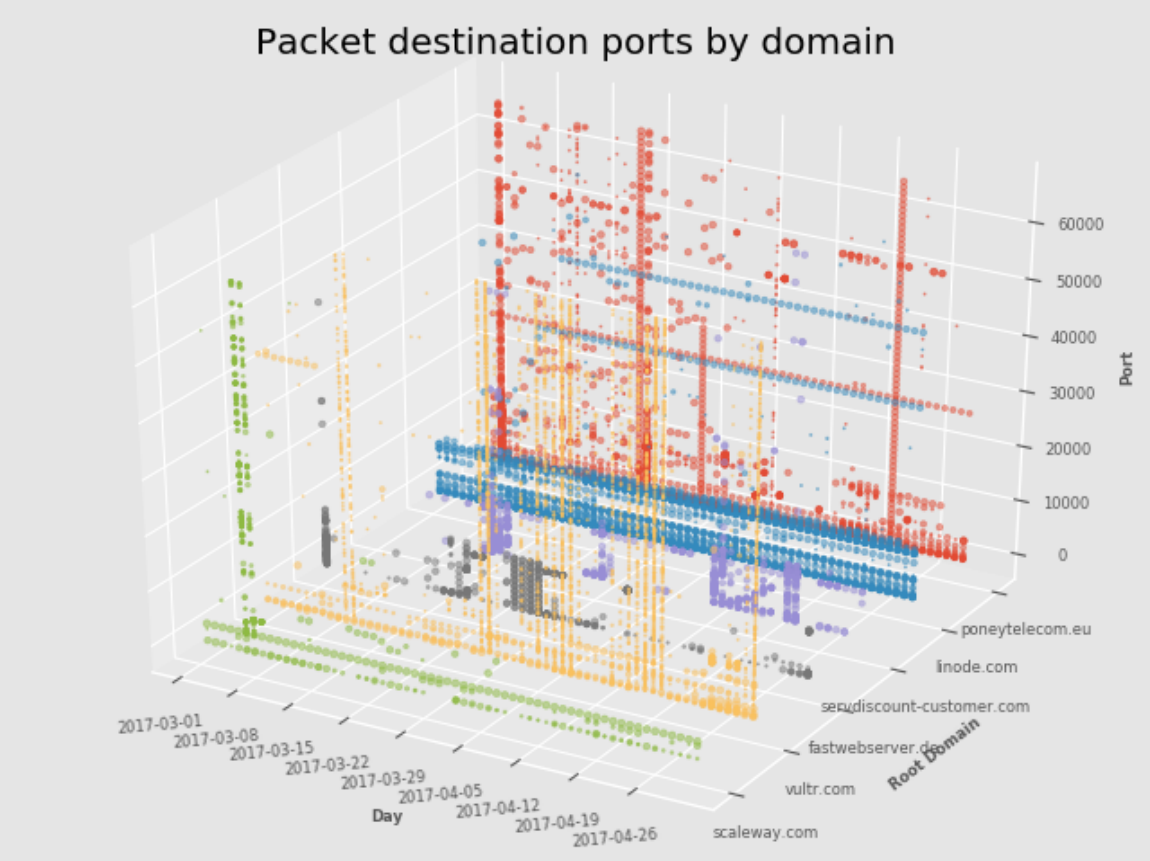


Figure 5.9: Daily traffic for the top 6 in the Hosting category, by port

naive ephemeral test for reflected packets.

In order to gain a better understanding of the differences in behaviour, two root domains were selected a more in-depth analysis of their associated packets. These domains were `linode.com` for its consistent behaviour, and `poneytelecom.eu` for its unusual behaviour.

5.7.3 linode.com

Linode is a reputable US-based VM provider, promising "High performance SSD Linux servers for all of your infrastructure needs". As such one can reasonably assume that packets originating from Linode are either reflected or originating from a VM hosted by Linode.

After performing the naive ephemeral test on packets with a `linode.com` DNS name, it was found that the reflected packet count is less than 0.1%. Due to this fact we treated the packets as if the source IP is correct, and the packet originated from with the Linode network.

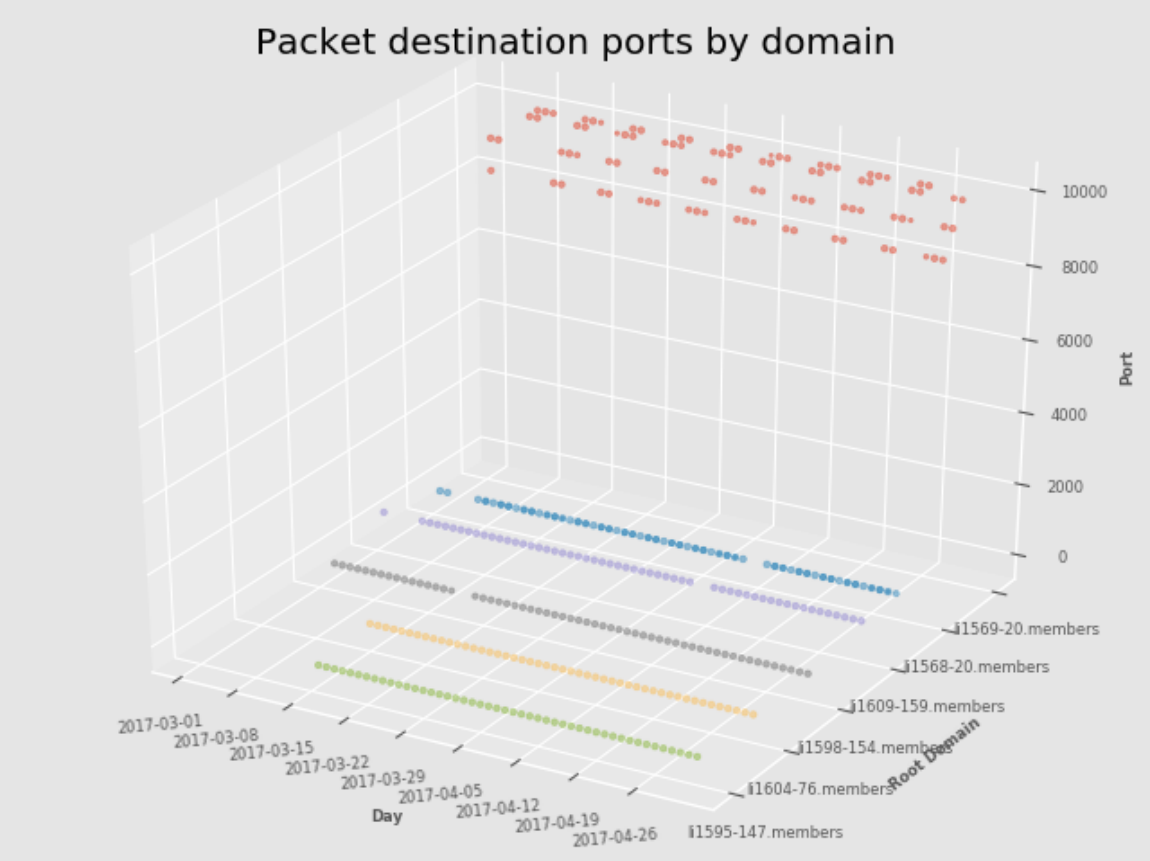


Figure 5.10: Daily traffic for linode.com, by port

Figure 5.10 shows the destination ports by day, for the top six linode domains. It shows regular scanning of a limited number of ports below 1024, which matches the initial assessment of consistent and regular scans.

The linode host `li1569-20.members` regularly, and sequentially scanned ports 8118/tcp, 9000/tcp, 9797/tcp and 9999/tcp. This shows as an interesting step function with traffic consistent with a system sequentially scanning a port on every Telescope IP address in order, then some time later scanning the next port, presumably scanning the rest of the Internet in the interim. This may be the result of inexpert use of a tool such as `nmap`, or `masscan`, or possibly a new tool written for personal use.

5.7.4 poneytelecom.eu

Poneytelecom.eu[4] is a French hosting company. If one searches for further information, what is found is a large list of third-party complaints regarding irresponsible hosting prac-

[4]http://www.poneytelecom.eu/

tices and lack of response to abuse notices, for example those discussed in (de Sacrobosco, 2016; Troy, 2017). It is not surprising to find the company at the top of the list of packet sources. The packet rate from this network, shown in Figure 5.8 has several peaks corresponding to what appear to be either comprehensive port scans or reflected packets shown on Figure 5.9.

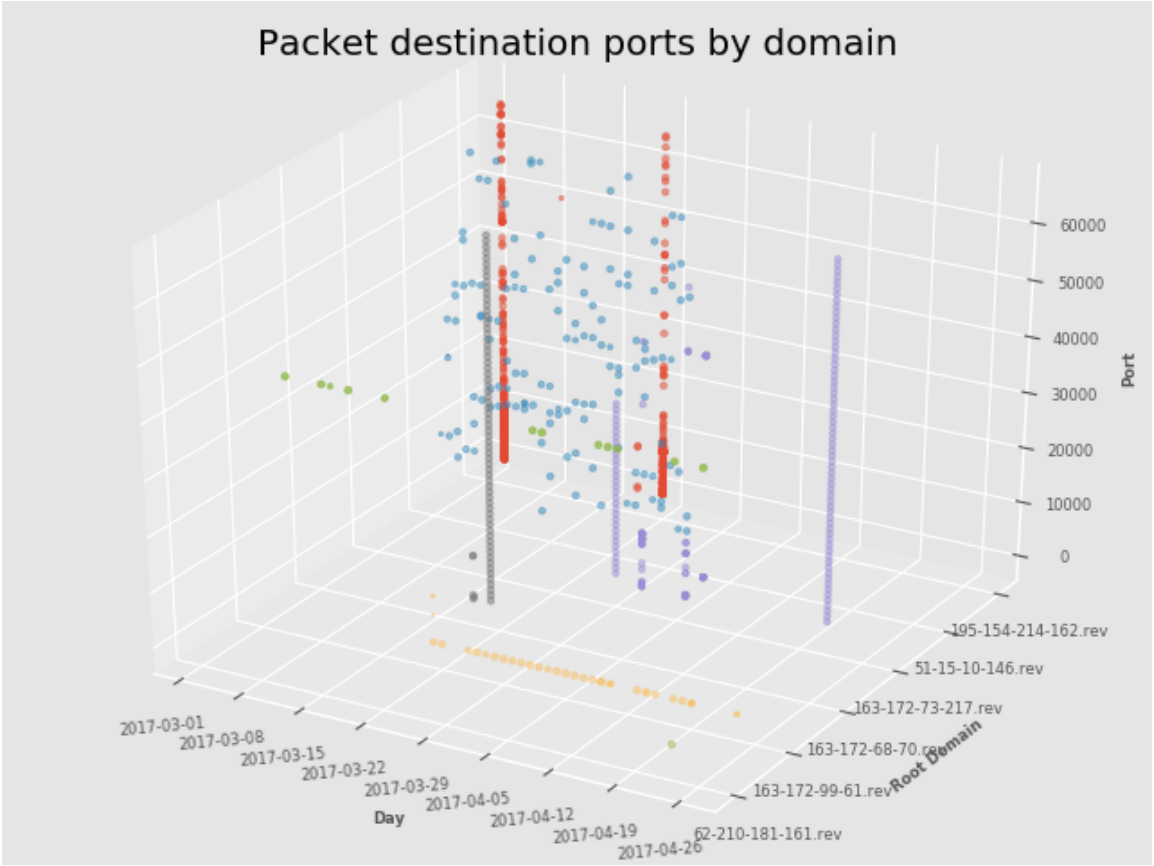


Figure 5.11: Daily traffic for poneytelecom.eu, by port

Figure 5.11 displays the top 6 originating hosts, and breaks down those by destination port. In this case it displays what may be comprehensive port scans, performed by single hosts. The host `195-154-214-162.rev.poneytelecom.eu` is responsible for three day-long scans, with an interesting characteristic whereby the source port is fixed for a scan over a sequential set of non-contiguous ports.

This hypothesis is strengthened by a period on the 21st March where over several iterations, a small number of destination ports are scanned before the source port changes and the scan is restarted. This may indicate some form of software or host failure in the process of being debugged, or where an intruder has been detected on a compromised host. Additionally, each of the 1280 Telescope destination IP addresses is

sent exactly one packet per port. Hosts `163-172-73-217.rev.poneytelecom.eu` and `163-172-68-70.rev.poneytelecom.eu` display similar characteristics, along with a period of destination port generation based on permutations of '443', prefixed or suffixed with random numbers.

These characteristics are unusual and are possibly the result of inexperienced users attempting to use large-scale scanning tools such as `zmap`[5] and `masscan`[6] to scan the Internet.

Examining these two root domains in detail allowed us to examine some of the unusual and otherwise interesting received packet trends. This approach is time consuming but yields a far better understanding of the intent and origin of packets received by the Telescope.

5.8 Residential

The Residential category selects for networks whose primary purpose is to provide broadband Internet access via mediums such as ADSL, VDSL or Fiber. Clients of these providers are typically home users with Internet connections ranging between 1Mbps ASDL and 1Gbps Fiber.

Table 5.6: Packets from a set of known residential networks

Rank	**Root Domain**	**Packet Count**	**% Total**
1	hinet.net	5 804 412	3.8
2	ttnet.com.tr	2 683 356	1.7
3	speedy.com.ar	2 561 841	1.7
4	vnpt.vn	1 454 584	0.9
5	rdsnet.ro	456 191	0.3
6	airtelbroadband.in	454 441	0.3
7	tedata.net	439 243	0.3
8	tpgi.com.au	404 132	0.3
9	superonline.net	367 007	0.2
10	comcast.net	352 698	0.2
	Sub-total	**14 977 905**	**9.7**

Unfortunately, routers for residential Internet connections are routinely compromised. In 2018 hundreds of thousands of different brands of home routers were compromised (US-CERT, 2018; Panda Security, 2018; Menn and Lynch, 2018). Malware such as Mirai takes advantage of easily compromised routers to form large self-propagating botnets (MalwareTech, 2016; Antonakakis *et al.*, 2017).

[5]https://zmap.io/
[6]https://github.com/robertdavidgraham/masscan

Having control of large numbers of Internet connections gives an attacker control of large amounts of readily available bandwidth, along with anonymity that comes with large numbers of unique IP addresses. This is especially useful in the case of DDoS attacks and large-scale scanning attempts.

Table 5.6 shows that the top three root domains account for just under 50% of the traffic in this category, and that the top 4 account for 8.1% of all packets.

5.8.1 Packet Distribution

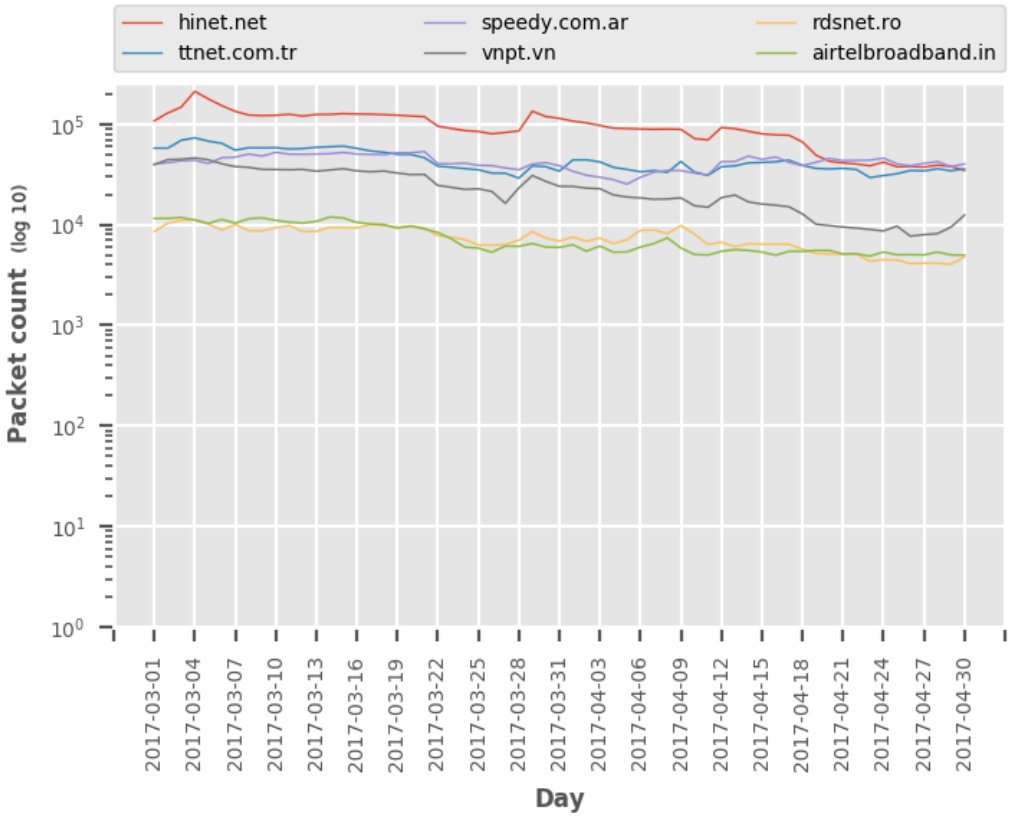


Figure 5.12: Daily traffic for the top root domains in the Residential category

Figure 5.12 shows traffic arriving at a fairly consistent rate. It would appear that, at least for the visible root domains, traffic is steadily decreasing over the period. This trend may be worth examining over a longer time period, which falls outside the scope of this research.

Visually it would appear that this decrease is part of a general correlation between traffic from `hinet.net`, `ttnet.com.tr` and `vnpt.vn`. These three providers are geographically

quite separate, with Hinet in Taiwan, TTNet in Turkey and vnpt in Vietnam. Thus is highly unlikely that there is a common path between the three providers and the Telescope that has no effect on any of the other root domains' traffic. A more likely explanation is some form of malware running on equipment within their networks. This malware could be performing scanning for new hosts to infect, or taking part in co-ordinated reflection DDoS attacks while spoofing the Telescope IP ranges.

In order to better understand the cause of the traffic, `hinet.net` is examined in detail in Section 5.8.3.

5.8.2 Destination Ports

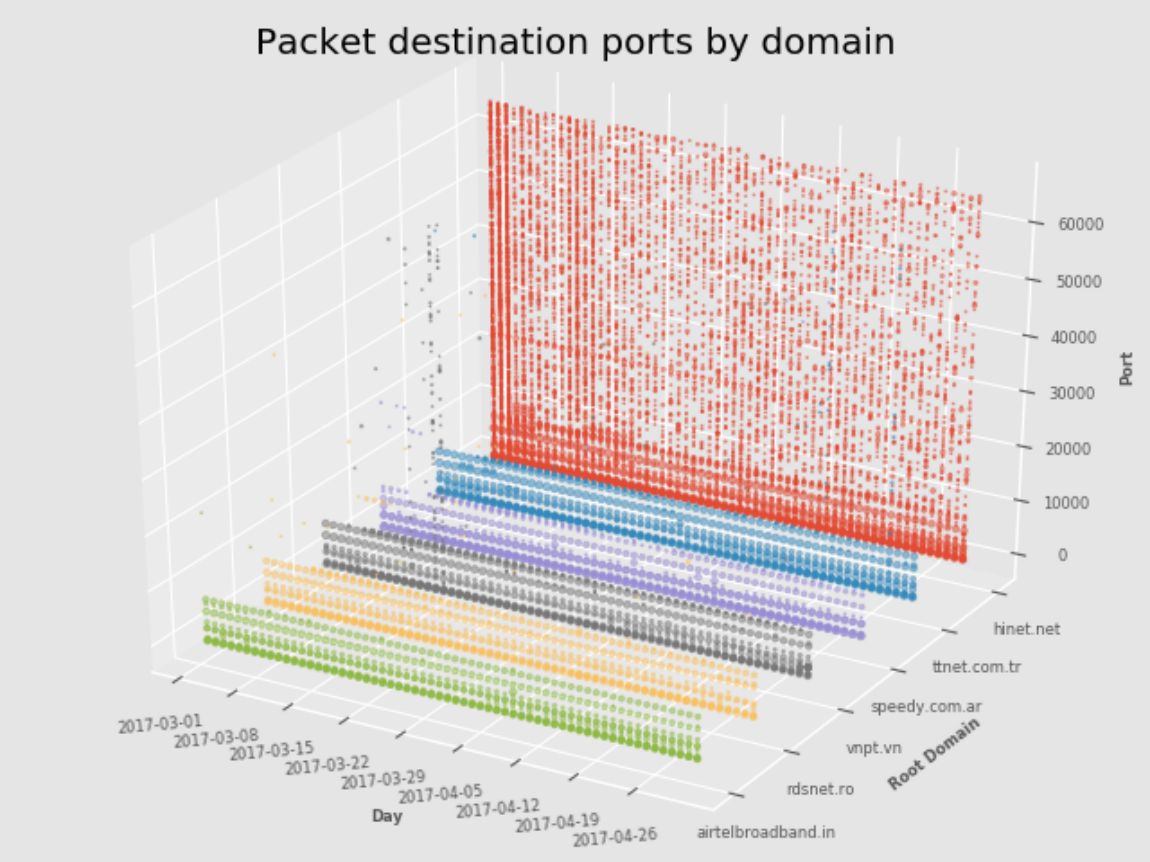


Figure 5.13: Daily traffic for the top 6 in the Residential category, by port

In Figure 5.13, the root domains `hinet.net` and `ttnet.com.tr` appear to be the source of a sustained and broad port scanning effort. `airtelbroadband.in` performs a large amount of scanning below port 10000/tcp.

In section 5.8.3 `hinet.net` is examined in greater detail.

5.8.3 hinet.net

HiNet is a Taiwanese ISP, primarily ADSL-based[7]. In Figure 5.13 it is shown to be performing a wide-ranging scan across the entire valid port range. When examined in closer detail, Figure 5.14 shows a very different story, with 4 046 840 (69.7%) of the packets destined for port 23/tcp, 983 946 (17%) for 5358/tcp and 325 793 (5.6%) for port 2323/tcp. These ports are commonly used by Mirai-style botnets when searching for further hosts to infect.

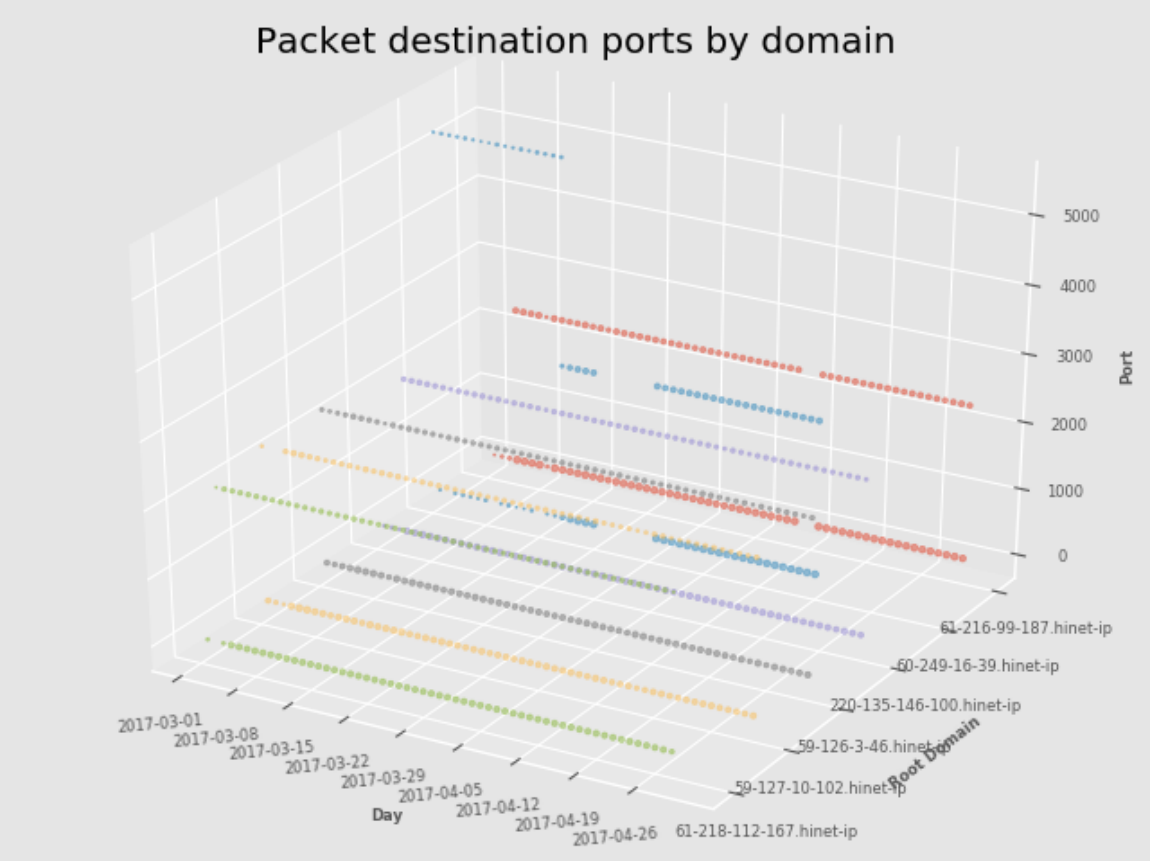


Figure 5.14: Daily traffic for hinet.net, by port

Assuming that this traffic is due to a botnet, the previously discussed traffic rate correlation between `hinet.net`, `ttnet.com.tr` and `vnpt.vn` could be the result of a single botnet owner managing the devices under the Malware's control.

The number of unique IP addresses sending packets to ports 23/tcp, 5358/tcp and 2323/tcp for these three networks was 187 736 during March and April 2017. Unfortunately, from a Telescope perspective, it difficult to estimate the number of Residential packet sources

[7]https://www.hinet.net/globe/en/about.html

given a number of unique IP addresses. This is due to IP address churn (Richter *et al.*, 2016). ISPs will often use some form of IP address pooling coupled with dynamic address allocation in order to reduce administrative load, and to optimally allocate addresses. These addresses have a limited lease which must be regularly renewed by the client. If the client fails to renew, the address is allocated to another client. This system causes what is known as Dynamic Host Configuration Protocol (DHCP) churn, whereby an IP address is not tightly linked to a particular client. Examples of attempts to track hosts across IP address changes can be found in Metwally and Paduano (2011); Moura *et al.* (2015). Because of this, the number of unique IPs must be interpreted as the upper limit of the number of infected systems in this potential botnet.

HiNet appears to have been experiencing some form of scanning/propagating malware. This area is worthy of future research.

5.9 Uncategorised

The remaining packets with root domains have a variety of associated reverse-DNS domain names. As the classification was a partially manual process, a portion of these domains were not classified due to factors such as foreign language, unresponsive or non-existent web-presence or it was not possible to understand which category a source should be included into.

Table 5.7: Packets from uncategorised domains

Rank	**Root Domain**	**Packet Count**	**% Total**
1	gvt.net.br	1 557 185	1.0
2	163data.com.cn	1 373 006	0.9
3	virtua.com.br	1 229 498	0.8
4	prod-infinitum.com.mx	818 472	0.5
5	telecomitalia.it	815 023	0.5
6	sl-reverse.com	516 031	0.3
7	server-hosting.expert	445 589	0.3
8	veloxzone.com.br	445 037	0.3
9	asianet.co.th	412 107	0.3
10	brasiltelecom.net.br	372 028	0.2
	Sub-total	**7 983 976**	**5.2**

The top domains by total number of packets are listed in Table 5.7. The top 10 entries together account for 36.7% of the packets in this category, which is quite different from the other categories, where the top few sources accounted for a far higher portion of the

total. Additionally, the packet rates shown are an order of magnitude lower than the other categories. This is possibly a result of bias in the categorisation process.

The total packet count for this category was 21 780 366 packets. Of this 10 703 119 (49.1%) packets were sent with a destination of 23/tcp, and 3 876 577 (17.8%) packets for ports 5358/tcp, 7547/tcp, and 2323/tcp. Packet sources in other categories sending packets to these ports were attributed to botnets such as Mirai.

5.9.1 Packet Distribution

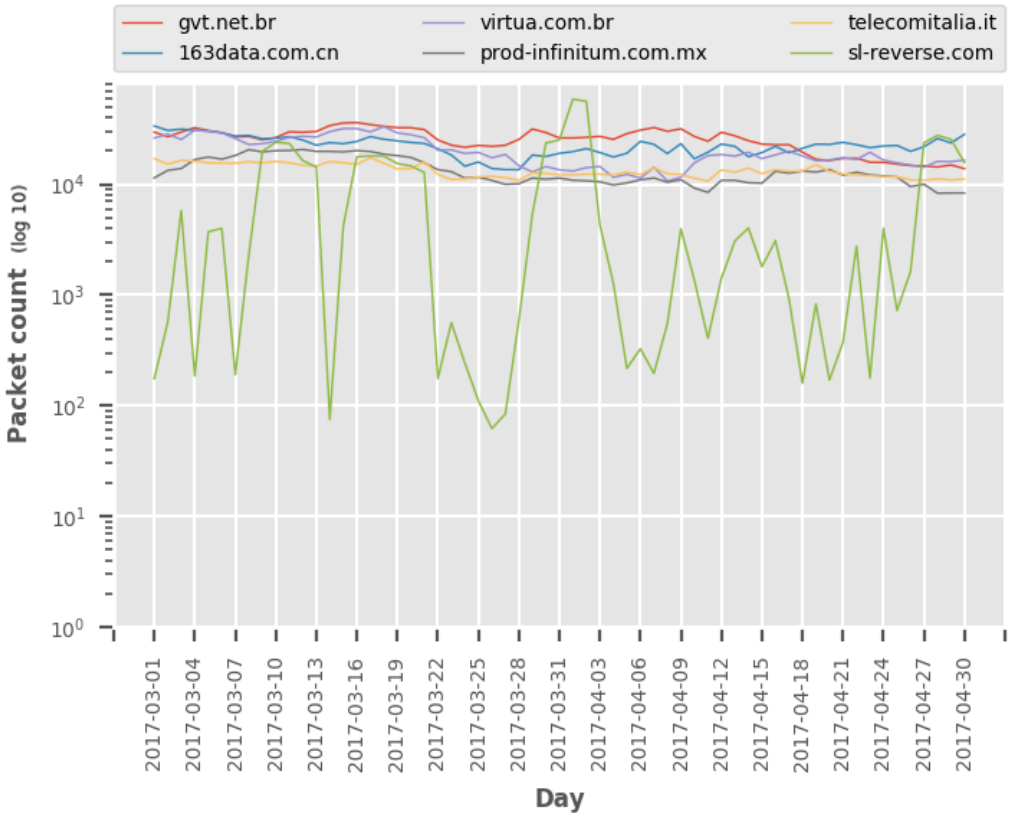


Figure 5.15: Daily traffic for the top 6 in the Unknown category

Figure 5.15 shows the overall traffic for the top root domains in this category. The packets were received at a constant rate with the exception of sl-reverse.com, which shows large changes in the daily packet rate. All of these domains sent packets to ports typically associated with a Mirai or similar botnet infection. This alone is not enough data to categorise these root domains, but does indicate the possibility that these packets originated from compromised home routers of the type usually infected by Mirai.

5.9.2 Summary

A large portion of the root domains in the Unclassified category exhibit behaviour (in terms of destination ports) that shows a high probability of being botnet propagation traffic caused by Malware such as Mirai. Further research could be performed in this area to uncover the categories of these domains. This was not performed in favour of time spent on other enrichment attributes such as BGP.

5.10 Unknown, Absent or Unreliable Enrichment Data

From Table 5.3 it can be seen that 47,6% of the incoming packets have no associated RDNS enrichment data. This includes scenarios where the timespan between the packet arrival and the enhancement data's generation time falls outside of an acceptable limit. This data cannot easily be examined using the methods discussed up to this point in this research.

The ASN mapping data is comprehensive, which allows us to group the packets discussed above in other ways. While this grouping does not strongly indicate the origin of the packets, it can assist with more general assertions as to the nature of the originator, as well as to understand the underlying reasons for the low DNS enrichment rate. This approach is discussed further in Chapter 7.

5.11 Summary

The primary goal of this Chapter was to explore methods of gaining a better understanding of the reason(s) for packets addressed to the Telescope with the intention of establishing a connection to a particular service. With this in mind, the RDNS enrichment approach was able to successfully catagorise a significant amount of the packets arriving at the Telescope. The success rate may be improved by further research into those root domains categorised in the Unknown category.

This form of categorisation enables targeted filtering of traffic, and the ability to focus on the underlying cause and source of traffic. In the active traffic category, the largest source of packets was the Residential category. This could be attributed to malware infections on the personal devices, but evidence exists for a large number of home routers infected

with Mirai or equivalent malware. A small number of hosting providers account for a significant portion of the categorised packets, which is cause for concern given that the packets are easily traced to the source. Better enforcement of complaints against hosting providers may curtail this form of activity. Research institions contribute a surprisingly large amount of traffic that may be misinterpreted as malicious.

The packets with no RDNS attributes will be addressed in Chapter 7, as part of efforts to use BGP to analyse the entire Telescope dataset. In the following Chapter 6, traffic categorised as Passive will be examined.

Chapter 6

Passive Traffic Categorisation

6.1 Overview

In the context of a Telescope, a naive assumption would be that every packet received was sent in order to create a connection. This assumption is unfortunately incorrect, as a substantial amount of packets are received with *invalid* state, or with destination ports that do not usually have services running.

Passive traffic is traffic for which the initiating packets were not sent to the Telescope IP addresses. Packets in this category are often the result of some form of DOS campaign, where the target is sent a large number of packets with spoofed source IP addresses. In this Chapter passive traffic received by the Telescope is examined and categorised.

In this Chapter the concept of *Reflected* packets is examined in Section 6.2, followed by packet distribution analysis in Section 6.3. In Section 6.4 an attempt is made to use the port-scan enrichment attribute to verify the hypothesis that a particular set of packets are reflected. Finally, Section 6.5 summarises the Chapter.

6.2 Reflected Packets

When an attacker performs a DOS attack, network packets are sent to the victim in an attempt to disable the targeted system or service. These are *resource exhaustion* attacks, where some part of the system is expected to fail when attempting to process the incoming

packets, as described in Sachdeva *et al.* (2010). Examples of resources that may fail are the receiving system's network stack, application stack or Internet connection. Modern DDoS attacks can consume enormous amounts of Internet bandwidth in an attempt to overwhelm the destination's Internet connection (Cloudflare, 2018). Application and Network stacks are vulnerable to attacks that consume memory, CPU or related resources. In other cases a flaw in system design allows a small number of received packets to disable relatively large systems. An historical example is the Slowloris attack where a web server could be disabled by consuming certain resources associated with open web server connections until the server was unable to accept further connections. This was performed with a very low number of packets (Aqil *et al.*, 2015).

These packets can be created with faked source addresses (*spoofed*) in order to circumvent some forms of DDoS-protection, or simply to avoid identification (Paxson, 2001). An effective attacker will use an IP address selection mechanism that is appropriate for the protocol under attack, and the particular resource that should be exhausted. For stateless protocols (typically UDP) it may be sufficient to randomise the source IP addresses. In the case of a stateful service or protocol, it may be more appropriate to select an unused set of IP addresses so as to avoid any stateful systems responding to the target system, and so reducing the effectiveness of the attack.

Because the source address has been spoofed, the recipient of these packets will respond to the source IP address specified by these packets, which in this case is an IP address within the Telescope's IP address range. This traffic is classified as *Reflected*, and is detected by selecting for root domains where *all* packets for that domain do not match the typical ephemeral source port and service-based destination port pattern (Team CYMRU, 2015), or where packets have unexpected TCP flags set (such as SYN + ACK).

Packets in this category arrive sporadically, as attacks on the targeted networks start and finish, or possibly as the attacker's source IP address randomisation algorithm walks through the global IP range.

Table 6.1 lists the packet count per root domain in the Reflected category. Google systems (`1e100.net` and `googleusercontent.com`) received the bulk of the traffic (65.2%).

6.3 Packet Distribution

Figure 6.1 shows the distribution of packets by *destination port* over time, per root domain. Included in this list are `1e100.net` and `googleusercontent.com`, both of which are part

Table 6.1: Reflected packets

Rank	Root Domain	Packet Count	% Total
1	1e100.net	1 786 883	1.2
2	dreamhost.com	759 059	0.5
3	googleusercontent.com	462 056	0.3
4	bluehost.com	222 961	0.1
5	secureserver.net	166 990	0.1
6	contabo.net	50 009	0.0
	Sub-total	**3 447 958**	**2.2**

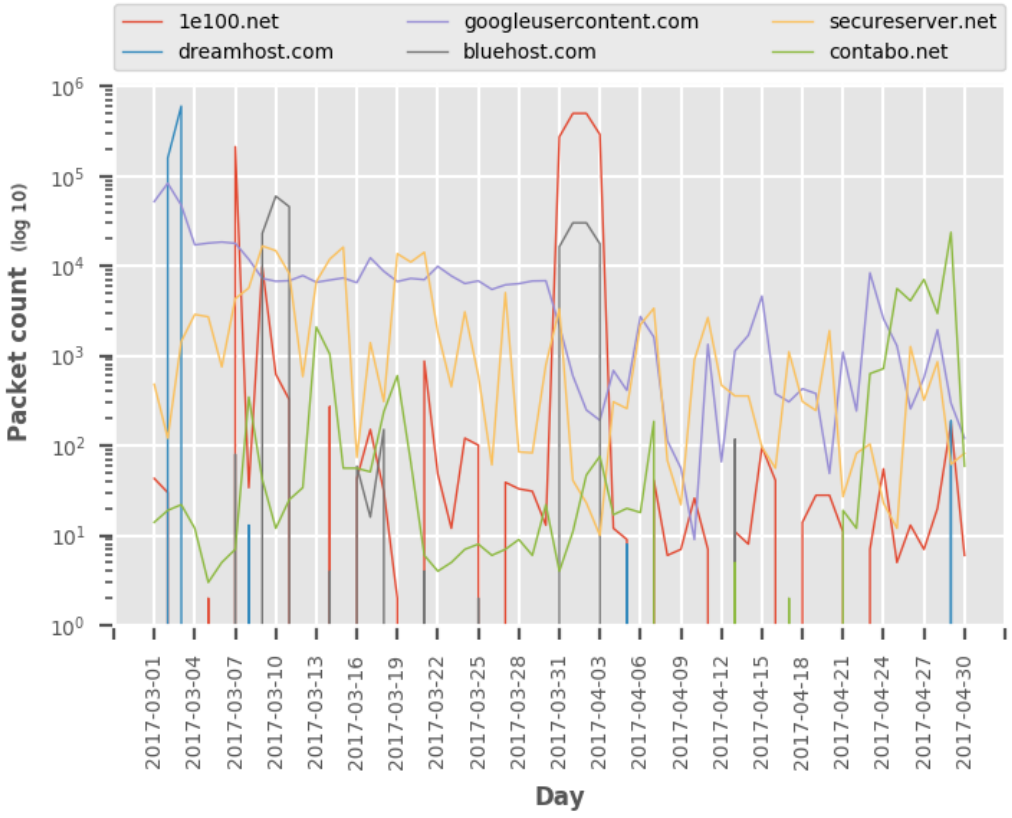


Figure 6.1: Daily traffic for the top 6 in the Reflected category

of Google's infrastructure, and as such it could reasonably be expected that the Telescope would not receive any packets from these domains. This further supports the hypothesis that packets in this category are the result of a reflection attack on Google's infrastructure, with the Telescope IP address used as the source IP.

The traffic seen in this category appears to arrive in relatively short-duration bursts. The traffic from `1e100.net` between the 31st March 2017 and the 3rd of April 2017 is a good example, with almost 1 million packets per day arriving at the Telescope. Given this data it is difficult to estimate the size of the actual attack, but as the packets were

received across all five of the Telescope IP ranges, either the attackers specifically used this particular IP range as source IP addresses, or a larger range was used and the Telescope received a fraction of the traffic.

6.3.1 TCP Ports

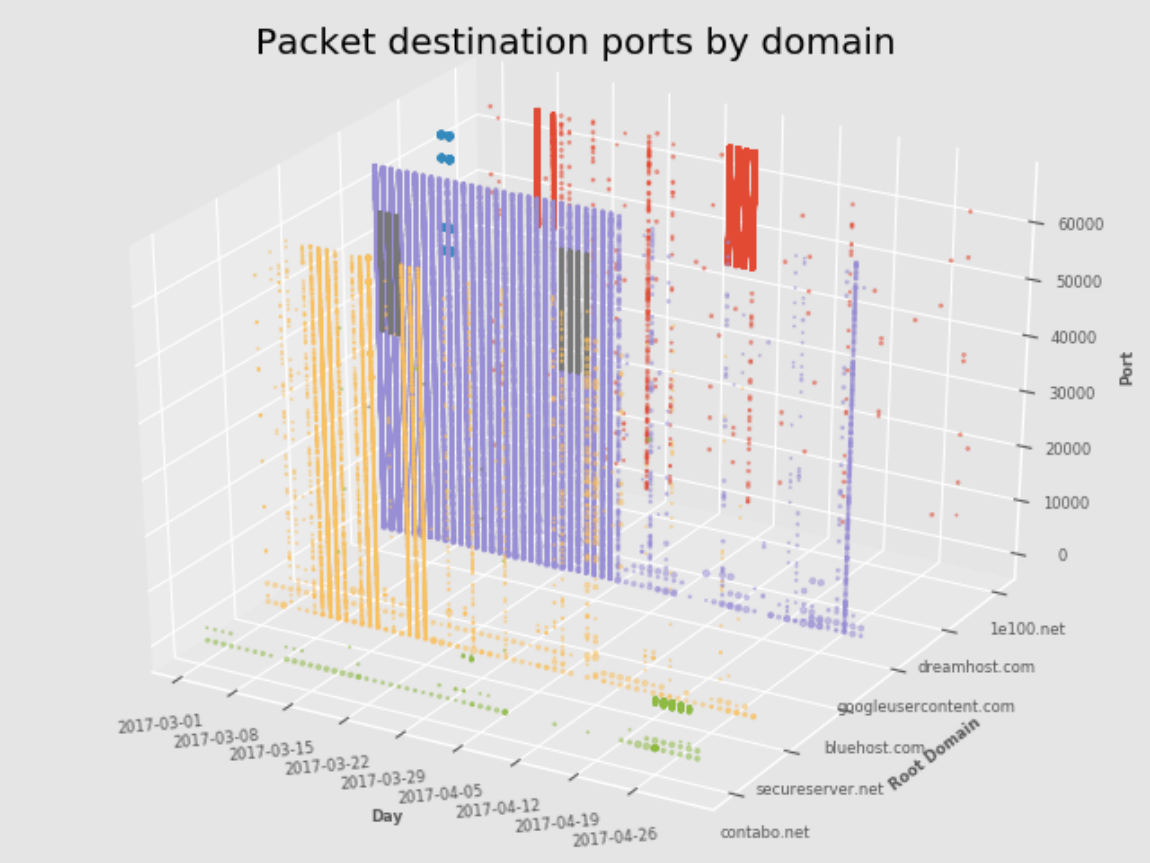


Figure 6.2: Daily traffic for the top 6 in the Reflected category, by destination port

The selection of destination ports can be seen in Figure 6.2, as the DDoS target replies to the faked Telescope source IP address and source port. When using a standard TCP or UDP stack, source ports are typically selected from the *ephemeral* range (Team CYMRU, 2015). Figure 6.2 shows the different source port selection strategies chosen by the attackers. While for `1e100.net` the ports appear constrained to above 42 000, both `dreamhost.com` and `bluehost.com` received traffic across the entire TCP port range. In the constrained cases, it is likely that the standard OS-based TCP stack of a compromised system was used. For the other cases, with ports chosen from across the port range, it is possible that a custom TCP stack was used for packet injection purposes.

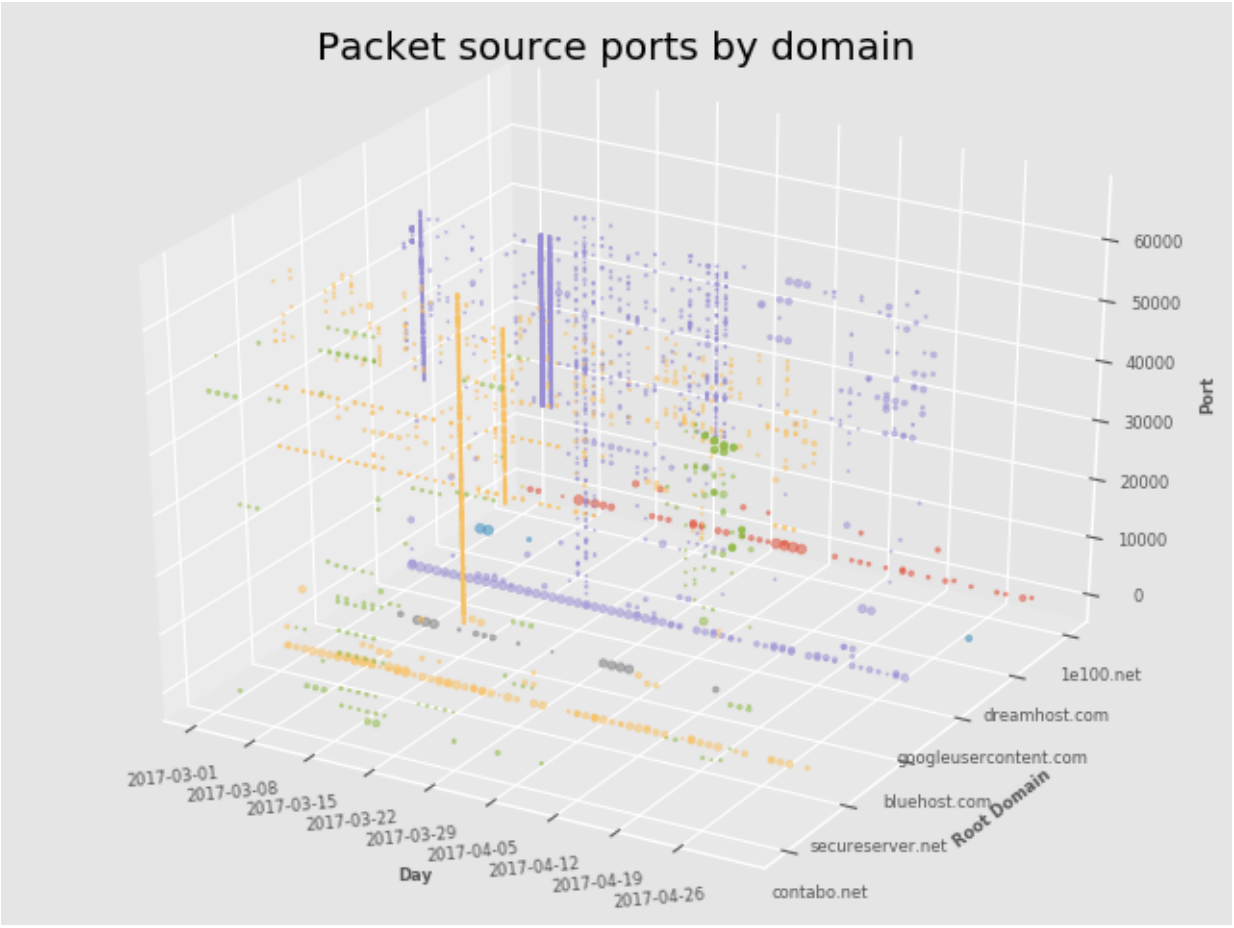


Figure 6.3: Classification by source port in the Reflected category

As discussed, in a Reflection attack, the source port received by the Telescope is that of the service under attack. In Figure 6.3, this principle can be seen by the limited number of source ports when compared to the list of destination ports. The root domain `1e100.net` is largely focused on port 80/tcp (HTTP), as are others, with exceptions such as `googleusercontent`, where for a short period between the 15th March 2017 and the 22nd March 2017 a selection of only higher ports is used.

6.3.2 1e100.net

This domain is used by Google as part of their infrastructure, and as such one would not expect this to be the source of unsolicited packets.

The majority of these packets are TCP packets, with a small portion having the TCP reset flag set. The remainder have the TCP syn/ack flags set, which typically occurs as a response to the initial TCP packet sent during connection establishment. Receiving these

packets on a Telescope IP address is either the result of an error on the originating system, or more likely due to an attack on the `1e100.net` IP systems, where the attacker's IP has been falsified to be an address (or addresses) within the Network Telescope's IP range.

6.4 Open Ports on the Source System

For the traffic categorised as reflected, the hypothesis is that a particular service on an IP address was attacked by sending large numbers of packets with a source IP address that was spoofed to be within the Telescope's IP range. This results in the recipient sending some form of response to the Telescope. In the case where the service is accepting connections, this response may be a packet with the SYN/ACK flags set. If the service has become unavailable due to the attack, or has deployed counter-measures, typically a packet with Reset Flag (RST) set will be returned.

This hypothesis can be tested by using the port scan enrichment attribute to check for an open port matching that of the source port in the packets received by the Telescope. Unfortunately, a negative result must be taken as equivalent to no available port scan data, as the port scan source data is limited both in terms of time span and the ports that were scanned. On the other hand, a positive match allows for an increase in confidence that the packets are indeed part of some form of reflection attack.

After port scan enrichment, every packet in the Reflected category will have a (possibly empty) list of open ports discovered by the port scan. The source port of each packet was checked against this list, and if found it was counted as a positive match. In total 35823051 packets match the criteria for reflected packets during March 2017. Of these, 1166431 (or 3.26%) packets have TCP source ports with matching port scans.

This approach is unfortunately not very successful when attempting to confirm the reflected categorisation of a small number of services available as port scan enrichment data. As such no further analysis was performed.

A potential improvement to this approach may be to include a comprehensive daily scan of ASNs actively sending packets to the Telescope.

6.5 Summary

In this Chapter, the concept of Reflected packets was explored, with packets successfully placed into this category and then analysed. The top 6 root domains account for the bulk of the passive traffic, and the pattern of source/destination port supports the hypothesis that these are due to reflected attacks. A longer study of this data could allow for analysis of long-term attack behaviour, and perhaps inform mitigation strategies.

An attempt was made to provide further evidence of the reflected nature of these packets by using port scan data to validate that the reflected packet's source port had an appropriate service running on it. Due to the narrow set of ports in the enrichment data, this attempt was inconclusive.

In Chapter 7, the BGP enrichment attribute is used to briefly re-examine the Telescope data, with the intention of using it to enhance or replace the RDNS approach.

Chapter 7

BGP ASN Enrichment

7.1 Overview

In this Chapter the BGP ASN enrichment attribute is explored and used for categorisation purposes. The BGP ASN enrichment attribute was present on close to 100% of the received packets. This presents an opportunity to categorise previously uncategorised packets. As described in Chapter 3, the BGP enrichment attribute links the received packet's source IP address with a routing *path* between the source IP address and the Telescope. The path is a list of BGP ASN entries, each of which represents a router's advertised ASN. This data was captured on the day of the packets' arrival.

In Section 7.2 the BGP enrichment attribute is analysed and its properties explained in terms of how it relates to the other enrichment attributes. The clear divide between ASNs with a high and low rate of RDNS enrichment is discussed. Section 7.3 discusses this divide in terms of previous enrichment analysis, and in Section 7.4 the top 3 ASNs below the median enrichment rate are analysed. Finally, Section 7.5 summarises the findings.

7.2 Analysis

For the purposes of this research, the intermediate routers along the path between the packet source and the Telescope are ignored, and only the final element is used. This element is the ASN of the packet source's first publicly visible router (or routers), which provide Internet connectivity for a particular organisation or group of organisations.

Packets were grouped by this ASN, after which it was possible to analyse these groups in terms of other packet attributes. The initial exploration examined the ratio between the packet count and DNS enrichment packet count per ASN, in order to validate the DNS enrichment approach. Once the ASN-grouped data is ordered by this ratio, a very clear pattern appears, whereby a large portion of the ASNs have either a high ratio of DNS enriched packets (greater than 80%), or a low number (less than 10%).

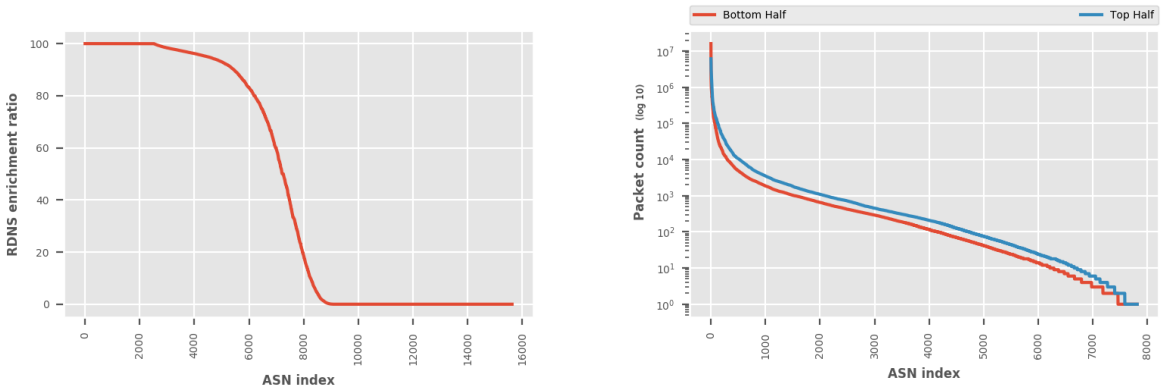
Table 7.1: Top 15 ASNs by packet count

ASN	Packet count	% Total	RDNS count	% Packet count	IP count	ASN Name
4 134	14 792 136	9.6	904	0.0	391 586	CHINANET-BACKBONE,CN
3 462	5 799 929	3.8	5 799 929	100.0	52 444	HINET,TW
12 876	4 571 598	3.0	4 571 598	100.0	358	AS12876,FR
4 837	4 130 273	2.7	22 216	0.5	152 980	CHINA UNICOM,CN
4 766	3 841 610	2.5	202	0.0	33 579	Korea Telecom,KR
24 961	2 684 037	1.7	2 684 037	100.0	103	MYLOC-AS,DE
9 121	2 683 356	1.7	2 683 356	100.0	93 680	TTNET,TR
22 927	2 561 294	1.7	2 561 294	100.0	303 470	Telefonica de Argentina,AR
6 939	2 304 662	1.5	2 304 662	100.0	225	Hurricane Electric,US
15 169	1 786 883	1.2	1 786 883	100.0	67	GOOGLE,US
12 880	1 705 431	1.1	85	0.0	488 017	DCI-AS,IR
7 552	1 662 360	1.1	520 142	31.3	90 694	Viettel Group,VN
14 061	1 612 661	1.0	1 612 661	100.0	660	DIGITALOCEAN,US
18 881	1 557 159	1.0	1 557 159	100.0	205 661	TELEFONICA BRASIL,BR
45 899	1 431 396	0.9	1 431 396	100.0	80 923	VNPT Corp,VN
Sub-total	**53 124 785**	**34.5**				

Table 7.1 shows the RDNS enrichment ratios for the top 15 ASNs. For the entries in the table, the RDNS enrichment rate is either 100%, or 0%, with two exceptions. This pattern is consistent throughout the data, which indicates that the organisations within this data tend to either correctly and completely configure RDNS or they provide no records for their IP addresses. The distribution of enrichment ratios can be seen in Figure 7.1a.

Given the unsolicited nature of the packets received by the Telescope, explanations for this sharp divide in enrichment rates need to account for the potential bias towards malicious, compromised, or victim hosts as packet sources. There is also the potential for bias in the RDNS data.

The average DNS enrichment success rate is close to 50%. Additionally, the enrichment ratios discussed above are evenly spread amongst the ASNs when ordered by packet count. Given this, and the sharp divide between high and low enrichment rates shown in Figure 7.1a, it was assumed that the median ASN would be a fair representative midpoint of the divide between the enriched ASNs and the unenriched ASNs, when ASNs were ordered by DNS enrichment ratio. Each ASN entry was placed into either a *top* bucket if they fell above the median value, or a *bottom* bucket if the fell below. The buckets were then re-ordered by packet count, and the resulting distribution graphed as per Figure 7.1b. Interestingly, both curves are roughly equivalent in terms of packet counts per ASN.



(a) ASN rank by RDNS enrichment ratio

(b) ASN rank by packet count, for Top/Bottom RDNS enrichment buckets

Figure 7.1: ASN rankings

The clear tendency towards high or low enrichment ratios per ASN lends weight to the results discussed in previous chapters, by suggesting that where enrichment data exists, it will tend towards more than 80% coverage of a particular ASN. While each ASN can only be assigned to a particular organisation, that organisation may possess multiple ASNs, and equally, the assigned organisation may in some way rent or assign IP blocks under that ASN to other organisations. This is the case with hosting providers, as well as with broadband providers. As such an ASN is not necessarily a perfect grouping of IP addresses, but with some manual checking, is sufficient for a limited exploration of the data.

7.3 Categorisation

From the above, we understand that the categorisation efforts in Chapter 5 and Chapter 6 are a reasonable representative of the current state for those ASNs that possess reverse DNS. Those packets without RDNS entries require a different approach, and do not overlap with the efforts in Chapter 5.

Every ASN has associated meta-data, similar to that provided by an Internet Domain Name registration. This data is stored by the relevant RIR when registering for an ASN, and is provided as a queryable interface by services such as those provided by Team Cymru (Team Cymru, 2019). A typical ASN whois lookup of this type is shown in Listing 17. Part of the response is an `AS Name` such as `TEAM-CYMRU - Team Cymru Inc., US`, which

can be used to identify a particular organisation or ISP. This name along with other data returned in the lookup can be used to discover useful information about the organisation responsible for the ASN.

```
\% whois -h whois.cymru.com " -v AS23028"
AS    | CC | Registry | Allocated  | Info      | AS Name
23028 | US | arin     | 2002-01-04 | -v AS4134 | TEAM-CYMRU - Team Cymru Inc., US
```

Listing 17: Team Cymru ASN Whois lookup

As shown in Section 7.4.1, the registered name (and the ASN) could refer to an ISP whose large size masks the source and intent of the packet. This is less of a problem with RDNS, as an ISP provides Internet services, including the provisioning of bandwidth and blocks of associated IP addresses, but will hand over the responsibility for DNS to the client of this ISP.

Figure 7.1a and Figure 7.1b highlight the uneven distribution of packets per ASN, where the top three ASNs account for almost 45% of the total traffic below the median. Because of this, the top three ASNs from Table 7.2 were selected and analysed.

7.4 ASN Analysis

The top three ASNs were selected from the top packet sources shown in Table 7.2. This table was generated by first ordering the list of ASNs by the RDNS enrichment ratio. All ASNs above the median were then discarded, and the remaining ASNs re-ordered by packet count. As discussed in Section 3.4.3, this approach selects ASNs with a low enrichment rate that were unlikely to have been previously categorised during the DNS enrichment activities.

Interestingly, half of the top 10 ASNs are registered to a Chinese organisation, accounting for 24 226 029 packets (15.8% of the Telescope packets). All of which have a relatively low RDNS enrichment rate, ranging between 0% and 15.9%. This may be the result of privacy concerns, or simply a difference in operational procedures within the country.

For each of these ASNs, basic analysis was performed, including destination ports and overall traffic rates. The purpose of this analysis is to provide insight into this method of categorisation, as compared to RDNS categorisation. Figure 7.2 shows the daily traffic for each ASN, where ASN 4134 shows large traffic spikes characteristic of either receiving

Table 7.2: Top 10 ASNs below the median, by packet count

Rank	ASN	Packet count	% Total	RDNS count	% Packet count	ASN Name
1	4 134	16 170 735	10.5	1 379 503	8.5	CHINANET-BACKBONE,CN
2	4 837	4 883 012	3.2	774 955	15.9	CHINA UNICOM,CN
3	4 766	3 843 610	2.5	2 202	0.1	Korea Telecom,KR
4	12 880	1 705 432	1.1	86	0.0	DCI-AS,IR
5	9 808	1 408 200	0.9	25 522	1.8	Guangdong Mobile,CN
6	18 403	1 241 866	0.8	80 319	6.5	Corporation for,VN
7	9 829	1 182 391	0.8	25 906	2.2	National Internet Backbone,IN
8	28 719	981 261	0.6	26 492	2.7	Khanty-Mansiysky department,RU
9	45 090	919 741	0.6	0	0.0	Tencent,CN
10	4 808	844 341	0.5	4 247	0.5	China Unicom,CN
	Sub-total	**33 180 589**	**21.6**			

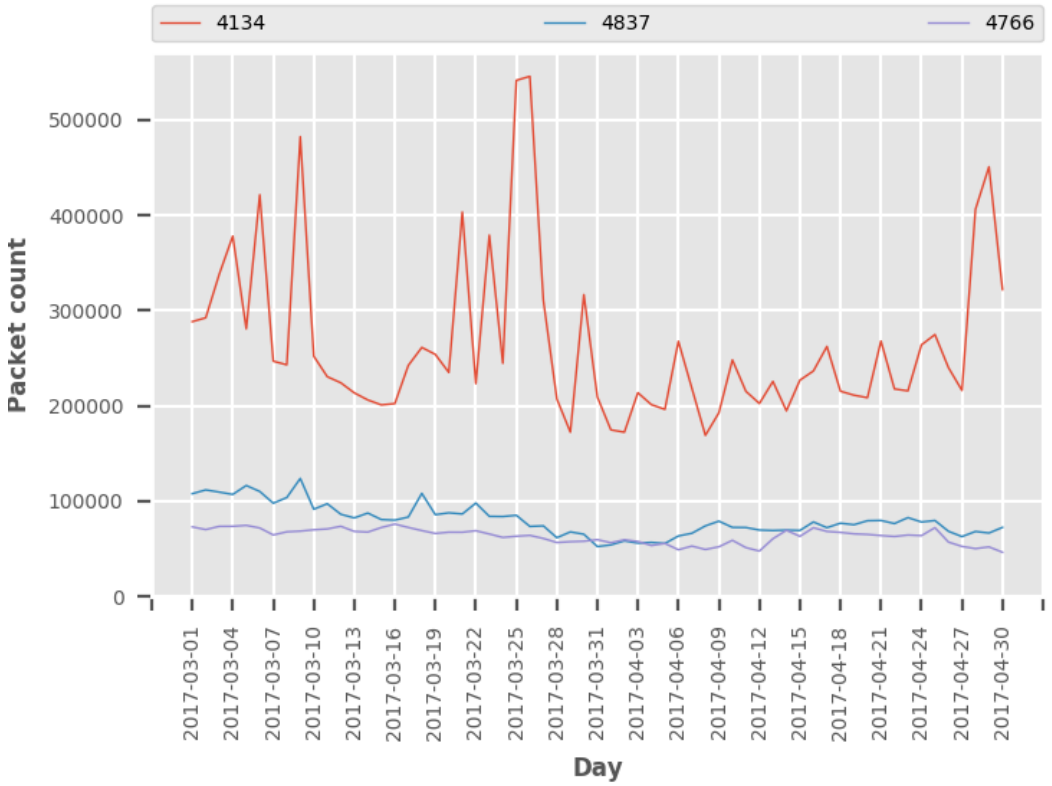


Figure 7.2: Daily traffic for ASN 4134, 4837 and 4766, by day

or transmitting DDoS attacks. ASNs 4837 and 4766 have substantially lower traffic rates, with a more consistent packet generation rate over this period.

7.4.1 ASN 4134

According to current information[1] [2], ASN 4134 is owned by China Telecom. China Telecom is an Network Service Provider (NSP) which provides large scale Internet services within China. It has been implicated in multiple BGP misconfiguration incidents in which traffic intended for US-based service providers was instead routed through the China Telecom network, as described in Hiran *et al.* (2013) and Demchak and Shavitt (2018).

ASN 4134 had 8.5% of the received packets with associated RDNS attributes, yet is the top source of packets reaching the Telescope, as seen in Table 7.1. The majority of the packets shown in Table 7.3 are those of popular Internet-based services such as 23/tcp (telnet), 22/tcp (ssh), or variations of these, such as 2323/tcp (an alternative for telnet). This scanning behaviour is typical of malware propagation techniques used by malware such as Mirai (Antonakakis *et al.*, 2017).

Rank	**TCP Port**	**Packets**	**% Total**
1	23	4 875 229	3.2
2	22	2 326 060	1.5
3	1 433	1 277 709	0.8
4	29 526	506 326	0.3
5	2 323	427 933	0.3
6	3 306	356 023	0.2
7	7 547	342 915	0.2
8	3 389	280 742	0.2
9	53 019	268 644	0.2
10	81	266 523	0.2
11	2 222	182 879	0.1
12	5 358	164 985	0.1
13	8 080	164 390	0.1
14	445	156 651	0.1
15	80	152 096	0.1
	Sub-total	**11 749 105**	**7.6**

(a) ASN 4134 top destination ports - all

Rank	**TCP Port**	**Packets**	**% Total**
1	29 526	506 326	0.3
2	53 019	268 644	0.2
3	10 758	149 603	0.1
4	63 733	148 542	0.1
5	38 976	123 416	0.1
6	18 146	109 071	0.1
7	42 528	104 430	0.1
8	65 057	67 810	0.0
9	37 459	59 557	0.0
10	10 871	49 431	0.0
11	10 762	40 295	0.0
12	63 582	33 482	0.0
13	65 215	31 886	0.0
14	10 931	31 885	0.0
15	63 723	30 691	0.0
	Sub-total	**1 755 069**	**1.1**

(b) ASN 4134 top destination ports - above 10000

Table 7.3: ASN 4134 top destination ports

Figure 7.3 shows the destination ports per day of packets received by the Telescope from IP addresses associated with ASN 4134. Activity is consistent throughout the period. The ports below 10000 are heavily scanned, while the ports above show comprehensive scanning throughout the period. Some of the visually interesting ports shown in Figure

[1] https://www.peeringdb.com/net/308
[2] https://bgp.he.net/AS4134

7.3 can be found in Table 7.3b. Examples are ports 18146/tcp, 38976/tcp and those above 63000/tcp, which all appear to receive packets consistently throughout the period.

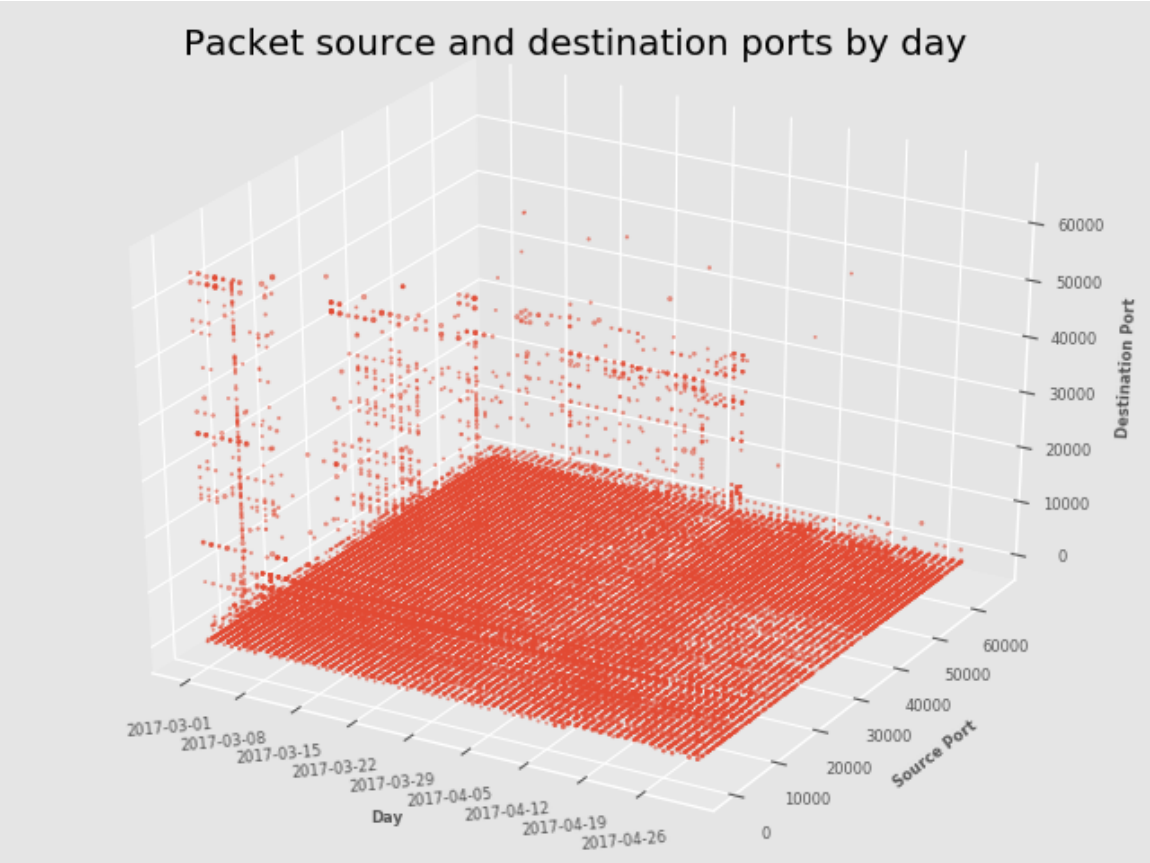


Figure 7.3: Source vs. destination ports for ASN 4134, by day

While it was not easily possible to categorise this packet source as anything apart from an ISP, some of the malware-based traffic indicates a good probability of home routers compromised by Mirai or similar malware, and as such this could be attributed to a home broadband category. Scanning on other service-based ports could indicate the possibility of other forms of malware hosted by compromised servers, but could equally be the result of automated scanning tools.

7.4.2 ASN 4837

ASN 4837 is currently owned by a company called China Unicom[3] [4], which is a Chinese Telecom that provides mostly Mobile services, as well as home-based Internet Services. In

[3]https://www.peeringdb.com/net/730
[4]https://bgp.he.net/AS4837

the context of the RDNS categorisation, this would place it in the Mobile-based networks category.

Rank	TCP Port	Packets	% Total
1	23	1 758 122	1.1
2	22	1 277 945	0.8
3	1 433	315 030	0.2
4	81	196 202	0.1
5	7 547	165 621	0.1
6	2 222	139 175	0.1
7	2 323	138 944	0.1
8	3 389	92 821	0.1
9	3 306	80 462	0.1
10	443	79 284	0.1
11	993	60 755	0.0
12	5 358	50 088	0.0
13	8 081	36 045	0.0
14	27 017	32 856	0.0
15	4 899	26 477	0.0
	Sub-total	**4 449 827**	**2.9**

(a) ASN 4837 top destination ports - all

Rank	TCP Port	Packets	% Total
1	27 017	32 856	0.0
2	23 231	3 084	0.0
3	37 383	2 898	0.0
4	63 507	2 738	0.0
5	35 300	1 765	0.0
6	34 055	1 719	0.0
7	32 777	1 672	0.0
8	17 997	1 615	0.0
9	17 939	1 301	0.0
10	37 569	1 250	0.0
11	10 050	1 236	0.0
12	10 051	1 192	0.0
13	37 518	1 097	0.0
14	56 274	1 061	0.0
15	22 022	777	0.0
	Sub-total	**56 261**	**0.0**

(b) ASN 4837 top destination ports - above 10000

Table 7.4: ASN 4837 top destination ports

As with Section 7.4.1, Table 7.4 shows the destination ports of packets associated with ASN 4837. This ASN generated a far lower number of packets, but shows a similar pattern to that of ASN 4134, with 23/tcp and 22/tcp dominating the packet generation table. The remainder of the packets lie largely below port 10000.

The traffic sent to port 23/tcp and 22/tcp is not typical of mobile malware, but is instead usually caused by either infected home routers, or compromised servers. Given that according to the ASN registration information, this ASN is largely Mobile-based, it could be the case that this traffic represents the low number of users using vulnerable Mobile routers at the home or office, or that this provider also provides broadband Internet services.

Figure 7.4 shows that most of the packets were again received for ports below 10000. The ports 27017/tcp and 23231/tcp are notable in that they are consistently scanned during a section of the period. Port 27017/tcp is the default port for MongoDB, and although the MongoDB security alerts page[5] does not mention a vulnerability at this time, it is possible these were attempts at abusing default credentials.

[5]https://www.mongodb.com/alerts

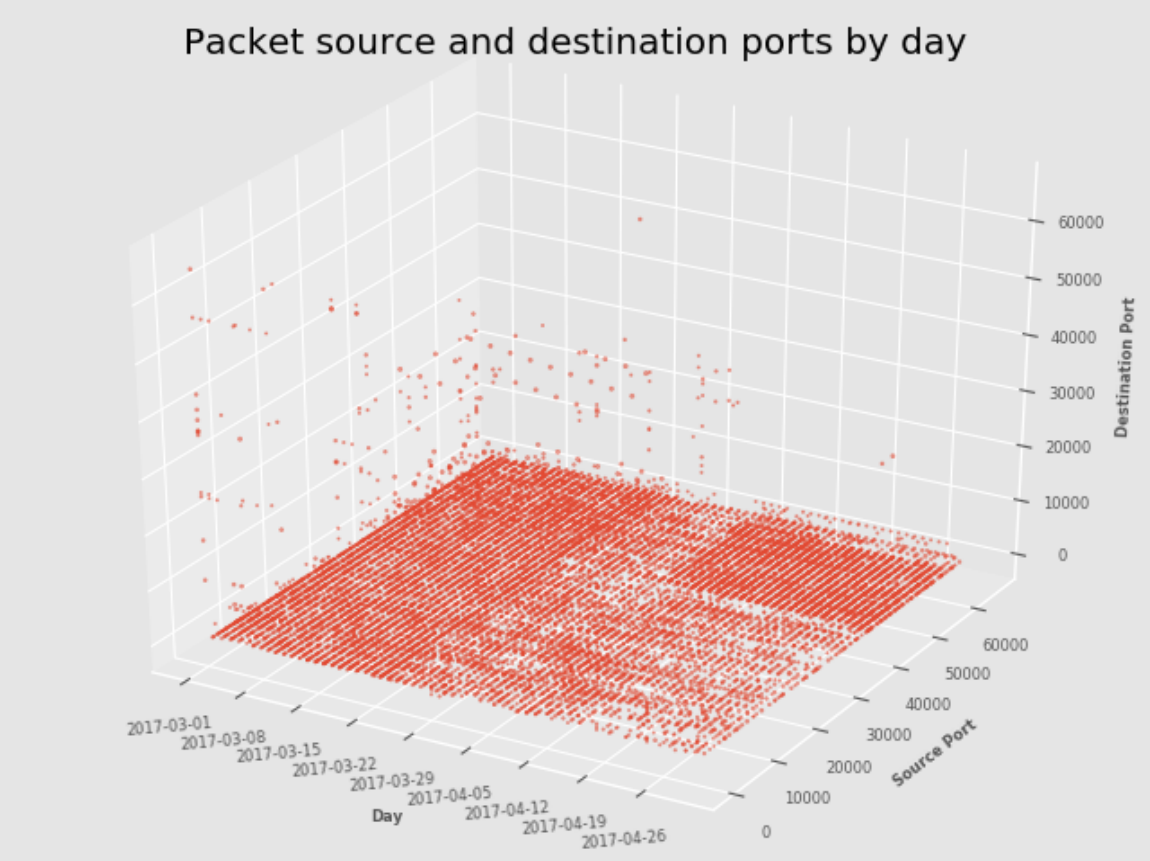


Figure 7.4: Source vs. destination ports for ASN 4837, by day

Given the available information, this ASN should be placed into the Mobile category.

7.4.3 ASN 4766

ASN 4766 is currently owned by Korea Telecom[6] [7], which is classified as a Residential ISP. As such it could be placed in the Residential category.

As seen in Table 7.5, the bulk of the packets received from this ASN were for port 23/tcp - another potential instance of Mirai or similar scanning for new routers to compromise. Port 7547 is also a port used by Mirai as part of its propagation approach (Lyndon, 2016).

The ports above 10000 represent effectively 0% of the received packets, but is interesting in that many of the port choices end in 389, which could be an attempt at scanning for high-port Lightweight Directory Access Protocol (LDAP) servers.

[6]https://www.peeringdb.com/net/23
[7]https://bgp.he.net/AS4766

Rank	TCP Port	Packets	% Total
1	23	2 144 377	1.4
2	5 358	569 848	0.4
3	7 547	327 155	0.2
4	2 323	84 375	0.1
5	22	77 304	0.1
6	81	33 639	0.0
7	1 433	18 335	0.0
8	3 389	13 626	0.0
9	8 080	6 974	0.0
10	10 913	6 576	0.0
11	80	6 418	0.0
12	2 222	6 343	0.0
13	21	4 204	0.0
14	1 099	3 840	0.0
15	8 081	2 845	0.0
	Sub-total	**3 305 859**	**2.1**

(a) ASN 4766 top destination ports - all

Rank	TCP Port	Packets	% Total
1	10 913	6 576	0.0
2	33 890	1 792	0.0
3	23 231	1 592	0.0
4	12 000	1 280	0.0
5	32 764	1 057	0.0
6	33 899	1 024	0.0
7	10 555	891	0.0
8	33 389	768	0.0
9	13 389	768	0.0
10	33 891	767	0.0
11	58 455	519	0.0
12	10 200	512	0.0
13	33 892	512	0.0
14	23 389	512	0.0
15	43 389	512	0.0
	Sub-total	**19 082**	**0.0**

(b) ASN 4766 top destination ports - above 10000

Table 7.5: ASN 4766 top destination ports

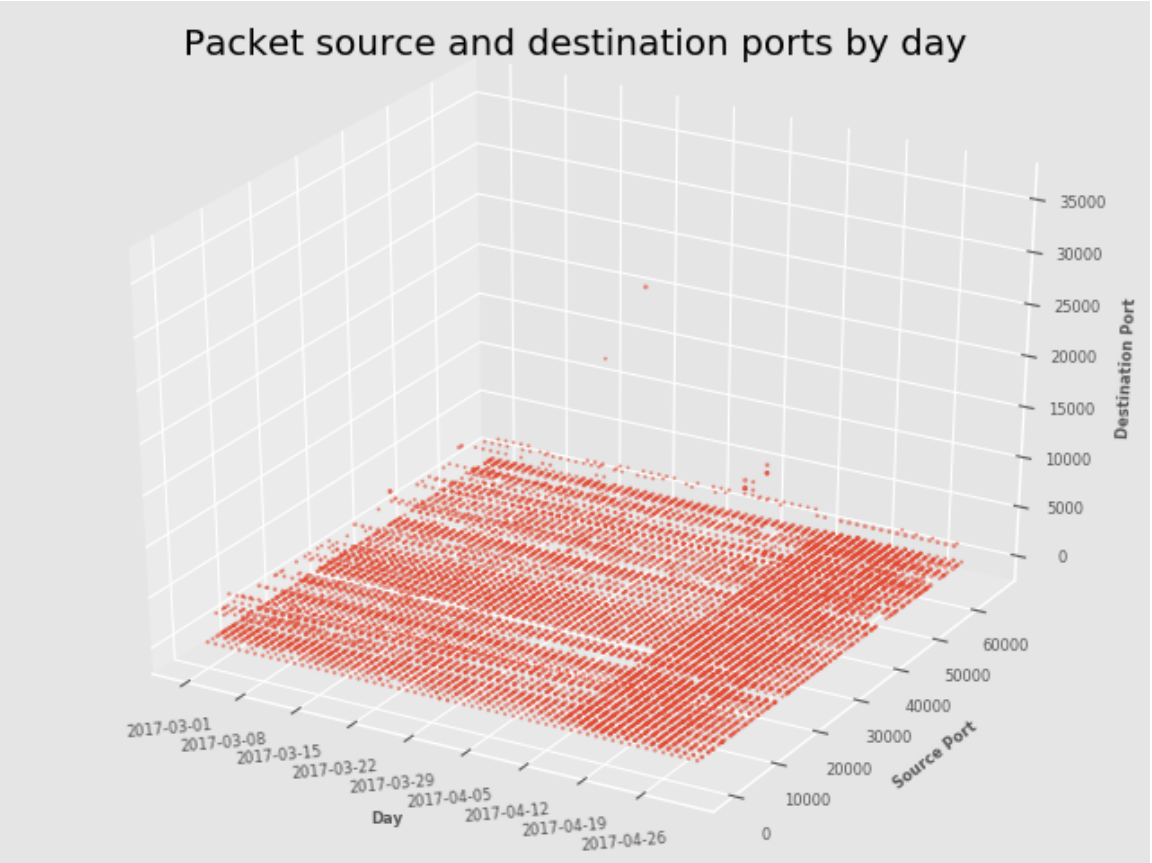


Figure 7.5: Source vs. destination ports for ASN 4766, by day

Despite receiving a relatively similar amount of packets as from ASN 4837, Figure 7.5 shows a far more focused set of destination ports, the majority of which are below 10000/tcp. Notably, on the 15th, 16th and to a small degree on the 17th of March it can be seen that a comprehensive scan of the Telescope was performed.

Both the available information, as well as the packet generation behaviour of this ASN match the Residential category.

7.5 Summary

The BGP enrichment attribute can successfully be used to further categorise packets where DNS enrichment attributes were not present. Categorisation involves an easily automated manual step (lookup of the ASN information) which includes useful information such as the organisation's name, broad categorisation and other information such as additional ASNs and BGP peers.

This method could result in less precise categorisation than with DNS. A client of an ISP represented by a single ASN may be able to set RDNS entries independently of the ISP, resulting in a potential sub-ASN categorisation. Another possibility is an organisation that is spread over multiple ISPs. In this case an organisation's packets could be present in multiple ASNs. It is also possible that route aggregation has taken place, and so from the Telescope router's point of view, multiple networks may appear to fall under a single ASN.

As long as the above is taken into account when analysing the Telescope data, this BGP ASN categorisation method is a very useful where no other categorisation is possible.

In the following Chapter 8, the research is discussed in terms of its goals, and how successful it was in meeting them.

Chapter 8

Conclusion

8.1 Overview

In this Chapter the research is retrospectively examined with respect to the initial goals, what was discovered, and what future work there may be. Section 8.2 briefly discusses the previous Chapters, Section 8.3 reiterates the stated objectives, and how or if they were met. The research is closed in Section 8.5 and future work is discussed in 8.4.

8.2 Recap

In Chapter 3, the enrichment process began. First the Telescope data and the enrichment data were cleaned and converted appropriate formats for merging. The initial approach using BigQuery was unsuccessful, and an alternative Python and PostgreSQL was used, which transformed all of the data into a set of enriched Telescope JSON files.

The files were loaded into BigQuery in Chapter 4. This was followed by some initial analysis and processing within BigQuery, giving a solid understanding of the data's shape and size, as well as testing the ability to successfully process the data with this system.

At this point the categorisation efforts began, by breaking the DNS data into root domains, and grouping packets by this domain. The top 500 root domains were manual categorised using screenshots of their websites along with other data such as Whois lookups, in an effort to understand what type of business they were, and what the primary clientele

may be. In Chapter 5, Passive traffic was removed, and root domains grouped into categories such as Research institutions, Hosting providers, Residential and Uncategorised. Each category was analysed in terms of the intent of packets received by the Telescope.

The Passive traffic packets excluded from Chapter 5 were analysed in Chapter 6. The concept of Reflected packets was introduced, and a domain (`1e100.net`) in the Reflected category was examined in detail, due to the extremely low likelihood of compromise. An attempt was made to validate the categorisation, by matching source ports to open port scans, but the results were inconclusive.

Finally, in Chapter 7, a separate categorisation effort was performed by using the RDNS attribute to find ASNs with low enrichment rates. It was discovered that ASN's tend to have either most of their IP addresses configured with RDNS entries, or very few. The ASNs were then ordered by RDNS enrichment ratios, and the domains above the median excluded. The remaining ASNs where then re-ordered by packet counts, and the top three were analysed.

8.3 Objectives

This research had three primary objectives: to enrich Telescope data using relevant third-party data-sets, to investigate the use of a Big Data platform such as BigQuery, in order to process and manipulate Telescope data, and finally to investigate methods for categorising Telescope data.

8.3.1 Enrich Network Telescope Data

Initially the Telescope and the enrichment data were both loaded into BigQuery, and attempts were made to join the data using BigQuery's SQL implementation. Unfortunately this approach could not be successfully implemented due to limitations in the implementation. At this stage a new approach using Python scripts was implemented. This approach was relatively slow, but eventually completed with a very high rate of success. After the process was complete, each Telescope file contained additional attributes mapping the source IP address to an enrichment data lookup result. This objective was fully met.

8.3.2 Use BigQuery To Manipulate Network Telescope Data

Once the enrichment process was complete, the results were loaded into a BigQuery database. Due to the nature of BigQuery, and the format of the enriched Telescope data (JSON), no further steps such as index creation or table partitioning were required in order to use the data. However, some cost optimisations were put in place, such as summary tables and aggressive local data caching. From this point onwards, only the data in BigQuery was used for exploration, analysis and graphing. This objective was fully met.

8.3.3 Methods For Categorising Network Telescope Data

Using RDNS, a successful categorisation method was developed and the top 500 of the packets were categorised. This represents 90% of the 52.4% of packets with a RDNS) attribute. Of these, 27% had the Unknown categorisation. With further effort, these root domains (and associated packets) could be completely classified. In short, while this approach appeared to have a relatively low success rate, from Section 7.2 (BGP Analysis), we know that where ASNs have RDNS entries, the rate tends to be above 80%. In other words, where this approach is successful, it will generally be very successful.

The BGP categorisation effort was technically quite successful, in that every packet has an associated ASN, and as such it can be categorised according to the information related to the ASN. It also has the advantage of readily available automated ASN meta-data lookups, from which the organisation and category can be retrieved. This may be a good classification method for those ASNs with little or no RDNS information. RDNS attributes are preferable due to likelihood of smaller organisations setting their own DNS entries, vs. ASNs which are usually aggressively aggregated.

Splitting Active and Passive packets, along with BGP data, allows for categorisation of organisations that have no RDNS data. The naive ephemeral port test was surprisingly effective at extracting reflected packets.

Unfortunately the utility of port-scan data was inconclusive due to the limited port ranges within the available enrichment data and the one to two week gaps between scans.

This objective was met in that a researcher using these techniques is able to easily and quickly isolate a particular organisation's packets and analyse them in detail. As a research

tool for understanding the source and intent of unsolicited packets, this approach is very useful. Further improvements to the approach, and the enrichment datasets will yield better results.

8.4 Future work

There are several potential areas of future work. The most promising would be to improve the categorisation rates by using additional data-sets such as whois lookups on the root domain names and IP addresses, as well as ASN routing data from other perspectives. More comprehensive port-scan data would enable the researcher to have greater certainty about a reflected packet, and better understand the nature of the packet source.

Given that for privacy and security reasons Netflow data is usually a more acceptable than full packet captures, it would be useful to replicate this research with a set of data from a Netflow-based sensor. This may then allow a researcher to gain access to concurrent Telescope data-sets from multiple organisations.

Live enrichment and automated categorisation of Telescope data would enable organisations to have better situational awareness. This approach would also benefit from higher efficiency, as enrichment data could be gathered directly from sources such as live DNS queries, ASN and routing state lookups.

This concept of live enrichment could be taken one step further, whereby a source IP that has sent more than a certain threshold of packets will have an immediate full port scan performed. This may have the side-effect of exposing the Telescope IP addresses to discovery by an alert botnet controller.

8.5 Closing

In conclusion, while the research did have some initial issues to resolve, the end results were quite successful. This is an area with scope for improvements, both in terms of the possible enrichment data-sets, and in the area of source categorisation.

References

Abley, J. and Lindqvist, K. Operation of Anycast Services. RFC 4786, December 2006. doi:10.17487/RFC4786.
URL `http://www.rfc-editor.org/rfc/rfc4786.txt`

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*. 2014.

Ahmad, R. W., Gani, A., Hamid, S. H. A., Xia, F., and Shiraz, M. A review on mobile application energy profiling: Taxonomy, state-of-the-art, and open research issues. *Journal of Network and Computer Applications*, 58:42–59, 2015.

Al-Musawi, B., Branch, P., and Armitage, G. BGP Anomaly Detection Techniques: A Survey. *IEEE Communications Surveys and Tutorials*, 19(1):377–396, 2017.

Alrwais, S., Liao, X., Mi, X., Wang, P., Wang, X., Qian, F., Beyah, R., and McCoy, D. Under the shadow of sunshine: Understanding and detecting bulletproof hosting on legitimate service provider networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 805–823. IEEE, 2017.

Amazon. Amazon EMR.
URL `https://aws.amazon.com/emr/`
Last Accessed: 4 October 2017

Amazon Alexa. The top 500 sites on the web. Top million is now a paid service.
URL `https://www.alexa.com/topsites`
Last Accessed: 12 January 2019

American National Standards Institute. ANSI/ISO/IEC 9075-2-2016: Information technology Database languages SQL Part 2: Foundation (SQL/Foundation). ANSI, 2016.
URL `http://www.ansi.org/`

ANSSI. ANSSI-FR/concerto.
URL `https://github.com/ANSSI-FR/concerto`
Last Accessed: 17 February 2018

ANSSI. ANSSI/parsifal.
URL `https://github.com/ANSSI-FR/parsifal`
Last Accessed: 17 February 2018

Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M. *et al.* Understanding the mirai botnet. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1093–1110. 2017.

Aqil, A., Atya, A. O., Jaeger, T., Krishnamurthy, S. V., Levitt, K., McDaniel, P. D., Rowe, J., and Swami, A. Detection of stealthy TCP-based DOS attacks. In *Military Communications Conference, MILCOM 2015-2015 IEEE*, pages 348–353. IEEE, 2015.

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.

Baker, F., Harrop, W., and Armitage, G. IPv4 and IPv6 Greynets. RFC 6018, September 2010. doi:10.17487/RFC6018.
URL `https://rfc-editor.org/rfc/rfc6018.txt`

Barr, D. Common DNS Operational and Configuration Errors. RFC 1912, February 1996. doi:10.17487/RFC1912.
URL `https://rfc-editor.org/rfc/rfc1912.txt`

Barrett, D. J., Silverman, R. E., and Byrnes, R. G. SSH, The Secure Shell: The Definitive Guide. O'Reilly Media, Inc., 2005.

Blunk, L., Labovitz, C., and Karir, M. Multi-Threaded Routing Toolkit (MRT) Routing Information Export Format. RFC 6396, October 2011. doi:10.17487/RFC6396.
URL `https://rfc-editor.org/rfc/rfc6396.txt`

CAIDA. DNS Research.
URL `https://www.caida.org/research/dns/`
Last Accessed: 12 January 2019

Chen, Z. and Ji, C. An information-theoretic view of network-aware malware attacks. *IEEE Transactions on Information Forensics and Security*, 4(3):530–541, 2009.

Cloudflare. What was the largest DDoS attack of all time?
URL `https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/`
Last Accessed: 18 January 2019

Codenomincon. The Heartbleed Bug.
URL `http://heartbleed.com/`
Last Accessed: 6 March 2018

Ćosović, M., Obradović, S., and Trajković, L. Performance evaluation of BGP anomaly classifiers. In *Digital Information, Networking, and Wireless Communications (DINWC), 2015 Third International Conference on*, pages 115–120. IEEE, 2015.

Cotton, M., Vegoda, L., Bonica, R., and Haberman, B. Special-Purpose IP Address Registries. RFC 6890, April 2013. doi:10.17487/RFC6890.
URL `https://rfc-editor.org/rfc/rfc6890.txt`

Czyz, J., Allman, M., Zhang, J., Iekel-Johnson, S., Osterweil, E., and Bailey, M. Measuring IPv6 Adoption. In *ACM SIGCOMM Computer Communication Review*, volume 44, pages 87–98. ACM, 2014.

Dabbagh, M., Ghandour, A. J., Fawaz, K., El Hajj, W., and Hajj, H. Slow port scanning detection. In *Information Assurance and Security (IAS), 2011 7th International Conference on*, pages 228–233. IEEE, 2011.

Dainotti, A., King, A., Claffy, K. C., Papale, F., and Pescapè, A. Analysis of a/0 stealth scan from a botnet. In *Proceedings of the 2012 ACM conference on Internet Measurement Conference*, pages 1–14. ACM, 2012.

de Sacrobosco, J. Poneytelecom.eu / Online SAS: Hosting Botnets, Scrappers, Spammers and Hacking Scripts.
URL `https://www.valueweb.gr/poneytelecom-eu-online-sas-hosting-botnets-scrappers-spammers-and-hacking-scripts/`
Last Accessed: 12 January 2019

De Vivo, M., Carrasco, E., Isern, G., and de Vivo, G. O. A review of port scanning techniques. *ACM SIGCOMM Computer Communication Review*, 29(2):41–48, 1999.

Demchak, C. C. and Shavitt, Y. Chinas Maxim–Leave No Access Point Unexploited: The Hidden Story of China Telecoms BGP Hijacking. *Military Cyber Affairs*, 3(1):7, 2018.

Dittrich, D., Karir, M., and Carpenter, K. An ethical examination of the Internet census 2012 dataset: a Menlo report case study. In *Proceedings of the IEEE 2014 International Symposium on Ethics in Engineering, Science, and Technology*, page 48. IEEE Press, 2014.

Dittrich, D., Kenneally, E. *et al.* The Menlo Report: Ethical principles guiding information and communication technology research. *US Department of Homeland Security*, 2012.

Durumeric, Z., Adrian, D., Mirian, A., Bailey, M., and Halderman, J., Alex. A search engine backed by Internet-wide scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 542–553. ACM, 2015.

Durumeric, Z., Kasten, J., Adrian, D., Halderman, J. A., Bailey, M., Li, F., Weaver, N., Amann, J., Beekman, J., Payer, M. *et al.* The matter of heartbleed. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 475–488. ACM, 2014. doi:10.1145/2663716.2663755.

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.

Eidnes, H., Runit, S., de Groot, G., and Vixie, P. Classless IN-ADDR.ARPA delegation. RFC 2317, March 1998. doi:10.17487/RFC2317.
URL `https://www.ietf.org/rfc/rfc2317.txt`

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

Fall, K. R. and Stevens, W. R. TCP/IP illustrated, volume 1: The Protocols. Addison-Wesley, 2011.

Fontugne, R., Borgnat, P., Abry, P., and Fukuda, K. MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking. In *ACM CoNEXT '10*. Philadelphia, PA, December 2010.

Franceschi-Bicchierai, L. How 1.5 million connected cameras were hijacked to make an unprecedented botnet.
URL `https://motherboard.vice.com/en_us/article/8q8dab/15-million-connected-cameras-ddos-botnet-brian-krebs`
Last Accessed: 16 September 2017

François, J., Lahmadi, A., Giannini, V., Cupif, D., Beck, F., and Wallrich, B. Optimizing Internet scanning for assessing industrial systems exposure. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International*, pages 516–522. IEEE, 2016.

Fyodor. The Art of Port Scanning. Phrack Magazine Volume 7, Issue 51 September 01, 1997.
URL `https://nmap.org/p51-11.html`
Last Accessed: 18 January 2019

Fyodor. Chapter 15. Nmap Reference Guide: Timing and Performance.
URL `https://nmap.org/book/man-performance.html`
Last Accessed: 18 January 2019

Gasser, O., Scheitle, Q., Gebhard, S., and Carle, G. Scanning the IPv6 internet: towards a comprehensive hitlist. *arXiv preprint arXiv:1607.05179*, 2016.

Genge, B. and Enăchescu, C. Non-intrusive historical assessment of internet-facing services in the Internet of Things. *International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics 2015*, 1(1):25–36, 2015.

Graham, R. D. MASSCAN: Mass IP port scanner.
URL `https://github.com/robertdavidgraham/masscan`
Last Accessed: 27 March 2019

Graham, R. D., Mcmillan, and Tentler, D. DEFCON 22 - Graham, Mcmillan, and Tentler - Mass Scanning the Internet: Tips, Tricks, Results.
URL `https://www.youtube.com/watch?v=nX9JXI4l3-E`
Last Accessed: 18 January 2019

GSMA. Definitive data and analysis for the mobile industry.
URL `https://www.gsmaintelligence.com/`
Last Accessed: 15 January 2019

Gu, G., Chen, Z., Porras, P., and Lee, W. Misleading and defeating importance-scanning malware propagation. In *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*, pages 250–259. IEEE, 2007.

Gulbrandsen, A., Vixie, P., and Esibov, L. A DNS RR for specifying the location of services (DNS SRV). RFC 2782, February 2000. doi:10.17487/RFC2782.
URL `https://rfc-editor.org/rfc/rfc2782.txt`

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

Hastings, M., Fried, J., and Heninger, N. Weak keys remain widespread in network devices. In *Proceedings of the 2016 ACM on Internet Measurement Conference*, pages 49–63. ACM, 2016.

Heisenberg. Heisenberg cowrie.
URL `https://scans.io/study/heisenberg.cowrie`
Last Accessed: 18 January 2019

Hiran, R., Carlsson, N., and Gill, P. Characterizing large-scale routing anomalies: A case study of the china telecom incident. In *International Conference on Passive and Active Network Measurement*, pages 229–238. Springer, 2013.

Holz, R., Braun, L., Kammenhuber, N., and Carle, G. The SSL landscape: a thorough analysis of the x. 509 PKI using active and passive measurements. In *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, pages 427–444. ACM, 2011.

Horton, C. What is BYOD (Bring Your Own Device)?
URL `https://www.tomshardware.com/reviews/byod-pros-and-cons,5812.html`
Last Accessed: 15 January 2019

Hubbard, D. Cisco Umbrella 1 Million.
URL `https://umbrella.cisco.com/blog/2016/12/14/cisco-umbrella-1-million/`
Last Accessed: 25 October 2018

IANA. IANA IPv4 Special-Purpose Address Registry.
URL `https://www.iana.org/assignments/iana-ipv4-special-registry/iana-`

ipv4-special-registry.xhtml
Last Accessed: 22 October 2018

IANA. Service name and transport protocol port number registry.
URL https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml
Last Accessed: 12 January 2019

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

Irwin, B. V. W. A framework for the application of network telescope sensors in a global IP network. Ph.D. thesis, Rhodes University, 2011.

Jiang, X. and Zhou, Y. Dissecting android malware: Characterization and evolution. In *2012 IEEE Symposium on Security and Privacy*, pages 95–109. IEEE, 2012.

Jicha, R., Patton, M. W., and Chen, H. Identifying devices across the IPv4 address space. In *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on In Intelligence and Security Informatics (ISI)*, pages 199–201. IEEE, 2016.

Jung, J., Sit, E., Balakrishnan, H., and Morris, R. DNS performance and the effectiveness of caching. *IEEE/ACM Transactions on networking*, 10(5):589–603, 2002.

Kadlec, T. The MongoDB hack and the importance of secure defaults.
URL https://snyk.io/blog/mongodb-hack-and-secure-defaults
Last Accessed: 24 October 2018

King, A., Huffaker, B., Dainotti, A. *et al.* A coordinated view of the temporal evolution of large-scale internet events. *Computing*, 96(1):53–65, 2014.

Krenc, T., Hohlfeld, O., and Feldmann, A. An internet census taken by an illegal botnet: a qualitative assessment of published measurements. *ACM SIGCOMM Computer Communication Review*, 44(3):103–111, 2014.

Laboratories, B. T., Kernighan, B., and McIlroy, M. UNIX programmer's manual. Bell Telephone Laboratories, Incorporated, 1979.

Larsen, M. and Gont, F. Recommendations for transport-protocol port randomization. RFC 6056, January 2011. doi:10.17487/RFC6056.
URL `https://rfc-editor.org/rfc/rfc6056.txt`

Lee, Y. and Spring, N. Identifying and Analyzing Broadband Internet Reverse DNS Names. In *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*, pages 35–40. ACM, 2017.

Levillain, O. Une étude de lécosystème TLS. Ph.D. thesis, Evry, Institut National des Télécommunications, 2016.

Lyndon, S. Mirai Evolving: New Attack Reveals Use of Port 7547.
URL `https://securityintelligence.com/mirai-evolving-new-attack-reveals-use-of-port-7547/`
Last Accessed: 14 January 2019

Maier, G., Feldmann, A., Paxson, V., Sommer, R., and Vallentin, M. An assessment of overt malicious activity manifest in residential networks. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 144–163. Springer, 2011.

MalwareTech. Mapping Mirai: A Botnet Case Study.
URL `https://www.malwaretech.com/2016/10/mapping-mirai-a-botnet-case-study.html`
Last Accessed: 16 September 2017

Markowsky, L. and Markowsky, G. Scanning for vulnerable devices in the Internet of Things. In *Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), 2015 IEEE 8th International Conference on In Intelligence and Security Informatics (ISI)*, volume 1, pages 463–467. IEEE, 2015.

Martin, O. Maglev: The Load Balancer behind Googles Infrastructure (Architectural Overview) Part 1/3.
URL `https://medium.com/martinomburajr/maglev-the-load-balancer-behind-googles-infrastructure-architectural-overview-part-1-3-3b9aab736f40`
Last Accessed: 12 January 2019

Menn, J. and Lynch, S. N. FBI warns Russians hacked hundreds of thousands of routers.
URL `https://www.reuters.com/article/us-usa-cyber-routers/fbi-warns-`

russians-hacked-hundreds-of-thousands-of-routers-idUSKCN1IQ2DY
Last Accessed: 16 January 2019

Metwally, A. and Paduano, M. Estimating the number of users behind IP addresses for combating abusive traffic. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge Discovery and Data Mining*, pages 249–257. ACM, 2011.

Microsoft. Understanding TCP/IP addressing and subnetting basics.
URL https://support.microsoft.com/en-us/help/164015/understanding-tcp-ip-addressing-and-subnetting-basics
Last Accessed: 16 October 2018

Mirian, A., Ma, Z., Adrian, D., Tischer, M., Chuenchujit, T., Yardley, T., Berthier, R., Mason, J., Durumeric, Z., Halderman, J. A. *et al.* An internet-wide view of ICS devices. In *Privacy, Security and Trust (PST), 2016 14th Annual Conference*, pages 96–103. IEEE, 2016.

Mogul, J. Internet subnets. RFC 0917, October 1984. doi:10.17487/RFC0917.
URL https://rfc-editor.org/rfc/rfc0917.txt

Mogul, J. and Postel, J. Internet Standard Subnetting Procedure. RFC 0950, August 1985. doi:10.17487/RFC0950.
URL https://rfc-editor.org/rfc/rfc0950.txt

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.

Moore, D., Shannon, C., Voelker, G., Savage, S. *et al.* Network telescopes: Technical report. Technical report, Cooperative Association for Internet Data Analysis (CAIDA), 2004.

Moura, G. C., Ganán, C., Lone, Q., Poursaied, P., Asghari, H., and van Eeten, M. How dynamic is the ISPs address space? towards Internet-wide DHCP churn estimation. In *IFIP Networking Conference (IFIP Networking), 2015*, pages 1–9. IEEE, 2015.

Munroe, R. Map of the Internet (Dec 2006).
URL https://xkcd.com/195/
Last Accessed: 17 February 2018

Oberheide, J., Karir, M., and Mao, Z. M. Characterizing dark DNS behavior. In *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pages 140–156. Springer, 2007.

Panda Security. Thousands of home routers hacked - what can you do?
URL `https://www.pandasecurity.com/mediacenter/mobile-news/routers-hacked/`
Last Accessed: 16 January 2019

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.

Paxson, V. An analysis of using reflectors for Distributed Denial-of-Service attacks. *ACM SIGCOMM Computer Communication Review*, 31(3):38–47, 2001.

Postel, J. User datagram protocol. RFC 0768, August 1980. doi:10.17487/RFC0768.
URL `https://rfc-editor.org/rfc/rfc0768.txt`

Postel, J. Internet Control Message Protocol. RFC 0792, September 1981. doi:10.17487/RFC0792.
URL `https://rfc-editor.org/rfc/rfc0792.txt`

Postel, J. *et al.* Transmission control protocol. RFC 0793, September 1981. doi:10.17487/RFC0793.
URL `https://rfc-editor.org/rfc/rfc0793.txt`

Prudhomme, A. S., Farinholt, B. R., and Sullivan, E. L. An Internet-wide Measurement and Security Analysis of IPsec. *University of California, San Diego*, 2013.

Rapid7. Rapid7 Sonar FDNS.
URL `https://scans.io/study/sonar.fdns_v2`
Last Accessed: 18 January 2019

Rapid7. Rapid7 Sonar port 80.
URL `https://scans.io/study/sonar.http`
Last Accessed: 18 January 2019

Rapid7. Rapid7 Sonar RDNS.
URL `https://scans.io/study/sonar.rdns_v2`
Last Accessed: 18 January 2019

Rapid7. Open Data.
URL `https://opendata.rapid7.com/`
Last Accessed: 12 January 2019

Rekhter, Y., Li, T., and Hares, S. A Border Gateway Protocol 4 (BGP-4). RFC 4271, January 2006. doi:10.17487/RFC4271.
URL `https://rfc-editor.org/rfc/rfc4271.txt`

Richter, P., Smaragdakis, G., Plonka, D., and Berger, A. Beyond counting: new perspectives on the active IPv4 address space. In *Proceedings of the 2016 Internet Measurement Conference*, pages 135–149. ACM, 2016.

RIPE NCC. Analysis of Egyptian Internet outage 27th January - 2nd February 2011.
URL `https://stat.ripe.net/events/egypt`
Last Accessed: 17 January 2019

Rudis, B. Mapping IPv4 Address (with Hilbert curves) in R.
URL `http://datadrivensecurity.info/blog/posts/2015/Jan/mapping-ipv4-address-in-hilbert-space/`
Last Accessed: 17 February 2018

Sachdeva, M., Singh, G., Kumar, K., and Singh, K. Measuring impact of ddos attacks on web services. 2010.

Sargent, M., Kristoff, J., Paxson, V., and Allman, M. On the Potential Abuse of IGMP. *ACM SIGCOMM Computer Communication Review*, 47(1):27–35, 2017.

Shirer, M. and Torchia, M. IDC Forecasts Worldwide Spending on the Internet of Things to Reach $772 Billion in 2018.
URL `https://www.idc.com/getdoc.jsp?containerId=prUS43295217`
Last Accessed: 18 January 2019

Shodan. The search engine for the web.
URL `https://www.shodan.io/`
Last Accessed: 18 January 2019

Swart, I., Irwin, B., and Grobler, M. Towards a platform to visualize the state of South Africa's information security. In *Information Security for South Africa (ISSA), 2014*, pages 1–8. IEEE, 2014.

Tcpdump. TCPDUMP & libpcap.
URL `https://www.tcpdump.org/`
Last Accessed: 18 January 2019

Team CYMRU. Ephemeral Source Port Selection Strategies.
URL `https://www.cymru.com/jtk/misc/ephemeralports.html`
Last Accessed: 18 January 2019

Team Cymru. IP to ASN Mapping.
URL `http://www.team-cymru.com/IP-ASN-mapping.html`
Last Accessed: 14 January 2019

Tozal, M. E. The Internet: A system of interconnected autonomous systems. In *Systems Conference (SysCon), 2016 Annual IEEE*, pages 1–8. IEEE, 2016.

Troy, M. Ongoing, large-scale SIP attack campaign coming from Online SAS (AS12876).
URL `https://badpackets.net/ongoing-large-scale-sip-attack-campaign-coming-from-online-sas-as12876/`
Last Accessed: 12 January 2019

UN Human Rights Council. GENERAL ASSEMBLY: Oral revisions of 30 June.
URL `https://www.article19.org/data/files/Internet_Statement_Adopted.pdf`
Last Accessed: 18 January 2019

US-CERT. OpenSSL 'Heartbleed' vulnerability (CVE-2014-0160).
URL `https://www.us-cert.gov/ncas/alerts/TA14-098A`
Last Accessed: 17 February 2018

US-CERT. TA18-106A: Russian State-Sponsored Cyber Actors Targeting Network Infrastructure Devices.
URL `https://www.us-cert.gov/ncas/alerts/TA18-106A`
Last Accessed: 16 January 2019

Vervier, P.-A., Thonnard, O., and Dacier, M. Mind Your Blocks: On the Stealthiness of Malicious BGP Hijacks. In *NDSS*. 2015.

Won, Y., Fontugne, R., Cho, K., Esaki, H., and Fukuda, K. Nine years of observing traffic anomalies: trending analysis in backbone networks. In *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium*, pages 636–642. IEEE, 2013.

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.

Zhang, J., Durumeric, Z., Bailey, M., Liu, M., and Karir, M. On the Mismanagement and Maliciousness of Networks. In *NDSS*. 2014.

Zheng, L., Joe-Wong, C., Andrews, M., and Chiang, M. Optimizing data plans: Usage dynamics in mobile data networks. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 2474–2482. IEEE, 2018.

Zorz, Z. Mirai Linux Trojan corrals IoT devices into DDoS botnets.
URL `https://www.helpnetsecurity.com/2016/09/07/mirai-linux-trojan-iot-ddos-botnets/`
Last Accessed: 16 September 2017