

AN EXPLORATION OF THE OVERLAP BETWEEN
OPEN SOURCE THREAT INTELLIGENCE AND
ACTIVE INTERNET BACKGROUND RADIATION

Submitted in fulfilment
of the requirements of the degree of

MASTER OF SCIENCE

of Rhodes University

Deon Turner Pearson

Grahamstown, South Africa
26th April, 2019

Abstract

Organisations and individuals are facing increasing persistent threats on the Internet from worms, port scanners, and malicious software (malware). These threats are constantly evolving as attack techniques are discovered. To aid in the detection and prevention of such threats, and to stay ahead of the adversaries conducting the attacks, security specialists are utilising Threat Intelligence (TI) data in their defense strategies.

TI data can be obtained from a variety of different sources such as private routers, firewall logs, public archives, and public or private network telescopes. However, at the rate and ease at which TI is produced and published, specifically Open Source Threat Intelligence (OSINT), the quality is dropping, resulting in fragmented, context-less and variable data.

This research utilised two sets of TI data, a collection of OSINT and active Internet Background Radiation (IBR). The data was collected over a period of 12 months, from 37 publicly available OSINT datasets and five IBR datasets. Through the identification and analysis of common data between the OSINT and IBR datasets, this research was able to gain insight into how effective OSINT is at detecting and potentially reducing ongoing malicious Internet traffic. As part of this research, a minimal framework for the collection, processing/analysis, and distribution of OSINT was developed and tested. The research focused on exploring areas in common between the two datasets, with the intention of creating an enriched, contextualised, and reduced set of malicious source IP addresses that could be published for consumers to use in their own environment.

The findings of this research pointed towards a persistent group of IP addresses observed on both datasets, over the period under research. Using these persistent IP addresses, the research was able to identify specific services being targeted. Amongst these persistent IP addresses were significant packets from Mirai like IoT Malware on port 23/tcp and 2323/tcp as well as general scanning activity on port 445/TCP.

Acknowledgements

I would like to acknowledge and extend a big thank you to all the individuals who believed in me and made this whole research project possible. Firstly I would like to thank my supervisor, Prof Barry Irwin whom without him none of this would be possible. His guidance and support and knowledge throughout the whole project was key to the success I had in this research and development. Additionally, his belief in me played a key role in me seeing the research to the end. I would like to extend my thanks and gratitude to my family for their continuous support and belief in me. Special mention must be made to my wife Danika Pearson who took on the task of supporting us financially throughout my Masters. Her endless patience she has had towards me during this lengthy process has been priceless. All the members of Red Lab for their help and guidance they gave to problems I faced. Furthermore, their continuous laughs and fun times added an element of enjoyment to this project.

I would like to thank the Rhodes University Computer Science Department for providing me with the hardware and access to equipment which I required for this project. To the administration staff, technical staff and teaching staff for the time they gave up to assist me.

Finally I would like to acknowledge the Armaments Corporation of South Africa (Arm-scor) and the Council for Scientific and Industrial Research (CSIR) for the financial support throughout this research project. This work was undertaken in the Distributed Multimedia Centre of Excellence at Rhodes University, with financial support from Telkom SA, THRIP, Coriant, Easttel and Bright Ideas 39. The author acknowledge that opinions, findings and conclusions or recommendations expressed here are those of the author and that none of the above mentioned sponsors accept liability whatsoever in this regard.

Contents

1	Introduction	1
1.1	Background	2
1.2	Problem Statement	3
1.3	Research Objectives	4
1.4	Research Approach and Scope	5
1.5	Document Conventions	6
1.6	Document Structure	6
2	Literature Review	8
2.1	Routing	9
2.1.1	Routing Information Base	9
2.1.2	Forwarding Information Base	10
2.2	Transmission Control Protocol (TCP)	10
2.2.1	TCP Flags	10
2.2.2	TCP Communication	11
2.3	Internet Control Messaging Protocol (ICMP)	12
2.4	Border Gateway Protocol (BGP)	12
2.4.1	IBGP	13
2.4.2	EBGP	14
2.5	Internet Traffic	15
2.5.1	Malicious Traffic	16
2.5.2	Other Traffic	17

2.5.3	Active and Passive IBR	17
2.6	Network Telescopes	18
2.7	Malicious Internet Activity	20
2.7.1	Botnets	20
2.7.2	Denial of Service Attacks	21
2.7.3	Port Scanning	22
2.7.4	Proxy Servers and Open Relays	24
2.7.5	IP Spoofing	24
2.8	Domain Name System Block List (DNSBL)	25
2.9	Threat Intelligence (TI)	27
2.9.1	Sources of Threat Intelligence	28
2.9.2	Open Source Threat Intelligence (OSINT)	28
2.10	Threat Intelligence Platforms (TIPs)	29
2.10.1	AlienVault Open Threat Exchange	29
2.10.2	Webroot Threat Intelligence Platform	30
2.10.3	GOSINT	30
2.10.4	Collective Threat Intelligence Gathering System	31
2.10.5	Recorded Future	31
2.11	Chapter Summary	32
3	Data	33
3.1	Data Sources	34
3.2	Open Source Threat Intelligence	34
3.2.1	Data Collection and Preprocessing	34
3.2.2	Dataset Overview	35
3.3	Network Telescope Sensors	38
3.3.1	Data Collection and Processing	40
3.3.2	Dataset Overview	40

3.4	Chapter Summary	44
4	Design and Implementation	46
4.1	System Design	47
4.2	Technologies Used	48
4.2.1	JavaScript Object Notation (JSON)	48
4.2.2	Comma Separated Values (CSV)	49
4.2.3	ExaBGP	50
4.2.4	RBLDNSD	51
4.2.5	PostgreSQL	52
4.3	Database Design	53
4.3.1	Sources Table	54
4.3.2	OSINT Tables	54
4.3.3	IBR Tables	55
4.4	Collection	55
4.5	Chapter Summary	58
5	Data Analysis	59
5.1	Initial Analysis	60
5.2	Change in Datasets	63
5.2.1	Monthly Change in Dataset Totals	64
5.2.2	Monthly Change in Dataset Composition	66
5.3	Offset Comparison	70
5.3.1	Saw Tooth	72
5.3.2	Bell Shape	72
5.3.3	Discussion	73
5.4	Persistent IP Addresses	75
5.5	Highly Persistent IP Addresses	77
5.5.1	Volume of Persistent IP Addresses	77

5.5.2	Persistent IP Addresses in Common	78
5.5.3	Persistent IP Address Services	79
5.5.4	Month-to-Month Change in Requests	83
5.6	Source IP Address Evaluation	86
5.6.1	Registry Grouping	86
5.6.2	Geopolitical Analysis	88
5.6.3	AS Names	89
5.7	Chapter Summary	90
6	Processing and Publishing	92
6.1	Processing	92
6.1.1	Whitelisting Module	93
6.2	Publishing	94
6.2.1	Flat File Module	95
6.2.2	Border Gateway Protocol (BGP) Module	96
6.2.3	Domain Name System Block List (DNSBL) Module	98
6.3	Chapter Summary	99
7	Conclusion	101
7.1	Document Summary	102
7.2	Evaluation of Research Goals	103
7.3	Research Contribution	104
7.4	Future Work	105
	References	106
A	Open Source Threat Intelligence	123
A.1	Open Source Threat Intelligence Feeds	123
A.2	Abuse.ch	123
A.3	Alexa	127

A.4	AlienVault	127
A.5	AutoShun	127
A.6	Bambenek Consulting	128
A.7	Binary Defense	128
A.8	Brute Force Blocker	129
A.9	Collective Intelligence Network Security (CINS)	129
A.10	Cisco Umbrella	130
A.11	Malware Domain List (MDL)	130
A.12	Proofpoint Emerging Threat Intelligence	130
A.13	Passive Spam Blocklist	131
A.14	SANS ISC	131
A.15	SpamHaus	132
B	Project Repository	135
B.1	System Code	135
B.2	Analysis Scripts	136
C	Design and Implementation	137
C.1	Process PCAPS Script	137
C.2	Database Design	138
C.3	Custom Libraries	138
C.4	Module Data	140
C.4.1	Collector Module	140
C.4.2	Processing Module	140
C.4.3	Publishing Module	141
C.5	DNSBL Configuration	142
C.6	BGP Configuration	142
C.6.1	ExaBGP Script	143

D Data Analysis	144
D.1 Monthly Change in Dataset Composition	144
D.2 Monthly Offset Comparison	148
D.3 Persistent IP Addresses	155
D.4 Highly Persistent IP Addresses	156
D.5 Additional Results	171

List of Figures

2.1	IP Datagram Format after Postel (1981)	9
2.2	TCP Datagram after Postal (1981)	11
2.3	TCP Three Way Handshake after Kurose and Ross (2013)	12
2.4	BGP Illustration after Kosem (2015)	13
2.5	Incoming Network Traffic on Host after Blenn <i>et al.</i> (2017)	15
2.6	Incoming Network Traffic on Network Telescopes after Blenn <i>et al.</i> (2017) .	16
2.7	Attack Phases after Gadge and Patil (2008)	23
3.1	Network Telescope Infrastructure (Irwin <i>et al.</i> , 2007)	40
3.2	Network Telescope Protocol Split	42
3.3	IBR Dataset Volume of Data	45
4.1	System Design	47
4.2	Database Schema	54
4.3	Open Source Threat Intelligence Feeds Collection Process	56
5.1	Percentage Of Observed IP Addresses In Common	63
5.2	Change in Data Set Example	66
5.3	Month-to-Month Change in OSINT Data	67
5.4	Month-to-Month Change in IBR Data	69
5.5	IBR1 Monthly Offset Comparison CINS	71
5.6	IBR1 Monthly Offset Comparison AlienVault	73
5.7	Persistent IP Addresses	76

5.8	Top 20 TCP Ports: RUSCOPE2	80
5.9	Top 20 TCP Ports for IBR Datasets	81
5.10	Top 20 TCP Ports For OSINT Datasets	82
5.11	OSINT and IBR Heatmaps	87
5.12	Countries	91
6.1	Flat File Module Design	95
6.2	BGP Module Design	96
6.3	ExaBGP Python Script Flow	98
6.4	DNSBL Module Design	99
D.1	IBR1 Monthly Offset Comparison	148
D.2	IBR2 Monthly Offset Comparison	149
D.3	IBR3 Monthly Offset Comparison	150
D.4	IBR4 Monthly Offset Comparison	151
D.5	IBR5 Monthly Offset Comparison	152
D.6	IBR6 Monthly Offset Comparison	153
D.7	IBR7 Monthly Offset Comparison	154
D.8	Top 20 TCP Ports For RUSCOPE1	157
D.9	Top 20 TCP Ports For RUSCOPE2	158
D.10	Top 20 TCP Ports For RUSCOPE3	159
D.11	Top 20 TCP Ports For RUSCOPE4	160
D.12	Top 20 TCP Ports For RUSCOPE5	161
D.13	Top 20 TCP Ports For AlienVault	162
D.14	Top 20 TCP Ports: CINS	163
D.15	Top 20 TCP Ports: IBR1	164
D.16	Top 20 TCP Ports: IBR2	165
D.17	Top 20 TCP Ports: IBR3	166
D.18	Top 20 TCP Ports: IBR4	167

D.19 Top 20 TCP Ports: IBR5	168
D.20 Top 20 TCP Ports: IBR6	169
D.21 Top 20 TCP Ports: IBR7	170

List of Tables

2.1	TCP Flags	11
2.2	TCP Packet Configurations (Irwin, 2011)	18
2.3	ICMP Packet Configurations (Irwin, 2011)	19
2.4	Levels of DOS Attacks	21
2.5	BrightCloud Services	30
2.6	Collective Threat Intelligence Gathering System	31
3.1	Initial OSINT Sources	35
3.2	Summary of OSINT Feeds Collected	36
3.3	OSINT Collection HTTP Errors	37
3.4	OSINT Feed Types	38
3.5	OSINT Volume of Data	39
3.6	Network Telescope Data	41
3.7	Network Telescope Active Passive Data	43
3.8	IBR Datasets of Active Unique IP Addresses	44
3.9	IBR Dataset Volume of Data	45
4.1	RBLDNSD Arguments (Tokarev, 2015)	51
5.1	Total IP Addresses in Common Between OSINT and IBR Datasets	61
5.2	IP Addresses in Common Between OSINT and IBR Datasets	62
5.3	Dataset Percentage Change Values Month-to-Month	65
5.4	IBR Persistent IP Addresses	78

5.5	IBR Persistent IP Addresses in OSINT Lists	78
5.6	Top Targeted TCP Ports	83
5.7	AlienVault Description	83
5.8	Top 20 TCP Ports Requested	84
5.9	Top 20 UDP Ports Requested	85
5.10	Top 10 Countries	88
5.11	Top 10 ASes: IBR6 Highly Persistent IP Addresses	89
5.12	Packet Count of Top 10 ASes IP Addresses	90
A.1	Open Source Threat Intelligence Source Data	124
A.2	Feodo Tracker	125
A.3	Ransomware Tracker	126
A.4	SSL Blacklist	126
A.5	ZeuS Tracker	126
A.6	IP Reputation List Data	127
A.7	Brute Force Blocker Block List Data	129
A.8	Malware Domain List Lists	130
A.9	SANS Lists	131
A.10	SANS IP Block List Data	132
A.11	DNSBL Sources	132
A.12	ZEN Result Codes	133
A.13	DROP Lists	134
C.1	Sources Table Design	138
C.2	DB Handler Functions	139
C.3	File Handler Functions	139
C.4	Source Handler Functions	140
C.5	Collection Library	141
C.6	Zone File Lines (Tokarev, 2015)	142

C.7	ExaBGP Script	143
D.1	Month-to-Month Change in OSINT Data Values	144
D.2	Month-to-Month Change in IBR Data Values	146
D.3	Persistent IP Addresses	155
D.4	AlienVault Top 20 IP Addresses by Unique Ports Targeted	171
D.5	IBR6 Top 20 IP Addresses by Unique Ports Targeted	172
D.6	AlienVault Monthly Top 20 Port Request Count	173
D.7	IBR6 Monthly Top 20 Port Request Count	174

List of Listings

2.1	DNS Question and Answer (Caudle, 2015)	25
2.2	Reverse DNS Question and Answer	26
4.1	CSV File Format (Shafranovich, 2005)	49
4.2	ExaBGP Terminal Command Example	50
4.3	RBLDNSD Command	51
4.4	Postgres Function Example	53
4.5	Commented Data File	57
6.1	Publish BGP Function	97
6.2	Route Example	97
6.3	ExaBGP Command Example	98
C.1	Zone File Line Options (Tokarev, 2015)	142

1

Introduction

IN recent years the volume, frequency, and intensity of Internet-based attacks against organizations have risen in number, with no sign of relenting (Modi *et al.*, 2016). Morgan (2016) predicts that in 2019, the annual costs of cybercrime will reach 2 trillion US dollars. This is largely due to the increasing connectivity of IT infrastructure, and the deployment of Internet of Things (IoT) devices (Ahrend *et al.*, 2016).

With the increase in the number of Internet-connected devices comes the parallel increase in the amount of malicious unwanted Internet activity. In this research, the source of a malicious and unwanted Internet request is referred to as a Threat Actor (TA). This definition includes all malicious hosts on the Internet as well as sources of malicious Internet packets, such as humans, worms, bots, scanners, rogue devices, and any other unwanted, potentially malicious traffic (Magee *et al.*, 2014, Bromiley, 2016).

Although it is more common for large organisations to be the target of Internet-directed attacks, TAs are constantly probing the entirety of the Internet looking for weaknesses and vulnerabilities to exploit and gain access to network infrastructure and sensitive data (Hutchins *et al.*, 2011). TAs have the upper hand when exploiting a system, since they only have to find one vulnerability or weakness, as opposed to security teams, who have to defend on every front. The TAs essentially make the first move in the process; then traditionally security teams respond to the moves of the TAs.

Due to the rapidly evolving and diverse selection of technologies and tools available to TAs, cyber-attacks are evolving and changing at a rapid pace (Bromiley, 2016). Proofpoint

(2016) identified the need to better understand the threats being faced by networks and systems. They are quoted saying “Knowing what types of threats exist is no longer enough to protect your people, data, and brand”. Having the ability to identify the source of the threat, who is behind it, and the methods being used greatly increases the chances of protecting against the threat. The type of information required includes source IP addresses, domains, URLs being used by TAs, as well as which ports or services are being targeted (Magee *et al.*, 2014). This information needs to be timely and up-to-date for it to be deemed useful (Ahrend *et al.*, 2016). Threat Intelligence (TI) is one way of gathering detailed, timely, and up-to-date information. TI includes information such as Indicators of Compromise (IOC)¹, botnet IPs, malware signatures, malicious URLs, malicious domains, and spam sources. TI that is well-crafted and accurate can provide an advantage to security teams trying to protect their critical infrastructure against attacks and unwanted traffic. It allows the security team to have ample information about the TAs and thus make an informed decision on how to defend against them.

1.1 Background

In recent years, security teams have come to realize the benefits TI can provide when protecting an internal network from external threats (Anomali, 2003, Barnum, 2012, Lee and Shon, 2016). TI is usually provided to a consumer, either as a free or paid service. Paid services typically publish TI via dedicated Threat Intelligence Platforms (TIPs) (Sauerwein *et al.*, 2017). Commercial TIPs such as Threat Connect², Recorded Future³, Anomali⁴, and IBM X-Force Exchange⁵, provide a comprehensive set of TI data for a fee. Some of these TIPs, such as IBM X-Force, offer a free tier subject to terms and services and usage policies. However, to get the most benefit of the TIP the service must be paid for. These platforms contain data obtained from thousands of different TI feeds that have been extensively processed, managed, and analysed into a set of contextualised TI data and made accessible to paying consumers (Robb, 2017). The TIPs mentioned above are able to provide this high quality of TI using income generated from consumers.

An alternative to paid TI is free or Open Source Threat Intelligence (OSINT) (Lee and Shon, 2016). OSINT is typically gathered from community sources such as public forums, network telescopes (telescopes), public databases, and other OSINT feeds. OSINT has the potential to significantly hinder the success rate of attacks on the Internet if it is used correctly and to its full potential. According to Bromiley (2016), research findings

¹Catakoglu *et al.* (2016) defines an IOC as “forensic artifacts that are used as signs that a system has been compromised by an attack or that it has been infected with a particular malicious software”

²<https://threatconnect.com/>

³<https://www.recordedfuture.com/>

⁴<https://www.recordedfuture.com/>

⁵<https://exchange.xforce.ibmcloud.com/>

published in the 2016 SANS Incident Response Survey show that 72% of the organisations interviewed realise the importance of TI and implement it to some extent. However, the majority of these organisations do not use TI to its full potential.

If security teams collaborate and share OSINT with one another, the detection rate of threats will improve and the prevention rate of attacks will increase. This is useful because if one organisation is attacked, the chances are high that another organisation will soon experience a similar attack, or attempted attack, from the same source or with the same pattern; thus sharing OSINT can be advantageous to multiple organisations (Ahrend *et al.*, 2016, Liao *et al.*, 2016). The major drawback of OSINT is that with sources emerging all over the Internet, the quality of the data gathered is dropping (Caltagirone, 2015b). As the quality drops the need for consumers to perform additional curation and processing rises. Uncurated data lacks context and relevance, which requires consumers to spend time and money processing it and turning it into usable data. These complexities inevitably discourage the consumer from using the data.

1.2 Problem Statement

Finding OSINT feeds on the Internet has become significantly easier in recent years, since both open source and paid services are readily available to the consumer (Sillaber *et al.*, 2016). Any entity has the ability to publish a dataset as “TI”, but without context, validation, and relevance, the data provides little to no benefit to the user. In addition, as the number of OSINT sources increases, a negative effect is being experienced by consumers of the data (Sillaber *et al.*, 2016, Sauerwein *et al.*, 2017). The quality of OSINT is reducing and there are more cases of redundant, uncategorised data being published. As a result, consumers are finding it harder to curate the data into something relevant and usable (Tounsi and Rais, 2017). This reduction in the quality of data, combined with the sheer amount of data that needs to be processed, means that OSINT is not being used to its full potential, or resulting in undesirable effects (Serrano *et al.*, 2014).

There are many considerations when it comes to evaluating TI. Lee and Shon (2016) identify two conditions TI must meet for it to be useful; 1) the content and information must be verified, and 2) the TI must be analysed and evaluated to ensure it is useful for its purpose. Caltagirone (2015a) and Sillaber *et al.* (2016) propose a method of evaluating the quality of OSINT using the CART (Completeness, Accuracy, Relevance, and Timeliness) framework:

1. **Completeness:** TI is complete if it is usable by more than one consumer on the Internet. If the data is too specific to be applied to multiple cases, or to be used by another consumer, it is incomplete.
2. **Accuracy:** if the TI has many errors and redundant data it is considered inaccurate

and the organisation will spend extra time and money processing the data before it can be used.

3. **Relevance:** the TI must relate to threats that the organisation is facing or will face. Additionally, the TI must be published in a method that is most relevant for the type of threat.
4. **Timeliness:** there are two aspects to timeliness. Firstly, the TI needs to be published and received fast enough to block the threats being faced. Secondly, the TI needs to be operationalised with enough time to prevent the threat.

1.3 Research Objectives

From the problem identified above, it is evident that having large volumes of OSINT to manage and process into actionable data can result in reduced quality of TI, reduced performance of systems processing the data, and reduced protection against threats. As such, this research had two aims. The first was to evaluate the effectiveness of OSINT against real-world Internet data. The second was to research how to disseminate TI to consumers in a method that is most appropriate for the type of TI being distributed. To achieve these goals, the research set out to investigate and achieve four primary and two secondary objectives.

Primary Objectives:

1. Develop suitable tools for the collection and analysis of OSINT and telescope TI. The recommended use of OSINT is to combine different feeds into a single dataset that can be operationalised into security strategies. This research collected a number of different OSINT feeds to use in the processing and analysis stages.
2. Evaluate the OSINT and telescope datasets to discover overlapping data and understand how effective existing OSINT data is at reducing the number of requests being recorded on the telescope sensors. This objective focuses on evaluating OSINT in terms of the CART framework discussed above.
3. Develop processing modules that are capable of turning the collected OSINT into enriched contextualised TI.
4. Develop tools to aid in the distribution of enriched TI to consumers in a way that is most suitable for the type of threat being faced. Publishing TI in a manner appropriate for the type of service being targeted allows the TI feeds to be tailored for the environment they will be used in.

Secondary Objectives:

1. Investigate ways to reduce OSINT data into smaller datasets and/or categories to allow consumers to subscribe only to data relevant to their needs. Categories include

the severity level of the attack and the type of threat detected.

2. Evaluate the performance of defence strategies using OSINT data in a selection of critical network hardware. Performance plays a key role in determining whether any defence strategy is effective. An effective defence solution is one in which the ratio of performance to threat coverage is balanced. Having a high threat coverage but a low performance can deter consumers from using the OSINT.

1.4 Research Approach and Scope

This research aimed to evaluate the effectiveness of OSINT by comparing it to real-world malicious Internet traffic. The basis for this research was formed by understanding the limitations inherent in OSINT data and reasons behind consumers not utilising OSINT to its full potential.

Two distinct datasets were collected over a period of 12-months. The first was a collection of OSINT from various sources on the Internet. The second was telescope data collected from five telescopes maintained by the Rhodes University Security Networks Research Group (SNRG) (Irwin, 2011). Tools and techniques were developed for the collection, analysis, processing, and dissemination of the data. During the course of this research, quantitative analysis was performed on both the OSINT and the telescope data to understand the composition of the datasets and the common data they share.

The scope of this research was the exploration of the overlap between OSINT and telescope data. The network telescope sensors are a set of passive network monitoring devices designed to record both active and passive Internet Background Radiation (IBR) (Section 2.5). However, the analysis and evaluation in this research was limited to active Internet traffic (Section 2.5.3).

Included in this research was the design of a system capable of collecting, processing, and disseminating OSINT. The collection component (Section 4.4) was limited to collecting OSINT that is freely available on the Internet as a downloadable set of data. During the data collection stage, a variety of different types of OSINT lists were collected (Appendix A.1). These lists contained data on malicious domains, IP addresses, URLs, and more. However, to allow for comparability, the analysis was limited to OSINT datasets that contain IPv4 source addresses, since the telescopes only record IPv4 packets from Internet hosts.

The design of the processing component (Section 6.1) was limited to a conceptual discussion of how modules can be added to the system to turn the collected OSINT into TI that can be published. The publishing component (Section 6.2) consisted of three modules; Flat File, BGP, and DNSBL. These modules were chosen to demonstrate how OSINT can

be published on various channels and various formats in a service appropriate way.

1.5 Document Conventions

In the remainder of this document, a number of conventions can be seen. This section describes the conventions used.

URLs: When a URL pertaining to any website, organisation or software is mentioned, it is given as a footnote on the page of mention. These are hyper-linked in the electronic version of this document.

IP Address: This research focuses primarily on IPv4 addresses. As such, when the term IP address is used, it refers to IPv4 addresses unless specifically stated. Examples in this research use IP addresses reserved for documentation purposes (192.0.2.0/24, 198.51.100.0/24, and 203.0.113.0/24) (Arkko *et al.*, 2010).

Dates: This research collected data from September 2017 through August 2018. As such, when two-digit numbers are used to identify months, the months are displayed on their own⁶. In 2017 the months are September through December (09 through 12). In 2018 the months are January through August (01 through 08). Date formats are as follows:

- Long Months - September 2017
- Short Month - 09/17
- Month Range - 09-10

1.6 Document Structure

The remainder of the document adheres to the following structure:

Chapter 2 provides a literature review for this research. The intention of this literature review is to give the reader an understanding of background material relevant to this research project.

Chapter 3 presents a discussion and quantitative analysis of the data used in this research project. The chapter focuses on describing the datasets collected over the 12-month period. There are two datasets used in this study; OSINT collected from external TI feeds, and internet background radiation data collected from five telescope sensors at Rhodes University.

Chapter 4 presents the design and implementation of a prototype system capable of

⁶This is possible because there are no duplicate months in the collection period

collecting, processing, analysing, and distributing enriched OSINT to consumers. This chapter is part one of two chapters that discuss the design of the system. The system developed as part of this research comprises three main components; collection, processing, and distribution. Included in this chapter is a discussion of various technologies and approaches used to achieve the development goals, as well as a detailed discussion of the *Collection* component.

Chapter 5 discusses the exploratory analysis of the data collected using the tools described in Chapter 4. The chapter starts with a high-level analysis of the two datasets before progressing down to more specific analysis cases. The chapter is separated into multiple sections; one for each type of analysis performed on the datasets.

Chapter 6 presents the design and implementation of the *Processing* and *Publishing* components of the system described in Chapter 4. The main focus of this chapter is on the *Publishing* component and the three modules designed to illustrate how TI can be distributed to consumers via different channels.

Chapter 7 concludes the research with a discussion of the results achieved during this study. The focus of this chapter is tying the results achieved to the goals set at the start of this research. The chapter concludes with a discussion of future development and research that could stem from this study.

2

Literature Review

THIS chapter presents the reader with the background knowledge required to understand this research both in terms of concepts and related work. The remainder of this chapter is layout out with **Section 2.1** discussing relevant information about routing on the Internet. Included in this section is a discussion of how a router or networking device determines where a packet is coming from and going to. **Section 2.2** briefly touches on relevant details of the Transmission Control Protocol (TCP) that is used in this research. **Section 2.3** briefly discusses the Internet Control Protocol (ICMP) information relevant to this research. **Section 2.4** discusses the Border Gateway Protocol (BGP) which is responsible for publishing known routes between border routers on the Internet. **Section 2.5** details background information on the three types of Internet traffic; backscatter, misconfiguration, and malicious traffic. These three types of traffic allow the classification of telescope data into active and passive data. Telescopes are detailed in **Section 2.6** and their traffic in **Section 2.7**. **Section 2.8** discusses Domain Name System Block Lists (DNSBL) and how they are used by mail clients to determine whether a source is malicious or not. **Section 2.9** details TI and what this research classifies as TI. Included in this section is a discussion of the sources of TI, as well as OSINT and what it is. **Section 2.10** discusses TIPs and some of the existing projects analysing and distributing TI to consumers.

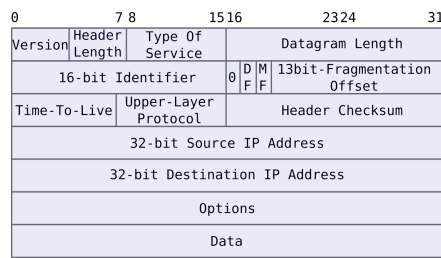


Figure 2.1: IP Datagram Format after Postel (1981)

2.1 Routing

For a router to successfully route a packet on a network, it must be able to determine the destination IP addresses, as well as the next-hop to pass the packet to (Postel, 1981, Slattery and Burton, 2000). The destination IP address and other packet related routing information are contained within the *IP Datagram* (Figure 2.1) section of the packet. The next-hop is typically an interface on the router that the packet must be sent to for it to reach the final destination. The next-hop details are maintained within the routing table of the router, and determines how the router should handle each packet. The *IP Datagram*, however, is specific to each packet, and contains information about the packet, its source and destination.

The IP addresses analysed in this research are the *32-bit Source IP Addresses* taken from the *IP Datagram* of Internet packets (Postel, 1981). The *32-bit Source IP Addresses* identifies the sender of the packet; firewalls and routers with source based rules will use this field to determine what action to perform on the packet.

2.1.1 Routing Information Base

The Routing Information Base (RIB) is a data table stored on a router or network connected device that contains a list of all known routes and paths to a specific destination (Scudder, 2007, Bahadur *et al.*, 2018). The RIB is used by the routing device to determine the best path towards the destination (Washburn and Evans, 1993). The RIB is populated by a number of different protocols such as Open Shortest Path First (OSPF) and Border Gateway Protocol (BGP). Each protocol maintains its own RIB database of known routes which it submits to the global RIB (routing table¹). The RIB can contain several routes to the same destination (Melazzi *et al.*, 2013, Fàbrega *et al.*, 2013), it is not concerned with which route is best. It does, however, know when routes become unavailable. The RIB is kept consistent and free from loops through the exchange of information between other instances on the control plane (Nadeau and Gray, 2013).

¹A phone book like structure used to determine the direction of the next physical hop

2.1.2 Forwarding Information Base

Forwarding is the process of sending a packet onto the correct interface to get it on the correct route towards its destination (Garud, 2019). When a router has determined the best route to send a message on, it determines how to forward the message accordingly. The Forwarding Information Base (FIB) contains a single route per destination network (or host) (Trotter, 2001, Fàbrega *et al.*, 2013). The route contained in the FIB is the best route (as determined by inspection of the RIB) to get the packet to its destination. The FIB draws its information from the RIB once the RIB is deemed to be consistent and stable (Nadeau and Gray, 2013).

When a routing device receives a packet, it uses the destination IP addresses to search the FIB to determine which interface to send the packet to (Melazzi *et al.*, 2013). The device will match the destination IP address to a list of known networks stored in the FIB (Schulting, 2014). There are two types of routes which can be seen in the FIB, network routes and host routes (Shinder, 2001). Network routes contain information on how to reach a specific network on the Internet, and host routes contain information on how to reach a specific host on the local network.

2.2 Transmission Control Protocol (TCP)

TCP can be considered the protocol of the Internet. TCP is responsible for the connection between two endpoints over which application data can be sent. According to Balram and Wiscy (2008), TCP data accounts for more than 85% of packets entering and leaving a system. A TCP connection is a point-to-point connection which has a dedicated communication channel between sender and receiver (Kurose and Ross, 2013). Each message sent using the TCP protocol is encapsulated with a *TCP Datagram* illustrated in Figure 2.2. Of interest to this research is the *Destination Port* and *Flags* part of the datagram. The destination port is used to signal which service the message must be passed to when the destination host receives the message. The flags signal the type of message being sent. The ports are primarily used in Chapter 5 during the discussion of the type of activities the telescopes are detecting.

2.2.1 TCP Flags

A TCP flag is a 6-bit string which is toggled based on the type of TCP request being sent (Postal, 1981). The six main flags considered in this research are described in Table 2.1. In the three-way handshake described below, the TCP SYN and TCP ACK flags are toggled in different messages. In this research, the TCP flags are used in Section 2.5.3,

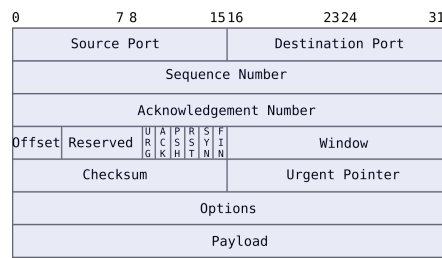


Figure 2.2: TCP Datagram after Postal (1981)

Table 2.1: TCP Flags

Flag	Description
URG	Indicates to the receiver that the data stored in the current request is marked as urgent.
ACK	Indicates that the previous request was valid and has been acknowledged.
PSH	Signals that the receiver should pass the data immediately to the upper layers.
RST	Indicates that the sender must reset the connection. This tells the sender that the request was received but there are no services listening on the requested port number.
SYN	Tells the receiver there is a synchronisation number in the TCP header. This signals to the receiver, the sender is trying to make a connection.
FIN	Tells the receiver there is no more data to be sent in the current connection.

to determine whether packets recorded on the telescopes are active or passive packets.

A receiver will respond to a TCP request based on the flags that are toggled. Examples of these responses are discussed in Sections 2.7.2 and 2.7.3.

2.2.2 TCP Communication

There are many different types of TCP messages that can be sent. With each communication that occurs, the sender and receiver must create a security relationship in the form of a three-way handshake. This relationship creates an assurance that the packet will arrive at the receiver. For example, if Alice and Bob wish to communicate they must first complete a three-way handshake. The TCP flags are a way to signal the type of message being sent. A simplified three-way handshake can be seen in Figure 2.3. Alice wants to send a message to Bob so Alice initiates the connection with a TCP SYN request. If Bob is listening on the specific port and Alice is allowed to make a connection, Bob replies to Alice with a TCP SYN+ACK response. Alice receives the TCP SYN+ACK response and sends a TCP ACK response to Bob to signal the connection has been created. This describes a typical connection process that is relevant in this research because the passive telescopes detect the first message in the in the TCP communication process.

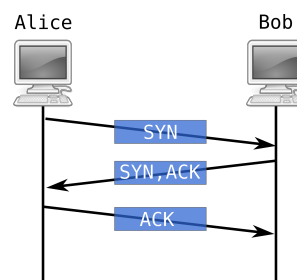


Figure 2.3: TCP Three Way Handshake after Kurose and Ross (2013)

2.3 Internet Control Messaging Protocol (ICMP)

The Internet Control Messaging Protocol (ICMP) is used to signal the status of a network or host on the Internet. A common ICMP request is the “ping” request (Blenn *et al.*, 2017). One of the characteristics of ICMP messages is they carry error messages or codes if the request does not reach its intended destination. Blenn *et al.* (2017) gives an example of how ICMP requests can be considered malicious or perceived as resulting from malicious activities. For example, in a DDoS (Section 2.7.2) attack, the TA targeting a router may send an ICMP request with a very large Time To Live (TTL) value. The TTL value determines how long the request should continue trying to reach its destination. If the router cannot find the destination it will continue trying to send the packet as long as the TTL is still valid. With each hop, the TTL value is decremented by one unit until it reaches zero, at which point, the packet is dropped.

Many network scanning tools use a “ping test” to determine if a host is reachable, as a precursor to a TCP or UDP port scan (Panjwani *et al.*, 2005, Irwin, 2011). The “ping test” scans are echo request (ICMP Type 8) messages that are simply asking the target host to reply if the message has been received. If the host is live and is able to respond to the packet it will send back an echo reply (ICMP type 0) message (Postel *et al.*, 2018). Further discussion of these types are presented in Section 2.5.3.

2.4 Border Gateway Protocol (BGP)

In general, there are two types of routing protocols in use, the Interior Gateway Protocol (IGP) and the Exterior Gateway Protocol (EGP) (Kosem, 2015). However, EGP did not last long before it was replaced by Border Gateway Protocol (BGP). BGP is used as one of the three distribution formats in Chapter 6. BGP is a routing protocol used to determine the best path from one Autonomous System (AS) to the next (Rekhter *et al.*, 2005). The most concise way to define an AS is given by Kosem (2015) as, “... a network or a collection of networks that are all managed by a single entity or organization”. The most common example of an AS is an Internet Service Provider (ISP) (Schulting, 2014).

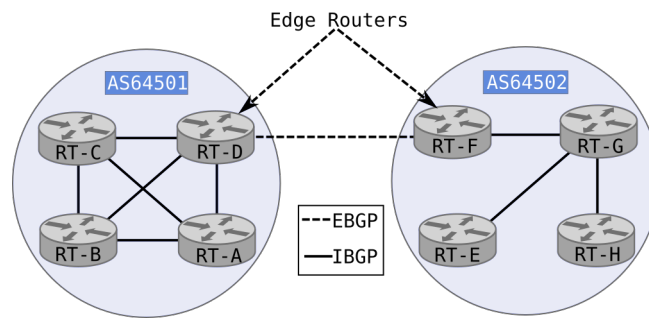


Figure 2.4: BGP Illustration after Kosem (2015)

Each AS is given a globally unique 16 digit number between 1 and 65535 known as the AS Number (ASN) (Techopedia, 2013). The analysis in Section 5.6.3 uses ASNs to evaluate the source of the threats. The ASN enables BGP routers to communicate with their neighbours and successfully forward packets from source to destination (Chandrashekar *et al.*, 2005). The Internet is made up of many ASes each with their own collection of IP networks. A simplified example of the Internet in terms of ASes and BGP routers is illustrated in Figure 2.4, in which there are two ASes, *AS64501* and *AS64502*. When data is moved between them, the edge routers are able to forward the packets because each router knows about its neighbouring AS.

To accomplish the goal of routing packets between ASes, at least one router in the AS must know how to reach a neighbouring router. BGP makes this possible by creating and sharing the FIB across the Internet (see Section 2.1.2).

For BGP to work, two neighbouring routers must be configured to recognise and trust each other. The term given to this is BGP neighbour or BGP peer (Kosem, 2015). There are two types of BGP peer establishments, interior/internal BGP (IBGP) or external/exterior BGP (EBGP), these two types can be seen in Figure 2.4. Figure 2.4, illustrates these two types. *RT-A*, *RT-B*, *RT-C* and *RT-D* within *AS64501* have an IBGP connection, denoted by the solid lines and the two edge routers (*RT-D* and *RT-F*) have an EBGP connection denoted by the dotted lines (Rekhter *et al.*, 2005). BGP routes are used in Figure 6.2 when the system published TI via the BGP module.

2.4.1 IBGP

Only edge routers are able to communicate with other ASes. Internal routes of the AS that are not edge routers cannot communicate directly with another AS. However, these internal routers still need to know how to reach the external ASes. IBGP allows routers within an AS to exchange reachability information. It allows internal routers to reach other routers on a different AS through the edge router (Griffin and Wilfong, 2002). Each router is aware of the edge router, therefore, any packet received by an internal router,

destined for an external router, will be passed onto the edge router for that AS. The following list contains two important constraints on BGP routers within the same AS:

- There must be an IBGP connection to allow neighbouring routers to communicate with each other.
- Routes learned from an internal peer must not be sent back to internal peers, only to external peers.

Within an AS, the BGP routers will, by default, create a full mesh topology network (Xiao and Nahrstedt, 2004), which is illustrated in Figure 2.4 in *AS64501*. A full mesh topology is often difficult and complex to implement because for n IBGP routers in an AS, there are $(n \cdot \frac{(n-1)}{2})$ unique IBGP sessions to maintain (Bates *et al.*, 2006). For example, if *RT-A* receives a route update from an EBGP peer it would have to advertise the route to *RT-B*, *RT-C*, and *RT-D*. Furthermore, if an additional router is added to the topology, all other routers need to be manually updated to include the new router.

To address the overheads and complexities of a full mesh topology, the IBGP routers can be configured in a route reflector topology (Bates *et al.*, 2006). A route reflector provides three main advantages:

1. **Simplicity:** It is simple to configure.
2. **Easy transition:** Transitioning from full-mesh topology is simple.
3. **Compatibility:** Non-compliant IBGP peers are still able to be part of the AS.

In Figure 2.4, *AS64502* is configured in a route reflector topology. *RT-G* is the route reflector which allows it to advertise IBGP routes to other IBGP routers. Consequently, if *RT-F* receives an EBGP route, it will pass it onto *RT-G* which will then publish the route to *RT-E* and *RT-H*.

2.4.2 EBG P

For two different ASes to communicate and exchange routing information, at least one BGP connection must exist between a router in each AS (Chandrashekar *et al.*, 2005). The BGP communication between two ASes is referred to as EBGP. The connection between the two edge routers, *RT-D* and *RT-F*, in Figure 2.4 illustrate an EBGP connection between the two ASes.

Kosem (2015) defines characteristics of EBGP routers as follows:

- Routers set-up with EBGP have a direct connection with their peers in the other AS.
- Routes learned from external peers can be sent to all other peers. For example, if *RT-F* in Figure 2.4 learns a new route from *RT-D* it will publish the new route to

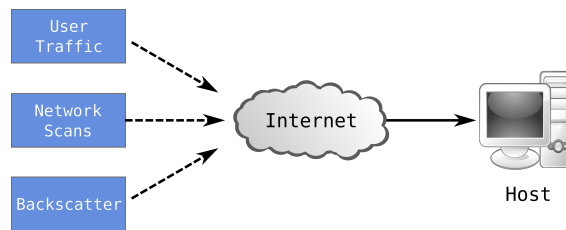


Figure 2.5: Incoming Network Traffic on Host after Blenn *et al.* (2017)

RT-G who will then publish it to all other IBGP peers in the AS.

- When one router sends a route to another router, it appends its local AS number and modifies the next-hop to its local IP addresses.
- BGP neighbours must be one network hop away by default (i.e. directly reachable). For example, in Figure 2.4 *RT-D* and *RT-F* are neighbours because they are connected by an EBGP connection. BGP neighbours are not necessarily directly connected to each other as in Figure 2.4. However, they do need to be able to reach the IP addresses they use in the relationship (Schulding, 2014).

2.5 Internet Traffic

Any Internet-connected host is capable of receiving Internet traffic. Some hosts with public IP addresses, such as routers and firewalls are exposed directly to the incoming Internet traffic. Other hosts, such as servers and workstations, sit behind routers and firewalls, and thus receive a filtered set of Internet traffic. The focus of this research is on the malicious traffic that Internet-facing devices are exposed to. This section provides a discussion of the relevant information behind Internet traffic in relation to this project.

Blenn *et al.* (2017) categorises the incoming Internet traffic as *User Requests*, *Network Scans* and *Backscatter*. Figure 2.5 illustrates the three types of traffic received by an Internet-facing device. In a simplified example, if a network administrator was to observe the network logs of the host, all three categories would be combined into a single file. *Backscatter* traffic is discussed further in Section 2.5.1. User requests are everyday traffic created by the user sending requests and receiving responses back. This is legitimate harmless traffic. Network Scans (port scanning, probing) are a form of information gathering used by TAs to determine what services are running on a particular host (de Vivo *et al.*, 1999). These Scans are used by third parties to probe for any active ports on a target host. Although not immediately a threat, these requests are potential precursors to an exploitation (Wustrow *et al.*, 2010). In this research, network scans are referred to as port scanning, included in this are probes, port scans, and vulnerability scans. Port scans are defined as packets sent to the telescopes that exhibit characteristics of known port scanning botnets (Section 2.7.1) or hosts.

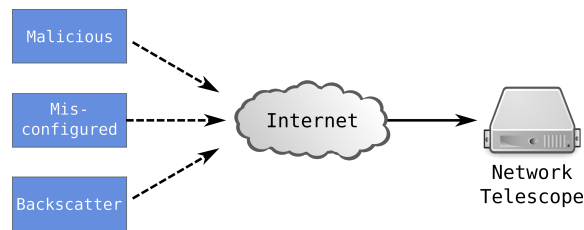


Figure 2.6: Incoming Network Traffic on Network Telescopes after Blenn *et al.* (2017)

2.5.1 Malicious Traffic

When researching traffic on the Internet, user requests are considered to be noise. It is this legitimate traffic which adds additional overheads in terms of processing and analysing. For certain types of research, researchers have developed telescopes to separate user requests from illegitimate traffic such as malicious network scanning, and backscatter (Pang *et al.*, 2004, Moore *et al.*, 2006, Irwin, 2011). Telescopes are discussed further in Section 3.3. With this in mind, Figure 2.6 illustrates the traffic recorded by telescopes as *Backscatter* (Wustrow *et al.*, 2010, Irwin, 2011, Benson *et al.*, 2015, Blenn *et al.*, 2017), *Misconfiguration*, and *Malicious*.

Backscatter is Internet traffic originating from hosts on the Internet who are responding to spoofed IP addresses (Pang *et al.*, 2004, Moore *et al.*, 2006, Wustrow *et al.*, 2010, Irwin, 2011). These hosts are likely the victim of a Denial of Service attack (Section 2.7.2). Backscatter does not present any threat to an organisation receiving the traffic. It does, however, get in the way of traffic analysis. With TCP connections it is significantly easier to determine which packets are backscatter. Wustrow *et al.* (2010) classifies TCP backscatter traffic to be any request with either the TCP SYN+ACK, TCP RST, TCP RST+ACK and TCP ACK flags set. This research project defines backscatter to be any TCP or ICMP passive requests that are recorded in the IBR datasets (see Section 2.5.3).

Misconfiguration according to Wustrow *et al.* (2010) is all traffic other than backscatter and port scans. It is the result of hardware or software errors causing requests to be sent to the telescope IP addresses range (Wustrow *et al.*, 2010, Rossi *et al.*, 2011).

Malicious traffic is traffic left over after backscatter and Misconfiguration traffic is removed. Malicious traffic includes requests and packets linked to port scanning, Internet worms and malware. It is the traffic which likely warrants some response from the host. Research by Irwin (2011), showed that malicious traffic accounts for more than 50% of the traffic recorded on the telescopes. However, results from this research suggest that the number has increased to more than 70%.

2.5.2 Other Traffic

There is recorded traffic that cannot be classified without further analysis into the context of the request. Such traffic includes TCP requests with the ACK or RST flags set. The reason for this is that these scans are either a misconfigured request or they are part of a TCP—ACK or RST request and the TA is performing a port scan.

UDP traffic also falls into this category of traffic. A UDP communication is connectionless and carries little information in the header as to the intent of the request (Blenn *et al.*, 2017). For this reason, it is difficult to determine the full intent of the request. For example, in a paper by (Moore *et al.*, 2003, Chindipha and Irwin, 2017), the authors discuss the SQL Slammer worm which attempts to exploit the 1434/udp port to replicate itself. The SQL Slammer worm does not attempt to make a connection, it simply broadcasts the exploit to vulnerable servers in an attempt to exploit them (Chindipha and Irwin, 2017).

2.5.3 Active and Passive IBR

There are two high-level categories of Internet traffic captured by the telescopes, *Active* and *Passive* (Irwin, 2011). Active traffic are requests that is sent with the expectation that the receiver will send back a response. Passive traffic, are requests that do not expect a response from the sender or are requests that do not conform to the standard communication guidelines and the receiver cannot determine the response to give (Moore *et al.*, 2006). Using the TCP three-way handshake in Figure 2.3 as an example, the first packet sent by Alice (TCP SYN) is an active packet because it is the first packet in the process, and Bob is able to determine the type of response to send back (TCP SYN+ACK). However, if the first packet Bob receives from Alice is a TCP ACK packet, 1) no TCP connection starts with a TCP ACK packet, and 2) Bob is not able to determine the response to this packet and therefore, the packet is classified as a passive packet. A real-world example of an active packet is a TA conducting a port scan on a host. Passive traffic is typically the result of a DoS (Section 2.7.2) attack on a host with a spoofed source IP address, or a malformed request. Backscatter traffic forms part of passive traffic.

Tables 2.2 and 2.3 describe the configuration of TCP and ICMP Passive and Active packets. The configurations are used to separate the IBR data for use in further analysis of the research project (Chapters 3 and 5). The TCP configurations in Table 2.2 define the combination of flags that fall into the category of active or passive traffic. Any combination of flags that do not fall within the values shown in the table are considered to be “other traffic”. This “other traffic” contributes a small proportion of the overall requests (0.01%).

Table 2.2: TCP Packet Configurations (Irwin, 2011)

Active Packets		
Flags	Name	BPF Syntax
NONE	NULL Scan	tcp[tcpflags]=0
FIN	FIN Scan	tcp[tcpflags]=tcp-fin
SYN	SYN Scan	tcp[tcpflags]=tcp-syn
PSH	PSH Scan	tcp[tcpflags]=tcp-psh
URG	URG Scan	tcp[tcpflags]=tcp-urg
URG—PSH—FIN	XMAS Scan	tcp[tcpflags]=tcp-urg & tcp-psh & tcp-fin
PSH—FIN	XMAS Scan	tcp[tcpflags]=tcp-psh & tcp-fin
URH—FIN	XMAS Scan	tcp[tcpflags]=tcp-urg & tcp-fin
URG—PSH	XMAS Scan	tcp[tcpflags]=tcp-urg & tcp-psh
Passive Packets		
Flags		BPF Syntax
SYN—ACK		tcp[tcpflags]=tcp-syn & tcp-ack
RST		tcp[tcpflags]=tcp-rst
RST—ACK		tcp[tcpflags]=tcp-rst & tcp-ack
ACK		tcp[tcpflags]=tcp-ack

The ICMP configurations shown in Table 2.3 describe which category the ICMP packets fall into, based on their type. The ICMP protocol makes use of the request response framework. For example an “ICMP type 8” message elicits an “ICMP type 0” response (Postel *et al.*, 2018). For this reason the ICMP packets can be classified into active and passive packets based on the type and to some extent the code of the message. The ICMP configurations in the table are a list of possible common message types.

2.6 Network Telescopes

Since the early 2000s, researchers and security teams have been using telescopes to monitor unsolicited Internet traffic (Harrop and Armitage, 2005, Irwin, 2011, Benson *et al.*, 2015). The telescopes act as a type of filter, eliminating the legitimate traffic from the illegitimate traffic. Telescopes go by many names, including darknets, greynets, telescopes, blackhole monitors, network sinks and network monitor sensors (Baker *et al.*, 2010, Wustrow *et al.*, 2010). Benson *et al.* (2015) defines a telescope as “... a collection of routed but unused IP address”. Based on this definition, it is possible to consider all traffic seen on these addresses as unsolicited.

Typically telescopes do not respond to any requests, they simply receive and log them (Benson *et al.*, 2015). The general term for the traffic recorded on these telescopes is Internet Background Radiation (IBR) (Pang *et al.*, 2004). IBR includes traffic such as Backscatter, Scans, vulnerabilities, and worms, even Misconfiguration of devices. In this research, the

Table 2.3: ICMP Packet Configurations (Irwin, 2011)

Active Packets			
Type	BPF Syntax		
8	Echo Request (ping)	icmp[icmptype]=icmp-echoreq	
13	Timestamp	icmp[icmptype]=icmp-echoreq	
15	Information Request	icmp[icmptype]=icmp-echoreq	
17	Address Mask Request	icmp[icmptype]=icmp-echoreq	
30	Traceroute	icmp[icmptype]=icmp-echoreq	
33	IPv6 where-are-you	icmp[icmptype]=icmp-echoreq	
35	Mobile Registration Request	icmp[icmptype]=icmp-echoreq	
Passive Packets			
Type	Code	Name	BPF Syntax
0	0	Echo Reply	icmp[icmptype]=icmp-echoreply
3	-	Destination Unreachable	icmp[icmptype]=icmp-unreach
4	0	Source Quench	icmp[icmptype]=icmp-sourcequench
11	-	Time to Live Exceeded	icmp[icmptype]=icmp-timexceed
12	-	Parameter problem	icmp[icmptype]=icmp-paramprob
14	0	Timestamp Reply	icmp[icmptype]=icmp-tstampreply
16	0	Information Reply	icmp[icmptype]=icmp-ireqreply
18	0	Address Mask Reply (ping)	icmp[icmptype]=icmp-maskreply
31	-	Datagram Conversion Error	icmp[icmptype]=icmp-31
34	-	IPv6 I-am-here	icmp[icmptype]=icmp-34
36	-	Mobile Registration Reply	icmp[icmptype]=icmp-36

telescope datasets used for analysis are given the name IBR1 through IBR7 (Section 3.3).

The three examples listed below describe how telescope data has been or can be used to study various types of malicious Internet activity:

Example 1: This example shows how telescope data can be used to study the spread of a malicious worm. Chindipha and Irwin (2017) presented a study of the SQL Slammer worm over 84 months of telescope data. The SQL Slammer worm randomly generates a set of IP addresses to attempt to infect (Symantec, 2007). Given that the IP addresses are randomly generated, the probability of the an IP address being generated that belongs to the telescopes is high. When this occurs, the SQL Slammer worm blindly attempts to upload itself to the 1434/udp port. It is these and other such attempts that are recorded by the telescopes.

Example 2: In this example the telescope data is used to study the occurrence of large scale DDoS (Section 2.7.2) attacks by monitoring passive (Section 2.5.3) IBR. In such attacks the TA spoofs the source IP address of the packets in an attempt to conceal their identify (Mao *et al.*, 2006). These spoofed IP addresses include those being monitored by the telescopes, which result in the victim sending a response to one of the telescope IP addresses and thus being recorded.

Example 3: This final example illustrates how multiple telescopes can be used to correlate the type of traffic being seen in multiple locations. Irwin (2013) presents a study of five telescope sensors monitoring a diverse range of IP addresses. In this study, the author describes how it is possible to detect a set of hosts (IP addresses) that are common across multiple sensors. Within this research, these persistent hosts are studied (Section 5.4) to observe the type of activity they are conducting.

2.7 Malicious Internet Activity

The Internet has seen the rise of many different categories of threats such as unsolicited emails (Spam), viruses, worms, botnets, denial of service (DoS) and open relays. This threat base continues to evolve and expand as new exploitation and attack methods are discovered. A large portion of these threats falls into the category of malware. According to Department of Homeland Security (2009), the term malware includes threats such as Viruses, Worms, Trojans, Spy-ware, and Bots. Malware was traditionally used for the purpose of stealing information from an entity. However, with the progression of technology and the availability of resources, malware has seen a shift towards taking control of a user's computer or web server for black market exploitation (Bayer *et al.*, 2009).

2.7.1 Botnets

Cooke *et al.* (2005) describes a botnet as a collection of hosts such as workstations or servers which have been infected by a piece of malware. The individual infected host is referred to as a bot. After infection, all the bots in the botnet wait dormant listening for further instructions from the bot controller (command and control server) to perform some malicious attack on a target (Abu Rajab *et al.*, 2006, Koliass *et al.*, 2017). Below is a list of botnet uses as presented by Ianelli and Hackworth (2005):

- DoS attacks
- Exploit scanning
- Proxies
- Phishing
- Packet capture

In 2006 Abu Rajab *et al.* (2006) titled “A Multifaceted Approach to Understanding the Botnet Phenomenon” released a study in which they found that botnets account for 27% of malicious traffic on the Internet. The majority of this botnet traffic was attributed to the replication of infected hosts and attack activities. A recent example of a botnet and its capabilities can be seen by looking at the Mirai Botnet (Antonakakis *et al.*, 2017,

Table 2.4: Levels of DOS Attacks

Attack	Description
Network level	Aim to disrupt the access to a server on the Internet. A network attack brings down the whole server
Application level	Aim to prevent the application running on the server from operating as it should. An application attack will only bring down a specific service that the web server is running

Kambourakis *et al.*, 2017, Kolas *et al.*, 2017). Evidence of the existence of the Mirai Botnet started surfacing in August 2016, a month prior to the large attacks it played a role in. The Mirai bot infects IoT devices through exploiting a vulnerability on the Telnet port (Antonakakis *et al.*, 2017). The bot infects a device in stages, the vulnerability is discovered and an attempt is made to gain access to the port 23/tcp or 2323/tcp using default Telnet credentials. If it is successful, the vulnerable host is recorded along with the credentials used to gain access. This information is passed to the source code loader which uploads the botnet source code to the device.

Initially, the bot uses port scanning (Section 2.7.3) to find any vulnerable hosts. Once a host is identified the bot conducts brute-force attacks to gain authorised access. Having successfully infected a host, the bot waits and listens for commands from the Command and Control (C2) server or botmaster (Gu *et al.*, 2008).

2.7.2 Denial of Service Attacks

Denial of Service (DoS) attacks are a type of aggressive intrusion attack focusing on online services. The attack aims to degrade or even cut off the availability of online resources such as routers, servers, entire networks, or Internet-based services (Heron, 2010). To an organisation or ISP who relies on their services always being on, a successful DoS attack can cause serious damage (Coppolino *et al.*, 2017). A DoS attack attempts to overwhelm the victim with computationally heavy tasks or a very large number of useless packets (Tan *et al.*, 2014). This results in the victim being forced out of service for a period of time whilst it handles the requests, or whilst human intervention brings it back online. Table 2.4 lists two levels of DoS attacks with a brief description of each.

Distributed Denial of Service (DDoS) attacks occur in much the same way as a DoS attacks, in that a target host is flooded with unmanageable requests. However, the difference in a DDoS attack is that the requests come from multiple bots (see Section 2.7.1) on the Internet (Park and Lee, 2001). Using a botnet to conduct a DDoS attack proportionally increases the severity of the attack. An example of this is the attack on the consultancy company Brian Kerbs where the Mirai botnet was responsible for sending 620 Gbps of traffic towards the servers (Kolas *et al.*, 2017),.

There have been many variations of DoS and DDoS attacks recorded. The following examples are two DoS or DDoS attacks which can be detected and prevented through the use of TI data.

TCP SYN Flood attacks occur after a TA has successfully infected hosts to join a botnet. The botnet is used to execute a TCP SYN Flood attack (Eddy, 2007). A single instance of the TCP SYN Flood attack occurs when a TA sends a TCP SYN request to a server, the server replies to the attacker with a TCP SYN+ACK response and, because TCP uses a three-way handshake (Figure 2.3) to establish a connection, the server waits for the client to send back a TCP ACK response. If the client does not respond with the expected TCP ACK message the server connection is kept open for a period whilst it waits to receive the TCP ACK message from the client (Adeyinka, 2008). When this attack is conducted from multiple bots (distributed), it causes the victims' resources to be consumed by all the requests from the botnet and thus denying service to legitimate users.

SlowLoris is an attack commonly launched from a client's machine. The SlowLoris tool issues legitimate HTTP requests to the victim server. However, the requests are only partial requests which never get completed. A partial HTTP request is sent, then a period of waiting occurs before more of the request is sent. This results in the server keeping the connection open for long periods of time whilst the request is in progress (Muscat, 2014). The client attempts to create as many requests as possible, thus exhausting the resources of the server with many concurrent connections that never complete.

2.7.3 Port Scanning

When a TA is planning an attack on a system, one of the first steps is to conduct a port scan of the target. A port scan is a technique whereby a TA sends one or more requests to various ports on a target system (Balram and Wiscy, 2008, Barnett and Irwin, 2008, van Heerden *et al.*, 2012) to discover any open and vulnerable ports (Rattan *et al.*, 2017). According to Gadge and Patil (2008) and van Heerden *et al.* (2012), a typical attack goes through a series of phases, these phases are illustrated in Figure 2.7. If a security team is able to prevent an the attack at the *Reconnaissance* phase, it potentially eliminates the more sever stages from being achieved. The *Reconnaissance* phase is primarily performed by botnets.

In 2004 Shah *et al.* (2004) described how it is possible to scan all IPv4 address on the Internet in under 38 hours. However, it is not likely that a TA will scan all these addresses so quickly, it is more likely the TA will be scanning over a longer period of time.

Gadge and Patil (2008) classify port scanning into two types, brute force and stealth scanning. Brute force scanning is an aggressive form of port scanning where each port

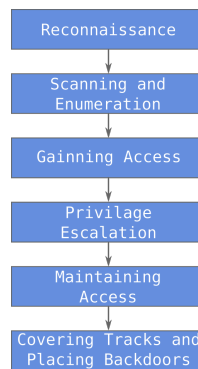


Figure 2.7: Attack Phases after Gadge and Patil (2008)

is scanned one after the other in a specific range. These types of port scans are easily detected because a monitor will detect a flood of TCP SYN requests arriving from a single IP address to multiple ports. Stealth scanners, on the other hand, are more targeted in their approach, they send specific requests with different flag sets, and the response to these requests will tell the TA whether the ports are open or not. Some examples of different stealth scanning requests described by Gadge and Patil (2008), Zhang *et al.* (2008) are listed below:

SYN Scan aims to discover whether a port is open or not by sending a TCP SYN packet (Bano *et al.*, 2018). Although this is the start of the TCP three-way handshake (Figure 2.3), the TA does not intend to complete the handshake, the TA is simply concerned with the response. If the response is a TCP SYN+ACK, it tells the TA that the port is open on that specific IP address and can potentially be exploited (NMAP, 2008). However, if the response is a TCP RST response, it tells the TA that the port is closed and cannot be exploited. Lastly, if the TA does not get a response back, it tells the TA that there is a firewall filtering traffic.

ACK Scan is when a TA actor sends a TCP ACK request to the receiver to determine whether there is a firewall on the current IP address (NMAP, 2008). This request might be sent if the SYN Scan (above) does not get a TCP SYN+ACK or TCP RST response. The only information that can be determined from this type of scan is whether there is a firewall on the other end, and whether that firewall is stateful or not (Sanz *et al.*, 2017). This is due to the fact that if a TCP ACK request is received which is not part of an ongoing connection setup, the response will be TCP RST, regardless of whether the port is open or closed. If there is no response to this request, it states that there is a stateful firewall on the receiver's end which is filtering traffic.

FIN, NULL, and XMAS Scans attempt to discover if there is an open port on the receiving host (NMAP, 2008). A FIN Scan sends a TCP FIN packet, a NULL scan sends a packet with no flags set, and an XMAS scan sends a TCP FIN+PSH+URG packet. If there is an open port, there will be no response. However, if the port is closed, the

response will be a TCP RST (De La Peña Montero and Hariri, 2017).

A comprehensive survey of port scanning types can be found in (Bou-Harb *et al.*, 2014).

2.7.4 Proxy Servers and Open Relays

Proxy servers are used by organisations to reduce network loads (Post Cast Server, 2017b). These proxy servers are often configured with little regard for security. TAs attempt to seek out these vulnerable servers to exploit and use for further malicious activities. More specifically, spammers seek out Open Relays to exploit and use as a means to disguise the source of spam emails (Krawetz, 2004). As such, an Open Relay can be defined as an email server that will accept and transfer emails on behalf of any user or third party (Post Cast Server, 2017a). These Open Relays are attractive to the spammers because there is no way to trace the emails back to the original sender. Furthermore, with the large availability of Open Relays, if one is taken down the spammer is able to exploit another and continue sending the spam (Jung and Sit, 2004).

2.7.5 IP Spoofing

IP spoofing is a technique used by TAs to conceal their identity (Senie and Ferguson, 1998, Doss *et al.*, 2017). When a malicious request is crafted, the TA will randomly generate an IP address to insert as the source of the request (Moore *et al.*, 2006). A TA can spoof an IP address in one of two ways:

1. The IP address can be chosen at random from all 2^{32} IP addresses available (Mao *et al.*, 2006).
2. The IP address can be chosen from a list of predefined IP addresses (Blenn *et al.*, 2017).

The first method is much harder to mitigate than the second. If the list of IP addresses to protect against is known, it would be easier to filter traffic based on IP addresses (Blenn *et al.*, 2017).

Before the attack is sent, the TA will spoof the source IP address of the packets being sent to the server (Blenn *et al.*, 2017). When the victim receives the packet they will attempt to send a reply back, but instead of sending it to the TA, the response is sent to the spoofed IP address.

Although spoofing an IP address seems a sure way to hide the identity of a TA, research by (Mao *et al.*, 2006) suggests that this technique is rarely used. This is because TAs avoid sending DoS attacks themselves, preferring to use a botnet, which is the more common

approach to launch attacks. The TA simply issues the attack command to the botnet, and the bots do the rest of the work, with little worry for the reputation of the infected host (Blenn *et al.*, 2017). For this reason, mitigation techniques such as IP filtering will not work to eliminate an attack that makes use of spoofed IP addresses because the network device will filter spoofed IP addresses and not TAs IP address.

2.8 Domain Name System Block List (DNSBL)

To understand how a Domain Name System Block List (DNSBL) works, one must have a basic understanding of how DNS (Domain Name Service) works. All devices connected to the Internet have a unique IP address, for example, *203.0.113.1*. This IP address is used to uniquely identify that specific device. When two devices wish to communicate over a network, they will use IP addresses to communicate. However, humans find it easier to identify a device by its hostname rather than its IP address. The DNS protocol provides hostnames to IP address resolution (Verisign, 2017).

A DNS server is a dedicated server on the Internet which keeps a record of known IP address-to-hostname mappings. On receiving a query, the DNS server will search its database of records to determine if it can provide the IP address of the hostname in question. If the DNS server does have a record of the hostname, it responds to the client with the IP address in the record. However, if the DNS server does not have a record of the hostname it can either send the DNS query onto another DNS server or tell the client the hostname does not exist.

An example DNS query can be seen in Listing 2.1. A DNS query is sent to the local DNS server asking for the IP address of the *example.com*² website (Lines 3 and 4). The local DNS server responds with an answer which is a resource record containing the IP address (Lines 6 and 7).

```

1 $ dig example_domain.com
2 ...
3 ;; QUESTION SECTION:
4 ;example_domain.com.          IN      A
5
6 ;; ANSWER SECTION:
7 example_domain.com.          4191    IN      A      203.0.113.1
8 ...

```

Listing 2.1: DNS Question and Answer (Caudle, 2015)

A DNS server is able to send various types of resource records back as the answer. Line 7 in Listing 2.1 shows the common type of resource record, an *A* record (Caudle, 2015). The two most common resource records are *A* and *TXT*. The *A* record returns the IP address

²www.example.com

that matches the given hostname, and the *TXT* record returns arbitrary text recorded for the hostname. When an answer is sent it can include one or more resource records, each containing information specific to the type of resource record. However, if the DNS server does not have a record of the IP address being queried it can choose to send back a *NXDOMAIN* answer to the client which states that the DNS query was not able to be resolved and the domain is non-existent (Vincenti, 2010).

In addition, the DNS protocol also allows reverse DNS queries to be performed (Jung *et al.*, 2002). This query allows an IP address to be mapped to a domain name. Listing 2.2 illustrates a reverse DNS lookup to find the domain name of an IP address (*203.0.113.1*). Reverse DNS is used in Section 5.6 to determine the hostnames of the IP addresses discovered during the data analysis stages.

```

1 $ dig -x 8.8.8.8 ...
2 ;; QUESTION SECTION:
3 ;1.113.0.203.in-addr.arpa.      IN      PTR
4
5 ;; ANSWER SECTION:
6 1.113.0.203.in-addr.arpa.      26862   IN      PTR      www.example_domain.com
7 ...

```

Listing 2.2: Reverse DNS Question and Answer

A DNSBL makes use of the already implemented DNS protocol to provide an anti-spam filtering service to mail servers and email clients. Contrary to a DNS server, DNSBLs keep a list of malicious IP addresses or hostnames of known spam or malware sources (Iverson, 2007). A DNSBL is not distributed to consumers, rather the consumers issue a DNS query to the DNSBL to determine if an IP address or domain is malicious (The SpamHaus Project, 2000). The system developed in this research implemented a DNSBL module (Section 6.2.3) to allow consumers to query the OSINT database. Vincenti (2010) describes the steps taken when a consumer issues a DNS query to a DNSBL:

- The consumer takes an IP address, for example, *203.0.113.100* and reverses the order of the octets resulting in *100.113.0.203*.
- The consumer then appends the DNSBL domain name to the IP address, for example, *100.113.0.203.dns.example.com*.
- The consumer issues a DNS query for this name in the DNS with an *A* record.
- The consumer receives either an IP address or a *NXDOMAIN* response.

The consumer is able to tell the status of the IP address based on the response received from the DNSBL. If the consumer receives an IP address, this shows that the DNSBL knows about the IP address and therefore can be considered a source of spam according to the DNSBL's data. However, if the response is *NXDOMAIN*, this signals to the client that the IP address does not exist in the DNSBL and can be treated as normal mail.

2.9 Threat Intelligence (TI)

A threat to a computer system is defined by Newman (2006) as “...any potential occurrence, either accidental or malicious, that can have an undesirable effect on the assets and resources of an organization”. Threats target vulnerabilities in systems and resources. A vulnerability is the absence or weakness of a safeguard in an asset or resource (Newman, 2006). When a vulnerability is exploited, a chain of events can normally be traced in much the same way as Figure 2.7:

- The vulnerability which exists within the system is discovered by the TA.
- The vulnerability is exploited in the actual attack.
- The result of the attack can pose a threat against a particular resource.

Intelligence can be defined as “the gathering of distributed information ...”, “the capacity for understanding” or the “ability to perceive and comprehend meaning” (Concise Oxford Dictionary, 1990). With this understanding, it is possible to define TI as the information gathered on malicious or undesirable cyber-criminal activities such as IP addresses, malicious domains, malicious networks via Autonomous Systems (ASes) and malicious botnets which can be used to inform decisions regarding a subject’s response to a threat (Hawkinson and Bates, 1996, McMillan, 2013, Tounsi and Rais, 2017).

TI is not only about understanding the threats on the Internet, it is also about understanding how these threats relate to the organisation (Bromiley, 2016). For example, an e-commerce organisation would be more interested in threats such as DoS attacks which can potentially bring down their service, rather than spam emails.

Despite the prevalence of TI in this era, there are still misconceptions on what TI is. Understanding what TI is not helps to understand what good TI is and how to provide it. The list below presents what TI is not:

- a list of IP addresses or domains a TA used at some point in time.
- outdated information on TAs.
- irrelevant information that an organisation obtains but ignores.

Caltagirone (2015a), Sillaber *et al.* (2016) propose a framework for evaluating the quality of TI. The framework proposes that good quality TI must be Complete, Accurate, Relevant and Timely (CART). The TI provider must consider this when creating TI to publish to consumers. TI providers must ensure the TI they provide is of the highest quality. Low-quality TI is not only costly to organisations, it also affects the perceived confidence in the TI provider (Sillaber *et al.*, 2016). Furthermore, if the TI provider is not trusted, consumers will not be willing to use their data (Tounsi and Rais, 2017).

2.9.1 Sources of Threat Intelligence

The source used to obtain TI is dependent on the organisational requirements and the type of TI needed. TI sources are either internal or external to the organisation (Bromiley, 2016).

- **Internal:** these sources are from within the organisation, for example, telescopes (see Section 3.3), firewalls, intrusion detection systems, and historical IOCs. These sources of TI provide information that is relevant to the organisation.
- **External:** these sources of TI are from outside the organisation. This is the most common form of TI. The advantage of using external sources is that the TI may provide an early warning of potential incoming attacks or exploitations (Bromiley, 2016). This is mainly relevant to threats which attempt to find any exploitable host on the Internet such as Mirai, Mirai like IoT botnets (Antonakakis *et al.*, 2017) and Conficker (Irwin, 2012).

To gain the most benefit from TI, the suggested approach is to combine both internal and external sources into a single TI feed. A TI feed is a collection of multiple sources of TI (Nichols, 2016, Recorded Future, 2017a), usually containing the same type of data. For example, IP addresses, domains, URLs and IOCs. A TI feed is maintained by a *TI provider*. A *TI provider* is an organisation or individual who organises multiple TI sources into one or more datasets, which are typically made available as downloadable resources in formats such as JSON³ or CSV⁴ files, APIs⁵, and TI reports. Appendix A lists the TI feeds used in this research project, mainly IP address feeds.

2.9.2 Open Source Threat Intelligence (OSINT)

As TI becomes more popular for organisational protection, research, and individual use, more and more TI providers are emerging with their own set of TI. There are two ways TI is made available to consumers, paid subscription or free-to-use services. Paid sources are typically published through TIPs (Section 2.10) which are commonly subscription based platforms (Sauerwein *et al.*, 2017).

Free-to-use services, also known as open source, are made available to anyone wishing to consume the data. This type of TI has been termed Open Source Threat Intelligence (OSINT). OSINT is described by Infosec Institute (2018) as being "... a collection of data from public sources used in an intelligence context". OSINT is typically gathered from publicly available TI sources such as forums, telescopes, databases, and other OSINT feeds. This research uses OSINT which is publicly downloadable from Internet feeds.

³Java Script Object Notation

⁴Comma Separated Values

⁵Application Programming Interface

Appendix A discusses the OSINT feeds collected during the 12-month research period, and Section 3.2 discusses the OSINT data itself.

2.10 Threat Intelligence Platforms (TIPs)

TIPs provide a single location through which consumers are able to gain access to TI from many different sources and feeds. Recorded Future (2017a) defines a TIP as, “... a piece of software typically developed by a security vendor, which organises one or more feeds into a single stream of threat intelligence”. TIPs provide an element of trust between the TI provider and consumer, which according to Sillaber *et al.* (2016) is crucial to the consumer utilising the TI. The revenue generated from subscriptions allows TIPs to provide quality TI that has undergone advanced analysis and processing (Robb, 2017). As such, the consumers are able to trust the TI published from the TIP.

There are very few TIPs that are open source, the majority of TIPs are paid services (Sauerwein *et al.*, 2017). When it comes to OSINT, most of the providers simply collect and disseminate the data. There is little effort put into the analysis and enrichment of the TI. Out of the top eight TIPs listed by Robb (2017), two have a free-to-use tier, IBM X-Force Exchange⁶ provides free usage up to 5000 records a month, and AlienVault provides the Open Threat Exchange platform (Section 2.10.1).

2.10.1 AlienVault Open Threat Exchange

AlienVault provides the Open Threat Exchange (OTX)⁷ platform to security teams and researchers. OTX provides users with a way to investigate and collaborate on new and potential threats (AlienVault, 2018). OTX is an open source platform which benefits from community collaboration and involvement. The data is collected from contributors all over the world into a single accessible set of feeds. For consumers of the platform, OTX provides TI in two forms, *Pulses* and *IP Reputation*. Pulses are collections of indicators of compromise (IOCs), created by the community. *IP Reputations* are lists of IP addresses considered to be linked to TAs on the Internet. Users are able to customize their own requirements based on the information they are looking for. This research uses the *IP Reputation* lists as part of the OSINT datasets.

⁶<https://exchange.xforce.ibmcloud.com/>

⁷<https://otx.alienvault.com/>

Table 2.5: BrightCloud Services

Service	Description
Web Classification and Reputation	This service categorises URLs into different categories based on their intended purpose. The reputation service provides customers with a security check for websites and URLs visited. The services allows websites to be evaluated by the customer to determine whether it poses a threat to the organisation (BrightCloud, 2005).
TI for IoT GateWays	This service offers organisations a way to secure their IoT devices against current threats being faced (Webroot, 2018c). The service provides consumers with real-time TI on malicious IP addresses and websites which target IoT devices. Webroot is constantly monitoring the Internet and collecting TI for the TIP.
IP Reputation Service	This service provides timely access to IP address related TI. It gives consumers access to Webroot's data on malicious IP addresses they have discovered and confirmed (Webroot, 2018a). Part of this service is the ranking of IP addresses according to their level of risk. An IP address can fall into one of five categories, trustworthy, low risk, moderate risk, suspicious or high risk.

2.10.2 Webroot Threat Intelligence Platform

Webroot⁸ identified that traditional security mechanisms, such as static and list-based protection mechanisms, are not able to keep up with the rate at which online threats are growing (Webroot, 2018b). To address this issue, Webroot provides a platform which integrates closely with their subscription based BrightCloud⁹ TI services.

The Webroot platform provides quality TI to its consumers and, with the help of the BrightCloud services, allows organisations to implement top-quality threat detection and prevention for their resources. Table 2.5 lists some of the services provided by BrightCloud (BrightCloud, 2005).

2.10.3 GOSINT

GOSINT¹⁰ is an OSINT platform designed to collect and process multiple sources of TI (Bollinger, 2017). The purpose of GOSINT is to provide security teams with a means of collecting TI and having it converted into a set of standardised data. GOSINT was created as a solution to vast amounts of structured and unstructured TI being available on the Internet with no easy way to collect and filter it. Additionally, GOSINT provides security teams with a means of collecting OSINT and analysing it to determine if it is suitable to use. The platform does not come predefined with OSINT sources, it relies on the user going out and finding sources to add to the system. Once added to the system, GOSINT automatically downloads the data at user-defined intervals. The downloaded data is then standardised and made available to the user.

⁸<https://www.webroot.com/us/en/business>

⁹<https://www.brightcloud.com/>

¹⁰<https://github.com/ciscocsirt/GOSINT>

Table 2.6: Collective Threat Intelligence Gathering System

Process	Description
Collection	Data is collected from predefined sources on the Internet and stored in database as raw intelligence data.
Aggregation	The collected data is parsed to identify the format it is in. This allows the system to convert all the data into a single format which is stored in a single location.
Normalisation	After the data is aggregated into a single format, it is normalised to reduce duplicate entries.
Storage	The aggregated, normalised data is stored in the database along with information which helps identify the source it was pulled from.
Enrichment	IP addresses are enriched by doing a reverse DNS lookup query. Domain names are enriched by doing a normal DNS lookup query.
Filtering and Validation	The data is filtered again to reduce the duplicates and false positives. It is also validated to confirm whether the threat is legitimate.
Scored	After filtering and validation, the threats are scored. Scoring quantifies the threat posed by the records.
Formatted and Distributed	Before the TI is distributed to consumers, it is formatted according to the consumer's desire. Distribution can happen on a number of pathways such as SFTP and Dropbox.

2.10.4 Collective Threat Intelligence Gathering System

Magee *et al.* (2014) describes a TI gathering system which aggregates multiple OSINT from various different sources. The system collects, aggregates, filters and enriches OSINT before making it available to consumers. The system focuses on OSINT sources which report on malicious IP addresses, domains, URLs, and emails. Table 2.6 describes the key components of the system.

2.10.5 Recorded Future

Recorded Future is a TIP available to consumers as a pay-to-use service (Recorded Future, 2017a), with the goal of delivering true contextualised TI. Recorded Future is committed to providing accurate and relevant TI from thousands of sources. Since the platform is pay-to-use, Recorded Future is capable of providing advanced TI gathering and discovery techniques. Furthermore, the TI provided has been verified and validated by complex processes and procedures. Recorded Future allows customers to create customised threat alerts based on their organisational requirements (Recorded Future, 2017b).

One of the claims made by Recorded Future is that they provide TI data which cannot be obtained from other sources. To achieve this, Recorded Future collects data from a large number of sources such as forums, security blogs, code repositories and more (Recorded Future, 2017b). A paper title “Acing the IOC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber ThreatIntelligence” by Liao *et al.* (2016) claims that Recorded Future gathers TI from over 650 000 Internet sources in seven different languages.

2.11 Chapter Summary

This chapter discussed the relevant background information required to understand this research. The chapter started off with **Section 2.1** presenting a discussion around routing and how Internet routers are able to move a packet from source to destination. Within this section is a discussion of the RIB (Section 2.1.1) and the FIB (Section 2.1.2) which are two tables used to help the router route and forward each packet to its destination. **Section 2.2** presented a discussion of TCP and the process of two hosts sending and receiving packets. **Section 2.3** discussed ICMP. The concept of BGP routing was discussed in **Section 2.4**. **Section 2.5** introduced the focus area of this research with an description of Internet traffic and how it can be classified. Within this section is a discussion of malicious and none malicious Internet traffic detected on the telescopes. In addition, the section describes active and passive Internet traffic (Section 2.5.3) and its relevance to the research. Following on from this was **Section 2.6** which sought to describe the concept of telescopes and their purpose. **Section 2.7** discusses the type of Internet activity recorded by the telescopes. **Section 2.8** described DNSBLs and how they operate. **Section 2.9** discussed TI and how it has been used and can be used to prevent Internet threats. Included in this section was a discussion of OSINT. This chapter concludes with **Section 2.10** discussing TIPs and how TI providers make TI available to consumers. The next chapter presents a quantitative analysis of the data sources and data sets used in the the analysis of this research in Chapter 5.

3

Data

THIS chapter describes the datasets used for analysis and testing in the remainder of this research project. The primary source of data was a collection of OSINT feeds. The secondary source was telescope data established through previous research by (Irwin, 2011). Since Irwin’s research, the telescope project has been extended and maintained which provided a reliable source of Internet Background Radiation (IBR) data for this research. This chapter focuses on quantifying the two datasets to understand the properties and composition of each. Understanding the properties and composition of the data aids in the selection of appropriate analysis tools and techniques used in Chapter 5.

The remainder of the chapter is structured as follows; **Section 3.1** details the two dataset groups used in this research, namely OSINT and telescope data. **Section 3.2** describes the primary source of data. Included in this section is a discussion of how the data was collected and processed. Additionally, the section quantifies the data to give the reader a better understanding of the information. **Section 3.3** discusses the telescope sources used, as well as how the data was processed. Additionally, the section describes the characteristics of the data including volume and composition.

3.1 Data Sources

The data used in this research was collected over a period of 12-months from September 2017 through to August 2018. The primary dataset was a collection of OSINT feeds collected from various sources on the Internet. The secondary dataset was IBR data taken from a set of telescope infrastructure previously developed and maintained.

Each OSINT feed was evaluated by researching existing projects to discover the most commonly used, reliable, and up-to-date sources. A GitHub repository by *hslatman*, “A curated list of Awesome Threat Intelligence resources”¹, provided most of the links used to access the OSINT feeds. However, the OSINT feeds’ URLs needed to be manually located. A full list of URLs can be found in Table A.1. The telescope data was chosen to allow a comparison between OSINT data and real-world Internet traffic.

3.2 Open Source Threat Intelligence

There are many sources of OSINT that could have been used in this project. However, the sources discussed in this section were regarded as being the most reliable as they appeared on a number of different websites, articles, and posts by other work in the field. Table 3.1 lists each of the 18 OSINT sources used in this research, along with the number of feeds downloaded from each. Each of the OSINT feeds are downloadable as a text file, with the most common being a CSV file. A list of each OSINT feed along with the URL where the data is held can be found in Table A.1, in Appendix A.

3.2.1 Data Collection and Preprocessing

Each of the OSINT feeds described above were downloaded daily at midnight. Although some of the feeds changed more frequently, it was decided to collect all the feeds at the same interval to allow for comparability.

The ability for anyone to release his or her own version of OSINT has resulted in TI start-ups proliferating (Caltagirone, 2015b). The downside of these startups is that there is often little regard for standards and quality, resulting in inconsistencies in how the data is published and lack of error checking. These issues are a result of a combination of the problems identified in Chapter 1. As such, there is a need for pre-processing of the data to get it ready for processing.

The collection and preprocessing is an important part of the research as a whole. It

¹<https://github.com/hslatman/awesome-threat-intelligence>

Table 3.1: Initial OSINT Sources

Source	Feeds
Alexa	1
AlienVault	1
Autoshun	1
Bambenek	1
Binarydefense	2
Bruteforceblocker	1
CINS	1
Feodo	2
SANS ICS	4
Malware Domain List	3
Proofpoint	1
PSBL	1
Ransomware	3
Spamhaus	4
SSL Blocklist	3
Statvoo	1
Umbrella	2
Zeus	5
Total	37

emphasises the additional work required to combine multiple OSINT feeds into a single dataset which can be used by an organisation. Section 4.4 gives a detailed discussion of how the data was collected, pre-processed and made available for use later in the research.

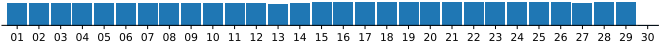
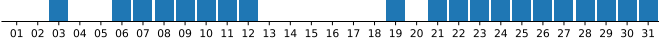
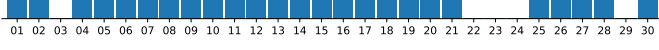
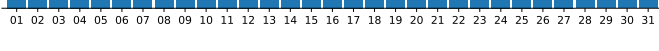
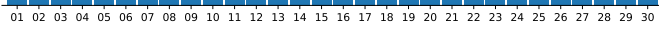
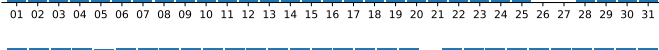
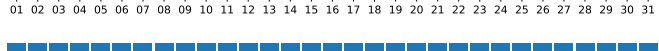
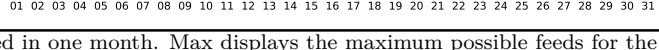
3.2.2 Dataset Overview

This section provides an overview of the characteristics of the OSINT collected over the 12-month period under study. A total of 37 OSINT feeds were collected from 18 different sources. Table 3.2 provides a summary of the volume of data collected for each month during that period. The table headings are as follows:

It is important to note the following points when interpreting Table 3.2:

- Although the data collected had periods of less than 100% collection, it did not impact the overall analysis of this research since the primary focus was the OSINT collected on a month-to-month basis.
- Only January 2018 and August 2018 had 100% collection. Due to the nature of OSINT, resources are not available 100% of the time, which is why only two of the 12-months achieved 100% collection. In addition to the resources not always being available, the collection process encountered a number of different errors which resulted in less than 100% collection of data. The more common errors can be seen in Table 3.3. The error of most importance is the HTTP/304 which shows how

Table 3.2: Summary of OSINT Feeds Collected

Month	Data	Max	%	Daily Feeds Collected
09/17	940	1020	92.16	
10/17	723	1147	63.03	
11/17	919	1110	82.79	
12/17	1138	1147	99.22	
01/18	1147	1147	100.00	
02/18	1034	1036	99.81	
03/18	1138	1147	99.22	
04/18	1104	1110	99.46	
05/18	1069	1147	93.2	
06/18	1105	1110	99.55	
07/18	1146	1147	99.91	
08/18	1147	1147	100.00	

Data displays the total number of feeds collected in one month. Max displays the maximum possible feeds for the month. For example, September 2017 has 30 days in total, with 37 possible feeds, the maximum possible feeds are $30 * 37 = 1110$.

% displays the total number of feeds collected for the month as a percentage of the maximum possible number. Daily Feeds Collected illustrates the number of feeds collected per day as a percentage of the total number of possible feeds in a month.

often a specific OSINT feed changes its information. The OSINT feed that had this error was Autoshun (Appendix A.5).

After each error was noticed, the scripts were adjusted to rectify the problem and prevent the same issues happening in the future. After the errors were identified and corrected, the data collection improved and became more consistent. Despite correcting the scripts, there were some errors which could not be corrected resulting in missing the data seen in Table 3.2.

The following points can be made to explain some of the periods of missing data:

- September 2017 did not have all 37 OSINT feeds added to the collection script. The total that month was only 34 because the research was still in the process of identifying and finding OSINT feeds to add to the collection process.
- Throughout the collection period, two sources gave continuous problems with down-

Table 3.3: OSINT Collection HTTP Errors

Code	Error	Freq	Description
104	Connection reset by peer.	3	This error results when there is a problem with the download script or the network connection. Some sources time out after a predetermined time to protect against potential DoS attacks.
110	Connection time out.	4	Similar to the 104 error code, this error occurs when the request for the resource times out whilst trying to download or request the resource.
304	Not Modified.	12	This error states that the resource has not changed in the last x amount of time (Fielding and Rescheke, 2014).
404	Not Found.	7	This error states that the server hosting the resource could not find the resource being requested (Fielding and Rescheke, 2014).
502	Bad Gateway	2	This error states that the proxy or gateway received an invalid response from the server hosting the resource (Fielding and Rescheke, 2014).
503	Service Not Available	7	The error signals that the server holding the resource can not handle the request being received (Fielding and Rescheke, 2014).
521	Origin Down	2	This error is signals that the content delivery service is unable to connect the server hosting the resource (Zatlyn, 2018).
522	Origin connection time out.	3	This error occurs when a proxy server cannot reach the origin server to retrieve the resource being requested (Zatlyn, 2018).

The Freq column shows the number of times the specific error was seen over the 12-month period.

loading the data, Malware Domain List (Appendix A.11) as a whole and Zeus Bad Domain (Appendix A.2).

- Between October and early November 2017 saw periods of 100% of the data missing. Most notable where six days in the middle of October. This missing data can be attributed to power outages. The power outages resulted in the computer used for collection not being on during the collection times. In November 2017 the scripts were moved onto a server connected to generator backups to eliminate the power factor.
- Late November 2017 saw consecutive errors as a result of the collection scripts not executing correctly after major changes had been made. These changes included adding feeds to the collection scripts which required additional processing to download because the data was inconsistent and abnormal. As such, the collection scripts were not able to execute fully and crashed, resulting in the less than 100% collection in late November 2017.
- The final noticeable lack of data was in May 2018 over a weekend. During this time, there was an incomplete collection process due to the collection scripts timing out whilst waiting for certain feeds to download due to a cascade of upstream network errors.

Although the research was focused on OSINT lists containing IPv4 addresses, there were other types of OSINT collected, including domains, AS numbers, URLs, etc. Table 3.4 details the types of the OSINT lists collected. The data analysis in Chapter 5 was limited to using OSINT lists that publish IPv4 addresses. As such, the OSINT datasets were reduced to those containing IPv4 addresses.

Table 3.4: OSINT Feed Types

Feed Type	Feeds
AS Numbers	1
IPv4	19
IPv6	1
Domain	11
SSL Fingerprint	1
TLD (Top Level Domains)	1
URL	4
Total	37

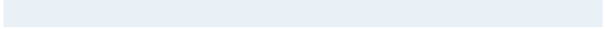
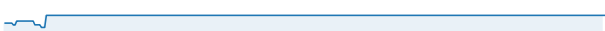
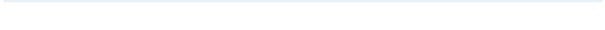
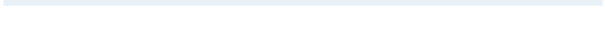
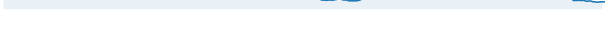
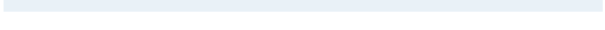
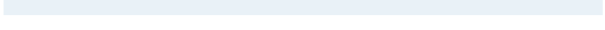
With the reduced OSINT datasets, further quantification was undertaken on the data to illustrate the volume of data downloaded. Table 3.5 illustrates the volume of data trend over the 12-month period. Included in this table is the average volume of data contained in the OSINT dataset on a monthly basis. Each graph illustrates the trend in the volume of data published by the different OSINT feeds. It is important to note that the graphs do not share the same y axis scale. The trends seen in Table 3.5 illustrate how different sources publish their data. For example, the trend seen for Binarydefence Banlist hints towards the lists being purged every so often causing the regular, sudden drops in volume. Conversely, the trends seen in AlienVault Reputation hint towards the lists having IP addresses added when they are identified and removed after a period of inactivity.

A curve such as that seen with AlienVault Reputation may suggest that the IP addresses published on the list are more general IP addresses covering a wider range of TA (supported by Section 5.1). This is evident in the shape of the graph which shows the volume of data steadily increasing and then staying high with only small reductions. Spikes in data are not present in OSINT feeds such as Autoshun IP, CINS CI Army, Malware Domain List Hosts, and Malware Domain List Active, likely due to the fact that the datasets are limited to a specific number of IP addresses so there is less variation in volume. On the other hand, OSINT feeds that do not limit their data, such as AlienVault Reputation see their volume increasing and decreasing over time.

3.3 Network Telescope Sensors

As discussed in Section 2.6, telescopes provide insight into the potentially malicious Internet traffic seen on a daily basis. The data collected from the telescopes used in this research are a subset of the data collected by Irwin (2011). Since late 2005 the Security and Networks Research Group (SNRG) has maintained a set of telescopes hosted on the Rhodes University network in a DMZ (Demilitarized Zone) between the Internet and the

Table 3.5: OSINT Volume of Data

IP Feeds	Avg	Data Volume Trend
AlienVault Reputation	1 676 896	
Autoshun IP	14 250	
Bambenek IP Master	13 429	
Binarydefense Banlist	133 126	
Binarydefense TOR	17 7767	
Bruteforceblocker BL	34 428	
CINS CI Army	422 605	
Feodo IP	36 276	
ISC IP	472	
MDL Active	26 743	
Proofpoint IP	60 783	
Ransomware IP	8 802	
Spamhaus DROP	23 564	
Spamhaus EDROP	2 535	
SSLBL IP	2 898	
SSLBL Aggressive	108 278	
Zeus Bad IP	3 086	
Zeus IP	3 300	

campus network (Irwin, 2011, 2013, Irwin and Nkhumeleni, 2015). Figure 3.1 illustrates the structure of the network and where the telescopes are located in relation to the other servers. The *Telescope Monitor* is a server configured to monitor a range of IPv4 /24 netblocks for unsolicited Internet traffic (Irwin, 2011). The *Telescope Monitor* is placed in the DMZ on the public side of the university network, ensuring that the *Campus Firewall* does not block traffic destined for the monitored IP addresses (Cheswick *et al.*, 1998).

The five telescopes used in this project are physically located on a single server as shown in Figure 3.1. However, the IP addresses are distributed across three distinct top-level

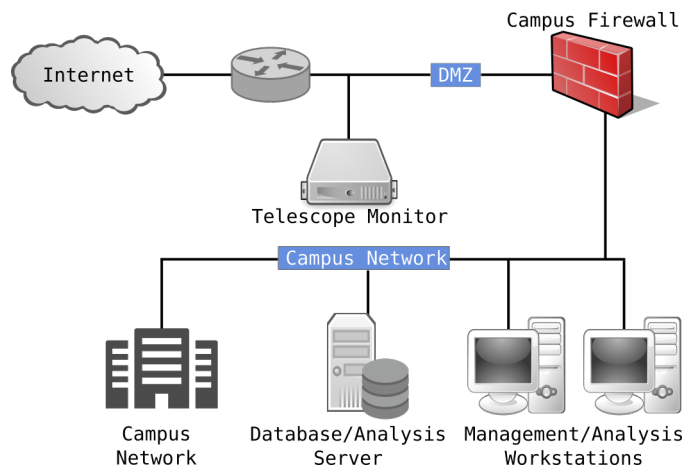


Figure 3.1: Network Telescope Infrastructure (Irwin *et al.*, 2007)

IPv4 network address blocks (netblocks). All telescopes are contained within the TENET² network. RUSCOPE1 is contained in a 146/8 network, RUSCOPE2 in a 155/8 network, and RUSCOPE3, RUSCOPE4, and RUSCOPE5 are contained within a 196/8 block of addresses in two different /16 netblocks.

3.3.1 Data Collection and Processing

The *Telescope Monitor* keeps a record of each request it receives destined for any of the IP addresses being monitored. These requests are saved to the *Database/Analysis Server* which carries out a number of different steps that are beyond the scope of this research. However, a feature of the *Database/Analysis Server* is that it is capable of storing the requests in a PCAP file. The PCAP file contains the full packet data of requests received (Garcia, 2000) which are made available to researchers within the university. The data in the *Database/Analysis Server*, commonly referred to as Internet Background Radiation (IBR) (Pang *et al.*, 2004), allows researchers access to unwanted, potentially malicious Internet traffic that is free from legitimate day-to-day requests.

3.3.2 Dataset Overview

This section provides an overview of the data collected from the telescopes that are used in this research. Given that the telescopes are continuously collecting data for other research, it was straightforward to download data during the same 12-month period as the OSINT data. The telescope data is 100% complete for every day of the month and therefore did not require consideration and processing to account for missing data. Table 3.6 presents information on the data recorded by the telescopes throughout the data collection period,

²<http://www.tenet.ac.za>

Table 3.6: Network Telescope Data

(a) Overall Packets				
Telescope	Minimum	Maximum	Total	Monthly Data
RUSCOPE1	11 789 601	20 644 146	191 365 032	
RUSCOPE2	12 031 381	37 175 035	237 612 324	
RUSCOPE3	13 276 828	22 651 961	210 892 601	
RUSCOPE4	13 535 861	22 272 803	210 785 844	
RUSCOPE5	13 292 053	23 038 760	207 250 360	
(b) Unique IP Addresses				
Telescope	Minimum	Maximum	Total	Monthly Data
RUSCOPE1	863 401	1 409 087	10 191 226	
RUSCOPE2	828 519	1 392 472	10 033 956	
RUSCOPE3	859 037	1 470 114	10 527 310	
RUSCOPE4	860 108	1 455 642	10 317 470	
RUSCOPE5	856 413	1 456 098	10 301 633	

Minimum and Maximum represent the lowest and highest numbers, respectively, of packets/unique IP addresses recorded on the telescope for a single month. Total Packets is the overall total number of packets/unique IP addresses. Monthly Data illustrates the volume of packets/unique IP Addresses. The y axis share the same scale.

with Table 3.6a describing the total volume of packets and Table 3.6b describing the volume of unique IP addresses recorded over the data collection period.

On further observation of the graphs in Tables 3.6a and 3.6b it can be seen that the unique IP addresses very similar trends, however, the overall packets recorded do not. The graphs do show that when one telescope has an increase in requests, the other telescopes experience a similar increase. Additionally, the overall packets and unique IP addresses do not always follow the same trend. When the total requests increase, the assumption is that unique IP addresses would also increase. This is not true for all cases, it is the general trend seen in telescope data. There are some months when the overall packets increase while the unique IP addresses decrease. One possible reason for this are worms and botnets such as the Conficker³ attempting to propagate by finding other vulnerable hosts (Irwin, 2012). Further analysis into the way the data changes is discussed in Section 5.2.

³Conficker has largely disappeared, without looking at the request data there is no way to say that this is the Conficker worm. However, the requests are targeting the same port 445/TCP

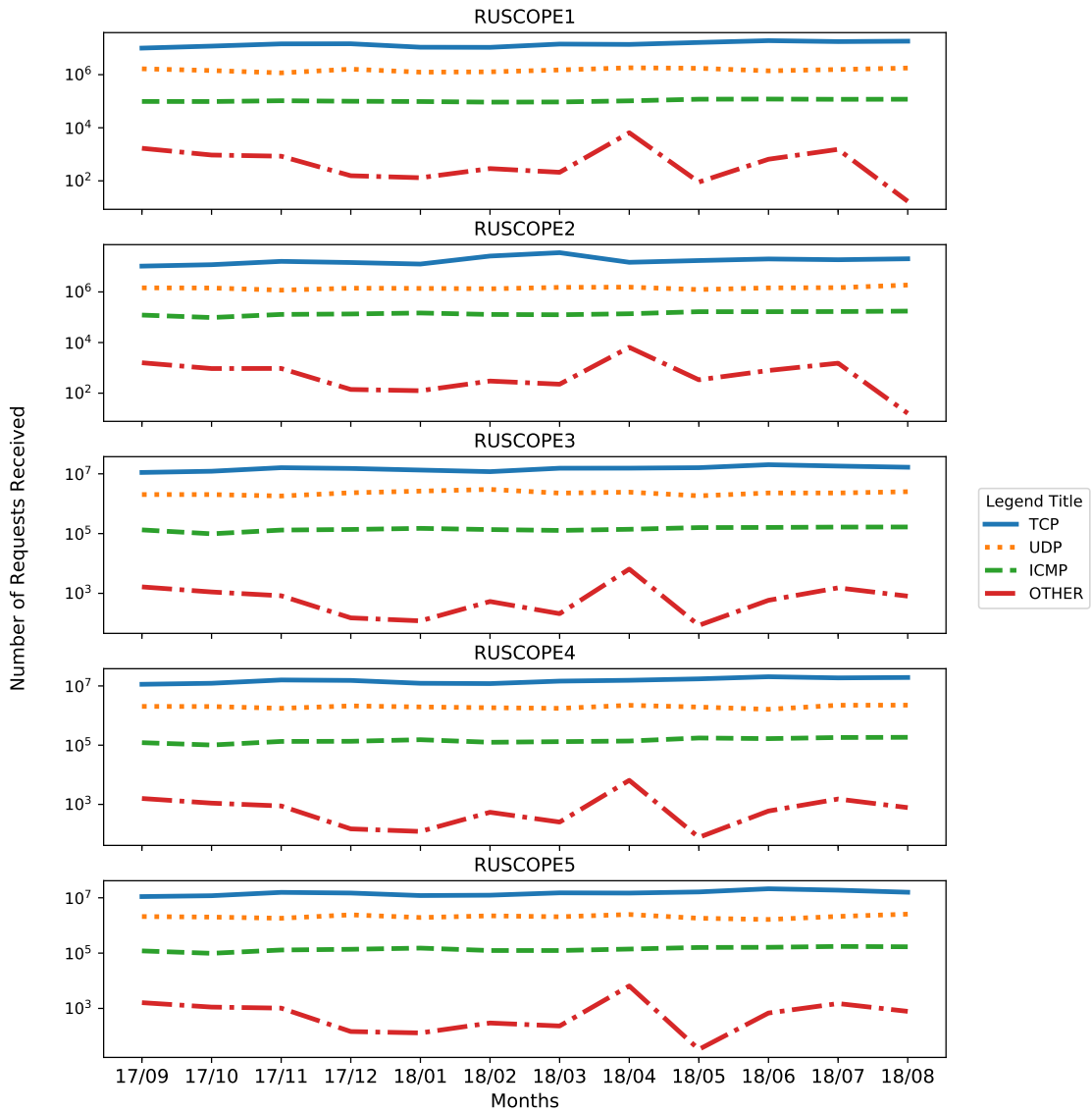


Figure 3.2: Network Telescope Protocol Split

The similarity in the shape of the curves suggest that all five telescopes share similar trends. However, the number of requests differ on the y axis.

The graphs illustrated in Figure 3.2 show the number of requests received on each telescope over the 12-months, split by protocol. The requests recorded on the telescopes represent a variety of different protocols including TCP, UDP, and ICMP. The *Other* category includes protocols such as, GRE (Farinacci *et al.*, 2000), SCTP (Stewart, 2007), IPv6

Table 3.7: Network Telescope Active Passive Data

(a) Requests					
Network Telescope	TCP %		ICMP %		UDP %
	Active	Passive	Active	Passive	–
RUSCOPE1	81.98	7.83	0.55	0.11	9.51
RUSCOPE2	70.85	21.16	0.62	0.09	7.27
RUSCOPE3	78.84	7.30	0.70	0.11	13.04
RUSCOPE4	80.68	7.21	0.71	0.12	11.27
RUSCOPE5	79.51	7.49	0.71	0.11	12.18

(b) Unique IP Addresses					
Network Telescope	TCP %		ICMP %		UDP %
	Active	Passive	Active	Passive	–
RUSCOPE1	97.35	0.75	0.43	0.27	1.77
RUSCOPE2	97.04	0.77	0.44	0.27	2.07
RUSCOPE3	96.28	0.73	0.43	0.26	2.92
RUSCOPE4	97.65	0.76	0.44	0.26	1.49
RUSCOPE5	97.0	0.75	0.44	0.26	2.15

The values are calculated as a percentage of the totals from Table 3.6

Encapsulation (Conta, 1998), ESP (Kent, 2005), and IPsec (Perkins, 1996). The trend is the same across all telescopes with TCP seeing the most requests every month, followed by UDP and ICMP. *Other* is a very small percentage of the total and received the least requests each month.

Despite having access to large quantities of data, not all data collected by the telescopes was relevant and useful. Some of the data is made up of backscatter (passive traffic), some is misconfiguration, and some is malicious requests (Section 2.5). This research focuses on the malicious requests (active traffic). The passive traffic is not relevant for this project.

Table 3.7 lists the total requests and unique IP address for the three most common protocols TCP, UDP, and ICMP. The number of requests for protocols other than these three was less than 1% of the total requests and as such considered negligible and not included in the table. It is important to note that it is significantly more difficult to classify UDP packets in terms of active or passive (Blenn *et al.*, 2017). For this reason, all UDP packets were considered to be active traffic when analysing and processing data.

The values in Table 3.7a give a percentage representation of the graphs in Figure 3.2. Table 3.7b shows the number of active and passive unique IP addresses recorded on the telescopes over the whole data collection period. The packets were divided using the criteria discussed in Section 2.5.3. The reader will note that the percentages do not total 100%. This is the result of some packets being malformed, or falling outside of the active and passive rules defined in Section 2.5.3.

Table 3.8: IBR Datasets of Active Unique IP Addresses

Dataset	Data Source	Total	%
IBR1	RUSCOPE1	10 113 380	75.24
IBR2	RUSCOPE2	9 955 211	75.06
IBR3	RUSCOPE3	10 450 707	75.45
IBR4	RUSCOPE4	10 240 514	75.02
IBR5	RUSCOPE5	10 224 608	75.15
IBR6	$T_1 \cap T_2 \cap T_3 \cap T_4 \cap T_5$	2 469 406	–
IBR7	$T_1 \cup T_2 \cup T_3 \cup T_4 \cup T_5$	21 427 726	–

T_n represents an IBR dataset where n is from 1 to 5.

Percentages are calculated using the total value from Table 3.6b

Having quantified the data in terms of the overall requests recorded on the telescopes, the datasets used in this research were defined and described. IBR datasets were created using the unique IP addresses from the TCP, UDP, and ICMP active requests recorded on the telescopes and are shown in Table 3.8. IBR6 and IBR7 were two additional synthetic datasets created, using the data from IBR1 through IBR5, to bring the total of IBR datasets to seven. The IP addresses in IBR6 provide a dataset of IP addresses that are common across all five telescopes. IBR6 provides a set of IP addresses which are more prevalent on the Internet. The IP addresses in IBR7 represent every IP address seen on all five telescopes. IBR7 is used to show the total number of IP addresses seen throughout the research.

Table 3.9 details the size of the IBR datasets for each month and Figure 3.3 represents the IBR datasets in graphical form. The Data Volume axis uses a logarithmic scale to display values of each IBR dataset. IBR1 through IBR5 have similar values throughout the 12-month period. IBR6 has significantly lower values than any other IBR dataset, due to the fact that IBR6 is made up of IP addresses which are seen across all five real-world datasets. This means that to be recorded in IBR6 an IP address had to score 5/5 in terms of how many real-world datasets it was seen on in the month. In comparison, IBR7 is a very large dataset, due to the fact that it is made up of all IP addresses seen across all five real-world datasets. This means that to be added to IBR7, an IP address had to be recorded on at least one of the real-world datasets. Despite the difference in the volume of data, all IBR datasets follow the same increase and decrease in the volume of data received.

3.4 Chapter Summary

This chapter described the data utilised in the analysis stage of this research. The data was collected from two different sources, OSINT feeds and telescopes. The OSINT came from various threat feeds around the Internet while the IBR data came from telescopes

Table 3.9: IBR Dataset Volume of Data

Date	IBR1	IBR2	IBR3	IBR4	IBR5	IBR6	IBR7
09/17	853 917	819 369	850 732	851 539	847 845	250 135	2 100 680
10/17	897 439	897 439	926 564	926 754	924 886	272 380	2 098 140
11/17	1 351 407	1 352 716	1 392 564	1 391 006	1 388 942	378 725	3 351 020
12/17	1 400 690	1 384 529	1 462 294	1 447 764	1 448 233	370 247	3 522 187
01/18	1 130 140	1 198 024	1 317 804	1 202 180	1 169 066	265 622	3 165 605
02/18	1 073 573	1 037 526	1 097 621	1 070 733	1 084 821	204 298	2 907 089
03/18	1 367 603	1 366 425	1 413 749	1 381 188	1 386 273	262 760	3 581 966
04/18	1 210 702	1 172 283	1 223 497	1 213 951	1 218 139	253 895	3 214 130
05/18	1 059 349	1 015 086	1 057 252	1 054 279	1 057 998	247 961	2 847 679
06/18	992 270	962 175	1 008 300	998 564	990 302	287 706	2 524 530
07/18	1 015 676	978 257	1 012 845	1 030 074	1 013 868	278 835	2 635 411
08/18	1 087 987	1 080 017	1 090 197	1 082 165	1 074 566	256 490	2 929 365

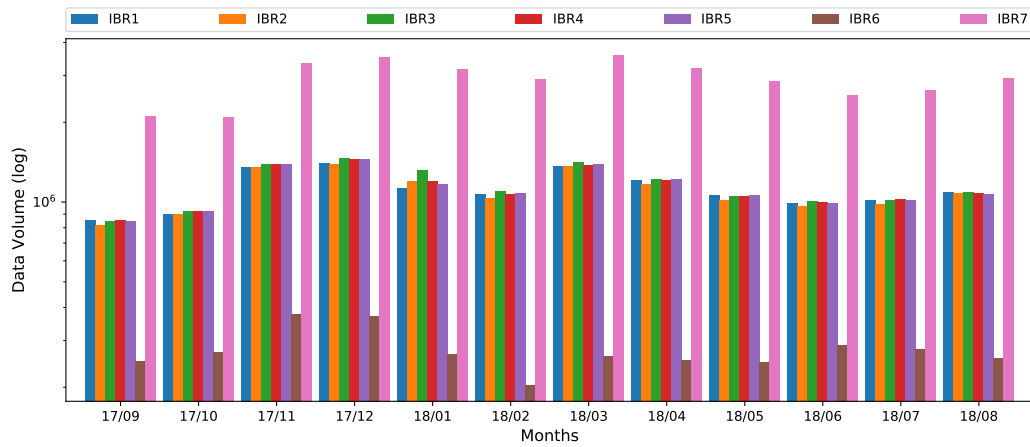


Figure 3.3: IBR Dataset Volume of Data

located on the Rhodes University network. The chapter was broken up into two parts. **Section 3.2** and **Section 3.3** described how the data from each source was collected and quantified. Quantifying the data provided an illustration of the amount of data used in the research, as well as an understanding of the composition of the data used during the analysis performed in Chapter 5.

4

Design and Implementation

THIS chapter is the first of two chapters discussing the design and implementation of a system capable of achieving two of the four primary research objectives (objective one and three) introduced in Section 1.3. The system described provides a platform for the collection, processing and dissemination objectives of this research. The full code used in this research is available in the system code on repository detailed in Appendix B. Additional information relating to the design and implementation of the system can be found in Appendix C.

The remainder of this chapter is structured with **Section 4.1** discussing the overall design of the system, the role of each component, and how they integrate to form a complete system. **Section 4.2** discusses the different technologies used, specifically focusing on the three different technologies used to publish the TI created by the system. **Section 4.3** discusses the design and implementation of the database used in the system. The database was primarily used to store the OSINT created by the system. A discussion of the schema of the database and the important tables used is also provided. **Section 4.4** describes the *Collection* component of the system and the various modules implemented to achieve OSINT collection and pre-processing.

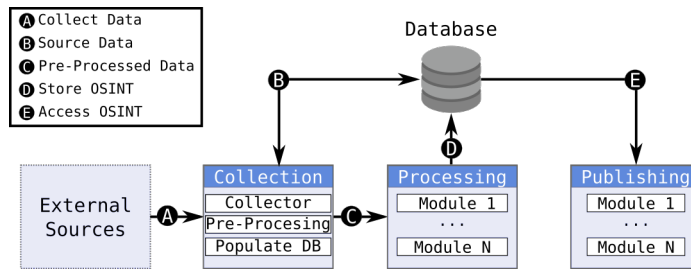


Figure 4.1: System Design

4.1 System Design

The system discussed in this chapter has been divided into three distinct components, namely *Collection*, *Processing*, and *Publishing*. The decision was made to divide the system into components due to the distinct nature of the tasks being performed at different stages of the process. Additionally, dividing the components of the system by task allows for modifications and changes to be made to a single component without affecting the others. Each component expects data to be available in a certain format and, as long as this data is present in the correct format, the component can continue performing its task. Figure 4.1 provides a high-level overview of the design of the system.

The three high-level components communicate with each other through two methods; 1) using a central database, and 2) by placing data in a shared location. This architecture enables components on both local or remote locations to operate in the same way. The shared location in this system is a folder on the local computer that each component has access to. If these components were to be distributed onto different physical machines, the shared folder can be synchronised using a utility such as Rsync. However, the synchronisation of the shared folder is dependent on the implementation and the needs of the organisation or security team at the time.

The *External Sources* illustrated in Figure 4.1 represent the OSINT data collected for use in the system. The primary external source of data is a collection of OSINT feeds made available on the Internet. The secondary external source of data is a collection of telescope log files, as described in Section 3.3. The data is collected from these external sources by the *Collection* component.

The *Collection* component is designed with its own individual modules which collect and pre-process the data downloaded from the external sources. The *Collection* component collects data from all external sources and converts it into a format which can be used by the *Processing* (Section 6.1) component (Figure 4.1 point A). All information relevant to the external sources of data is stored in the database of the system. The *Collection* component communicates with the database to retrieve and update information about the external sources, such as URL and data type (Figure 4.1 point B).

The *Processing* component receives data from the *Collection* component through the shared local folder (Figure 4.1 point C). Each module within the *Processing* component operates independently from the next. The data created by the *Processing* component is inserted or updated into the database (Figure 4.1 point D), at which point the OSINT is ready to be published by the *Publishing* component.

The *Publishing* component is responsible for making the OSINT available to consumers through a number of different channels. The *Publishing* component contains a number of different modules. The system used for this research implemented three publishing modules, Flat File (Section 6.2.1), ExaBGP (Section 6.2.2), and DNSBL (Section 6.2.3) to publish the OSINT created during the *Processing* component. Each module is able to pull OSINT from the database and publish it in its own respective format (Figure 4.1 point E). The design of the *Publishing* component is such that modules can be added or removed without affecting any other module.

4.2 Technologies Used

This section discusses the main technologies used in the system. The technologies discussed in this section were determined to be those most suitable for the functionality required. Additionally, the technologies were the most common technologies used by the Python community to achieve the functionality required. The system was developed using Python 3.6.x, with the exception of the *process PCAPS* script (Appendix C.1) which was developed in Python 2.7.x. This script was written in Python 2.7.x because, at the time of writing, the *dpkt*¹ Python library was not fully implemented in Python 3.6.x and gave continuous errors when working with the telescope data.

4.2.1 JavaScript Object Notation (JSON)

JSON² is the format used to store information about the external sources of TI. JSON is a lightweight, standard format for representing data objects as human-readable text (JSON, 2002). At its core, JSON is a collection of key:value pairs of data. For example, “name”:“john” is a JSON object where the key is *name* and the value is *john*. The JSON library for Python provides a set of easy-to-use functions which make reading and writing to files simple and straightforward. The system uses the JSON library to save objects to file using the *json.dump()* function, as well as reading saved objects from file, using the *json.load()* function.

¹<https://dpkt.readthedocs.io/en/latest/>

²<https://www.json.org/>

4.2.2 Comma Separated Values (CSV)

CSV format is widely used in data processing applications for data file interchange and flat file storage (Shafranovich, 2005). The most common usage for CSV files is in flat file databases where each line of text holds a single record separated by delimiters. According to Avrach (2015), CSV files are categorised as flat files. However, the main characteristic which sets CSV files apart from other flat files is the use of a delimiter to separate data fields in each line. These delimiters are generally a comma, but any character can be used, provided that character does not occur in the data itself.

The CSV format has no formal specification or documentation defining standard implementation. However, RFC 4180 defines the most common format for CSV files (Shafranovich, 2005), an example of which can be seen in Listing 4.1. The characteristics of a CSV in terms of RFC 4180 are described as follows:

```

1 field_name1,field_name2,field_name3 CRLF
2 aaa,bbb,ccc CRLF
3 ddd,eee,fff CRLF
4 hhh,,iii CRLF
5 'jjj','k CRLF
6 kk','lll' CRLF
7 'm','m','m','nnn','ooo' CRLF
8 zzz,yyy,xxx

```

Listing 4.1: CSV File Format (Shafranovich, 2005)

- The first line in the file can be an optional header showing names corresponding to the fields in the file (see Line 1). There must be the same number of field names as there are data fields.
- Each line represents a single record and at the end of each record is a line break - “CRLF” (see Line 2). The last record in the file does not require a line break, but one can be added if the author wants one (see Line 8).
- Each line can contain one or more fields separated by commas, but each line must contain the same number of fields. Line 4 shows a valid line with a record where not all the data is present. The line still has the required three fields, with the second being an empty field.
- If a field contains special characters such as double-quotation marks, commas, or line breaks, it must be enclosed in double quotation marks. Lines 5 and 6 is a single record that spans two lines, the “CRLF” is contained within the third field³. This is treated as a single record.
- If a quotation mark is found within the field, it must be escaped by another quotation mark. Line 7 shows an example where the second “m” in the first field is enclosed in double-quotation marks.

Regardless of the subtle differences in formats and delimiters used, CSV files provide

³This record would be written as “jjj”,“kCRLFkk”,“lll” CRLF on one line

a standard format which can effectively manipulate and process flat file data from one process to the next (Python Software Foundation, 2007). The Python CSV module⁴ is a standard library which provides the functionality required to create CSV flat files for this research.

4.2.3 ExaBGP

ExaBGP is a tool designed to allow third-party applications to communicate with physical BGP routers (Mangin, 2010). The ExaBGP Python library⁵ provides a way to peer with BGP (Section 2.4) routers and control the advertising of networks and routes (thePacketGeek, 2015). ExaBGP is widely used by security professionals as a means to protect their internal network from external threats such as DDoS attacks. Mangin (2010) describes how NetFlow can be used to monitor a network for potential DDoS attacks and ExaBGP can be used to blackhole the IP address of the TA before it reaches the internal network. The system used in this research uses ExaBGP as a way of publishing potentially malicious TA IP addresses to other neighbouring BGP routers on the Internet.

ExaBGP requires two files to be passed to it at runtime; a configuration file and an environment file (Mangin, 2010) (Appendix C.6). The configuration file holds all the configuration information used by ExaBGP when it is running. Configuration information includes:

- The neighbouring routers which are peered with the ExaBGP router.
- Settings that apply during start-up, restart and shut down.
- The script files to execute when it starts up.

The environment file holds settings for executing the daemon. Much of this file is kept as default, however, some of the settings can be changed according to performance requirements.

To create an ExaBGP instance from the terminal, a command such as the one seen in Listing 4.2 is used. The first argument of the ExaBGP command is the configuration file to use (*exaBGP.conf*), and the second argument is the environment file to use (*exaBGP.env*) which is preceded by *-env*. The ExaBGP configuration and environment files used in the system (Section 6.2.2) can be seen in Appendix C, Appendix C.6.

```
1 # exabgp ./etc/exabgp/exaBGP.conf --env ./etc/exabgp/exaBGP.env
```

Listing 4.2: ExaBGP Terminal Command Example

⁴<https://docs.python.org/3.6/library/csv.html>

⁵<https://github.com/Exa-Networks/exabgp>

Table 4.1: RBLDNSD Arguments (Tokarev, 2015)

Arguments	Description
-r rootdir	changes directory to <i>/path/to/root/dir</i> if the argument is supplied.
-w workdir	changes the working directory to <i>/path/to/work/dir</i> if the argument is supplied.
dnsbl.example.org: ip4set:data.dns	The argument is in the format <i>zone:dataset</i> . The zone is <i>dnsbl.example.org</i> . The dataset <i>ip4set:data.dns</i> states that the file containing the data is <i>data.dns</i> and it contains one IPv4 addresses per line. Each line in the dataset can be a combination of A and TXT values. The dataset file is found in the working directory.
-b198.51.100.1/53	Specifies the IP address (<i>198.51.100.1</i>) and port (<i>53</i>) which RBLDNSD will bind to.

4.2.4 RBLDNSD

To provide the DNSBL (Section 2.8) service, the system uses a Linux service called RBLDNSD⁶ (Tokarev, 2015). RBLDNSD is a small DNS daemon designed specifically for DNSBLs. The daemon provides the ability to serve both malicious IP addresses and domain names. RBLDNSD runs as a service on the server, answering DNS queries. When RBLDNSD is running, any DNS query destined for the IP address of the server it is on will be handled by RBLDNSD.

Executing RBLDNSD requires a simple command in the terminal. Listing 4.3 shows the command used to run the daemon and Table 4.1 describes each argument in the command. The *-r* and *-w* arguments are optional.

```
1 rbltdnsd -r /path/to/root/dir -w /path/to/work/dir dnsbl.example.org:ip4set:
  data.dns -b198.51.100.1/53
```

Listing 4.3: RBLDNSD Command

Once RBLDNSD is running, it must have a zone file to reference when a DNS query is received. The zone file contains the list of IP addresses that RBLDNSD knows about, which are potentially IP addresses that need to be blocked. Appendix C.5 of Appendix C illustrates the zone file used in this system, along with a description of the different lines in the file.

⁶<https://rbltdnsd.io/>

4.2.5 PostgreSQL

PostgreSQL (Postgres) is an open-source object-relational database system. It was chosen as the database for this system based on the following advantages (The PostgreSQL Global Development Group, 1996):

- The database has a reputation of being reliable and correct, as well as ensuring data integrity throughout the database.
- It is compatible with all major operating systems and programming interfaces.
- It has native support for IPv4 and IPv6 addresses.
- The database size is limited by the physical hardware on which it is being run. Tables are limited to 32TB, fields are limited to 1GB and tables can contain an unlimited number of rows as long as it is within the table size limit.

A standard feature across any relational database is the use of indexing to provide faster and more efficient queries on the data. Indexing is added to the table to speed up the query (The PostgreSQL Global Development Group, 1996) and allows more efficient search and retrieval of a small number of records from the table. However, if the query returns more than 10% of the total records, indexing can slow down the process (Harrison and Feuerstein, 2006). Indexes are added to a single table by using one or more columns of the table. Due to the very large number of records stored in each OSINT and IBR table, indexing was added to the primary key field (typically the IP address field) of the table. Without indexing enabled queries were taking a very long time to execute, causing the data processing times to be high.

When choosing the type of index to create, it is important to consider the type of operations that will be performed on the table, as well as the type of data being used. B-Tree indexes are the default index type used in Postgres when creating an index, if no index type is specified. These indexes are used whenever equality comparisons ($<$, $<=$, $=$, $>=$, $>$) and range (*BETWEEN* and *IN*) queries are done. The database of the system will store IPv4 related data, so the *B-Tree* index with the *cidr_ops* operation class was used (Stinson, 2001).

In many instances of a system, certain CRUD (Create, Read, Update, and Delete) operations are performed repeatedly on the database in the same order or after a certain event. Having to rewrite or replicate these operations each time the query is needed can be time consuming. To account for this, a feature provided by Postgres and utilised in the system is that of stored procedures. Robidoux (2012) defines a stored procedure as “...nothing more than prepared SQL code that you save so you can reuse the code over and over again.”

Postgres refers to stored procedures as *functions*. These functions are written using the *PL/pgSQL* language, which is an extension of SQL specifically designed to be used with

Postgres databases when writing functions (Geschwinde and Schöning, 2002). An example PL/pgSQL function is illustrated in Listing 4.4. The function updates all rows in a given table (*arg1*) whose *id* matches the specified id (*arg3*) by setting *col1* to a given value (*arg1*). This listing illustrates one of the advantages of the PL/pgSQL used within the system, namely the ability to create dynamic queries. A dynamic query is a SQL statement which is composed at run-time, with arguments provided to the function when it is called (The PostgreSQL Global Development Group, 1996). These queries are used during the population of the database discussed in Section 4.4.

```
1 CREATE FUNCTION public.example(arg1 text arg2 text arg3 text)
2 RETURNS void
3 LANGUAGE 'plpgsql'
4
5 AS $BODY$
6 DECLARE
7 BEGIN
8 EXECUTE format('UPDATE %I SET col1 = $1 WHERE id = $2', arg1)
9 USING arg2, arg3;
10 END;
11 $BODY$;
```

Listing 4.4: Postgres Function Example

4.3 Database Design

As previously mentioned in Section 4.1, the system utilises a central database which is accessible to every component described in Figure 4.1. The *inet* and *cidr* data types provided by Postgres allow the system to manipulate the IP address data directly from within the database. Having this ability greatly reduces the number of external Python scripts and libraries needed to accomplish the analysis and processing tasks. The support for IP addresses built into Postgres also reduces the amount of space needed to store the IP addresses since they are stored directly as IP addresses and not strings of text.

Each system component has access to the central database through the *DB Handler* library discussed in Appendix C.3. The components are able to add this library to their scripts and perform CRUD operations on the database. Figure 4.2 illustrates the database schema created for the system. This schema illustrates the various tables created in the database and their relationship to each other. The *sources* table in the schema is discussed in Appendix C.2. The remaining tables are discussed in the list below:

- The *categories* table holds information about the type of TA the OSINT feed reports about. This table has a relationship with the *sources* table, as shown by the line joining the two tables.
- The *telescopes* table holds information about the telescopes used by the system. It includes the IP address of the telescope, the common name (neatly formatted

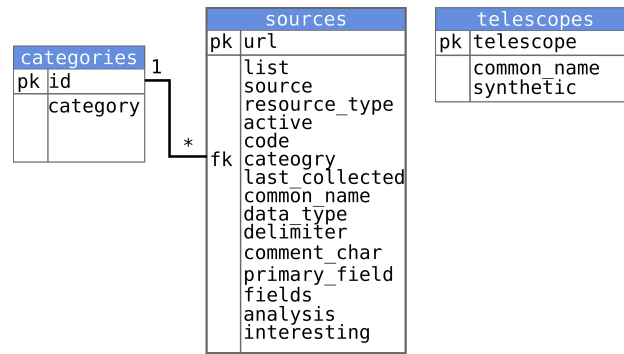


Figure 4.2: Database Schema

name), and whether it is a real world or synthetic telescope.

4.3.1 Sources Table

The *sources* table in the database holds a list of all the external OSINT feeds known to the system. These external sources are discussed in Appendix A.1. Table C.1 gives a description of the fields in the table. Any new OSINT feed must be added to the *sources* table before it can be used by the system.

The *SourceHandler* library is responsible for managing the OSINT feeds. The *SourceHandler* is described in Appendix C.3. The *sources* table is populated through a combination of the JSON input file and the *add_feeds_to_db()* function. The function takes a JSON file containing a list of OSINT feeds to insert into the database. The *key:value* pairs are populated with the relevant information pertaining to the OSINT feed. Once the JSON file is loaded, the function iterates through each JSON object in the file and issues a SQL insert query to insert the data into the database. In addition, a unique code is assigned to each OSINT feed implemented by the function. The code is used to identify the lists within the system during the *Processing* stage. At this point, the rest of the system interacts with the database to retrieve metadata about a specific OSINT feed.

4.3.2 OSINT Tables

The database has a table for each OSINT feed containing information about IPv4 addresses. The purpose of the OSINT tables is to store all the unique IP addresses published by the OSINT feed. Each OSINT list contains different information, so the fields in each table are different. However, all OSINT tables contain an IP address field as the primary key field for the table. Similarly, there are three fields common in all OSINT tables, *first_seen*, *last_seen*, and *days_inactive*. These fields provide the ability to keep track of when the IP address was first seen on the OSINT feed and when it was last seen

on the feed. The *days_inactive* field is updated after new IP addresses are added to the table and records how long an IP address has been inactive for.

4.3.3 IBR Tables

The IBR tables store data on all active requests recorded on the telescopes. There are seven IBR tables in the database, one for each IBR dataset described in Section 3.3.2. Each of the IBR tables contain the following fields:

- *srcip* is the unique IP addresses taken from the PCAP file.
- *year_month* records the year and month of the corresponding PCAP file the IP address was taken from.
- *packet_count* records how many packets were seen from the specific IP address. IBR6 and IBR7 do not have any packet counts for the IP addresses because they are synthetic datasets.
- *tcp*, *udp*, *icmp*, and *other* are boolean fields that state whether the IP address sent any packets for the protocol. If an IP address sent one or more TCP and UDP packets, the *tcp* and *udp* fields are set to *True*.

The *srcip* and *year_month* fields are a composite primary key. This means that the table will only store unique IP addresses for each month of IBR data, so that an IP address is not recorded twice for the same month.

The IBR tables also contain enrichment fields for whois and reverse DNS data. The whois fields are *allocated*, *cc*, *as_name*, *registry*, *bgp_prefix*, and *asn*. The reverse DNS fields are *ip2hostname* and *hostname2ip*. The whois and reverse DNS fields are used internally during the data analysis (Chapter 5) stage of this research.

4.4 Collection

The *Collection* component is responsible for gathering data from all the external sources and converting each dataset into a format which can be used by the *Processing* component. The *Collection* component is divided into three modules, the *Collector*, the *Pre-Processor*, and the *Populate DB* modules. Each module performs a specific process on the OSINT data. Figure 4.3 describes a complete execution of the *Collection* cycle which occurs each day at midnight.

The Collector is the first module to be executed. At the start of the collection process, the *Collector* determines which OSINT feeds need to be collected. For each URL in *sources* table (Section 4.3.1), the *Collector* determines whether the resource is a download-

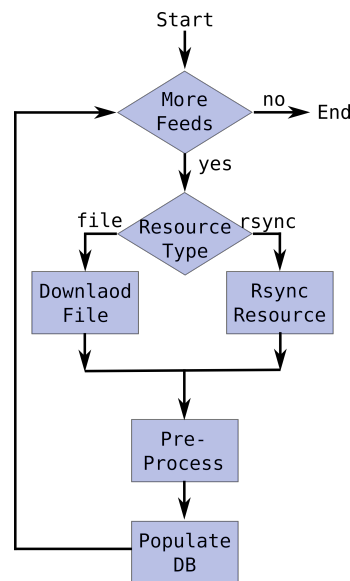


Figure 4.3: Open Source Threat Intelligence Feeds Collection Process

able HTTP file or an Rsync resource. Depending on the type of resource, the appropriate request is sent to the OSINT feed URL in an attempt to download the external resource. The HTTP request attempts to download the OSINT data, and the Rsync request attempts to synchronise the remote data to the local computer. The HTTP request is performed using the Python URLLIB⁷ library, and the Rsync request is performed using the Rsync service (Tridgel, 2007). Having downloaded the file or Rsync’ed the resource, it is saved to a folder that each of modules has access to. Once the *Collector* has downloaded the data, the *Pre-Processor* is initiated to prepare the data to be added to the database and used during the data analysis stages described in Chapter 5.

The Pre-Processor is responsible for cleaning the data and converting it into a usable format. The objective of the cleaning process is to take the OSINT data collected and remove any inconsistencies and formatting errors within the files. Pre-processing OSINT files is necessary because:

- the majority of the data files collected contain some form of comment before the actual data. Listing 4.5 shows an example of comments before the data from Bam-benek Consulting . There is no guarantee that the source will use a “#” for comments so the pre-processing module keeps a record of what escape characters are used for comments on a feed-by-feed basis.
- not all OSINT feeds use comma delimiter values for separating fields. To overcome this complication, the *Pre-Processor* is configured to keep a record of all feeds that do not use a “,” (comma) as the delimiter. When it collects and pre-processes one of these data files, it will perform the necessary steps to change the delimiter to a

⁷<https://docs.python.org/3/library/urllib.html>

“,” (comma) so that all data files can be stored as CSV files.

- any data file which contains a description field as part of the record is modified to remove any commas that may appear in the description since these commas will cause errors in later processing stages.

```

1 #####
2 ## Master Feed of known, active and non-sinkholed C&Cs IP
3 ## addresses
4 ...
5 ## All times are in UTC
6 #####
7 5.79.79.211,IP used by banjori C&C,2018-07-31 21:04,http://osint.
   bambenekconsulting.com/manual/banjori.txt
8 23.107.124.53,IP used by banjori C&C,2018-07-31 21:04,http://osint.
   bambenekconsulting.com/manual/banjori.txt
9 ...

```

Listing 4.5: Commented Data File

After the *Pre-Processor* completes execution, CSV files are created for each for each of the OSINT feeds. Next, the *Populate DB* component is called on to add the newly collected data to the database.

The **Populate DB** module is executed after the OSINT data has been pre-processed into CSV files. This component takes each of the downloaded OSINT lists and inserts the data into the database. To insert the data into the database, the following steps are executed:

- a temporary table with headers specific to the OSINT list being inserted is created. This temporary table is used to insert the data into the database.
- `Psycpg2` provides a `copy_expert("query",file)` function with two inputs, an SQL query and a file. The function adds the contents of the file into a given table in the database. An example of the function call used in this component is `copy_expert("COPY temp FROM STDIN WITH CSV HEADER", "/path/to/data.csv")`. This function call will insert the contents of `data.csv` into the `temp` table.
- two functions within the database are called with the data in the `temp` table:
 - `insert_from_temp` checks if any IP addresses in the `temp` table do not exist in the OSINT table. Any IP address that is not present in the OSINT table is added, along with the `first_seen` and `last_seen` timestamps.
 - `update_from_temp` updates the `first_seen` and `last_seen` fields for all IP addresses that are already present in the OSINT table, using the given timestamp.
- after adding the IP addresses to the database, the `calculate_days_inactive` Postgres function is called on to calculate how many days since the IP address was last seen.

4.5 Chapter Summary

This chapter discussed the design and implementation of the system used for the collection of OSINT. **Section 4.1** started the chapter by describing a high-level overview of the design of the system, along with a discussion of the *Collection* component of the system. A discussion of the specific technologies used in this research to achieve the described system was provided in **Section 4.2**. The focus of this subsection was the three distribution technologies chosen; CSV, ExaBGP, and RBLDNSD.

Section 4.3 went on to discuss the design and implementation of the database used in the system. **Section 4.4** discussed the *Collector*, *Pre-Processor*, and *Populate DB* components of the *Collection* module used in the system. The remaining two components of the system (*Process* and *Publishing*) are discussed in Chapter 6. Next, Chapter 5 provides a discussion of the OSINT and IBR data used during this research.

5

Data Analysis

THIS chapter presents an exploratory analysis of the OSINT and IBR data discussed in Chapter 3. The primary focus in this chapter is the OSINT feeds containing IPv4 addresses as described in Section 3.2. The IBR data is used as the basis for comparative analysis between IP addresses classified as malicious by TI providers, and the IP addresses actually observed in the real world. Using the data collected during the period under study, this chapter explores how effective the OSINT data is at reporting current, active and malicious Internet traffic. The analysis in this chapter helps to form an understanding of how effective OSINT data can be to a consumer. All the scripts developed are available in the project repository detailed in Appendix B.

The remainder of the chapter is structured with **Section 5.1** presents the initial analysis of the OSINT and IBR data from Chapter 3. **Section 5.2** discusses how the datasets change from month-to-month and how this can affect the way they are used. **Section 5.3** presents an analysis of how OSINT data from each month compares to all 12-months of IBR data. **Section 5.4** discusses the presence of a set of persistent IP addresses published in the OSINT datasets and recorded on the telescope sensors. **Section 5.5** takes the persistent IP addresses identified from the previous section and further analyses the types of activities they are linked to. **Section 5.6** completes this chapter with an analysis of the source of the IP addresses.

5.1 Initial Analysis

To achieve the analysis discussed in this chapter, a series of scripts were developed. These scripts made use of two core libraries, Pandas¹ and Matplotlib². These two libraries were chosen because of their efficient data processing and data manipulation features.

Before exploration of the data was conducted, the OSINT datasets discussed in Chapter 3 needed to be further reduced to a smaller subset to eliminate data which falls outside the scope of this research. To achieve this, two questions were answered:

1. Are there any IP addresses in common between the OSINT feeds and the IBR datasets?
2. If there are IP addresses in common, what percentage of the total IBR and OSINT datasets do these IP addresses make up?

To answer the first question, the number of IP addresses that the OSINT and IBR datasets have in common over the 12-month period under research was calculated. The results are displayed in Table 5.1. To determine which lists to remove, the values detailed in Table 5.1 were converted to percentages of the total number of IP addresses. These percentage values are displayed in Table 5.2. This initial analysis provided a way to assess the OSINT data against the CART framework presented in Section 1.2 (Caltagirone, 2015a). In particular, it provided a way to determine the completeness and relevance of the OSINT collected.

A portion of the values listed in Table 5.1 are very low or zero. For example, Feodo IP, Malware Domain List Active, Spamhaus DROP, and Zeus IP show less than 20 IP addresses in common with all IBR datasets. For example, the values for Malware Domain List Active might be low because malware targets active hosts, while Zeus IP might be low because the Zeus botnet was inactive or eliminated. Although there is a reason for these low values, this research excluded them from further analysis because they do not provide sufficient data to allow accurate comparison of OSINT against IBR data.

Table 5.2 shows the percentages of IP addresses in common by total OSINT data (Table 5.2a), and total IBR data (Table 5.2b). The OSINT datasets listed in Tables 5.2a and 5.2b are limited to those with a percentage value greater than or equal to 0.01% for any of the seven IBR datasets. The OSINT datasets with a percentage value less than 0.01% were removed to reduce the amount of information displayed in the tables and allow easier reading and interpretation.

Given that IBR6 only contains IP addresses which appear in all five of the real-world datasets (IBR1-IBR5), the percentage of IP addresses in common was lower than

¹<https://pandas.pydata.org/>

²<https://matplotlib.org/>

Table 5.1: Total IP Addresses in Common Between OSINT and IBR Datasets

OSINT	IBR1	IBR2	IBR3	IBR4	IBR5	IBR6	IBR7
AlienVault	512 143	542323	542502	542 289	541 711	488 938	568 374
Autoshun	15 606	15 531	15 608	15 695	15 581	12 660	18 060
Bambenek	16	20	21	20	18	13	32
Binary Defense Banlist	23 494	30 041	29 633	29 688	29 337	16 396	36 412
Binary Defense TOR	749	791	848	881	880	190	1876
Bruteforce Blocker	5 761	5 796	5 814	5 834	5 806	4 803	6 837
CINS	477 226	507 161	507 477	507 250	506 903	456 416	531 307
Feodo IP	12	14	12	12	13	6	18
ISC IP	18	18	19	19	19	18	19
MDL Active	2	2	2	2	3	1	4
Proofpoint	12	15	12	12	14	6	19
Ransomware	5	5	5	6	4	3	7
SSLBL Aggressive	63	62	66	67	64	31	106
SSLBL IP	38	39	40	40	38	20	61
Spamhaus DROP	0	0	0	0	0	0	0
Spamhaus EDROP	0	0	0	0	0	0	0
Zeus Bad IP	0	1	1	1	2	0	2
Zeus IP	0	1	1	1	2	0	2

average. In contrast, IBR7 recorded a higher percentage of IP addresses in common since it contained a larger set of IP addresses, namely all the IP addresses seen in the real-world datasets. The inverse is seen when using the IBR dataset totals to calculate the percentages. In particular, IBR6 has fewer total records and therefore the percentage in common with the OSINT datasets is higher.

Based on the nature of the telescopes used to source the IBR datasets, the OSINT datasets can be arranged into different categories using the percentages in common:

1. The first category contains datasets such as AlienVault and CINS which report on general malicious activity, such as port scans and self-propagating worms.
2. The second category contains datasets such as Malware Domain List, Ransomware, and Zeus which report on malware, botnets, C2 servers, and specific types of ransomware.
3. The third category contains datasets such as Autoshun, Bruteforce Blocker, and Binary Defense which display a much smaller percentage of IP addresses in common. This was likely the result of the datasets containing fewer total records (particularly for Autoshun), or the IP addresses in the list being seen less often because they are linked to a more specific type of TA which was not detected by the telescope sensors.

Considering the percentage values in Table 5.2, it is possible to conclude that not all OSINT datasets are relevant to the type of requests being recorded on the telescopes. Datasets such as those from SpamHause, and to some extent Zeus and Malware Domain

Table 5.2: IP Addresses in Common Between OSINT and IBR Datasets

(a) Percentage by Total OSINT Data							
OSINT	IBR1	IBR2	IBR3	IBR4	IBR5	IBR6	IBR7
AlienVault	80.24	84.97	85.00	84.97	84.88	76.61	89.05
Autoshun	44.54	44.33	44.55	44.80	44.47	36.13	51.55
Bambenek	0.26	0.33	0.34	0.33	0.29	0.21	0.52
Binary Defense Banlist	35.88	45.88	45.26	45.34	44.81	25.04	55.61
Binary Defense TOR	0.97	1.02	1.09	1.14	1.13	0.24	2.42
Bruteforce Blocker	36.75	36.97	37.09	37.21	37.04	30.64	43.61
CINS	85.04	90.37	90.43	90.39	90.33	81.33	94.68
Feodo IP	0.75	0.87	0.75	0.75	0.81	0.37	1.12
ISC IP	4.83	4.83	5.09	5.09	5.09	4.83	5.09
MDL Active	0.19	0.19	0.19	0.19	0.29	0.10	0.39
Proofpoint	0.37	0.47	0.37	0.37	0.43	0.19	0.59
Ransomware	1.08	1.08	1.08	1.29	0.86	0.65	1.51
SSLBL Aggressive	1.54	1.52	1.62	1.64	1.57	0.76	2.60
SSLBL IP	3.25	3.34	3.42	3.42	3.25	1.71	5.22
Zeus Bad IP	0.00	0.43	0.43	0.43	0.87	0.00	0.87
Zeus IP	0.00	0.39	0.39	0.39	0.78	0.00	0.78

(b) Percentage by Total IBR Data							
OSINT	IBR1	IBR2	IBR3	IBR4	IBR5	IBR6	IBR7
AlienVault	5.06	5.45	5.19	5.30	5.30	19.80	2.65
Autoshun	0.15	0.16	0.15	0.15	0.15	0.51	0.08
Binary Defense Banlist	0.23	0.30	0.28	0.29	0.29	0.66	0.17
Bruteforce Blocker	0.06	0.06	0.06	0.06	0.06	0.19	0.03
CINS	4.72	5.09	4.86	4.95	4.96	18.48	2.48

Percentages are for those greater than 0.01 for any of the seven IBR datasets.

List Active, are of little relevance due to very low percentages. However, of the OSINT datasets that are relevant, the majority satisfy the completeness criteria and all have similar values across all IBR datasets.

Of the three categories described above, the five OSINT datasets listed in Table 5.2b were identified to be used in later stages of analysis since they recorded the highest percentages in common by both total OSINT data and total IBR data. Figure 5.1 shows a graphical illustration of values in these datasets. The graphs use the five most relevant OSINT and three of the seven IBR datasets (IBR1, IBR6, and IBR7). These three IBR datasets were chosen for the following reasons:

1. Referring to Table 5.2 it can be seen that IBR1 to IBR5 have very similar percentage values. As such, it was decided that presenting only one would be a sufficient illustration of the trend. IBR1 was chosen as it is the first on the list.
2. IBR6 and IBR7 were chosen because these are the two synthetic datasets, and they have very different percentage values to each other and the other IBR datasets.

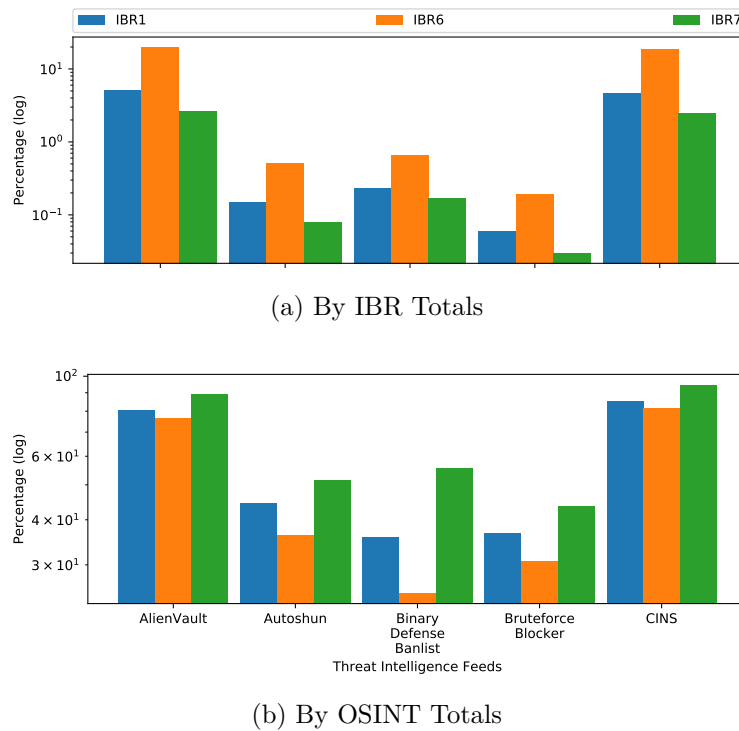


Figure 5.1: Percentage Of Observed IP Addresses In Common

5.2 Change in Datasets

As discussed in Section 2.7.5, TAs regularly change IP addresses in an attempt to conceal their identity (Moore *et al.*, 2006). An IP address may appear for a short period of time to perform a specific exploitation and then once the exploitation is complete it will disappear or change. This turn-over time can be a matter of hours or days. To be the most effective, OSINT is required to keep up-to-date with these changes and report on current IP addresses linked to TAs. To test this, the collected OSINT and IBR datasets were analysed to determine how the data changed on a month-to-month basis over the duration of the study. It is important to consider how the OSINT datasets change because outdated TI adds to the volume of data being processed, which can result in negative effects when it comes to consuming and operationalising TI (Agbenyegah and Asante, 2017). For example, adding too many rules to an edge router has the potential to slow down the processing speed of packets, which in turn affects the request and response time a user experiences. On the other hand, not filtering traffic allows potentially malicious or harmful traffic an avenue into the internal network.

5.2.1 Monthly Change in Dataset Totals

The first test performed determined how the volume of data changed from month-to-month. In the real-world, TA IP addresses change more often than monthly, but analysis into how the OSINT datasets changed from month-to-month provided insight into the general change in the volume of data.

This test was performed on both the OSINT and IBR datasets to create a table of percentage changes in the volume of the datasets. The test took each OSINT and IBR dataset in each month and calculated the total number of IP addresses. With these totals, the percentage change in the composition of the data was calculated from one month to the next. The results of this process are shown in Table 5.3. September 2017 is excluded from the results because this was the first month of data collection and thus there was no previous data to compare against. Table 5.3a displays the percentage change values for the OSINT datasets, Table 5.3b displays the percentage change values for the IBR datasets. These values were plotted as sparklines in Tables 5.3c and 5.3d respectively. The sparkline plots assist to illustrate the trend evident across all datasets over the collection period. Each graph includes a red line indicating the zero axes. The reader should be aware that the scale for the y axis is not the same for all sparklines.

The most significant characteristic noted in Table 5.3d is that the IBR datasets all share a common trend in the increase and decrease of data volume and inflection points, whereas, the OSINT datasets do not. This difference between the OSINT and IBR datasets can be attributed to the fact that, although the IBR datasets are taken from distinct and numerically distant telescopes, the nature of backscatter and IIBR is such that it can target any IP addresses on the Internet. This is likely the reason for the similarity between the change in volume of data seen on all real-world IBR datasets (Irwin and Barnett, 2009). The OSINT datasets, however, share less in common with each other. This is likely due to the fact that although they all report on malicious IP addresses, they each have their own criteria for inclusion of the IP address on their list and, to some extent, different criteria for what is considered a threat.

Although the graphs in Table 5.3c do not share the same trends, they do loosely follow the trends seen in the IBR graphs of Table 5.3d. Aside from Binary Defense (BD) and Autoshun (AS), the OSINT datasets share a large increase in the volume of data between October and November 2017, with AlienVault (AV) and CINS (CI) showing a trend most similar to the IBR dataset. This trend seen in AlienVault (AV) and CINS (CI) data possibly reflects some of their underlying criteria for including an IP address in the list. This is more likely with AlienVault (AV) since they allow the community to contribute IP addresses to their database (AlienVault, 2005). As the volume of data in the IBR datasets decreased between February and March 2018, the OSINT datasets showed a similar reduction in the volume, after which the volume of data started to level out to

Table 5.3: Dataset Percentage Change Values Month-to-Month

(a) OSINT Datasets

Dataset	Months										
	09-10	10-11	11-12	12-01	01-02	02-03	03-04	04-05	05-06	06-07	07-08
AV	72.32	42.21	-1.99	-17.24	-43.29	9.74	40.69	4.80	-0.77	8.78	-2.47
AS	-33.80	-17.33	-44.51	-18.91	-15.08	6.07	47.02	-13.36	-7.74	-19.45	69.34
BD	-25.69	14.09	-19.11	41.26	-7.59	49.74	45.17	-31.90	-1.78	-8.92	2.04
BB	47.86	19.72	-27.30	-16.46	7.37	5.78	-16.73	-32.71	-20.91	3.33	6.60
CI	75.29	26.09	4.06	-16.95	-39.33	50.27	-1.31	3.25	2.74	6.96	-7.15

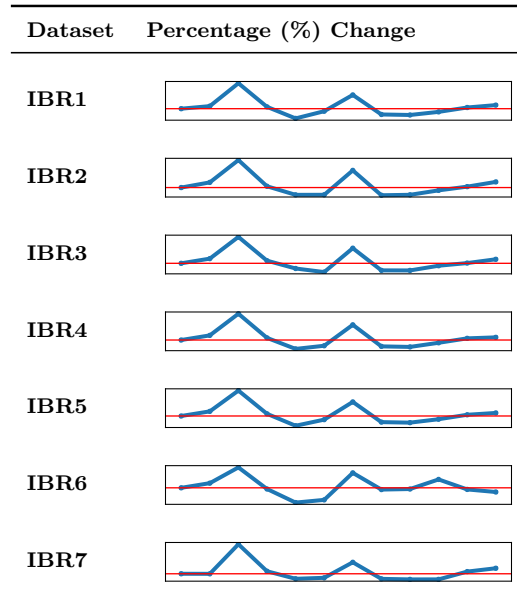
(b) IBR Datasets

Dataset	Months										
	09-10	10-11	11-12	12-01	01-02	02-03	03-04	04-05	05-06	06-07	07-08
IBR1	5.10	50.58	3.65	-19.32	-5.01	27.39	-11.47	-12.50	-6.33	2.36	7.12
IBR2	9.53	50.73	2.35	-13.47	-13.40	31.70	-14.21	-13.41	-5.21	1.67	10.40
IBR3	8.91	50.29	5.01	-9.88	-16.71	28.80	-13.46	-13.59	-4.63	0.45	7.64
IBR4	8.83	50.09	4.08	-16.96	-10.93	28.99	-12.11	-13.15	-5.28	3.16	5.06
IBR6	8.89	39.04	-2.24	-28.26	-23.09	28.62	-3.37	-2.34	16.03	-3.08	-8.01
IBR7	-0.12	59.71	5.11	-10.12	-8.17	23.21	-10.27	-11.40	-11.35	4.39	11.15

(c) OSINT Datasets Sparklines



(d) IBR Datasets Sparklines



Legend: AV; AlienVault; AS: Autoshun; BD: Binary Defense Banlist; BB: Bruteforce Blocker; CI: CINS

a point of increase. In March/April 2018 when the IBR datasets increased in volume, the OSINT datasets once again showed a similar trend, also showing an increase in the volume of data.

Although some of the datasets share a similar trend in the change in the volume of data, the lowest and highest points of inflection are offset. Initially, the OSINT datasets peaked

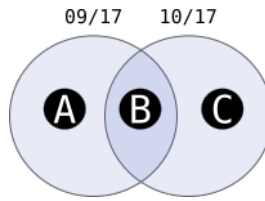


Figure 5.2: Change in Data Set Example

in October 2017 with a gradual increase in November 2017, while the IBR datasets experienced a small increase in October 2017 with the largest increase and peak in November 2017. At the next major point of inflection, the IBR datasets peaked first in March 2018, with the OSINT datasets peaking later in April/May 2018.

The results of this first test showed that some OSINT datasets share characteristics with the IBR datasets. These characteristics start to suggest that IBR data can provide a form of usable OSINT to protect against a certain category of TAs on the Internet.

5.2.2 Monthly Change in Dataset Composition

To further explore how the composition of the OSINT and IBR datasets change from month-to-month, the data was decomposed for further analysis. The goal was to categorise the IP addresses into three different categories. Figure 5.2 illustrates the following three categories using September and October 2017 as example datasets:

1. **Non-Recurring IPs (point A):** the IP addresses that have been removed from the list. These IP addresses were listed in the previous month but are not listed in the current month.
2. **Recurring IPs (point B):** the IP addresses which have stayed in the list. These IP addresses were listed in both the previous month and the current month.
3. **New IPs (point C):** the IP addresses that are newly added to the list. These IP addresses were not listed in the previous month's data, but are seen in the current month.

To sort the IP addresses into categories, the script developed iterated through each of the five OSINT and IBR datasets placing the IP addresses into one of the three categories discussed above.

The results of this analysis start from October 2017. Once again, this is due to the fact that September 2017 was the first month of data collection, and thus there is not data from previous months to determine the change. A graphical illustration of the results is shown in Figures 5.3 and 5.4, with each sub-figure containing two graphs. The top graph illustrates the *New IPs* and *Recurring IPs*, whilst the bottom graph illustrates the *Non-Recurring IPs*. The values used to plot these graphs can be found in Appendix D.1,

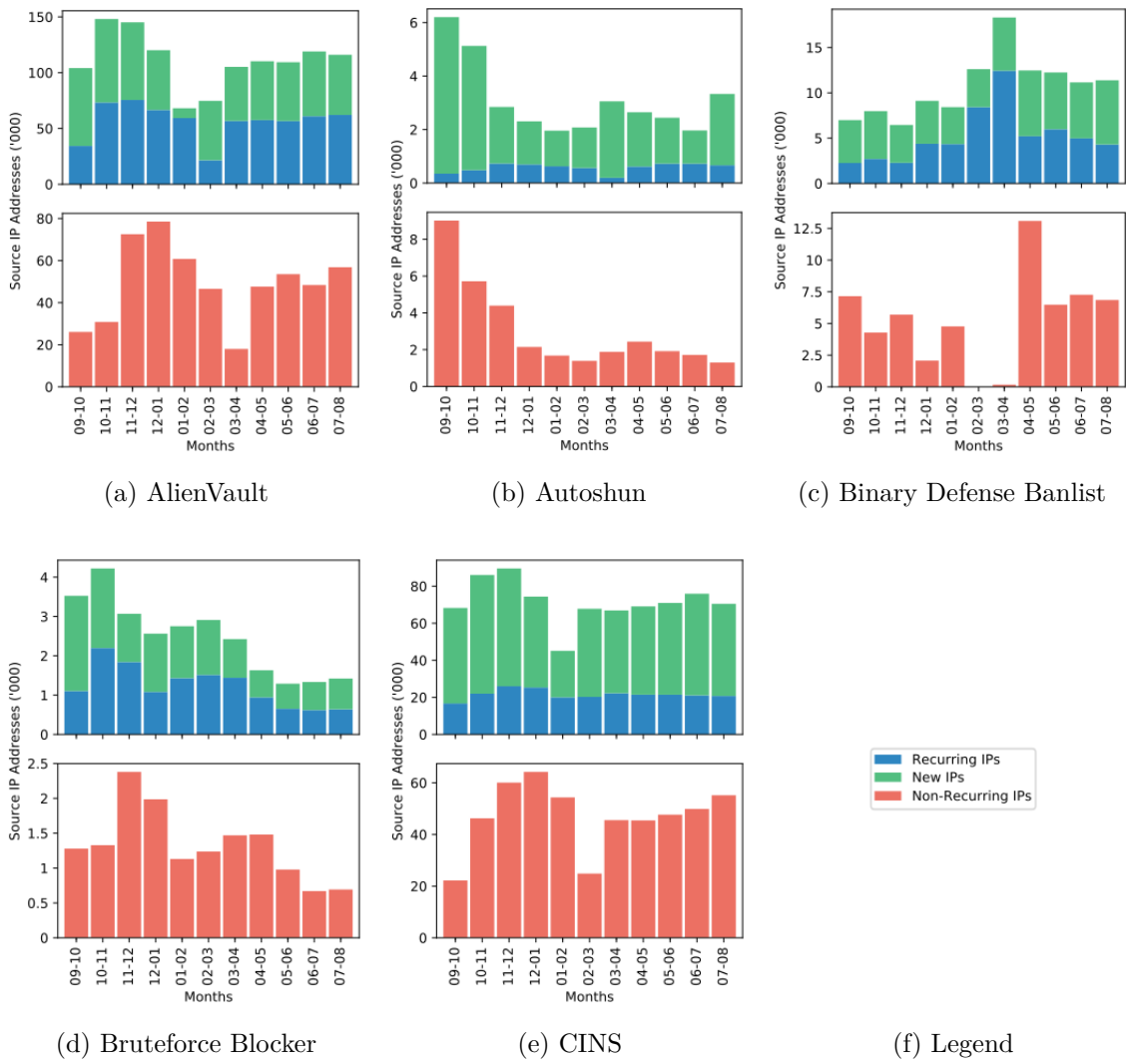


Figure 5.3: Month-to-Month Change in OSINT Data

Tables D.1 and D.2.

The following points must be noted when interpreting the graphs:

- Each point on the months axis represents the change in data between two months. For example, in Figure 5.4a the first point on the axis (09–10) is the change in data between September and October 2017.
- Due to maximum values differing by an order of magnitude, the IP addresses axis for IBR1 through IBR5 in Figure 5.4 have been scaled for readability. Since the range of values for IBR6 and IBR7 were different, they were not set to the same scale. Similarly, the OSINT graphs in Figure 5.3 have a larger range of values so these were set to a different scale.
- The combined total of *New IPs* and *Recurring IPs* make up the total number of unique IP addresses recorded in the current month.

The most significant trend observed in all the graphs in Figure 5.3, with the exception of Autoshun (Figure 5.3b) and Binary Defense Banlist (Figure 5.3c), is a distinctive increase in the volume of data between October 2017 and January 2018, with some graphs seeing this increase continue into February and March 2018. This increase is the result of an increase in the number of *New IPs*, whilst the number of *Non-Recurring IPs* remains low. With Binary Defense Banlist and Autoshun the trend appears to be opposite, with the volume of data in these datasets decreasing when the majority of the OSINT datasets increase in volume.

Another interesting characteristic of the OSINT graphs illustrated in Figure 5.3 is the variation in the number of *Recurring IPs*. On average AlienVault has 51.69% of the total IP addresses recurring from the previous month, Bruteforce Blocker has 49.91% and CINS has 30.83%. The assumption made about the OSINT datasets with higher percentages of *Recurring IPs* is that these IP addresses are linked to persistent TAs (Section 5.4). Conversely, the OSINT datasets with lower percentages of *Recurring IPs* are likely to be linked to a more specific type of TA such as C&C servers and DoS attacks. Comparing the percentage of *Recurring IPs* in the OSINT data to that of the IBR datasets showed that the IBR datasets have fewer IP addresses recurring from month-to-month. IBR1 through IBR5 all have an average of 15% of total IP addresses recurring from month-to-month. However, the combined sets of IBR6 and IBR7 had an average of 24.4% and 22.96% respectively. The higher *Recurring IPs* seen in the OSINT datasets compared to the IBR datasets, suggests that the IBR datasets have a higher churn rate of IP addresses than the OSINT datasets cater for.

The IBR dataset graphs in Figure 5.4 display a similar trend to the OSINT graphs discussed in Figure 5.3. In comparison to IBR1 through IBR5, IBR6 and IBR7 have significantly different volumes of data, with IBR6 displaying a lower than average volume, and IBR7 displaying a higher than average volume. This difference in the volume of data

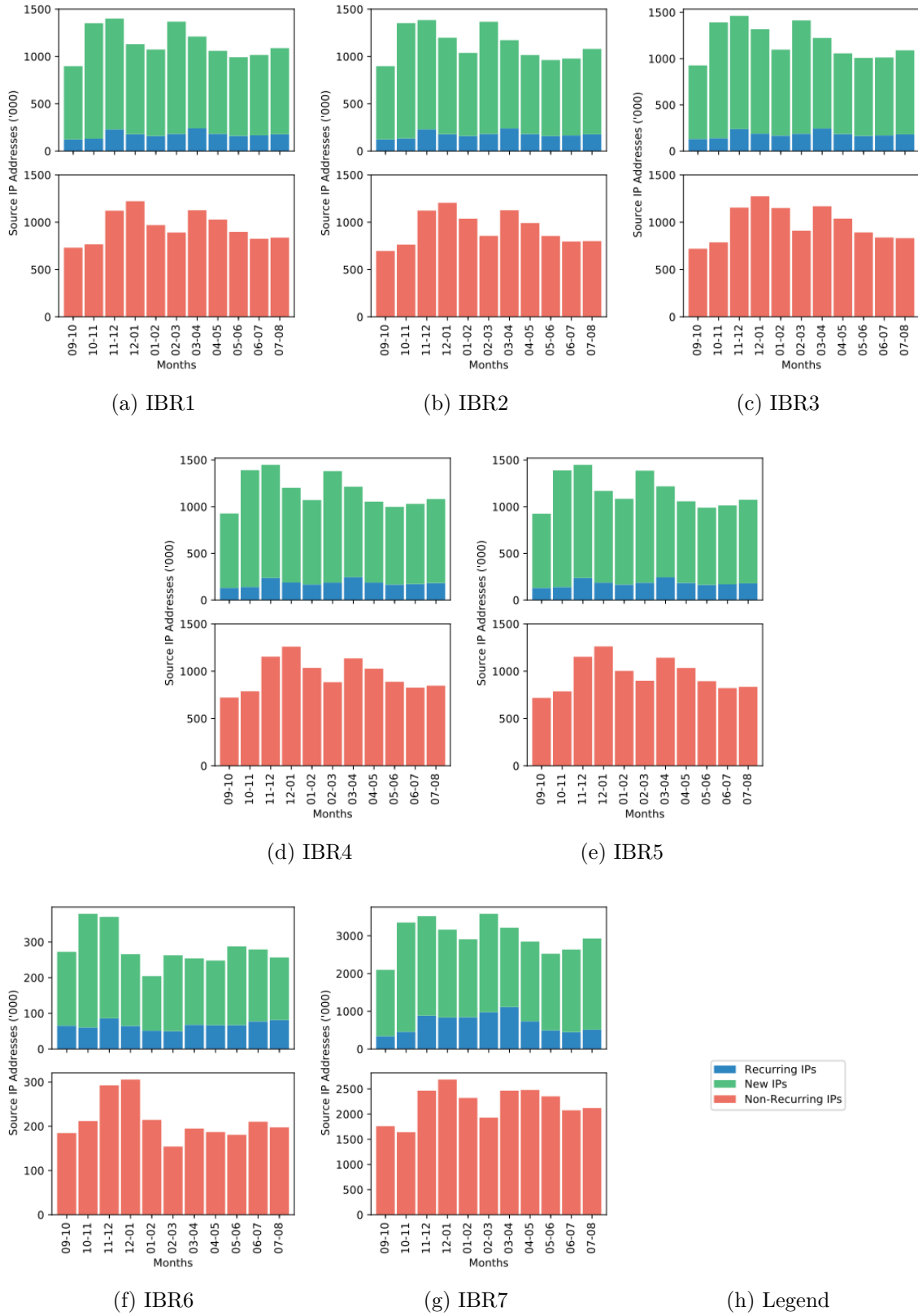


Figure 5.4: Month-to-Month Change in IBR Data

is expected in IBR6 and IBR7 because of the nature of the datasets (see Section 3.3.2). Despite, this difference, all the datasets display an increase in the volume data during the period of October through December 2017, with the highest levels recorded during November and December 2017.

All seven IBR datasets displayed a decrease in the volume of observed data until February and March 2018. After this, all IBR datasets, excluding IBR6 (Figure 5.4f), recorded another increase in the volume of data. Given that IBR6 does not record this increase in the volume of IP addresses, it is evident that the increase seen originates from IP addresses that were not common across all telescope sensors³.

Figures 5.3 and 5.4 show that it is possible to form an understanding of monthly change in composition of the datasets and how the OSINT data relates to the IBR data. The change in OSINT datasets provides useful insight from two perspectives. Firstly, it illustrates the churn rate of IP addresses being used by TAs. Secondly, it provides consumers of OSINT with an understanding of how long it takes for the data to become outdated. It is important for a consumer to consider these characteristics when managing their OSINT strategies. If a consumer is adding IP addresses from an OSINT feed into a blocklist, they need to know how relevant the data will be in a day, a week or a months time (Sillaber *et al.*, 2016). If the consumer is using outdated information, the OSINT strategy becomes less effective or even redundant. This type of information is also relevant to consider for the *Publishing* component of the system described in Chapter 6 since the churn rate of an OSINT dataset will determine how often the system publishes new TI.

5.3 Offset Comparison

One aspect of the CART framework discussed in Section 1.2 is timeliness. The greater the time between a threat emerging on the Internet and an OSINT feed publishing it, the less effective the OSINT will be in protecting against the threat (Rattan *et al.*, 2017). To evaluate how effective the OSINT feeds are at publishing timely data, an offset comparison was done on the two datasets.

To achieve this evaluation, a comparison was done between each month of IBR and OSINT data. The script executed takes one month of OSINT data and determines the number of IP addresses in common for each of the 12-months of IBR data. The purpose of this was to determine how many IP addresses in the OSINT datasets appear before the list was published, and how long the IP addresses stay active after the list was published. For example, November 2017 data from IBR1 was compared against data from AlienVault for September 2017 through August 2018.

³Investigation into this is beyond the scope of this research

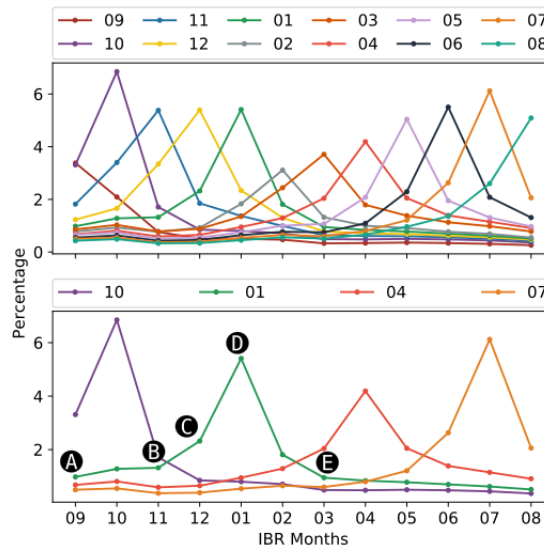


Figure 5.5: IBR1 Monthly Offset Comparison CINS

After analysing all OSINT datasets, the results illustrated two distinctive styles of plots, Saw Tooth (Section 5.3.1) and Bell Shape (Section 5.3.2). This section uses one of each style for discussion, but the full results can be found in Appendix D.2. Plots such as Autoshun, and to some extent Binary Defense Banlist, fall into the "uncategorized" category since these plots do not produce a trend relevant for this analysis.

AlienVault and CINS were chosen to be discussed in this section because they provide clear examples of the two different styles of graphs. The graphs were taken from the IBR1 set of results because it is a real-world dataset and it shares very similar trends to the other four datasets.

Figures 5.5 and 5.6 each contain two graphs. The top graph plots all 12-months of data and the bottom graph plots a selection of the 12-months. The bottom graph has been presented to illustrate the trends displayed in the top graph, but with fewer curves so it is clearer. The months in the bottom plot were chosen based on relevance to the discussion points made. Each curve in the plot represents a single month of OSINT data. When discussing the curves, reference is made to the persistent level, which is the minimum value on the y axis. These persistent IP addresses are discussed further in Section 5.4. The highest inflection point is also referred to in the discussion. This signifies the point when the lists are published since and is the highest because datasets for OSINT and IBR datasets from the same month.

5.3.1 Saw Tooth

In this style, the highest point of inflection is when the OSINT and IBR datasets of the same month are compared. This point moves with each month compared. An example of this style is illustrated in Figure 5.5. A Saw Tooth trend is characterised by a rising curve in the months leading up to the highest inflection point, and a declining curve leading from the highest inflection point. To illustrate this, consider the January 2018 curve in Figure 5.5. Points **A** and **E** represent the persistent level. There is a subtle increase in the percentages between **A** and **B** of 0.34% over two months. As the curve approaches **D**, the percentages increase faster, 1% in a month. In the month before **D** the percentages significantly increase by 3.09%. Conversely, leading from the highest inflection point (**D**), the inverse behaviour can be seen with the percentages decreasingly rapidly in the first month then slower in the months following. The main difference between the increase leading up and decrease leading away from the highest inflection point being the time it takes to get back to the persistent level (**E**).

The sharp increase in the percentage of IP addresses in common prior to the highest point of inflection suggests that a portion of the IP addresses published in the OSINT list started appearing on the telescopes in the months prior to the OSINT list being published. This trend is repeated for each of the other months of OSINT data used.

Of the six OSINT datasets compared, CINS was the only dataset to illustrate this style or curve. This type of behaviour suggests that OSINT list like CINS have a higher churn rate of IP addresses. This is supported by the results seen in Section 5.2.1, where CINS has a high percentage of IP addresses changing from month-to-month, with little fluctuation in values compared to other lists.

5.3.2 Bell Shape

In this style, the highest point of inflection occurs when OSINT and IBR datasets of the same month are compared. However, unlike the Saw Tooth style, the inflection points are surrounded by high percentages in the months before and after. An example of this plot is illustrated in Figure 5.6. Using April 2018 as an example for description of this style, there is a steady increase in the percentages between **A** and **B** of 1.12% over two months. However, two months prior to **D** (the common month) the percentages increase between **B** and **C** by 1.86%. In the month prior to **D**, the curve continues to increase but by a smaller amount (0.79%). After the highest inflection point (**D**), the curve drops to a percentage closer to the persistent level (**E**) in a month, though slightly slower than in the Saw Tooth style. However, similar to the Saw Tooth style, is the long steady decline towards the persistent level at **F**.

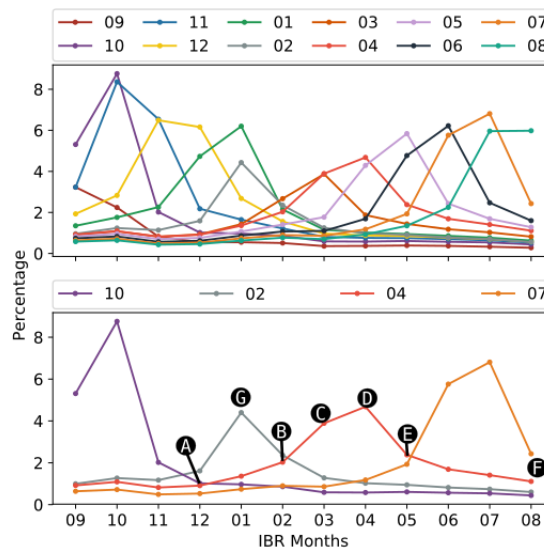


Figure 5.6: IBR1 Monthly Offset Comparison AlienVault

One anomaly in Figure 5.6 is the February 2018 curve. The highest inflection point (**G**) occurs in January 2018, with 4% of IP addresses in common, compared to 2% of IP addresses in common when the OSINT and IBR months are the same. Evaluation of why this occurs was not performed, but this curve illustrates an example of how collecting OSINT and implementing it without understanding of the context can be a waste of time and money since the information may not be relevant to the organisations requirements.

Of the additional figures in Appendix D.2, Binary Defense Banlist and Bruteforce Blocker display this Bell Shape pattern. However, there are some differences in the style of these curves:

- Some of the curves in Binary Defense Banlist have more than one highest inflection point in the percentage values.
- The persistent level of Bruteforce Blocker stays constant as a straight line up until the point close to the common months when the points start to increase. The curves lack the gradual increase in prior months characteristic of a typical Bell Shape.
- Both Binary Defense Banlist and Bruteforce Blocker have significantly lower percentage values than AlienVault and CINS. Therefore, the change in percentage values is smaller for Binary Defense Banlist and Bruteforce Blocker.

5.3.3 Discussion

The general trend displayed from this analysis was that the percentage of IP addresses in common between the OSINT and IBR datasets started to increase in the months prior

to publishing the OSINT lists⁴. In Saw Tooth style curves, the volume of IP addresses increased at a slow rate up until a month before the list was published when they increased sharply. Despite the delay between the IP address appearing in the IBR data and being published in the OSINT feed, the largest increase was between the month prior to publishing and the month in which the data was published. In comparison, the Bell Shape style showed the same small increase in IP addresses in the months prior to the publishing date, with a large increase between two and three months prior to the data being published. The exception to this was lists such as Binary Defense Banlist which saw a large increase up to four months prior to the publishing date.

In an attempt to find possible causes for these increases, three OSINT feeds (AlienVault, CINS, and Binary Defense Banlist) were used to identify the top five destination ports being targeted. Using January 2018 as a randomly chosen month of OSINT data, the top five ports were identified by counting the number of IP addresses which attempted to communicate with each port. The results showed that of the top five ports, AlienVault and CINS had 23/tcp, 2323/tcp, 81/tcp, and 22/tcp in common, in rank order. The only difference between the two feeds was the fifth port, 9000/tcp for AlienVault and 80/tcp CINS. Binary Defense Banlist showed a very different set of results, with the top five ports in rank order being 22/tcp, 2222/tcp, 445/tcp, 23/tcp, and 3389/tcp. What all these ports have in common is that they are linked to scanning attempts. A discussion on these and other ports can be found in Section 5.5.3.

The ports identified suggest that the OSINT lists are publishing IP addresses linked to malicious scanning activity on the Internet. In addition to this, the high percentage of IP addresses in common between OSINT lists such as AlienVault, CINS, and Bruteforce Blocker Banlist in Table 5.2 suggest that a large portion of some OSINT lists are made up of IP addresses linked to malicious scanning or botnets.

The two styles of curves discussed above highlight the need for consumers to understand the types of OSINT feeds available and how they can be used effectively. Furthermore, it highlights the fact that TA IP addresses are ever-changing, and that today's TI may not be relevant tomorrow (Tounsi and Rais, 2017). This analysis also highlighted the importance of collecting, analysing, and disseminating TI in the shortest amount of time (Rattan *et al.*, 2017).

In terms of the timeliness aspect of (CART) (Sillaber *et al.*, 2016), this analysis showed that a one month delay between the IP address appearing on the telescopes and the OSINT feed published is enough time for the quality of the OSINT to be affected. In comparison to the Bell Shape curve in Figure 5.6, the Saw Tooth pattern illustrated in Figure 5.5 shows a relatively small increase in the IP addresses leading up to the month in which data was published. This small increase before publishing is more desirable because it means there are less IP addresses evading the OSINT lists. However, since this lag is seen

⁴signified by the highest inflection point on the graph

```

input_a: An OSINT or IBR dataset to analyse
input_b: A list of  $n$  dates
output: A list of persistent IP addresses.

persistent = create an empty list to store persistent IP addresses;
for dataset in input_a do
  for date in input_b do
    ips = get a unique list of IP addresses from the dataset for the given date;
    if there are no IP addresses in the persistent list then
      | add all IP addresses from ips into the list;
    else
      | perform an intersection update on the persistent with the IP addresses
      | in ips;
    end
  end
  return the number of persistent IP addresses in the list;
end

```

Algorithm 1: Persistent IP Addresses Algorithm

throughout all OSINT datasets, it can be suggested that a characteristic of OSINT is that there will always be a lag between the time the IP address appears on the telescopes and the time it is published on the OSINT feeds.

The delay seen contributes to the notion that the more TI is shared, the better the chances are of threats being stopped. For example, if a set of telescope sensors detect a surge in requests from a given IP address or netblock, this data can be analysed and disseminated fast enough to protect legitimate networks from the threat if the information is shared⁵.

5.4 Persistent IP Addresses

The analysis described in Sections 5.2 and 5.3 discovered the existence of a portion of the IP addresses on both the OSINT and IBR datasets that stay active from month-to-month, characterised by the persistent level in Figures 5.5 and 5.6. This part of the analysis set out to discover more about these persistent IP addresses. The goal was to discover the number of persistent IP addresses seen over a range of months. Using September 2017 as the starting month, a script was developed to find the persistent IP addresses for n months between September 2017 and August 2018 where n is one of $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ month windows. A window is the number of months of data used. Algorithm 1 describes the steps taken to identify the persistent IP addresses for each dataset over all 11 windows.

⁵Further analysis into this is beyond the scope of this research

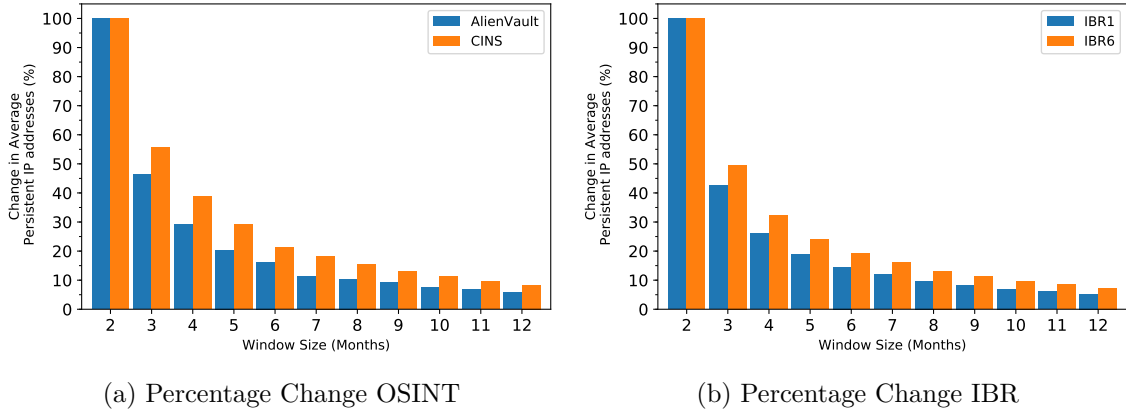


Figure 5.7: Persistent IP Addresses

To illustrate the results of this script, Figure 5.7 plots the values as a percentage using the two-month window size ($x = 2$) as 100%. The full set of values for each of the OSINT and IBR datasets can be found in Appendix D.3. In this discussion, AlienVault and CINS were chosen to represent the OSINT data because they were used in the analysis performed in Section 5.3. IBR1 was chosen because it represents one of the five real-world datasets and the characteristics of the data are very similar to the other four real-world datasets (IBR2 through IBR4). IBR6 was chosen because it holds data on the IP addresses in common across all five telescopes.

Figures 5.7a and 5.7b illustrate an exponential decay in the number of persistent IP addresses as the months increase. This suggests that as the window size increases, the persistent IP addresses become more filtered, leading to a set of only highly persistent IP addresses across 12-months. Using the values for IBR1 from Table D.3, the line of best fit can be approximated as $y = 109943e^{-0.258x}$ with an R^2 of 0.907, where x is the window size. With this equation it is possible to estimate that after 24 (x) months, the persistent IP addresses are predicted to reach 224 (y) on IBR1. It is expected to take approximately 100 months before the number of persistent IP addresses seen on IBR1 reaches less than 1. Applying this to the other real-world IBR datasets to estimate future values produced similar results. These estimates are calculated without considering other factors inherent to IP addresses and therefore can only be used as a loose estimation. However, this prediction serves as a way to show how the data can be used to predict possible future trends.

The remaining sections present further analysis conducted on these persistent IP address, with a specific focus on the highly persistent IP addresses⁶.

⁶IP addresses common across 12-months of data

5.5 Highly Persistent IP Addresses

Having identified a set of highly persistent IP addresses⁷ on both the OSINT and IBR datasets, this research moved to discover more about the IP addresses themselves and the types of activities they are linked to. The aim was to understand why these IP addresses are persisting and how this data can be used to provide suggestions towards blocking them. This analysis will assist in creating a form of TI that can be used for a specific type of threat.

This section focused on the IBR datasets because they record the real-world data and provide true insight into the potentially malicious IP addresses.

In this analysis the following questions were asked:

1. What portion of the total unique IP addresses and total requests do the persistent IBR IP addresses constitute?
2. What percentage of the persistent IBR IP addresses appear in the OSINT lists?
3. What services are the persistent IP addresses targeting from both OSINT and IBR datasets?
4. On a month-to-month basis, how does the volume of requests to a port change?

5.5.1 Volume of Persistent IP Addresses

Highly persistent IP addresses are significantly easier to block because they are active for longer (more than one month) periods of time. This test iterated through each of the IBR datasets and determined what percentage of the unique IP addresses and the total packets these IP addresses constitute.

The results of the test are displayed in Table 5.4, which shows the number of IP addresses for each of the seven IBR datasets, the percentage of unique IP addresses and the percentage of total requests these persistent IP addresses constitute. The percentages were not calculated for IBR6 and IBR7 since these are synthetic datasets.

The values in Table 5.4 show that the IP addresses contribute a significant portion of traffic in comparison to the portion of the datasets they represent. The percentage of unique IP addresses suggests that on average the persistent IP addresses constitute a relatively small percentage (0.635%) of the total IBR data. However, they account for an average of 7.79% of the total packets on the telescopes.

This test highlighted that, although OSINT lists do a good job of publishing data on these IP addresses, there is a scope for the amount of highly persistent IP addresses being

⁷Simply referred to as IP addresses in this section

Table 5.4: IBR Persistent IP Addresses

Dataset	Persistent	Unique IP Addresses (%)	Packets (%)
IBR1	6 535	0.065	7.85
IBR2	6 718	0.068	7.48
IBR3	7 827	0.075	7.82
IBR4	7 914	0.077	7.85
IBR5	7 811	0.076	7.96
IBR6	4 818	0.195	–
IBR7	16 949	0.079	–

Unique IP addresses (%) is calculated using the totals from Table 3.8.

Packets (%) is calculated using the totals from Table 3.6a.

Table 5.5: IBR Persistent IP Addresses in OSINT Lists

OSINT List	IBR1	IBR2	IBR3	IBR4	IBR5	IBR6	IBR7
AlienVault	59.42	59.97	51.32	51.11	51.97	68.18	29.10
CINS	60.50	60.87	52.29	51.95	52.77	70.34	29.22

recorded on the OSINT feeds can be improved. The results of test case highlighted that the highly persistent IP addresses in the IBR datasets constitute a fraction of the total IP addresses but a significant percentage of the total packets.

5.5.2 Persistent IP Addresses in Common

In this test case, the aim was to discover how many of the IP addresses seen in the IBR datasets, are published in the OSINT lists. This test compared each set of IP addresses from the IBR datasets against all IP addresses recorded in AlienVault and CINS for the 12-month period. The results of this test are listed in Table 5.5.

The results in Table 5.5 show that the OSINT lists published an average of 53% of the IP addresses from the IBR datasets. The averages from this test are lower than the percentages listed in Table 5.2a, but still above 50% for the five real-world telescopes, and close to 70% for the IBR6 telescope. The reasons for the lower percentages is possibly related to the criteria governing the inclusion of IP addresses in the list⁸. However, the results discussed in the following test will help shed light on the activities these IP addresses are involved in.

The test further identified that there is a need to expand OSINT to be better equipped to eliminate persistent threats on the Internet, by ensuring a larger percentage of persistent IP addresses seen in IBR datasets are published on OSINT lists.

⁸This is beyond the scope of this research

5.5.3 Persistent IP Address Services

Sections 5.5.1 and 5.5.2 showed that the highly persistent IP addresses recorded on the telescopes constitute a significant percentage of the total packets, but the OSINT lists only publish an average of 53% of these IP addresses. The aim of this test case was to identify which services the persistent IP addresses are targeting to determine what type of malicious hosts are being seen on the Internet. With this information, it is possible to make suggestions of ways in which the OSINT can be contextualised and disseminated in a service suitable format.

The AlienVault, CINS, IBR2 and IBR6 datasets were used in this test. AlienVault was used because the OSINT data includes a description field which can be used to match what the IP addresses are said to be doing and what they actually are doing. CINS was used because it has a high percentage of IP addresses in common with the IBR datasets. IBR2 was randomly chosen from the five real-world telescopes, but it shares similar results to IBR1 through IBR5. IBR6 was chosen because it contains IP addresses common on all five telescopes.

UDP ports were not included in this test because there were so few of them. The test started by discovering the top 20 TCP ports by number of IP address hits for each of the five real-world telescopes (RUSCOPE1 through RUSCOPE5). Figure 5.8 illustrates the results of this test for RUSCOPE2 as a treemap. Figures D.8 to D.12 illustrate the values for all five telescopes. All five telescopes illustrate a very similar relative distribution of ports, with some variation in the actual ports being requested. For this reason, it is possible to use RUSCOPE2 to represent the five telescope datasets and as a comparison for the results discussed further in this section. The results in Figure 5.8 show that the most common port being probed is 23/tcp (discussed later), followed by 80/tcp, 8080/tcp, and 445/tcp. The number of IP addresses sending requests to 23/tcp is more than double the second most requested port (80/tcp) as seen in Figure 5.8. This observation is seen throughout all five real-world telescope datasets.

After observing the results in Figure 5.8, the same test was performed on IBR2 and IBR6 using the highly persistent IP addresses. These results are shown in Figure 5.9, with Figures D.15 to D.21 illustrating the full results and values for each of the seven IBR datasets.

Comparing the results in Figure 5.8 and Figure 5.9a to discover the difference between the packets of all traffic and active traffic, shows the change in the type of ports being requested by the IP addresses. The most notable change is the reduction in the ratio of 23/tcp against the other requests. Given that IBR2 is RUSCOPE2 with the passive IP addresses removed, this reduction could suggest the presence of passive IP packets being received on the ports such as 23/tcp. In addition to this, the reduction in the volume of requests to 80/tcp results in an increase in the volume of requests to 445/tcp and

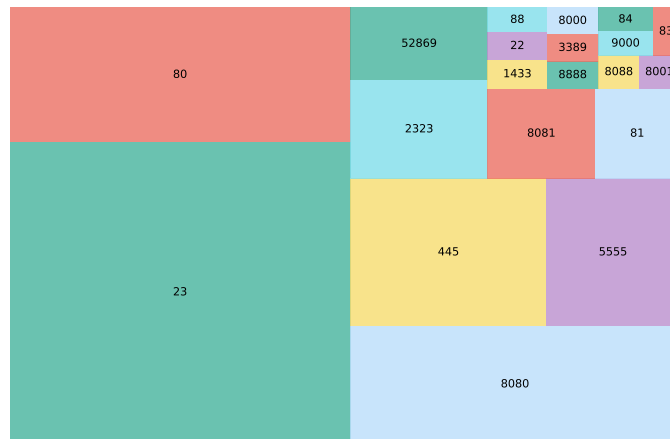


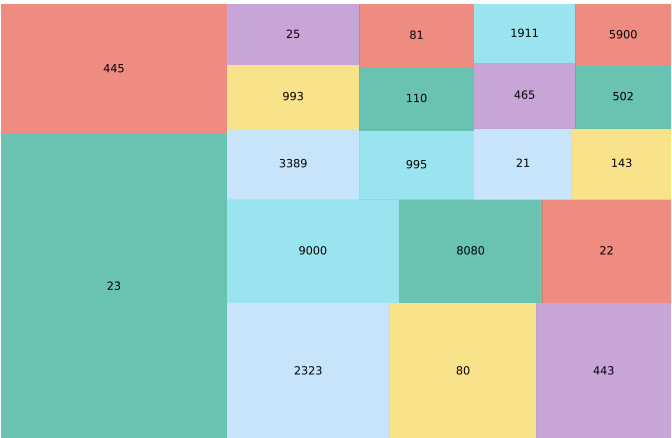
Figure 5.8: Top 20 TCP Ports: RUSCOPE2

2323/tcp. With the exception of 23/tcp and 445/tcp, Figure 5.9a illustrates a more even spread of ports requested compared to Figure 5.8. From Figure 5.9b it is possible to the change in ports being requested by IP addresses seen across all five telescopes. Notably, 2323/tcp and 443/tcp are second and third in IBR6 respectively.

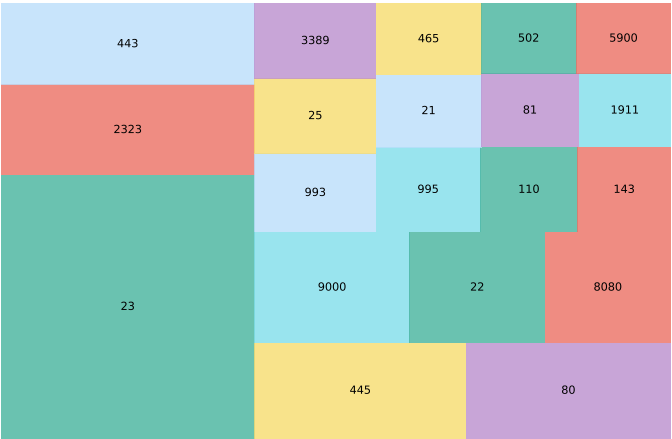
The results discussed above helped identify a set of common TCP ports being requested. However, to discover what the IP addresses on the OSINT feeds are publishing, the same test was performed on AlienVault and CINS. The results of this can be seen in Figure 5.10. AlienVault (Figure 5.10a) and CINS (Figure 5.10b) display similar trends to IBR6 (Figure 5.9b), rather than IBR2, with the relative size of the blocks being closer to the average size of all the other blocks. Figures D.13 and D.14 illustrate the same treemaps with the addition of the values used to plot them.

From Figures 5.9 and 5.10 it is possible to identify the top 10 most popular ports. 23/tcp ranked highest in all the datasets, with 2323/tcp and 445/tcp coming second and third respectively on IBR6, AlienVault and CINS. On IBR2, the order was swapped, with 445/tcp ranking second and 2323/tcp third. Table 5.6 lists the ports that appeared in the top 20 for all four datasets. The port assignments are taken from (Touch *et al.*, 2018).

Of the top 20 ports identified in this test all but 1911/tcp (MTP), 502/tcp (MBAP), 1883/tcp (MQTT), 10250/tcp, 4786/tcp (Smart-Install), 7547/tcp (CWMP), 6379/tcp (REDIS), and 11211/tcp (memcache) rank in the top 1000 ports the Nmap tool uses during its scanning (NullSec, 2016). Additionally, eight of these ports (445/tcp, 1433/tcp, 23/tcp, 443/tcp, 80/tcp, and 5900/tcp) appear in research by Choi *et al.* (2014) where they discuss the top 10 destination pots detected on darknet traffic. However, the rank of these IP addresses have changed, 445/tcp used to top charts but was overtaken by 23/tcp



(a) IBR2

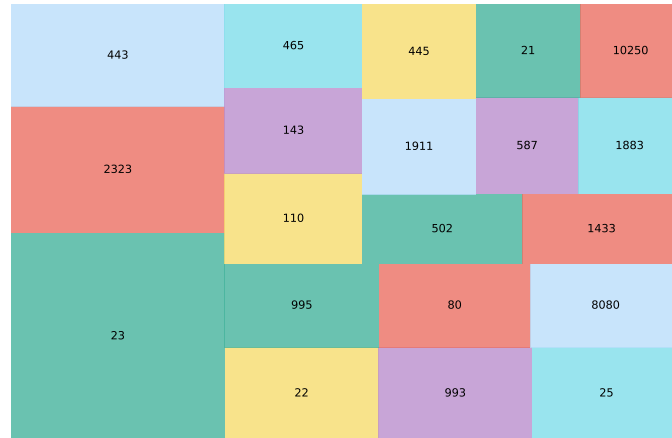


(b) IBR6

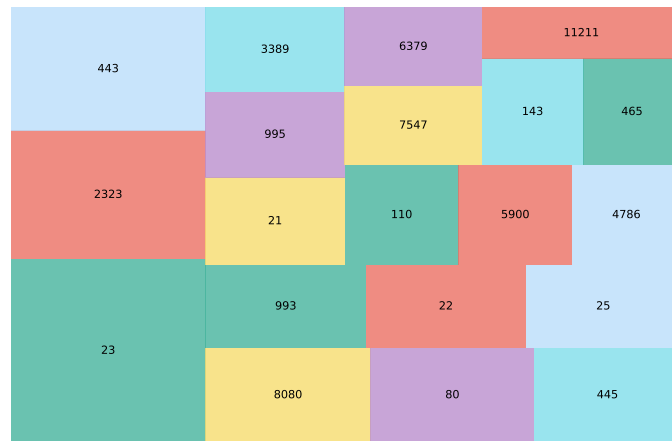
Figure 5.9: Top 20 TCP Ports for IBR Datasets

and 2323/tcp in late 2016 when the Mirai botnet emerged (Antonakakis *et al.*, 2017).

The results of this test highlighted that the majority of the highly persistent IP addresses on both OSINT and IBR datasets are linked port scanning (Section 2.7.3) detected on the telescopes. Using this information, a comparison was made with the descriptions published in AlienVault. As part of the OSINT data, AlienVault includes a description field that states the reasons for publishing the IP addresses (AlienVault, 2005). To determine how these IP addresses are categorised on AlienVault, the description for each IP address was pulled from the database. The OSINT data from AlienVault contains three



(a) AlienVault



(b) CINS

Figure 5.10: Top 20 TCP Ports For OSINT Datasets

categories of threats, *malicious host*, *scanning host*, and *spamming*. From Table 5.7 it can be seen that close to all the IP addresses have been classified as a malicious host. The results of the analysis performed in this section suggested that the majority of IP addresses seen on the telescopes are port scanners. However, there is not enough information to form a definitive conclusion⁹.

⁹This research has been included in the future work section (Section 7.4)

Table 5.6: Top Targeted TCP Ports

Port	Service	Description
21/tcp	FTP	One of the most common ports scanned when probing an Internet host to determine what services are running on it (Hacker Target, 2011, Durumeric <i>et al.</i> , 2014).
22/tcp	SSH	See 21/tcp
23/tcp	Telnet	One of the most common ports used in scanning tools. However, scans to this port have increased exponentially from Mirai and Mirai variant botnets attempting to self propagate (Antonakakis <i>et al.</i> , 2017).
25/tcp	SMTP	See 21/tcp
80/tcp	HTTP	See 21/tcp
110/tcp	POP3	See 21/tcp
143/tcp	IMAP	See 21/tcp
443/tcp	HTTPS	See 21/tcp
445/tcp	MS-DS	One of the most common ports used in scanning tools. Scans targeting this port are potentially probing for the Windows SMB service was exploited by the WannaCry ransomware worm on this port (Hodgman, 2017), as well as the Conficker(Irwin, 2013).
465/tcp	SMTPS	Description
993/tcp	IMAPS	Description
995/tcp	POP3S	Description
2323/tcp	3D-NFSD	This port is often used as an alternative port for 23/tcp for Telnet. As a result, Mirai and Mirai like botnets are probing this port in an attempt to propagate themselves (Kambourakis <i>et al.</i> , 2017).
8080/tcp	HTTP-ALT	See 21/tcp

Table 5.7: AlienVault Description

Dataset	Description	Count
AlienVault	Malicious Host	1953
	Scanning Host	3
	Spamming	20
CINS	Malicious Host	1335
	Scanning Host	2
IBR2	Malicious Host	4027
	Scanning Host	2
IBR6	Malicious Host	3283
	Scanning Host	1

5.5.4 Month-to-Month Change in Requests

While the previous test looked at the ports over the whole 12-month period, in this test, the ports are analysed on a month-to-month basis. The goal was to discover the change in the volume of IP addresses sending packets to a specific port. This test used the highly persistent datasets for AlienVault and IBR6 to determine the number of IP addresses sending requests to a given port. This test includes UDP packets because the data was available to use. Tables 5.8 and 5.9 present the results of this analysis.

The graphs in Table 5.8 show that, on average, the top five TCP ports in Table 5.8a stay at a constant level. As the rank decreases, there is more fluctuation in the volume of

Table 5.8: Top 20 TCP Ports Requested

(a) AlienVault					(b) IBR6			
Rank	Port	Avg	Total	Graph	Port	Avg	Total	Graph
1	23	1322	15870		23	3302	39625	
2	2323	994	11933		2323	1477	17724	
3	443	614	7369		80	1021	12261	
4	80	532	6386		443	993	11916	
5	25	441	5298		445	970	11643	
6	993	425	5110		9000	864	10379	
7	22	373	4483		22	567	6813	
8	110	353	4240		25	510	6122	
9	143	352	4230		993	482	5792	
1	995	351	4214		3389	421	5054	
11	502	330	3968		995	413	4956	
12	1433	328	3938		8080	401	4815	
13	465	327	3924		81	398	4780	
14	1911	296	3563		110	396	4755	
15	445	291	3494		143	394	4728	
16	587	275	3311		21	376	4516	
17	21	274	3292		465	370	4446	
18	1883	266	3202		502	361	4335	
19	7547	265	3186		1883	328	3937	
20	8080	253	3043		7547	321	3858	

Total is over the 12-month collection period, and Avg is by month.

packets per month. The same is evident in Table 5.8b, but the lower ranking ports exhibit larger changes in the values. Comparing these results with the UDP ports in Table 5.9 shows similar trends. The top five ports stay relatively constant with the exception of 19/udp which appears to drop close to zero in the final months of collection.

Table 5.9: Top 20 UDP Ports Requested

(a) AlienVault					(b) IBR6				
Rank	Port	Avg	Total	Graph	Port	Avg	Total	Graph	
1	53	402	4824		53	450	5400		
2	123	342	4112		123	375	4506		
3	47808	317	3807		47808	368	4422		
4	161	286	3432		19	307	3693		
5	19	266	3201		161	289	3477		
6	1900	151	1823		1900	182	2190		
7	5353	78	936		137	110	1324		
8	137	72	875		1434	95	1147		
9	1434	64	769		5353	94	1137		
10	5060	63	762		389	85	1031		
11	389	58	704		623	82	984		
12	5351	58	703		5351	76	913		
13	623	54	658		5060	74	892		
14	5632	44	536		520	67	814		
15	17185	44	535		17	65	788		
16	44818	43	524		111	64	777		
17	520	40	487		17185	62	745		
18	17	37	455		11211	44	538		
19	69	35	426		500	41	494		
20	523	34	411		69	35	429		

Total is over the 12-month collection period, and Avg is by month.

By comparing Tables 5.8 and 5.9 it is possible to see that there is no month where scanning is not occurring. This analysis was limited to the number of IP addresses sending packets to the ports. However, based on the results from this test case, there is reason to believe it is not the same IP addresses probing the network ports on a month-to-month basis¹⁰.

5.6 Source IP Address Evaluation

Having performed analysis into the ports being targeted by probes and port scanners, this section provides additional data obtained from the OSINT and IBR dataset analysis. The section details the analysis performed on the highly persistent AlienVault and IBR6 datasets described in the section above. The analysis and results described in this section present a set of principles that can be applied to larger datasets (such as IBR1 through IBR5) during further research or enrichment.

Before the analysis could be performed, the IP addresses were enriched with additional information such as country code (CC), AS number (ASN), registry, AS name, and BGP prefix number. This was achieved using the Python2 BulkWhoIs library¹¹ to query the Team Cymru¹² database (see Appendix B.2). With the IP addresses enriched, the analysis could be performed.

5.6.1 Registry Grouping

Using the *ipv4-heatmap*¹³ tool, the highly persistent IP addresses were plotted on a heatmap with the intention of illustrating the TA source. Figure 5.11 illustrates the results for AlienVault (Figure 5.11a) and IBR6 (Figure 5.11b). Comparing the two heatmaps reveals that AlienVault has fewer IP addresses in general, while both figures illustrate a grouping of IP address in similar regions. The largest grouping of IP addresses is in APNIC¹⁴; an Asian Pacific registry, followed by RIPE¹⁵; a Europe, West Asia and USSR registry, and ARIN¹⁶; an American registry for Canada, United States, Caribbean and North Atlantic islands.

¹⁰Research into this is beyond the scope of this paper, and is put in future work

¹¹<https://pypi.org/project/BulkWhois/>

¹²<https://www.team-cymru.com/>

¹³*ipv4-heatmap*

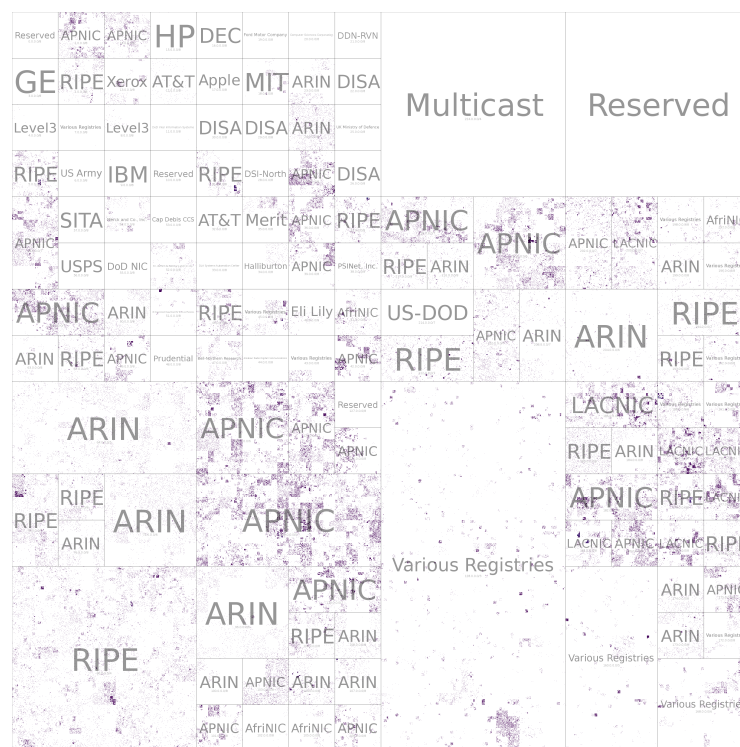
¹⁴<https://www.apnic.net/>

¹⁵<https://www.ripe.net/>

¹⁶<https://www.arin.net/>



(a) IBR6



(b) AlienVault

Figure 5.11: OSINT and IBR Heatmaps

Table 5.10: Top 10 Countries

(a) A			(b) B			(c) C		
CC	IPs	%	CC	IPs	%	CC	IPs	%
CN	591 199	17.76	US	514	26.01	US	1 322	27.44
BR	316 819	9.52	CN	308	15.59	TR	403	8.36
US	203 084	6.10	CH	227	11.49	CN	328	6.81
JP	198 253	5.96	RO	74	3.74	MN	256	5.31
RU	192 082	5.77	PL	69	3.49	CH	242	5.02
AR	153 648	4.62	IN	67	3.39	IN	214	4.44
MX	148 805	4.47	TW	62	3.14	TW	181	3.76
IN	130 982	3.93	RU	61	3.09	RU	163	3.38
CO	123 178	3.70	BR	60	3.04	BR	134	2.78
KR	99 361	2.98	TR	58	2.94	KR	131	2.72

A: IBR6 for the whole 12-months

B: Highly persistent IP addresses from AlienVault

C: Highly persistent IP addresses from IBR6

Percentages for A are calculated using IBR6 values from Table 3.8, percentages for B and C are calculated using values from Table 5.4.

This grouping information alone is not enough to draw conclusion as to which region the highest threat comes from. APNIC, ARIN, and RIPE are three of the five regional registries which assign IP addresses to many countries around the world (RIPE NCC, 2015). For this reason, stating that IP addresses originating from a particular region are malicious, is not an effective ranking criteria, however, it is a start to drawing conclusions about the source, or for future research. In addition to this, TAs spoofing source IP addresses (Section 2.7.5) may use a list of IP addresses from a single network or regional registry, resulting in higher values being seen from certain regional registries.

5.6.2 Geopolitical Analysis

Continuing the analysis from the previous subsection, this analysis evaluated the source IP addresses by country of origin. For the purpose of this analysis, countries are referred to using the ISO 3166¹⁷ two-letter short-code. This test used three datasets, IBR6 IP addresses for all 12-months, highly persistent IBR6 and highly persistent AlienVault IP addresses. Table 5.10 lists the top 10 countries for each of the three datasets used.

The complete set of results of the evaluation were used to create a choropleth map illustrated in Figure 5.12. Figure 5.12a showed the widest distribution of IP addresses, with the highest volume originating from China (CN), followed by various countries in North and South America. Figures 5.12b and 5.12c illustrate the distribution of IP addresses for the highly persistent OSINT and IBR6 datasets respectively. Comparing Figure 5.12b to Figure 5.12a, it can be seen that the US has the highest number of highly persistent

¹⁷http://www.iso.org/iso/country_codes.htm

ASN	AS	IPs	%
9121	TTNET, TR	316	6.56
7018	ATT-INTERNET4 - AT&T Services, Inc., US	293	6.08
45204	GEMNET-MN GEMNET LLC, MN	253	5.25
42570	EU-CH-KS.PORTSCANNING, CH	242	5.02
6939	HURRICANE - Hurricane Electric LLC, US	229	4.75
36375	UMICH-AS-5 - University of Michigan, US	224	4.65
4134	CHINANET-BACKBONE No.31,Jin-rong Street, CN	187	3.88
7922	COMCAST-7922 - Comcast Cable Communications, LLC, US	187	3.88
3462	HINET Data Communication Business Group, TW	153	3.18
NA	NA	90	1.87
Total		2174	45.12%

Percentages are calculated using the 12-month value for IBR6 from Table D.3b

Table 5.11: Top 10 ASes: IBR6 Highly Persistent IP Addresses

IP addresses, overtaking China by close to 10%. From the discussion in Section 5.5 it is possible to conclude that these IP addresses are port scanners.

A comparison of Figure 5.12b to Figure 5.12c was done to understand how the IP addresses on OSINT feeds relate to IBR datasets. The figures show that AlienVault has a large portion of IP addresses from North and South America that are not seen in the IBR6 datasets. These results can aid in the decision making process on what OSINT feeds to use for individual organisations.

5.6.3 AS Names

The aim of this analysis was to discover which ASes were responsible for the most amount of IP addresses recorded on the telescopes. The test sought to identify the top 10 ASes by the number of IP addresses contained in each, and the percentage of the total packets that these IP addresses constitute.

Table 5.11 presents the top 10 ASes by the number of IP addresses contained within each. From this table it is evident that the top 10 ASes account for 45.12% of the highly persistent IP addresses on IBR6. Given that these 10 ASes are 1.11% of the total (899) number of ASes in the dataset, the percentage of IP addresses is significantly high.

With the IP addresses from these 10 ASes, this test continued to count the number of packets the total of 2174 IP addresses had in each of the five telescopes. The results are illustrated in Table 5.12. From Table 3.8 presented earlier in this research, it is possible to calculate that these 2174 IP addresses are only 0.09% of the unique, active IP addresses. However, these IP addresses account for an average of 2.38% of the packets sent to the telescopes. These results further illustrate the notion that eliminating the small percentage of highly persistent IP addresses could significantly reduce the volume

Telescope	Packets(%)
RUSCOPE1	2.61
RUSCOPE2	2.11
RUSCOPE3	2.37
RUSCOPE4	2.38
RUSCOPE5	2.41

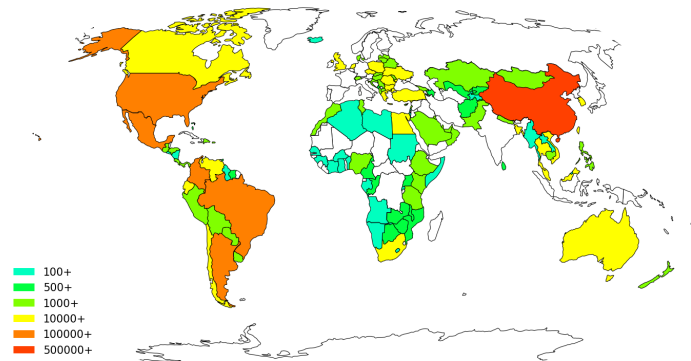
Percentage is calculated using the 12-month value for IBR6 from Table 3.6a

Table 5.12: Packet Count of Top 10 ASes IP Addresses

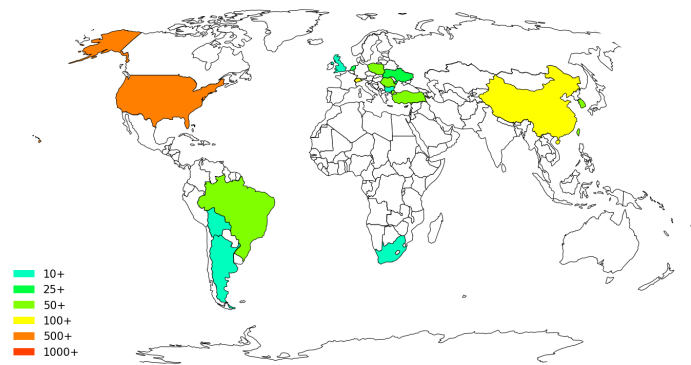
of potentially malicious or unwanted Internet traffic.

5.7 Chapter Summary

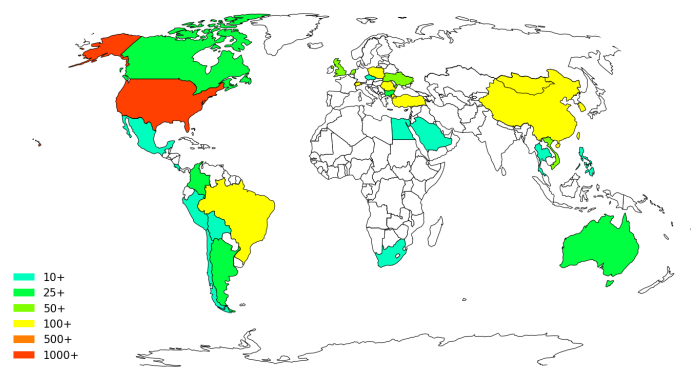
This chapter discussed the analysis of the OSINT and IBR data, with a specific focus on exploring the overlap between the OSINT and IBR datasets. The chapter started with **Section 5.1** detailing the initial analysis of all the datasets with the aim of reducing them to a subset of data which was used in later analysis. **Section 5.2** discussed the change in data on a month-by-month basis. **Section 5.3** discussed the results of the offset comparison performed between the OSINT and IBR datasets. In this section, it was proven that timely access to OSINT data is crucial to the effectiveness of TI. **Section 5.4** discussed the presence of a set of highly persistent IP addresses published on the OSINT datasets and recorded on the telescope sensors. **Section 5.5** concludes the analysis with a discussion of types of requests the persistent IP addresses are conducting. **Section 5.6** discussed the evaluation of the source of the highly persistent IP addresses. The next chapter discusses the second part of the design and implementation chapter, the *Processing* and *Publishing* components.



(a) IBR6



(b) AlienVault Highly Persistent



(c) IBR6 Highly Persistent

Figure 5.12: Countries

Colouring is based on the number of IP addresses registered to the country.

6

Processing and Publishing

THIS chapter discusses how OSINT data (Section 3.2) collected using the system detailed in Chapter 4, can be processed and distributed to consumers

The *Processing* and *Publishing* components of the system work together to publish a collection of enriched TI. Code listings are made available in an online repository detailed in Appendix B.

This chapter is structured as follows: **Section 6.1** discusses the design of the *Processing* component with an example module developed as a proof of concept. **Section 6.2** introduces the *Publishing* component of the system. **Section 6.2.1** details the design of the flat file module used to disseminate OSINT to consumers. **Section 6.2.2** details the design of the BGP module used to disseminate OSINT to neighbouring BGP routers. **Section 6.2.3** details the DNSBL module designed to make OSINT available to consumers in response to DNS queries to the system.

6.1 Processing

The *Processing* component provides the user with the ability to add specific modules to the system to create an enriched, relevant, and useful set of TI that can be distributed by the *Publishing* component. Given that different organisations have varying requirements, the purpose of the *Processing* component is to allow modules to be added to the system to meet

the organisation of security teams requirements. A module is a set of rules and processes that transform the collected TI into data that can be distributed. Any module developed for the *Processing* component must conform to the following data input assumptions:

- Data collected during the *Collection* phase (Section 4.4) is available on the database.
- The *sources* table (Section 4.3.1) holds information such as fields and field types for each of the OSINT tables in the database.

The output requirements of a *Processing* component are dependant on the format and channel being used to publish the TI. The *Flat File* module (Section 6.2.1) is capable of publishing data with any number of fields and rows. The *BGP* module (Section 6.2.2) in this research is limited to publishing IP addresses to neighbouring routers. The *DNSBL* module (Section 6.2.3) is limited to a database that can be queried through the DNS protocol. The remainder of this section describes a whitelisting module developed to illustrate the *Processing* component.

6.1.1 Whitelisting Module

Due to the nature of OSINT, there is the possibility that some legitimate IP address make their way into the OSINT lists. Given this, an organisation may decide that an IP or a set of IP addresses must not be added to blocklists or firewall rules even if they appear on the OSINT lists. In such scenarios, the security team may decide to filter the OSINT data and remove any IP addresses that are contained in a “Whitelist”. For example, Sections 5.4 and 5.5 discussed the presence of port scanners both in the persistent and highly persistent OSINT and IBR datasets. Looking into these IP addresses it was discovered that scanners from Shodan¹ and Shadowserver² appeared in the list of “malicious” hosts. These port scanners are known to be non-malicious and obey the “rules” of port scanning, in such cases the security team may “trust” these organisation and decide not to block these IP addresses from scanning their network. To achieve this, the OSINT lists can be filtered through a whitelist to remove any IP address that are “trusted”. Similarly, any critical services such as organisational or cloud infrastructure can be whitelisted.

This module was designed to take the persistent IP addresses from AlienVault and CINS (Section 5.4), and remove any IP addresses that appear in a predefined whitelist. The module follows a series of steps when filtering the TI:

- The persistent IP addresses from AlienVault and CINS are added to a single comprehensive list of IP addresses.
- The OSINT list of IP addresses is filtered to remove any IP address that appears in the whitelist provided.

¹<https://www.shodan.io/>

²<https://www.shadowserver.org/wiki/>

- Each IP address is given a status of active so that the BGP module will announce the new route.
- The filtered data is added to a new table in the database called *filtered_ip_addresses*. The TI in this table is used by the three modules in the *Publishing* component discussed below.

This minimalistic example of a processing component illustrates how one or more modules can be added to the *Processing* component to allow a security team to create their own set of TI based on their organisational requirements.

6.2 Publishing

The *Publishing* component is responsible for distributing the OSINT created by the system. The objective of this component is to provide OSINT to consumers in a format that can easily be incorporated into their systems. The *Publishing* component is designed to allow additional modules to be developed and added to the system with relative ease to meet organisational requirements.

The *BGP* and *DNSBL* modules in the *Publishing* component rely on external tools to be executed. BGP relies on the ExaBGP service, and DNSBL relies on the RBLDNSD service. The *Flat File* module does not rely on external tools, but it does rely on a web server or other service capable of serving plain text files to consumers. A requirement for each of the *Publishing* modules is that it accepts a list of database table names where the TI is stored. These tables are created during the *Processing* component and must be defined by the user. The ExaBGP and RBLDNSD tools run as services on the local machine. These two services are controlled through the system using the *subprocess.Popen*³ function to send terminal commands to the operating system. The code listings discussed in this section are taken from the *Publishing Library* (Appendix B.1). The full script can be found in the project repository online.

The system designed in this research used three different formats to distribute the data; JSON flat files, BGP routes, and DNSBLs. The format chosen to publish TI is crucial to effectively and efficiently disseminate the data to consumers (Rattan *et al.*, 2017). The JSON format provides a standard format which can be easily manipulated and adapted to the consumers needs. The BGP format was chosen to illustrate how malicious IP addresses can be added directly to a router's routing table. DNSBLs were chosen to provide the service of blocklists that store the data locally and allow consumers to query the database.

³<https://docs.python.org/3/library/subprocess.html#subprocess.Popen>

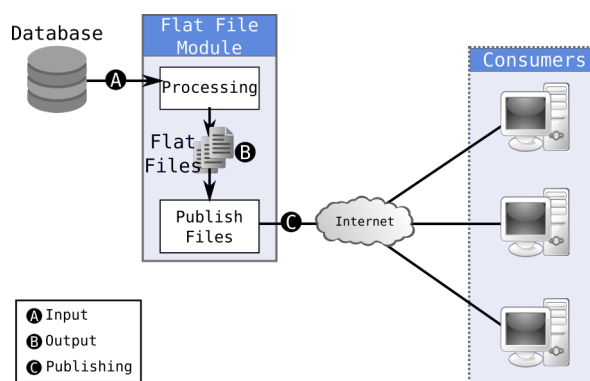


Figure 6.1: Flat File Module Design

6.2.1 Flat File Module

The Flat File module is responsible for publishing OSINT from the database to one or more flat files. Figure 6.1 illustrates the design of the module, giving a high-level overview of how the OSINT is made available to consumers.

The module can be conceptualised using input, output, and publishing stages. The input stage pulls OSINT data from the *filtered_ip_addresses* table into the processing sub-component. The data must be processed by the sub-component to transform it into a format which can be written to a flat file. The output stage takes the processed OSINT and writes it to the flat file. If the format of the flat file is CSV, the function writes a header row to the file to allow ease of processing on the consumer side. Each record is written to the file one line at a time using a comma as the delimiter. If the format of the flat file is JSON, the function dumps the whole list of records to a JSON file using the *json.dump* function from the JSON library. The *publish_flat_file* function is responsible for the input and output stages of the module. The flat files are made available in a location on the local computer for distribution. The distribution can be through tools such as an Apache web server, or an Rsync service.

The default output format for the Flat File module is JSON. The JSON format was chosen as the default because it:

- Is more suited to showing hierarchical data.
- Has stricter string handling. For example, bad CSV characters such as line breaks and quotes are handled better with JSON than with CSV.
- Has built-in support in modern-day APIs and applications for its input data
- Is more flexible than CSV since the structure allows for records of varying lengths to be stored in the same location without losing data.

It is important to note that there are areas where CSV would be a better format, namely when there are a large number of records being published on low bandwidth. For this

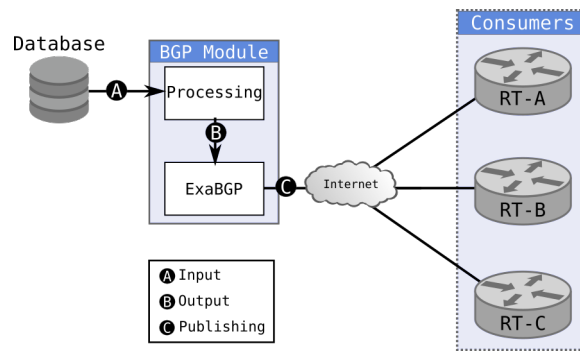


Figure 6.2: BGP Module Design

reason, the system provides the ability to change the output format if necessary.

6.2.2 Border Gateway Protocol (BGP) Module

The BGP module provided a way to disseminate OSINT data to consumers at the BGP route announcement level. A consumer has the ability to peer with the BGP router provided by the system and receive route announcements of TA IP addresses. One consideration with BGP is that if it fails, the consequences can potentially be replicated throughout the Internet (Kent, 2003). One such example was seen in an article by McCullagh (2008), where a small misconfiguration by an ISP caused worldwide disruption to YouTube access. Incorrectly configured EBGP routers have the potential to not only affect the Autonomous System (AS) itself, but all other EBGP routers on the Internet (Kent, 2003). For this reason, the BGP module of this system was implemented in a closed controlled environment, removing the risk of negative consequences across networks.

BGP peering operates on a trust relationship where peers know each other and trust the route announcements being sent and received. The trust relationship requires the consumers to be registered with the BGP module before they can receive routes.

The main component of the BGP module is the ExaBGP service which allows the system to automatically announce new routes to neighbouring routers. ExaBGP is responsible for disseminating OSINT created by the *Processing* component described in Section 6.1. Figure 6.2 illustrates the design of the BGP module in terms of the processing that occurs to retrieve the data from the database and publish it to the consumers.

The BGP module is divided into three stages:

1. **Input:** Retrieving the OSINT from the *filtered_ip_addresses* table to disseminate to BGP peers.
2. **Processing:** Processing the OSINT into a format that can be announced by the ExaBGP service to BGP peers.

3. Output Publishing the OSINT to BGP peers.

Listing 6.1 illustrates the main code of the *publish_bgp* function responsible for the Input and Output of the OSINT data. The *commands* dictionary holds two sets of commands, one for announcing routes (“add”) and one for withdrawing routes (“remove”).

```

53 for table in tables:
54     for item in dbh.custom_select("select ip, masklen(ip) as netmask from
      {0} limit {1}".format(table, limit)):
55         route = "{0}/{1} next-hop {2}".format(item["ip"], item["netmask"],
      next_hop)
56         if item["active"]:
57             if route not in announced_routes:
58                 commands["add"].add(route)
59             elif not item["active"] and route in announced_routes:
60                 commands["remove"].add(route)

```

Listing 6.1: Publish BGP Function

The function illustrated in Listing 6.1 retrieves OSINT data from the database. For each of the records in the table, the *publish_bgp* function determines whether a new route must be announced for the IP address (Line 56), or if the route must be withdrawn from the FIB (Line 59). A route example can be seen in Listing 6.2.

```

1 203.0.113.0/32 next-hop 198.51.100.1

```

Listing 6.2: Route Example

The *publish_bgp* function outputs a collection of these routes to a JSON file which is used by ExaBGP. The publishing of BGP TI is handled by the ExaBGP service. The ExaBGP configuration file for the system includes a section called *process add-remove* which tells ExaBGP to execute the *exabgp.py* (Appendix C.6) Python script after the service has successfully started. The Python script is responsible for taking the commands created in the processing stage of the BGP module and sending them out on the ExaBGP service.

The ExaBGP service is designed to continue running as long as there are still commands to execute. When there are no more commands, the ExaBGP service terminates. If the service terminates, the routes previously announced are lost. To overcome this, the Python script was designed to loop continuously and check for any new commands to send to the ExaBGP service.

The flow of the *exabgp.py* script is illustrated in Figure 6.3. When the script is executed for the first time, or after a reboot of the service, it must check to determine if there were any routes announced previously. This process ensures that all previous routes are restored to ExaBGP. Once the routes have been restored, the script moves into the loop body which runs forever. As mentioned above, the script needs to run continuously so the script embodies all code within a *while* loop which does not terminate. Within the loop, the following steps are performed:

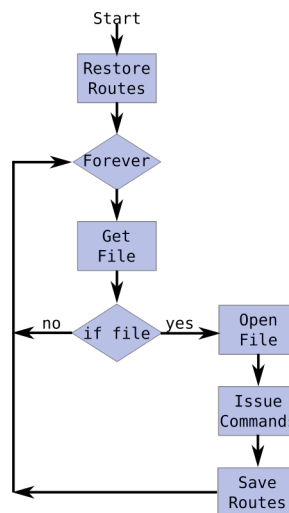


Figure 6.3: ExaBGP Python Script Flow

- **Get File:** The script checks to see if there are any new JSON files to read commands from. If there are no files, the script sleeps for a period of time before checking again.
- **Open File:** If there is a file to read from, it is opened and the commands are read into a list to be processed.
- **Issue Commands:** The script issues one of two commands; announce route or withdraw route. An example of the command issued can be seen in Listing 6.3 where *command* is either “announce route” or “withdraw route”, depending on whether the route must be added or withdrawn from the FIB. ExaBGP listens to STDOUT for any command from the script. When ExaBGP is running and a command such as `sys.stdout.write(“command”)` is given, ExaBGP intercepts this as an instruction it has to act upon.

```
1 command 203.0.113.0/32 next-hop 198.51.100.1
```

Listing 6.3: ExaBGP Command Example

6.2.3 Domain Name System Block List (DNSBL) Module

The DNSBL module provides the ability for consumers to query the system to determine whether a specific IP address is recorded as malicious. For the DNSBL module to function, it requires access to a zone file that is a list of IP addresses classified as malicious. As this module is a proof of concept idea, the zone file is a plain text file such as the one discussed in Appendix C.5. Figure 6.4 illustrates the design and data flow of the DNSBL module. The module can be conceptualised as having three stages:

1. **Input:** TI data is pulled from the database to be processed. The *Processing* component transforms the data into a form which can be written to a zone file.

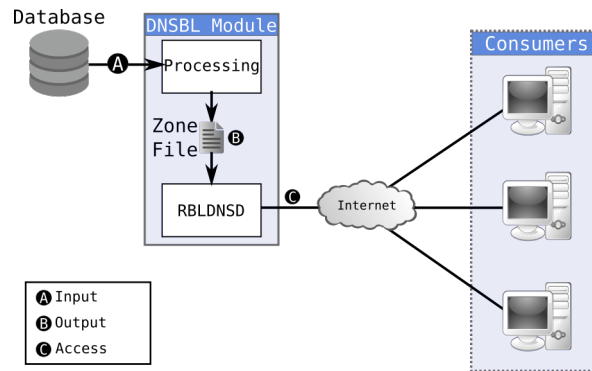


Figure 6.4: DNSBL Module Design

2. **Output:** the TI is written to the zone file.
3. **Access:** the DNSBL module is running and able to receive DNS queries from consumers.

The module periodically creates a zone file from the data stored in the database. The function responsible for publishing the OSINT to the DNSBL zone file is called *publish_zone_file* and can be found in the system code repository (Appendix B.1). The function performs the following steps:

- The IP addresses are pulled from each table in the list of tables provided.
- These IP addresses are then added to a list of other IP addresses from other tables.
- After adding all IP addresses to a single list and removing duplicate entries, a zone file is created with all the IP addresses from the tables. An example of the zone file can be seen in Appendix C.5.

With the zone file created, the module ensures that the RBLDNSD service is running so consumers can query the database. This is achieved by executing a command similar to that seen in Listing 4.3.

6.3 Chapter Summary

This chapter discussed the design and implementation of the *Processing* and *Publishing* components of the system used in this research. The *Processing* component provides consumers with the ability to add modules to the system to create a usable subset of the collected OSINT data for distribution. The *Publishing* component distributes OSINT to consumers in three prototype formats; Flat File, DNSBL, and ExaBGP. The chapter was introduced with the *Processing* component discussed in **Section 6.1**. **Section 6.2** presented the *Publishing* component with the three prototype formats implemented; the Flat File module (Section 6.2.1), the BGP module (Section 6.2.2), and the DNSBL module

(Section 6.2.3). This chapter concludes the discussion of Collection, Processing and Publishing of OSINT. The next chapter (Chapter 7) concludes this paper with the research conclusion and future work recommendations.

7

Conclusion

THE research presented in this work evaluated how effective OSINT is at reporting on real-world malicious Internet traffic, and how valid the OSINT is when dealing with current threats. During the course of this research, a prototype system was designed and developed. The purpose of the system was to collect (Section 4.4) OSINT from Internet-accessible feeds, analyse (Chapter 5) and process (Section 6.1) the data, and finally, publish (Section 6.2) the data via three distribution methods.

The collection period spanned 12-months (from September 2017 through to August 2018), during which a total of 37 OSINT feeds were downloaded and stored. The 37 OSINT feeds were reduced to a collection of 18 datasets containing IPv4 addresses linked to malicious activities.

Analysis of the OSINT data was performed against a collection of active IBR data taken from five different network telescope sensors over the same 12-month period. The IBR data was used to explore the overlap between the OSINT and network telescope data. By comparing OSINT to IBR data, this study was able to identify how effective OSINT can be at reducing the amount of malicious Internet traffic attacking an organisation's border routers on a daily basis.

Furthermore, this research discovered the existence of a collection of persistent IP addresses both in the OSINT and IBR data from month to month. On further analysis, a collection of highly persistent IP addresses, that were present throughout the 12-month period, were identified. The research found that these highly persistent IP addresses con-

stitute an average of 0.0646% of the IP addresses recorded on the network telescopes, but 7.85% of the total packets sent to the network telescopes. An evaluation of the highly persistent IP addresses revealed that the OSINT and IBR datasets only shared an average 53% of the IP addresses. Given that these highly persistent IP addresses have been present for 12 months, and possibly longer, the expectation was that the OSINT lists would have a higher percentage in common with the IBR datasets. Further research into the highly persistent IP addresses on both the OSINT and IBR datasets revealed that the majority of these IP addresses were port scanners probing for any vulnerabilities on the targeted hosts. Of these port scanners identified, a significant portion of the IP addresses were probing 23/tcp and 2323/tcp, which have been linked to Mirai and Mirai-like IoT botnets.

7.1 Document Summary

This section gives an overview of the research presented in this document.

Chapter 1 introduced the area of research and provided background information. Included in this chapter was an outline of the research objectives, research approach adopted, and the scope of the research.

Chapter 2 provided the current evidence required to give context to this research. This chapter specifically detailed what a network telescope (Section 2.6) is and the type of malicious activity (Section 2.7) they record. This chapter introduced the concept of TI (Section 2.9) as a data source for the detection and prevention of malicious activity.

Chapter 3 presented a discussion and quantitative analysis of the data used in this research project. The chapter discussed the 37 OSINT feeds (Section 3.2) and five network telescope sensors (Section 3.3). The main focus of this chapter was to describe the data collected during the 12-month period. Of specific note in this chapter is the description of the OSINT and IBR datasets used in the analysis stage of the research.

Chapter 4 detailed part one of the design and implementation of the system developed to meet the research objectives. The overall system design (Section 4.1) was presented, along with the technologies used (Section 4.2). The remainder of the chapter discussed the design of the database (Section 4.3) and collection (Section 4.4) components.

Chapter 5 presented the main body of the research, discussing the exploratory analysis of the two datasets. Each section of the chapter discussed a different type of analysis that was performed on the data, along with the results obtained. The main goal of this chapter was to identify common data in the OSINT and IBR datasets, and to describe how it can help in the evaluation of OSINT.

Chapter 6 detailed part two of the design and implementation of the system developed to meet the research objectives. In this chapter was a discussion of the *Processing* (Section 6.1) and *Publishing* (Section 6.2) components of the system. An example module was presented to illustrate how modules can be added to the *Processing* component.

7.2 Evaluation of Research Goals

This section provided a reflection on the objectives set out in Section 1.3, and the extent to which they were achieved. Each of the primary and secondary objectives is revisited below:

Primary Objectives:

1. The design of the system provided a mechanism for collecting and analysing OSINT from a number of different Internet sources. The *Collection* component discussed in Section 4.4 provided the ability to download OSINT regardless of the content and format of the data. Due to errors and formatting inconsistencies in the downloaded data, the *Collection* component required a *Pre-Processing* module that converted OSINT data into a consistently-formatted dataset. This highlights the additional work required by consumers if they are making use of a number of different OSINT feeds in their security plans.
2. In Chapter 5 the OSINT and IBR datasets were analysed to determine the areas where they overlap. The initial analysis performed identified that not all OSINT sources publish data that is seen on the network telescopes. However, the majority of the OSINT lists had some degree of commonality with the IBR datasets. The majority of the analysis was performed in Sections 5.2 to 5.5, where the OSINT and IBR datasets were evaluated against the CART framework (Caltagirone, 2015a, Sillaber *et al.*, 2016). The results showed that there is indeed a need for OSINT to be collected, analysed, and disseminated in the least possible amount of time. The most significant discovery was the portion of highly persistent IP addresses seen on the OSINT and IBR datasets in Section 5.5 and theneed to report more these IP addresses in the OSINT feeds.
3. The purpose of the *Processing* component was to process OSINT collected by the *Collector* into contextualised TI that can be published to consumers. The *Processing* component discussed in Section 6.1 describes a platform that allows security teams to develop custom processing modules based on their requirements. To illustrate this, a whitelisting module was developed and discussed in this section.
4. The *Publishing* component of the system detailed in Chapter 6 was designed as a modular component, with the ability to have multiple sub-components operating independently to publish TI to consumers. The example modules added to this component included a flat file module (Section 6.2.1), a BGP module (Section 6.2.2),

and a DNSBL module (Section 6.2.3). Each of these components provided the ability to publish TI to consumers in a manner that is most appropriate for the type of data and threat.

Secondary Objectives:

1. The objective to investigate ways to reduce OSINT data into smaller datasets and/or categories was partially met. The analysis in Section 5.4 identified a collection of persistent IP addresses. Further analysis in Section 5.5 identified a collection of highly persistent IP addresses that were present across all 12 months. These highly persistent IP addresses contribute a small portion of the total IP addresses seen on the network telescopes but a large portion of the packets received. Creating a reduced list of these IP addresses can provide a minimal set of OSINT that can be operationalised by the consumer.

7.3 Research Contribution

The results of this research provided useful insight into the effectiveness of OSINT and the need for good quality TI through evaluation of the overlap between OSINT and active IBR data. OSINT datasets only report, on average, 53% of the persistent IP addresses seen in the IBR datasets. Despite only being 53% of the IBR datasets, these IP addresses are still potentially malicious and unwanted, and account for a significant portion of the traffic (Section 5.5). As such, blocking these will prove beneficial to the consumer. It provided a way to reduce the amount of Internet noise seen on the telescopes.

The research also showed that, since the IBR data is primarily made up of port scanning traffic, using OSINT to identify and block these requests has the potential to exclude organisations from future attacks. For example, if a firewall blocks malicious scans targeting the Telnet service, the threat actor will assume the port is inactive and will not attempt the next stage of the exploitations.

Furthermore, the research showed that simply having a way to eliminate the persistent IP addresses seen on the network telescopes will significantly reduce the number of potentially malicious requests. Blocking an average of 8 367 (0.09%) IP addresses of the total unique active IP addresses on network telescopes will reduce the potentially malicious packets by up to 7.85%.

This research identified that, although OSINT adds value to a security strategy policy, it is still limited by the time factor. The time lag between an IP addresses appearing on the Internet, the OSINT source collecting, analysing and disseminating the data, and a consumer obtaining and operationalising the data is still a major factor in the effectiveness of utilising OSINT to improve security.

7.4 Future Work

The quantitative data analysis conducted in Chapter 3 and subsequent exploratory data analysis in Chapter 5 highlighted the fact that OSINT and IBR datasets contain a vast amount of data.

- This research showed that a portion of the IP addresses on the network telescopes can be classified as persistent hosts. The research also showed that these persistent IP addresses reduced in number as the months increased. Further research into these persistent IP addresses and how they behave will provide useful information towards developing a way to produce TI targeting persistent hosts on the Internet. Although the current research only identified the presence of these persistent IP addresses, with historical data from the network telescopes it is possible to track these IP addresses over months and even years to discover how persistent they are and how often they change.
- The analysis performed in this research was primarily focused on active hosts on the network telescopes. Further research into the comparison of OSINT data against passive hosts could potentially highlight whether OSINT publishing IP addresses is linked to passive requests.
- The results show that AlienVault classifies more than 90% of the persistent IP addresses as malicious hosts, but the network telescopes suggest that the persistent IP addresses are linked to port scanning hosts. The assumption here is that the persistent IP addresses are conducting more than one type of activity, and AlienVault is publishing it for reasons other than those seen on the network telescopes. Further research into these persistent IP addresses needs to be done to uncover the actual types of packets being sent. Similarly, research into the source of the request would uncover more about the TA, which would enable better classification of the IP addresses.
- The results from Section 5.5 revealed evidence of highly persistent IP addresses on both the OSINT and IBR datasets. Included in this section was the number of IP addresses probing a port on a month-to-month basis. The number of probing IP addresses was never zero; however, there is reason to believe that it was not the same IP addresses probing the ports each month. Using this data, further research can be conducted to discover changes in IP addresses scanning the top 20 ports on the network telescopes.
- Performance plays a significant role in the acceptance of OSINT. Adding too many rules to a firewall has the potential to hinder the processing time of traffic, and thus affect a security team's willingness to use certain OSINT. Further evaluation of the OSINT performance will allow better reduction processes on the OSINT being published to occur.
- In Section 5.3.3, the time lag between network telescopes recording potentially ma-

licious IP addresses and the OSINT feeds publishing the IP address was discussed. For some OSINT feeds, the IP addresses started appearing two or three months prior to the publishing of the list. Further research into how critical this lag could be will provide a better understanding of the quality of the OSINT feeds. In addition, a theory was raised in Section 5.3.3 that network telescopes can be used as an early warning system for potential attacks to legitimate network infrastructure. For example, when a network telescope detects a surge in malicious port scans, this could potentially be fed into a system that would operationalise the TI to prevent the scans from discovering information about the network infrastructure.

- In Section 5.5.3 it was seen that AlienVault classifies the majority of the IP addresses as malicious hosts. However, with the telescope data collected, it is possible to assign more specific categories to IP addresses published by the OSINT feeds or seen on the telescope sensors. By using results from Section 5.6 and further analysis, it is possible to assign labels with priority to IP addresses. For example, Shodan web crawlers are harmless and non-intrusive; these can be assigned Scanning Host (Low). Another example would be IP addresses linked to 23/tcp and 2323/tcp. which can be labelled as *Botnet Replication*, such as the case with the Mirai-derivative botnets.

References

- Abu Rajab, M., Zarfoss, J., Monrose, F., and Terzis, A.** A Multifaceted Approach to Understanding the Botnet Phenomenon. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, (IMC '06), pages 41–52. ACM, 2006. ISBN 1-59593-561-4. doi:10.1145/1177080.1177086.
- Abuse.ch.** abuse.ch. Online Blog, May 2007. [Accessed: 8 August 2017].
URL <https://www.abuse.ch/>
- Adeyinka, O.** Internet Attack Methods and Internet Security Technology. In *Modeling & Simulation, 2008. AICMS 08. Second Asia International Conference on*, pages 77–82. IEEE, 2008. doi:10.1109/AMS.2008.68.
- Agbenyegah, F. K. and Asante, M.** Impact Of Firewall On Network Performance. *International Journal of Scientific & Technology Research*, 6:32–38, March 2017.
- Ahrend, J. M., Jirotko, M., and Jones, K.** On the Collaborative Practices of Cyber Threat Intelligence Analysts to Develop and Utilize Tacit Threat and Defence Knowledge. In *Cyber Situational Awareness, Data Analytics And Assessment (CyberSA), 2016 International Conference On*, pages 1–10. IEEE, 2016. doi:10.1109/CyberSA.2016.7503279.
- Alexa.** Alexa. Online, May 1997. [Accessed: 31 August 2017].
URL <https://www.alexa.com/>
- AlienVault.** AlienVault. Online, March 2005. [Accessed: 15 August 2017].
URL <https://www.alienvault.com/>
- AlienVault.** AlienVault Open Threat Exchange (OTX)TM User Guide. Technical report, AlienVault, January 2016.
URL <https://www.alienvault.com/doc-repo/OTX/user-guides/AlienVault-OTX-User-Guide.pdf>
- AlienVault.** What is Open Threat Exchange. Online, June 2018. [Accessed: 16 June 2018].

- URL <https://www.alienvault.com/documentation/usm-appliance/otx/about-otx.htm>
- Anomali.** What is a Threat Intelligence Platform (TIP)? Online, May 2003. [Accessed: 12 November 2018].
URL <https://www.anomali.com/platform/what-is-a-tip>
- 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 *USENIX Security Symposium*, pages 1092–1110. 2017.
- Arkko, J., Vegoda, L., and Cotton, M.** RFC 5737: IPv4 Address Blocks Reserved For Documentation. Technical report, Internet Engineering Task Force (IETF), 2010.
URL <https://tools.ietf.org/html/rfc5737>
- AutoShun.** AutoShun. Online, June 2007. [Accessed: 30 August 2017].
URL <https://www.autoshun.org/>
- Avrach, J.** CSV vs. Delimited Flat Files: How to Choose. Online, January 2015. [Accessed: 12 April 2017].
URL <http://www.thoughtspot.com/blog/csv-vs-delimited-flat-files-how-choose>
- Bahadur, N., Kini, S., and Medved, J.** RFC 8430: RIB Information Model. Technical report, Internet Engineering Task Force (IETF), 2018.
URL <https://tools.ietf.org/html/rfc8430>
- Baker, F., Harrop, W., and Armitage, G.** IPv4 and IPv6 Greynets. Technical report, Internet Engineering Task Force (IETF), 2010.
URL <https://tools.ietf.org/html/rfc6018>
- Balram, S. and Wiscy, M.** Detection of TCP SYN Scanning Using Packet Counts and Neural Network. In *Signal Image Technology and Internet Based Systems, 2008. SITIS'08. IEEE International Conference on*, pages 646–649. IEEE, 2008. doi:10.1109/SITIS.2008.33.
- Bambenek Consulting.** Bambenek Consulting. Online, February 2011. [Accessed: 31 August 2017].
URL <http://www.bambenekconsulting.com/>
- Bano, S., Richter, P., Javed, M., Sundaresan, S., Durumeric, Z., Murdoch, S. J., Mortier, R., and Paxson, V.** Scanning the Internet for Liveness. *ACM SIGCOMM Computer Communication Review*, 48(2):2–9, 2018.

- Barnett, R. J. and Irwin, B.** Towards a taxonomy of network scanning techniques. In *Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology*, pages 1–7. ACM, 2008.
- Barnum, S.** Standardizing cyber threat intelligence information with the Structured Threat Information eXpression (STIX). *MITRE Corporation*, 11:1–22, 2012.
- Bates, T., Chen, E., and Chandra, R.** RFC 4456: BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP). Technical report, Internet Engineering Task Force (IETF), 2006.
URL <https://tools.ietf.org/html/rfc4456>
- Bayer, U., Habibi, I., Balzarotti, D., Kirda, E., and Kruegel, C.** A View on Current Malware Behaviors. In *Proceedings of Large-Scale Exploits and Emergent Threats*. 2009.
- Benson, K., Dainotti, A., claffy, k., Snoeren, A. C., and Kallitsis, M.** Leveraging Internet Background Radiation for Opportunistic Network Analysis. In *Proceedings of the 2015 Internet Measurement Conference*, (IMC '15), pages 423–436. ACM, 2015. doi:10.1145/2815675.2815702.
- Binary Defense.** Binary Defense. Online, April 2013. [Accessed: 17 April 2018].
URL <https://www.binarydefense.com/>
- Binsalleeh, H., Ormerod, T., Boukhtouta, A., Sinha, P., Youssef, A., Debbabi, M., and Wang, L.** On the analysis of the zeus botnet crimeware toolkit. In *Privacy Security and Trust (PST), 2010 Eighth Annual International Conference*, pages 31–38. IEEE, 2010.
- Blenn, N., Ghi tte, V., and Doerr, C.** Quantifying the Spectrum of Denial-of-Service Attacks Through Internet Backscatter. In *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ARES '17, pages 21:1–21:10. ACM, New York, NY, USA, 2017.
- Bollinger, J.** Open Source Threat Intel: GOSINT. Online, August 2017. [Accessed: 11 May 2018].
URL <https://blogs.cisco.com/security/open-source-threat-intel-gosint>
- Bou-Harb, E., Debbabi, M., and Assi, C.** Cyber scanning: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 16(3):1496–1519, 2014. doi:10.1109/SURV.2013.102913.00020.
- BrightCloud.** Webroot BrightCloud Threat Intelligence. Online, February 2005. [Accessed: 16 June 2018].
URL <https://www.brightcloud.com/>

- Bromiley, M.** Threat Intelligence: What It Is, and How to Use It Effectively. White paper, SANS Institute, September 2016.
URL <https://www.sans.org/reading-room/whitepapers/analyst/threat-intelligence-is-effectively-37282>
- Brute Force Blocker.** Brute Force Blocker. Online, January 2006. [Accessed: 31 August 2017].
URL <http://danger.rulez.sk/projects/bruteforceblocker/>
- Caltagirone, S.** CART: The 4 Qualities of Good Threat Intelligence. Online, July 2015a. [Accessed: 18 April 2018].
URL <http://www.activeresponse.org/the-4-qualities-of-good-threat-intelligence/>
- Caltagirone, S.** The Cost of Bad Threat Intelligence. Online, May 2015b. [Accessed: 18 April 2018].
URL <http://www.activeresponse.org/the-cost-of-bad-threat-intelligence/>
- Catakoglu, O., Balduzzi, M., and Balzarotti, D.** Automatic Extraction of Indicators of Compromise for Web Applications. In *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, pages 333–343. 2016. ISBN 978-1-4503-4143-1. doi:10.1145/2872427.2883056.
- Caudle, J.** What's in a DNS response. Online, March 2015. [Accessed: 27 July 2017].
URL <https://blog.dnsimple.com/2015/03/whats-in-a-dns-response/>
- Chandrashekar, J., Duan, Z., Zhang, Z.-L., and Krasky, J.** Limiting path exploration in BGP. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 4, pages 2337–2348. IEEE, 2005.
- Cheswick, W., Bellovin, S. M., Ford, W., and Gosling, J.** How computer security works. *Scientific American*, 279(4):106–109, 1998.
- Chindipha, S. D. and Irwin, B. V.** An analysis on the re-emergence of SQL Slammer worm using network telescope data. In *Southern African Telecommunications Networks and Applications Conference (SATNAC)*, pages 218–222. SATNAC, 2017.
- Choi, S.-s., Song, J., Kim, S., and Kim, S.** A model of analyzing cyber threats trend and tracing potential attackers based on darknet traffic. *Security and Communication Networks*, 7(10):1612–1621, 2014.
- CINS.** The CINS Score. Online, May 2013. [Accessed: 31 August 2017].
URL <http://cinsscore.com/>

- Cisco Umbrella.** Cisco Umbrella. December 2016. [Accessed: 31 August 2017].
URL <https://umbrella.cisco.com/>
- Concise Oxford Dictionary.** The Concise Oxford Dictionary. Oxford: Clarendon Press, 8 edition, 1990. ISBN-10: 0199601089.
- Conta, A.** RFC 2473: Generic Packet Tunneling in IPv6 Specification. Technical report, Internet Engineering Task Force (IETF), 1998.
URL <https://tools.ietf.org/html/rfc2473>
- Cooke, E., Jahanian, F., and McPherson, D.** The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. *Steps to Reducing Unwanted Traffic on Internet (SRUTI)*, 5:6–6, 2005.
- Coppolino, L., D’Antonio, S., Mazzeo, G., and Romano, L.** Cloud security: Emerging threats and current solutions. *Computers & Electrical Engineering*, 59:126–140, 2017.
- De La Peña Montero, F. and Hariri, S.** Autonomic and Integrated Management for Proactive Cyber Security (AIM-PSC). In *Companion Proceedings of the 10th International Conference on Utility and Cloud Computing*, pages 107–112. ACM, 2017.
- de Vivo, M., Carrasco, E., Isern, G., and de Vivo, G. O.** A Review of Port Scanning Techniques. *SIGCOMM Comput. Commun. Rev.*, 29(2):41–48, April 1999. ISSN 0146-4833. doi:10.1145/505733.505737.
- Department of Homeland Security.** A Roadmap for Cybersecurity Research. Online, November 2009. [Accessed: 30 May 2018].
URL <https://www.dhs.gov/sites/default/files/publications/CSD-DHS-Cybersecurity-Roadmap.pdf>
- Doss, S., Narayanan, S., and Anand, J.** Detecting IP Spoofing using Hop Count Filtering based dynamic path update approach. *Journal of Multidisciplinary Engineering Science Studies*, 3(1):1241–1246, 2017.
- Durumeric, Z., Bailey, M., and Halderman, J. A.** An Internet-Wide View of Internet-Wide Scanning. In *USENIX Security Symposium*, pages 65–78. 2014.
- Eddy, W.** RFC:4987 TCP SYN flooding attacks and common mitigations. Technical report, Internet Engineering Task Force (IETF), 2007.
URL <https://tools.ietf.org/html/rfc4987>
- Fàbrega, L., Vilà, P., Careglio, D., and Papadimitriou, D.** Measurement Methodology and Tools: First European Workshop, FP7 FIRE/EULER Project, May 9, 2012, Aalborg, Denmark, Invited Papers, volume 7586. Springer, 2013. ISBN-10: 9783642412950.

- Farinacci, D., Li, T., Hanks, S., Meyer, D., and Traina, P.** RFC 2784: Generic Routing Encapsulation (GRE). Technical report, Internet Engineering Task Force (IETF), 2000.
URL <https://tools.ietf.org/html/rfc2784>
- Fielding, R. and Rescheke, J.** Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content. Technical report, Internet Engineering Task Force (IETF), June 2014.
- Gadge, J. and Patil, A. A.** Port Scan Detection. In *2008 16th IEEE International Conference on Networks*, pages 1–6. December 2008. doi:10.1109/ICON.2008.4772622.
- Garcia, L. M.** TCPDUMP. Online, June 2000. [Accessed: 19 April 2018].
URL <http://www.tcpdump.org/>
- Garud, J.** Forwarding and Routing in Network Layer. Online, May 2019. [Accessed: 29 May 2018].
URL <https://electronicspost.com/forwarding-and-routing-in-network-layer/>
- Geschwinde, E. and Schönig, H.-J.** PostgreSQL Developer’s Handbook. Sams Publishing, 2002. ISBN:0-672-32260-9.
- Griffin, T. G. and Wilfong, G.** On the Correctness of IBGP Configuration. *ACM SIGCOMM Computer Communication Review*, 32(4):17–29, 2002. doi:10.1145/964725.633028.
- Gu, G., Perdisci, R., Zhang, J., Lee, W. et al.** BotMiner: Clustering Analysis of Network Traffic for Protocol-and Structure-Independent Botnet Detection. In *USENIX Security Symposium*, volume 5, pages 139–154. 2008.
- Hacker Target.** Port Scanner Tutorial. Online, September 2011.
URL <https://hackertarget.com/port-scanner/>
- Harrison, G. and Feuerstein, S.** MySQL stored procedure programming. O’Reilly Media, Inc., 2006. ISBN-10: 0-596-10089-2.
- 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*, MineNet ’05, pages 171–172. 2005. ISBN 1-59593-026-4. doi:10.1145/1080173.1080177.
- Hawkinson, J. and Bates, T.** RFC 1930: Guidelines for creation, selection, and registration of an Autonomous System (AS). Technical report, Internet Engineering Task Force (IETF), 1996.
URL <https://tools.ietf.org/html/rfc1930>

- Heron, S.** Denial of Service: Motivations and Trends. *Network Security*, 2010(5):10–12, 2010.
- Hodgman, R.** WannaCry Update: Vulnerable SMB Shares Are Widely Deployed And People Are Scanning For Them. Online, May 2017. [Accessed: 21 November 2018].
URL <https://blog.rapid7.com/2017/05/16/update-on-wannacry-vulnerable-smb-shares-are-widely-deployed-and-people-are-scanning-for-them/>
- Hughes, T.** Code Red: A New Threat. Technical report, SANS Institute, August 2001.
URL <https://www.sans.org/reading-room/whitepapers/malicious/paper/84>
- Hutchins, E. M., Cloppert, M. J., and Amin, R. M.** Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1(1):80, 2011.
- Ianelli, N. and Hackworth, A.** Botnets as a vehicle for online crime. *International Journal of Forensic Computer Science*, 19:19–39, 2005.
- Infosec Institute.** Top Five Open Source Intelligence (OSINT) Tools. Online, February 2018. [Accessed: 13 June 2018].
URL <https://resources.infosecinstitute.com/top-five-open-source-intelligence-osint-tools/#gref>
- Irwin, B.** A framework for the application of network telescope sensors in a global IP network. Ph.D. thesis, Rhodes University, January 2011.
- Irwin, B.** A network telescope perspective of the Conficker outbreak. In *Information Security for South Africa (ISSA)*, 2012, pages 1–8. IEEE, 2012.
- Irwin, B.** A baseline study of potentially malicious activity across five network telescopes. In *Cyber Conflict (CyCon)*, 2013 5th International Conference on Cyber Conflict, pages 1–17. IEEE, 2013.
- Irwin, B. and Barnett, R. J.** An analysis of logical network distance on observed packet counts for network telescope data. In *Southern African Telecommunications Networks and Applications Conference (SATNAC)*. 2009.
- Irwin, B. and Nkhumeleni, T.** Observed Correlations of Unsolicited IP Traffic across Five Distinct Network Telescopes. *Journal of Information Warfare*, 14(3):1–14, 2015.
- Irwin, B., Pilkington, N., Barnett, R., and Friedman, B.** A Geopolitical Analysis of long term Internet Network Telescope Traffic. In *Southern African Telecommunications Networks and Applications Conference (SATNAC)*. SATNAC, 2007.
- ISC.** Suspicious Domains. Online, July 2012. [Accessed: 19 August 2017].
URL https://isc.sans.edu/suspicious_domains.html

- Iverson, A.** What is DNSBL. Online, January 2007. [Accessed: 12 April 2017].
URL <http://www.dnsbl.com/2007/01/what-is-dnsbl.html>
- JSON.** Introducing JSON. Online, Dec 2002. [Accessed: 30 October 2017].
URL <http://json.org/>
- Jung, J. and Sit, E.** An Empirical Study of Spam Traffic and the Use of DNS Black Lists. In *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement, IMC '04*, pages 370–375. ACM, New York, NY, USA, 2004. doi:10.1145/1028788.1028838.
- 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.
- Kambourakis, G., Kolias, C., and Stavrou, A.** The Mirai botnet and the IoT Zombie Armies. In *Military Communications Conference, MILCOM 2017-2017 IEEE*, pages 267–272. IEEE, 2017.
- Kennedy, D.** Project Artillery - Now a Binary Defense Project! Online, November 2014. [Accessed: 4 May 2018].
URL <https://blog.binarydefense.com/project-artillery-now-a-binary-defense-project>
- Kent, S.** RFC 4303: IP Encapsulating Security Payload (ESP). Technical report, Internet Engineering Task Force (IETF), 2005.
URL <https://tools.ietf.org/html/rfc4303>
- Kent, S. T.** Securing the Border Gateway Protocol: A Status Update. In *IFIP International Conference on Communications and Multimedia Security*, pages 40–53. Springer, 2003.
- Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J.** DDoS in the IoT: Mirai and Other Botnets. *Computer*, 50(7):80–84, 2017. doi:10.1109/MC.2017.201.
- Kosem, G.** BGP. Online, September 2015. [Accessed: 3 March 2017].
URL <http://ipcisco.com/bgp-part-1/>
- Krawetz, N.** Anti-honeypot technology. *IEEE Security & Privacy*, 2(1):76–79, 2004. doi:10.1109/MSECP.2004.1264861.
- Kurose, J. F. and Ross, K. W.** Computer networking: A Top-Down Approach. Pearson Education Limited, 6th edition, 2013. ISBN-10: 0-273-76896-4.
- Lee, S. and Shon, T.** Open source intelligence base cyber threat inspection framework for critical infrastructures. In *Future Technologies Conference (FTC)*, pages 1030–1033. IEEE, 2016. doi:10.1109/FTC.2016.7821730.

- Liao, X., Yuan, K., Wang, X., Li, Z., Xing, L., and Beyah, R.** Acing the IOC Game: Toward Automatic Discovery and Analysis of Open-Source Cyber Threat Intelligence. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS '16*, pages 755–766. ACM, New York, NY, USA, 2016. doi:10.1145/2976749.2978315.
- Magee, J. C., Andrews, A. M., Nicholson, M. W., James, J. L., Li, H. C., Stevenson, C. L., and Lathrop, J.** Collective Threat Intelligence Gathering System. August 2014. US Patent 8,813,228.
- Malware Domain List.** Malware Domain List. Online, October 2007. [Accessed: 15 August 2017].
URL <https://www.malwaredomainlist.com/>
- Mangin, T.** ExaBGP - A new Tool to Interact with BGP. Online, July 2010. [Accessed: 23 June 2017].
URL https://labs.ripe.net/Members/thomas_mangin/content-exabgp-new-tool-interact-bgp
- Mao, Z. M., Sekar, V., Spatscheck, O., van der Merwe, J., and Vasudevan, R.** Analyzing Large DDoS Attacks Using Multiple Data Sources. In *Proceedings of the 2006 SIGCOMM Workshop on Large-scale Attack Defense, LSAD '06*, pages 161–168. ACM, 2006.
- McCullagh, D.** How Pakistan knocked YouTube offline (and how to make sure it never happens again). Online, February 2008. [Accessed: 13 February 2017].
URL <https://www.cnet.com/news/how-pakistan-knocked-youtube-offline-and-how-to-make-sure-it-never-happens-again/>
- McMillan, R.** Threat Intelligence: What is it, and How Can it Protect You from Today's Advanced Cyber-Attacks? Technical report, Gartner, May 2013.
URL https://www.gartner.com/imagesrv/media-products/pdf/webroot/issue1_webroot.pdf
- Melazzi, N. B., Detti, A., and Pomposini, M.** Scalability Measurements in an Information-Centric Network. In *Measurement Methodology and Tools*, pages 81–106. Springer, 2013.
- Modi, A., Sun, Z., Panwar, A., Khairnar, T., Zhao, Z., Doupé, A., Ahn, G.-J., and Black, P.** Towards Automated Threat Intelligence Fusion. In *2016 IEEE 2nd International Conference on Collaboration and Internet Computing (CIC)*, pages 408–416. IEEE, 2016.
- Moore, D., Paxson, V., Savage, S., Shannon, C., Staniford, S., and Weaver, N.** Inside the Slammer Worm. *IEEE Security & Privacy*, 99(4):33–39, 2003.

- 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, May 2006. ISSN 0734-2071. doi:10.1145/1132026.1132027.
- Morgan, S.** Cyber Crime Costs Projected To Reach 2 Trillion by 2019. Online, January 2016. [Accessed: 18 June 2018].
URL <https://www.forbes.com/sites/stevemorgan/2016/01/17/cyber-crime-costs-projected-to-reach-2-trillion-by-2019/#6f079d223a91>
- Muscat, I.** How to Mitigate Slow HTTP DoS Attacks in Apache HTTP Server. Online, 2014. [Accessed: 31 May 2018].
URL <http://www.acunetix.com/blog/web-security-zone/articles/slowhttp-dos-attacks-mitigate-apache-http-server>
- Nadeau, T. D. and Gray, K.** SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies. O'Reilly Media, Inc., 2013. ISBN-10: 9781449342302.
- Newman, R. C.** Cybercrime, Identity Theft, and Fraud: Practicing Safe Internet - Network Security Threats and Vulnerabilities. In *Proceedings of the 3rd Annual Conference on Information Security Curriculum Development*, InfoSecCD 2006, pages 68–78. ACM, New York, NY, USA, 2006. doi:10.1145/1231047.1231064.
- Nichols, A.** The Best Threat Intelligence Feeds. Online, September 2016. [Accessed: 8 May 2018].
URL <https://www.anomali.com/blog/the-best-threat-intelligence-feeds>
- NMAP.** Port Scanning Techniques. Online, May 2008. [Accessed: 1 December 2018].
URL <https://nmap.org/book/man-port-scanning-techniques.html>
- NullSec.** Top 1,000 TCP and UDP ports (nmap default). Online, January 2016. [Accessed: 15 November 2018].
URL <https://nullsec.us/top-1-000-tcp-and-udp-ports-nmap-default/>
- 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, New York, NY, USA, 2004. doi:10.1145/1028788.1028794.
- Panjwani, S., Tan, S., Jarrin, K. M., and Cukier, M.** An experimental evaluation to determine if port scans are precursors to an attack. In *In Proceedings International Conference on Dependable Systems and Networks, 2005*, pages 602–611. IEEE, 2005. doi:10.1109/DSN.2005.18.

- Park, K. and Lee, H.** On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. In *ACM SIGCOMM computer communication review*, volume 31, pages 15–26. ACM, 2001. doi:10.1145/964723.383061.
- Perkins, C.** RFC 2003: IP Encapsulation within IP. Technical report, Internet Engineering Task Force (IETF), 1996.
URL <https://tools.ietf.org/html/rfc2003>
- Post Cast Server.** Blocking Open Relays. Online, 2017a. [Accessed: 8 March 2017].
URL http://www.postcastserver.com/help/Blocking_Open_Relays.aspx
- Post Cast Server.** Open Proxy Servers. Online, 2017b. [Accessed: 8 March 2017].
URL http://www.postcastserver.com/help/open_proxy_servers.aspx
- Postal, J.** RFC 793: Transmission Control Protocol. Technical report, Internet Engineering Task Force (IETF), 1981.
URL <https://tools.ietf.org/html/rfc793>
- Postel, J.** RFC 791: Internet Protocol. Technical report, Internet Engineering Task Force (IETF), 1981.
URL <https://tools.ietf.org/html/rfc791>
- Postel, J., Jackson, D., Markson, T., Simpson, B., and Su, Z.-S.** Internet Control Message Protocol (ICMP) Parameters. Online, May 2018. [Accessed: 6 December 2018].
URL <https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml#icmp-parameters-codes-8>
- Proofpoint.** Emerging Threats Intelligence. Online, October 2016. [Accessed: 14 September 2017].
URL <https://www.proofpoint.com/us/products/et-intelligence>
- PSBL.** Passive Spam Block List. Online, August 2002. [Accessed: 15 August 2017].
URL <https://psbl.org/>
- Python Software Foundation.** The Python Standard Library. Online, September 2007. [Accessed: 18 April 2018].
URL <https://www.python.org/download/releases/3.0/>
- Rattan, A., Kaur, N., Chamotra, S., and Bhushan, S.** Attack Data Usability and Challenges in its Capturing and Sharing. *International Journal of Advanced Studies in Computer Science and Engineering (IJASCSE)*, 6:35–42, September 2017.
- Recorded Future.** Threat Intelligence: Difference Between Platforms and Providers. Online, October 2017a. [Accessed: 18 April 2018].
URL <https://www.recordedfuture.com/threat-intelligence-platform/>

- Recorded Future.** Threat Intelligence Feeds: Overview, Best Practices, and Examples. Online, March 2017b. [Accessed: 18 April 2018].
URL <https://www.recordedfuture.com/threat-intelligence-feeds/>
- Rekhter, Y., Li, T., and Hares, S.** RFC 4271: A Border Gateway Protocol 4 (BGP). Technical report, Internet Engineering Task Force (IETF), 2005.
URL <https://tools.ietf.org/html/rfc4271>
- RIPE NCC.** The Internet Registry System. Online, June 2015. [Accessed: 7 December 2018].
URL <https://www.ripe.net/participate/internet-governance/internet-technical-community/the-rir-system>
- Robb, D.** Eight Top Threat Intelligence Platforms. Online, July 2017. [Accessed: 12 November 2018].
URL <https://www.esecurityplanet.com/products/top-threat-intelligence-companies.html>
- Robidoux, G.** SQL Server Stored Procedure Tutorial. Online, May 2012. [Accessed: 3 November 2017].
URL <https://www.mssqltips.com/sqlservertutorial/160/sql-server-stored-procedure-tutorial/>
- Rossi, M., Armitage, G., and Huston, G.** How device misconfiguration drives TCP traffic to parts of 1.0.0.0/8 – an initial investigation. Technical report, Centre for Advanced Internet Architectures, Swinburne University of Technology, July 2011.
- Sanz, I. J., Lopez, M. A., Mattos, D. M. F., and Duarte, O. C. M. B.** A Cooperation-Aware virtual network function for proactive detection of distributed port scanning. In *2017 1st Cyber Security in Networking Conference (CSNet)*, pages 1–8. IEEE, 2017.
- Sauerwein, C., Sillaber, C., Mussmann, A., and Breu, R.** Threat Intelligence Sharing Platforms: An Exploratory Study of Software Vendors and Research Perspectives. In *International Conference on Wirtschaftsinformatik*, volume 13, pages 837–851. 2017.
- Schulting, C.** Networking 101: Understanding BGP Routing. Online, May 2014. Explains what BGP is and some of the basic information behind it.
URL <http://www.enterprisenetworkingplanet.com/netsp/article.php/3615896/Networking-101-Understanding-BGP-Routing.htm>
- Scudder, J.** Router Scaling Trends. Technical report, Juniper Networks, May 2007.

- Senie, D. and Ferguson, P.** RFC 2827: Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. Technical report, Internet Engineering Task Force (IETF), 1998.
URL <https://tools.ietf.org/html/rfc2827>
- Serrano, O., Dandurand, L., and Brown, S.** On the design of a cyber security data sharing system. In *In Proceedings of the 2014 ACM Workshop on Information Sharing & Collaborative Security*, pages 61–69. ACM, 2014. doi:10.1109/CSNET.2017.8242000.
- Shafranovich, Y.** RFC 4180: Common format and MIME type for comma-separated values (CSV) files. Informational, Internet Engineering Task Force (IETF), 2005.
URL <https://tools.ietf.org/html/rfc4180>
- Shah, K., Bohacek, S., and Broido, A.** Feasibility of Detecting TCP-SYN Scanning at a Backbone Router. In *Proceedings of the 2004 American Control Conference*, volume 2, pages 988–995. June 2004.
- Shinder, D.** Understanding Routing Tables. Online, May 2001. [Accessed: 29 May 2018].
URL <https://www.techrepublic.com/article/understanding-routing-tables/>
- Sillaber, C., Sauerwein, C., Mussmann, A., and Breu, R.** Data quality challenges and future research directions in threat intelligence sharing practice. In *In Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security*, pages 65–70. ACM, 2016. doi:10.1145/2994539.2994546.
- Slattery, T. and Burton, W.** Advanced IP routing in Cisco networks. McGraw-Hill, 2000. ISBN: 0-07-212591-8.
- Stewart, R.** RFC 4960: Stream Control Transmission Protocol. Technical report, Internet Engineering Task Force (IETF), 2007.
URL <https://tools.ietf.org/html/rfc4960>
- Stinson, B.** PostgreSQL Essential Reference. Sams Publishing, 2001. ISBN-10: 9780735711211.
- Symantec.** W32.SQLEXP.Worm. February 2007. [Accessed: 6 December 2018].
URL https://www.symantec.com/security_response/writeup.jsp?docid=2003-012502-3306-99
- Tan, Z., Jamdagni, A., He, X., Nanda, P., and Liu, R. P.** A System for Denial-of-Service Attack Detection Based on Multivariate Correlation Analysis. *IEEE Technology and Society Magazine Transactions on Parallel and Distributed Systems*, 25(2):447–456, 2014. doi:10.1109/TPDS.2013.146.

- Techopedia.** Autonomous System (AS). Online, January 2013. [Accessed: 11 June 2018].
URL <https://www.techopedia.com/definition/11063/autonomous-system-as>
- The PostgreSQL Global Development Group.** PostgreSQL. Online, December 1996. [Accessed: 26 September 2017].
URL <https://www.postgresql.org/>
- The SpamHaus Project.** SpamHaus. Online, April 2000. [Accessed: 11 April 2017].
URL <https://www.spamhaus.org>
- thePacketGeek.** ExaBGP and python: Getting Started. Online, May 2015. [Accessed: 23 June 2017].
URL <https://thepacketgeek.com/influence-routing-decisions-with-python-and-exabgp/>
- Tokarev, M.** rbldnsd. June 2015. [Accessed: 27 July 2017].
URL <http://www.corpit.ru/mjt/rbldnsd.html>
- Touch, J., Lear, E., Mankin, A., Kojo, M., Ono, K., Stiernerling, M., Eggert, L., Melnikov, A., Eddy, W., Zimmermann, A., Trammell, B., Iyengar, J., Mankin, A., Tuexen, M., Kohler, E., and Nishida, Y.** Service Name and Transport Protocol Port Number Registry. 2018. [Accessed: 24 November 2018].
URL <https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml>
- Tounsi, W. and Rais, H.** A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & Security*, 72:212–233, 2017. doi:10.1016/j.cose.2017.09.001.
- Tridgel, A.** Rsync. Online Manual, October 2007. [Accessed: 25 October 2017].
URL <https://rsync.samba.org/>
- Trotter, G.** RFC 3222: Terminology for Forwarding Information Base (FIB) based Router Performance. Technical report, Internet Engineering Task Force (IETF), 2001.
URL <https://tools.ietf.org/html/rfc3222>
- van Heerden, R. P., Irwin, B., and Burke, I.** Classifying network attack scenarios using an Ontology. In *Proceedings of the 7th International Conference on Information-Warfare & Security (ICIW 2012)*, pages 311–324. 2012.
- Verisign.** How the Domain Name System (DNS) Works. Online, 2017. [Accessed: 27 July 2017].
URL https://www.verisign.com/en_US/website-presence/online/how-dns-works/index.xhtml

- Vincenti, B.** DNS Blacklists: What are they, how do they work and why do we need them? Online, October 2010. [Accessed: 18 April 2017].
URL <https://techtalk.gfi.com/dns-blacklists-work/>
- Washburn, K. and Evans, J.** TCP/IP: Running a Successful Network. Addison-Wesley, 1993. ISBN-10: 0-201-87711-2.
- Webroot.** BrightCloud IP Reputation Service. Technical report, Webroot, 2018a.
URL https://www-cdn.webroot.com/5715/2511/0532/BC-IP-Reputation-DS_us.pdf
- Webroot.** BrightCloud Web Classification and Reputation Services. Technical report, Webroot, 2018b.
URL https://www-cdn.webroot.com/4415/2511/0672/BC-Web-Classification-and-Reputation-DS_us.pdf
- Webroot.** BrightCloud® Threat Intelligence for IoT Gateways. Technical report, Webroot, 2018c.
URL https://www-cdn.webroot.com/8614/8537/1409/BrightCloud_IoT_Gateway_DS_us.pdf
- Wustrow, E., Karir, M., Bailey, M., Jahanian, F., and Huston, G.** Internet Background Radiation Revisited. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pages 62–74. ACM, 2010. doi:10.1145/1879141.1879149.
- Xiao, L. and Nahrstedt, K.** Reliability Models and Evaluation of Internal BGP Networks. In *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1593–1604. IEEE, 2004.
- Zatlyn, M.** Error 521: Web server is down. September 2018. [Accessed: 11 October 2018].
URL <https://support.cloudflare.com/hc/en-us/articles/200171916-Error-521-Web-server-is-down>
- Zhang, W., Teng, S., and Fu, X.** Scan Attack Detection Based on Distributed Cooperative Model. In *Computer Supported Cooperative Work in Design, 2008. CSCWD 2008. 12th International Conference on*, pages 743–748. IEEE, 2008.

Appendix



Open Source Threat Intelligence

THIS appendix contains additional information relating to the open source threat intelligence data used in this research project. This appendix includes information on the OSINT sources in **Section A.1** and the OSINT sources themselves in **Sections A.2 to A.15**.

A.1 Open Source Threat Intelligence Feeds

This section lists all open source threat intelligence sources used in this project along with the URL which can be used to download or query the feed. Table A.1 lists the OSINT feeds and the URLs where they can be accessed.

A.2 Abuse.ch

Abuse.ch maintains four different projects that track ransomware, malware and malicious domains (Abuse.ch, 2007). The project is focused on providing open source threat intelligence about specific threats on the Internet. Each project provides a list of data specific to the type of threat being monitored. The lists are free for both commercial and non-commercial use. However, commercial vendors are requested to notify the distributor if the lists are to be used in software/services.

Table A.1: Open Source Threat Intelligence Source Data

Source	List	URL
Alexa	Top 1 Million	http://s3.amazonaws.com/alexastatic/top-1m.csv.zip
AlienVault	IP Reputation	http://reputation.alienvault.com/reputation.data
Autoshun	IP Blocklist	https://www.autoshun.org/download/?api_key=[api_key]&format=csv
Bambenek	IP Master	http://osint.bambenekconsulting.com/feeds/c2-ipmasterlist.txt
Binary Defence	Banlist	https://www.binarydefense.com/banlist.txt
Binary Defence	TOR	https://www.binarydefense.com/tor.txt
Brute Force Blocker	Blocklist	http://danger.rulez.sk/projects/bruteforceblocker/blist.php
CINS	CI Army	http://cinsscore.com/list/ci-badguys.txt
Feodo	DBL	https://feodotracker.abuse.ch/blocklist/?download=domainblocklist
Feodo	IPBL	https://feodotracker.abuse.ch/blocklist/?download=ipblocklist
MDL	Complete	http://www.malwaredomainlist.com/mdlcsv.php
MDL	Cleaned	http://www.malwaredomainlist.com/hostslist/delisted.txt
MDL	Active	http://www.malwaredomainlist.com/hostslist/ip.txt
Proofpoint	IPBL	http://rules.emergingthreats.net/fwrules/emerging-Block-IPs.txt
PSBL	PSBL	psbl-mirror.surriel.com:psbl/psbl.txt
Ransomware	URLBL	https://ransomwaretracker.abuse.ch/downloads/RW_URLBL.txt
Ransomware	IPBL	https://ransomwaretracker.abuse.ch/downloads/RW_IPBL.txt
Ransomware	DOMBL	https://ransomwaretracker.abuse.ch/downloads/RW_DOMBL.txt
SANS ICS	High	https://isc.sans.edu/feeds/suspiciousdomains_High.txt
SANS ICS	Medium	https://isc.sans.edu/feeds/suspiciousdomains_Medium.txt
SANS ICS	Low	https://isc.sans.edu/feeds/suspiciousdomains_Low.txt
SANS ICS	IPBL	https://isc.sans.edu/block.txt
SORBS	DNSBL	dnsbl.sorbs.net
SpamHaus	DROP	https://www.spamhaus.org/drop/drop.txt
SpamHaus	EDROP	https://www.spamhaus.org/drop/edrop.txt
SpamHaus	DROPV6	https://www.spamhaus.org/drop/droptv6.txt
SpamHaus	ASN-DROP	https://www.spamhaus.org/drop/asndrop.txt
SpamHaus	SBL	sbl.spamhaus.org
SpamHaus	XBL	xbl.spamhaus.org
SpamHaus	PBL	pbl.spamhaus.org
SpamHaus	ZEN	zen.spamhaus.org
SpamHaus	DBL	dbl.spamhaus.org
SSLBL	Fingerprint	https://sslbl.abuse.ch/blacklist/sslblacklist.csv
SSLBL	IP	https://sslbl.abuse.ch/blacklist/sslipblacklist.csv
SSLBL	Aggressive IP	https://sslbl.abuse.ch/blacklist/sslipblacklist_aggressive.csv
Statvoo	Top 1 Million	https://statvoo.com/dl/top-1million-sites.csv.zip
Umbrella	Top 1 Million	http://s3-us-west-1.amazonaws.com/umbrella-static/top-1m.csv.zip
Umbrella	Top TLDs	http://s3-us-west-1.amazonaws.com/umbrella-static/top-1m-TLD.csv.zip
Zeus	Bad Domains	https://zeustracker.abuse.ch/blocklist.php?download=baddomains
Zeus	Bad IP	https://zeustracker.abuse.ch/blocklist.php?download=badips
Zeus	Domains	https://zeustracker.abuse.ch/blocklist.php?download=domainblocklist
Zeus	IPs	https://zeustracker.abuse.ch/blocklist.php?download=ipblocklist
Zeus	Compromised	https://zeustracker.abuse.ch/blocklist.php?download=compromised

Feodo Tracker is designed to track the Feodo (also known as Cridex or Bugat), these Trojan. The Feodo Trojan is used to conduct e-banking fraud such as stealing credit card information from individuals (Abuse.ch, 2007). The tracker currently monitors five

different versions of the Feodo Trojan:

- **Version A:** is found on hijacked web servers running nginx¹ with 8080/tcp open. This Trojan can be seen targeting these ports from a source which has no domain name.
- **Version B:** is found on web servers with the specific purpose of hosting the Feodo botnet controller. These web servers usually have a domain name within the Country Code TLD (ccTLD) such as “.za”. These web servers should be considered malicious in all cases.
- **Version C:** a spin-off from Feodo currently known as Geodo and Emotet.
- **Version D:** a successor to Cridex called Drindex.
- **Version E:** a new Trojan called Heodo, which was first seen in March 2017. It was built from the Geodo I Emotet Trojan.

Table A.2 details the two feeds published by Abuse.ch containing information on the Feodo Tracker.

Table A.2: Feodo Tracker

List	Description
Domain Blocklist	This list contains a collection of domain names used for Command and Control (C&C) botnet communications. This blocklist only covers version B of the Feodo trojan (Abuse.ch, 2007).
IP Blocklist	This list contains a collection of IP addresses used in the C&C communication processes of the Feodo trojan (Abuse.ch, 2007). The list contains two types of IP addresses. Firstly, IP addresses used by Feodo version A, C and D. Secondly, IP addresses used by Feodo version B servers which are exclusively used for hosting a Feodo C&C server.

Ransomware Tracker is a tracker which monitors malicious domains, IP addresses, and URLs. Typically it monitors known ransomware sources such as botnet Command and Control (C&C) servers, distribution sites and payments sites (Abuse.ch, 2007). There are two categories of lists, combined and separated. The combined list contains a full set of malicious sources. The separated list provides a separated blocklist for each type of malware. The separated list provides more control as to which malware to block. The advantage of using the combined list is the false positive rates are much lower. Table A.3 details the three different lists provided by the Ransomware Tracker.

SSL Blacklist provides a list of bad SSL traffic which has been linked to malware or botnet activities (Abuse.ch, 2007). Table A.4 details the three lists provided by SSL Blacklist.

ZeuS Tracker is designed to track servers on the Internet which have been infected by the ZeuS crimeware kit (Binsalleeh *et al.*, 2010). ZeuS is a botnet created by a collection of servers injected by the Zbot or WSNPoem bot (Abuse.ch, 2007). The purpose of the

¹<https://nginx.org>

Table A.3: Ransomware Tracker

List	Description
Domain Blocklist	This combined list contains a collection of domains identified as being sources of malware.
URL Blocklist	This combined list contains a collection of URLs identified as being malicious URLs.
IP Blocklist	This combined list contains a collection of IP addresses identified as being sources of malware.

Table A.4: SSL Blacklist

List	Description
SSL Fingerprint Blacklist	Contains a collection of SSL certificate fingerprints of all blacklisted SSL certificates (Abuse.ch, 2007). The certificates recorded in this list are identified by Abuse.ch as being associated with malware or botnet activities on the Internet.
SSL IP Blacklist	Contains all the IP address of hosts seen to be using malicious SSL certificates for a period of 30 days prior to the list being updated (Abuse.ch, 2007).
SSL Aggressive Blacklist	Contains a collection of all IP addresses identified by SSL Blacklist that were ever detected to be associated with malicious SSL certificates (Abuse.ch, 2007). This list is aggressive because IP addresses can be reused, and for this reason, the chances of the IP address being reassigned to a new non-malicious host is very likely. As a result, the chances of false positives are much higher.

Zeus botnet is to steal credentials from online services such as social networks and online banking accounts. Table A.5 details the five different lists with information on the Zeus Tracker.

Table A.5: Zeus Tracker

List	Description
Bad Domain Blocklist	Provides a collection of domain names used by the Zeus trojan. The domains in this list are only used for Zeus Trojans, there are no listings of hijacked domains in this list (Abuse.ch, 2007).
Bad IP Blocklist	This list is the IPv4 version of the Bad Domain list.
Domain Blocklist	Contains the same information as the Bad Domain list, with the addition of hijacked websites (Abuse.ch, 2007). For this reason, the chance of false positive domains being recorded are higher.
IP Blocklist	Contains the same information as the Bad IP list, with the addition of hijacked websites (Abuse.ch, 2007). For this reason, the chance of false positive domains being recorded are higher.
Compromised Blocklist	Contains websites which have been hijacked or compromised by cybercriminals to host the Zeus botnet controller (Abuse.ch, 2007). The list contains the exact URL of the configuration or binary file which is used as the controller. This provides a means to block only the controller and not the whole domain.

A.3 Alexa

Alexa is an organization owned by Amazon, the organization provides website ranking based on popularity and visiting traffic (Alexa, 1997). As part of their server, Alexa provides a downloadable list of the top one million sites identified by Amazon. The websites recorded in this list are whitelisted. The list ranks websites based on the number of requests for the website.

A.4 AlienVault

AlienVault is an organization who identified the struggles of inadequate security solutions for organizations or individuals on a budget (AlienVault, 2005). AlienVault provides an affordable threat detection and response platform for customers called Open Threat Exchange (OTX) (AlienVault, 2016). The purpose of OTX is to aid security and IT communities to collaboratively develop and share threat data. An additional service provided is a publicly downloadable threat intelligence dataset. The AlienVault IP Reputation List is a collection IP addresses identified by AlienVault as being sources of cyber-criminal activities or malicious threats. Table A.6 details the information contained in the IP Reputation list.

Table A.6: IP Reputation List Data

Column	Description
1	The malicious IPv4 address.
2	Gives a rating (1–10) of how risky the source is.
3	Gives a rating (1–10) of how confident AlienVault is of the IP address.
4	States the type of activity the IP address is listed for.
5	States the country of origin of the IP address.
6	States the city of origin of the IP address.
7	Is the latitude and longitude location of the IP address.
8	Unknown Column.

A.5 AutoShun

Autoshun is an open source threat intelligence sharing platform (AutoShun, 2007). The organization started out as an email subscription service which slowly grew to a downloadable list and then finally a threat intelligence database in 2007. Autoshun has conducted research into threat intelligence and how beneficial it can be when used on security

devices. From their findings, AutoShun concluded that overloading a firewall with thousands of rules to block IP addresses can have adverse effects when attempting to defend a network (AutoShun, 2007). Furthermore, every Indicator Of Compromise (IOC) or IP address loaded onto a firewall impacts the performance of the device. For this reason, AutoShun limits its list to no more than 2000 IP addresses.

AutoShun IP Block List is a list of no more than 2000 IP addresses considered malicious. AutoShun identified that there is a cost to every IP address added to a firewall. The list is kept to a small size to reduce the impact on speed when it is added to a firewalls rules.

A.6 Bambenek Consulting

Bambenek Consulting is an IT consulting company whose focus is on cyber-security and cyber-crime (Bambenek Consulting, 2011). The founders of the company identified the overwhelming presence of online banking fraud, infrastructure attacks and identify theft affecting the public on a daily basis. Furthermore, they identified the need for good advice and how valuable it can be to the consumer.

For a security administrator to effectively defend against current threats, they need up-to-date and comprehensive threat intelligence. Bambenek Consulting identified this need and addressed it by providing a list of IP addresses of known Command and Control (C&C) servers. The C&C servers in this list of active and non-sinkholed servers.

A.7 Binary Defense

Binary Defense is a security organization specializing in software and services to stop the next generation of attacks (Binary Defense, 2013). Their focus is on providing security for organizations servers and endpoint infrastructure, one such service is the Artillery Project (Kennedy, 2014). Artillery is an open source project maintained by Binary Defense, it was created to provide threat intelligence and early warning indicators to security administrators. The service provides a list called the “Artillery Threat Intelligence Feed and Banlist Feed” which is a set of IP addresses linked to TAs. The list is intended to aid in the creating of rules to block or filter TA IP Addresses.

A.8 Brute Force Blocker

Brute Force Blocker developed a script which monitors a server to identify if it is the victim of a brute force attack (Brute Force Blocker, 2006). Brute Force Blocker will automatically create its own firewall rules and implement them on the server it is monitoring. When it detects and confirms a potential TA IP address, the script will submit the IP address of the brute force attack to its central database. From this central database, a publicly downloadable list of IP addresses is created and published. Table A.7 shows the information contained in the file.

Table A.7: Brute Force Blocker Block List Data

Column	Description
1	The IP address of the malicious client.
2	The date when the brute force attack was last reported.
3	A count of how many times the IP was recorded
4	An ID used by Brute Force Blocker.

A.9 Collective Intelligence Network Security (CINS)

CINS is an IP reputation database that provides accurate and timely threat intelligence to its customers (CINS, 2013). CINS has a collection of sensor units which collect and report on attack data such as TA IP addresses. By collecting more than just the malicious IP address, CINS is able to add a severity level to a malicious IP address. CINS (2013) claims that Internet security should be a fundamental human right and that everyone has the right to be secure on the Internet. Therefore, CINS publishes the “CINS CI Army List” which is a public list containing a subset of their paid-for threat intelligence. For an IP address to be included in this list it must meet two criteria:

1. The IP addresses Rogue Packet Score must be poor. This means the more rouge packets an IP address is sending, the more likely it is to be a malicious source (CINS, 2013).
2. The IP address must not be seen on other threat intelligence lists. This prevents the list from recording IP addresses already identified by other sources such as AlienVault. This ensures the size of the list stays small.

A.10 Cisco Umbrella

Cisco Umbrella is a cloud-based security Internet gateway organization that provides security services to its clients (Cisco Umbrella, 2016). Although the services provided by Cisco Umbrella are pay-to-use, the organization provides two open source lists. The “Top 1 Million Domains” is a list containing a record of the top one million domains currently active on the Internet. The “Top TLDs” is a list containing a record of the top one million Top Level Domains (TLDs) on the Internet for a period of time (Cisco Umbrella, 2016). The lists are created using passive DNS statistics from the Cisco Umbrella global network of routers. By using this data Umbrella is able to report on the popularity ranking of domains based on the domain activity rather than the site activity (Cisco Umbrella, 2016).

A.11 Malware Domain List (MDL)

MDL is a provider of downloadable threat intelligence lists (Malware Domain List, 2007). The lists are made available in CSV or TXT format via a URL. Table A.8 details the lists provided by MDL.

Table A.8: Malware Domain List Lists

List	Description
Hosts	Contains the hostnames of TAs MDL has identified as being sources of malware.
Cleaned	Contains a record of offline or cleaned websites which were previously used as sources of malware.
Active	Includes IP Addresses identified by MDL as actively releasing malware.

A.12 Proofpoint Emerging Threat Intelligence

Proofpoint is an organization which publishes threat intelligence on emerging threats (Proofpoint, 2016). The focus of their threat intelligence is providing information which can be used in the various security hardware tools the organizations provides. A consumer has a choice of paying for a comprehensive threat intelligence list or using the open source threat intelligence list. The pay-to-use list can be inserted directly into the variety of firewalls, IDS, IPS and authentication systems sold by Proofpoint. The open source list called the “Block IP List” is a collection of IP addresses involved in suspicious and malicious activity (Proofpoint, 2016). This list is a collection of unformatted IP addresses in a TXT format. The “Block IP List” is created by collecting open source threat intelligence from various sources including sources such as Abuse.ch (Section A.2).

A.13 Passive Spam Blocklist

Passive Spam Blocklist (PSBL) is an open-access database of known spam sources. Its primary purpose is to provide a platform that mail servers can use to determine if a sender is linked to spam mail (PSBL, 2002). The service has a number of spam traps deployed which are designed to detect spam and report the spam mail to a central database. If one of the spam traps receive a spam email, the IP addresses are recorded into the database. When an IP address is added to the database it is given an expiration date. The purpose of this expiration date is to prevent compromised IP addresses from being permanently listed as spam sources. Spam When the expiry time ends, the IP address is removed from the list. An additional service provided by PSBL is the ability to “rsync” their database to a local computer (PSBL, 2002). This list provides a dataset of all the IP addresses that are currently active in the database.

A.14 SANS ISC

SANS ISC is the Internet Storm Center of SANS. Its purpose is to provide a free analysis and warning service to Internet users and organizations (ISC, 2012). The ISC was formed as a result of the “LiOn Worm” (Hughes, 2001) attack in March 2001. After the emergence of the “LiOn Worm”, the ISC increased the number of sensors deployed in the Internet. The sensors cover over 500 000 IP address, monitoring for signs of malicious activity. The activity captured by these sensors are then made available in four different threat intelligence lists detailed in Table A.9.

Table A.9: SANS Lists

List	Description
High Level	High risk domains identified by the ISC. This list contains fewer false positives than the other lists.
Medium Level	Medium risk domains identified by the ISC.
Low Level	Low risk domains identified by the ISC. This list will contain the most amount of false positive domains.
IP Blocklist	An IP block list which ISC suggests blocking. The list contains the top 20 attacking class C subnets identified in the last three days of being release.

All lists except for the *IP Blocklist* contain a single column of data. Table A.10 shows the information contained in the *IP Blocklist*.

Table A.10: SANS IP Block List Data

Column	Description
1	The starting IP address for the subnet.
2	The end IP address for the subnet.
3	The netmask which the IP address subnet belongs to.
4	The number of targets scanned on the subnet.
5	The name of the subnet.
6	The country where the subnet is registered.
7	A contact email address for the subnet.

A.15 SpamHaus

SpamHaus is an international non-profit organization which tracks spam, phishing, malware, and botnets around the world (The SpamHaus Project, 2000). SpamHaus provides real-time threat intelligence on two main formats, DNSBLs and BPG feeds.

DNS Blocklists (DNSBLs) are IP and domain lists stored on SpamHaus's servers which can be queried by email servers and DNS clients. The blocklists are free to use provided certain criteria are met. Table A.11 details the four different DNSBLs provided by SpamHaus. If a consumer violates the criteria they are blocked from issuing further queries. Free usage is aimed at private mail servers with low amounts of traffic. The list below states the three criteria which must be met to allow free use of the DNSBLs (The SpamHaus Project, 2000). If one of the following criteria is not met, the consumer must register and pay for a commercial license.

1. Use of the SpamHaus DNSBL is for non-commercial use.
2. Email traffic is less than 100 000 SMTP connections per day.
3. DNSBL query volume to SpamHaus is less than 300 000 queries per day.

Table A.11: DNSBL Sources

List	Description
SpamHaus Blocklist (SBL)	A collection of IP addresses which are known to be sources of malicious emails. If a client receives emails from an IP address on this list it is recommended to block the email because the chances of it being spam are high.
Exploits Blocklist (XBL)	A collection of IP addresses of hijacked computers. These computers have been hijacked by exploitation's such as open proxies, worms or viruses and other forms of Trojans.
Policy Blocklist (PBL)	A collection of end-user IP address ranges which should not be delivering unauthenticated SMTP email. The IP addresses in this list are both dynamic and static IP addresses.
Domain Blocklist (DBL)	A list of domains that should be considered to be sources of spam (The SpamHaus Project, 2000). The DBL is suitable for mail servers which are capable of scanning the body of the emails to identify the domain from which the email is being sent.

ZEN combines the DNSBLs detailed in Table A.11 into a single blocklist. ZEN can

be queried by mail servers to determine whether a IP Address is malicious or not. By combining the individual lists into one comprehensive database, queries become simpler and faster. Based on whether the IP address is listed or not, Zen will return a result code. Table A.12 lists the result codes which can be returned from a query.

Table A.12: ZEN Result Codes

Return Code	Zone	Description
127.0.0.0/24	N/A	The response is from the SpamHaus IP Blocklists
127.0.1.0/24	N/A	SpamHaus Domain Blocklist
127.0.2.0/24	N/A	SpamHaus Whitelists
127.0.0.2	SBL	The response is from SBL
127.0.0.3	SBL	The response is from SBL CSS. The CSS blocklists is a sub-list part of SBL
127.0.0.4	XBL	The response is from CBL ²
127.0.0.9	SBL	From the DROP and EDROP list
127.0.0.10	PBL	From the ISP maintained list
127.0.0.11	PBL	From the SpamHaus maintained list

DROP List is an additional service provided by SpamHaus is a collection of downloadable datasets called DROP (Don't Route or Peer) lists detailed in Table A.13. These lists contain a collection of netblocks which have been identified as hijacked or rented for cyber-crime activities. The IP addresses in the DROP list are a small subset of the SBL which can be used on routers and firewalls to filter malicious traffic. The DROP lists are free to use by the public. However, SpamHaus does state the following in terms of usage for the lists.

1. If the lists are used in products or services, credit is given to SpamHaus.
2. Do not completely modify the lists comment headers. For example, leave the date and copyright symbol in the files.
3. The DROP lists must be kept up-to-date regularly.
4. Drop lists should not be downloaded more than once an hour and less than once per day.

²cbl.abuseat.org

Table A.13: DROP Lists

List	Description
DROP List	The standard list of malicious IP addresses. This list will only include IP addresses of netblocks associated with malicious activity. It will not include IP addresses of legitimate networks. The list contains two columns, one with the IPv4 address and another with an internal reference number used by SpamHaus
Extended DROP (EDROP)	An extended version of the DROP list. EDROP will contain a list of netblocks which are controlled by spammers and cybercriminals. The list contains two columns, one with the IPv4 address or netblock and another with an internal reference number used by SpamHaus
IPv6 DROP List (DROPv6)	Includes IPv6 addresses allocated to cybercriminals. The list contains two columns, one with the IPv6 address and another with an internal reference number used by SpamHaus
ASN DROP List (ASN-DROP)	A collection of Autonomous Systems (AS) controlled by cybercriminals or hijacked by cybercriminals. The list contains two columns, one with the IANA (Internet Assigned Numbers Authority) number of the malicious AS and another with information about the AS. The country it is registered in and the name of the registered entity.

B

Project Repository

THIS appendix details electronic resources relating to this this research project. There are two repositories for this research. The system code repository contains only the relevant scripts and libraries needed to run the system (Collection, Processing and Publishing). The scripts repository contains the analysis scripts used during the data quantification and data analysis. Further information about the contents of the repositories can be found in the README files online.

B.1 System Code

This online repository contains the code for a working implementation of the system described in Chapters 4 and 6. The system can be executed within a pipenv¹ environment. The repository is made available at:

https://bitbucket.org/deontp/masters_system.

¹<https://pipenv.readthedocs.io/en/latest/>

B.2 Analysis Scripts

This online repository contains a collection of the scripts used throughout this research. Within this repository are a collection of folders, one for each of the chapters where the scripts appear. The repository is made available at:

https://bitbucket.org/deontp/masters_scripts/



Design and Implementation

THIS appendix provides additional information on the design and implementation of the system described in this research. This appendix includes an overview of the PCAP processing script (**Section C.1**), database design (**Section C.2**), custom libraries (**Section C.3**), module specific data (**Section C.4**), DNSBL (**Section C.5**), and BGP (**Section C.6**) specific configurations.

C.1 Process PCAPS Script

The telescopes record data in PCAP format (Section 2.6) which was made accessible to this project. The data analysis performed during this research required the PCAP files to be processed into JSON files. This script performed the follow functionality:

- Create a dictionary of unique source IP addresses for each telescope sensor.
- Each IP address key in the dictionary contained a count for the number of packets it received, and a dictionary with the packet information.

The purpose of this script is to speed up the accessing of data during the analysis and processing.

C.2 Database Design

This section presents additional database design information referred to from the main document. Table C.1 describes the fields used in the *sources* table.

Table C.1: Sources Table Design

Field	Type	Description
url	[PK]text	The URL where the data is downloaded from.
list	text	The name of the OSINT feed. This field is specific to the system and not necessarily the name of the feed given by the sources.
source	text	The organisation or maintainer of the open source intelligence feed.
resource_type	text	The type of resource, either file or rsync.
active	boolean	Sets whether the <i>Collector</i> script must include the OSINT feed in the next download cycle.
code	character(4)	A unique four-character code used to identify the list source.
category	text	The category that the data falls into. This is a foreign key from the <i>categories</i> table.
last_collected	date	This field records when the OSINT feed was last downloaded.
common_name	text	Is the common name give to the OSINT feed, i.e it is the name used in documentation
data_type	text	Is the type of data being distributed, for example, IP addresses, URLs, domains, etc.
delimiter	character(1)	This field records the delimiter that is used by the OSINT feed.
comment_char	character(1)	This field records the character used to start a comment in OSINT feed.
primary_field	text	This field records the field in the OSINT feed that is or should be used as the primary key field. It is the filed that holds the most important information in the OSINT feed.
fields	text	This field holds a string representation of a dictionary of <i>field:type</i> pairs. The field value is taken from the fields in the OSINT feed file, and the type is decided based on the primary field value.
analysis	boolean	sets whether the OSINT feed will be used in the processing and analysis component.
interesting	boolean	sets whether the OSINT feed will be used in the data analysis chapter.

C.3 Custom Libraries

This research made use of a number of custom libraries that were designed to group similar functions into one location. This section briefly describes the various libraries to give the reader an understanding of their purpose. Not included in this section are the module specific libraries discussed in Appendix C.4.

DB Handler is the the database handler library. It groups together commonly used functions to manage the database. To allow communication between Python and the Postgres database, the *Psycopg2* library is used as the Postgres adaptor. Table C.2 lists the main functions within the DB Handler Library. In addition to these functions, the library is responsible for making a connection to the postgres database.

Table C.2: DB Handler Functions

Function	Description
<code>get_tables</code>	Queries the database for a list of all the user created tables.
<code>get_feed_tables</code>	Returns a list of all the OSINT feed tables in the database.
<code>create_table</code>	Given a table name and a dictionary of fields, creates a table in the database.
<code>drop_table</code>	Given a table name, drops the table from the database.
<code>get_column_names</code>	Returns a list containing the columns of a given table in the database
<code>insert_from_file</code>	Given a CSV file pointer and a table name, the function inserts the contents of the file into the table.
<code>conditional_get_records</code>	Gets a record from the database which match a condition.
<code>get_records</code>	Retrieves all the records from a given table. A limit can be applied to retrieve a set amount of records.
<code>get_record_count</code>	Gets the total number of records in a specific table.
<code>custom_select</code>	Provides the ability to send specific select queries to the database. The type of queries sent will be those which do not have specific functions in the DB Handler library.
<code>call_stored_procedure</code>	In Postgres a stored procedure is called a function. Calling a function is performed in a select query. For example, “ <code>SELECT example(arg1, arg2)</code> ” will call the <code>example</code> function with <code>arg1</code> and <code>arg2</code> .
<code>close_connection</code>	Closes the Pyscopg2 cursor and connection to the database once.
<code>commit_transaction</code>	Commits the transaction and saves the database
<code>execute</code>	Given an SQL query and a set of parameters, the function issues the query to the database.

File Handler library provides commonly used file functions. The functions extend the standard I/O functions in Python. Table C.3 lists the main File Handler functions in the library and their purpose.

Table C.3: File Handler Functions

Function	Description
<code>read_raw_file</code>	A RAW file is a file containing threat intelligence which has not been modified by the system. The file is in the format which it was sent from the origin source.
<code>output_to_csv</code>	This function is used to format a give file into a CSV file. The purpose of this function is to get all the different RAW files into a standard format.
<code>save_file</code>	This functions accepts a file name and list of data. It will save the data contents to a file without any special formatting.
<code>create_folder</code>	This function creates a new folder from a given path.
<code>process_file_name</code>	This function is given a filename in a format specific to the filename created during the collection process. The returned data is a list of OSINT feed name, OSINT list name and timestamp.
<code>extract_contents</code>	This function is used in conjunction with the <code>save_file</code> function. Its purpose is to extract the contents of a compressed file. For some of sources of threat intelligence, the data is downloaded in a compressed file. This function will extract the contents of the file and save it to the local machine.

Logger library provides the logging functionality used throughout the system. It implements the Python Logging¹ library. The library provides four main log levels in the form of functions, *info*, *debug*, *warning*, and *error*.

¹<https://docs.python.org/3/library/logging.html>

Source Handler library provides functions to manage the OSINT metadata. The list metadata holds information about each of the OSINT sources. Table C.4 lists the main functions and what they do. This library communicates with the *sources* table of the database to retrieve information about the various OSINT feeds.

Table C.4: Source Handler Functions

Function	Description
<code>get_feed_metadata</code>	All the lists used in the system have a metadata. This function is able to accept a source and list name and return a list back.
<code>get_feed_metadata</code>	This function returns the full OSINT feed metadata (Appendix C.2).
<code>get_unique_codes</code>	Each list has a unique code. This function generates a random code that can be used when adding new feeds to the database.
<code>get_category_id</code>	Gets the category of a given OSINT feed from the database.
<code>add_feeds_to_db</code>	Provides the functionality to add a collection of JSON formatted OSINT feeds to the database.

Source Specific Functions library provides functions specifically designed to deal with OSINT feeds that are badly formatted or saved in a none standard format.

C.4 Module Data

This section presents additional information about the three system modules (Collection, Processing, and Publishing).

C.4.1 Collector Module

The collection process described in Section 4.4 is achieved using the *Collection* library. Table C.5 lists the functions contained within the library.

C.4.2 Processing Module

This library is the controller for the processing module. The main function that is called by the system is the *process* function. This function simply calls each implemented module one at a time. In the case of the whitelisting module discussed in Section 6.1.1 this *process* function will call the *white_listing* function to perform the whitelisting processing. The *white_listing* function is given a list of tables to filter out trusted IP addresses.

Table C.5: Collection Library

Function	Description
download_list	Issues the HTTP get request using the Python urllib ² library. it returns the response to calling function.
rsync_list	Is used in the event that the OSINT feed uses the Rsync service to distribute the data.
update_db_with_failed	After the OSINT lists have been downloaded, this function updates the database with the lists that failed.
strip_data	Removes any additional carriage-return line-feeds.
pre_process_data	The function processes the newly download OSINT data into a single common format.
collect	Is the main function of the collection library that calls the other functions described in this table.
consolidate_tables	The function ensures that the database contains a table for all the OSINT feeds about to be collected. This ensures that when the populate_db function is called, there will not be any errors.
manage_index	controls the creation and deletion of indexing on a given table.
backup_files	After the collection and database population has occurred, this function is called to move the pre-processed files to a backup folder.
populate_db	This function take the download OSINT data and adds the records to the database. The function uses a temporary table to speed up the database insertion.

C.4.3 Publishing Module

This subsection describes the three distribution modules designed in this system. The publishing script has one standard function call *run*, which controls the execution of the implemented modules. To conform to a single standard, a module (function) being called from the *run* function must accept a list of tables in the database to pull information from, anything else the module does is no concern to the *run* function. In the system code repository is the *data* folder which contains example output for each of the three modules.

Flat File function (*publish_flat_file*) simple takes the records contained in each of the provided tables and outputs them to a JSON file. The module is designed to create separate files for each table, however, it can be modified to create on large file. The *write_file* function is called by the flat file function to write the JSON file to disk.

DNSBL function (*publish_zone_file*) creates a zone file that RBLDNSD uses when DNS queries are made to the service.

BGP function (*publish_bgp*) creates the temporary route instruction file that ExaBGP uses when adding or removing routes from the router.

C.5 DNSBL Configuration

For the DNSBL to perform its function, the RBLDNS service is used. Listing C.1 illustrates the zone file format and an explanation given in Table C.6.

```

1  :127.0.0.2:$ is a known spam source
2  203.0.113.5
3  192.0.2.0/24 :127.0.0.3:Address $ is from a private IP range
4  203/4 Reserved multicast address
5  192.0
6  198.51.100.0-129
7  !195.51.100.50
8  example.com
9  *.example.com
10 $NS 3000 ns1.example.com ns2.example.com

```

Listing C.1: Zone File Line Options (Tokarev, 2015)

Table C.6: Zone File Lines (Tokarev, 2015)

Line	Description
Line 1	The first line containing a “:” is the default response which will be sent if the subsequent lines do not have a value associated with an IP address. The “\$” will be replaced with the IP address.
Line 2	The simplest entry, a single IP address with no value associated with it.
Line 3	An entry which contains a netblock of IP addresses and an associated value.
Line 4	A netblock of IP addresses with an explicit TXT record associated with it.
Line 5	An abbreviation of IP addresses, this is the same as “192.0.0.0/16”.
Line 6	The last octet is repeated. This entry includes IP addresses from “198.51.100.0” up to “198.51.100.129”.
Line 7	Excludes the IP address. This entry goes along with the entry in Line 6.
Line 8	A simple domain name entry which will be returned with the default TXT value.
Line 9	A wildcard domain name entry which will be returned with the default TXT value. A wildcard will match or sub-domains of “example.com”
Line 10	An entry which includes two name servers.

C.6 BGP Configuration

As described in Section 2.4, ExaBGP is the service used to communicate with the BGP routers. This service requires two files when it starts up, the environment file and the configuration file.

ExaBGP Environment File contains daemon specific configurations such as the the user that is executing the daemon, whether the service should run in the background or not, and other additional options. The example in the project repository is the file used for the system designed in this research.

ExaBGP Configuration File is used by the ExaBGP service to configure the neighbouring routers that must be used in the peering session, it also contains location of the

Python script that must be executed when the ExaBGP service has successfully started. There are three important sections of the configuration file that must be present.

- **Process** tells ExaBGP what script must be executed when the daemon starts up. In this instance it is a Python script in the `./bin/` folder.
- **Neighbor** sets the peering information for BGP neighbouring routers. This example shows one neighbor with the following settings:
 - The neighbor IP address is `192.0.2.2`, this is the BGP router ExaBGP is set to peer with.
 - The router ID of ExaBGP is `192.0.2.10`
 - Local AS and Peer AS show the AS numbers to assign to each peer.
- **API** calls the process script to run. Telling ExaBGP that routers will be added and removed.

C.6.1 ExaBGP Script

The ExaBGP script provides the route addition and removal functionality to the BGP module. Appendix C.6.1 lists the functions contained in the script with a brief description of them.

Table C.7: ExaBGP Script

Line	Description
<code>reload_routes</code>	The function is designed to be called when the script first executes. It checks if there are any routes that were previously announced to peers. If there were any routes, the function restores them and announces them to the routing table.
<code>get_file</code>	This function retrieves a new file from the temporary file storage. The file is a new set of commands created by the BGP publishing module (Appendix C.4.2). The function returns a path to the file.
<code>save_routes</code>	This function saves the list of currently added routes to a text file on the local system.
<code>announce_route</code>	This function is used to announce the route to ExaBGP. It formats the IP addresses into the correct format.
<code>withdraw_route</code>	Same as <code>announce_route</code> above, except this function formats the IP addresses for withdrawing from ExaBGP.
<code>issue_commands</code>	Determines which function (<code>announce_route</code> or <code>withdraw_route</code>) based on the list of command it is given.
<code>main</code>	The main function calls all the above mentioned functions whilst in an endless loop.

D

Data Analysis

THIS appendix contains additional information and results from the data analysis chapter (Chapter 5). The main body of discussion uses a selection of OSINT and IBR analysis results; the other results are included in this chapter. The structure of this appendix follows that of Chapter 5, with additional analysis results in **Section D.5**.

D.1 Monthly Change in Dataset Composition

This section contains two tables with the values used to plot the OSINT (Figure 5.3) and IBR (Figure 5.4) graphs.

Table D.1: Month-to-Month Change in OSINT Data Values

Month	OSINT Feed	Non-Recurring IPs	New IPs	Recurring IPs	Total IPs
09/17	AlienVault	0	60 424	0	60 424
	Autoshun	0	9 369	0	9 369
	Binary Defense Banlist	0	9 407	0	9 407
	Bruteforce Blocker	0	2 384	0	2 384
	CINS	0	38 942	0	38 942
10/17	AlienVault	26 087	69 786	34 337	104 123
	Autoshun	9 018	5 851	351	6 202
	Binary Defense Banlist	7 152	4 735	2 255	6 990
	Bruteforce Blocker	1 279	2 420	1 105	3 525
	CINS	22 218	51 537	16 724	68 261

11/17	AlienVault	30 821	74 774	73 302	148 076
	Autoshun	5 718	4 643	484	5 127
	Binary Defense Banlist	4 286	5 271	2 704	7 975
	Bruteforce Blocker	1 328	2 023	2 197	4 220
	CINS	46 268	64 074	21 993	86 067
12/17	AlienVault	72 544	69 604	75 532	145 136
	Autoshun	4 395	2 113	732	2 845
	Binary Defense Banlist	5 696	4 172	2 279	6 451
	Bruteforce Blocker	2 379	1 227	1 841	3 068
	CINS	60 070	63 568	25 997	89 565
01/18	AlienVault	78 537	53 517	66 599	120 116
	Autoshun	2 148	1 610	697	2 307
	Binary Defense Banlist	2 071	4 733	4 380	9 113
	Bruteforce Blocker	1 986	1 481	1 082	2 563
	CINS	64 222	49 040	25 343	74 383
02/18	AlienVault	60 795	8 801	59 321	68 122
	Autoshun	1 679	1 331	628	1 959
	Binary Defense Banlist	4 767	4 075	4 346	8 421
	Bruteforce Blocker	1 131	1 320	1 432	2 752
	CINS	54 347	25 091	20 036	45 127
03/18	AlienVault	46 574	53 208	21 548	74 756
	Autoshun	1 393	1 512	566	2 078
	Binary Defense Banlist	0	4 189	8 421	12 610
	Bruteforce Blocker	1 238	1 397	1 514	2 911
	CINS	24 869	47 553	20 258	67 811
04/18	AlienVault	17 998	48 419	56 758	105 177
	Autoshun	1 881	2 858	197	3 055
	Binary Defense Banlist	171	5 867	12 439	18 306
	Bruteforce Blocker	1 470	983	1 441	2 424
	CINS	45 558	44 670	22 253	66 923
05/18	AlienVault	47 649	52 702	57 528	110 230
	Autoshun	2 437	2 029	618	2 647
	Binary Defense Banlist	13 087	7 247	5 219	12 466
	Bruteforce Blocker	1 481	688	943	1 631
	CINS	45 458	47 633	21 465	69 098
06/18	AlienVault	53 552	52 706	56 678	109 384
	Autoshun	1 920	1 715	727	2 442
	Binary Defense Banlist	6 481	6 259	5 985	12 244
	Bruteforce Blocker	979	638	652	1 290
	CINS	47 662	49 553	21 436	70 989
07/18	AlienVault	48 388	57 989	60 996	118 985
	Autoshun	1 717	1 242	725	1 967
	Binary Defense Banlist	7 263	6 171	4 981	11 152
	Bruteforce Blocker	670	713	620	1 333
	CINS	49 885	54 829	21 104	75 933
08/18	AlienVault	56 844	53 905	62 141	116 046
	Autoshun	1 302	2 666	665	3 331
	Binary Defense Banlist	6 851	7 079	4 301	11 380
	Bruteforce Blocker	693	781	640	1 421
	CINS	55 181	49 754	20 752	70 506

Table D.2: Month-to-Month Change in IBR Data Values

Month	OSINT Feed	Non-Recurring IPs	New IPs	Recurring IPs	Total IPs
09/17	IBR1	0	853 917	0	853 917
	IBR2	0	819 369	0	819 369
	IBR3	0	850 732	0	850 732
	IBR4	0	851 539	0	851 539
	IBR5	0	847 845	0	847 845
	IBR6	0	250 135	0	250 135
	IBR7	0	2 100 680	0	2 100 680
10/17	IBR1	730 784	774 306	123 133	897 439
	IBR2	695 958	774 028	123 411	897 439
	IBR3	720 067	795 899	130 665	926 564
	IBR4	721 299	796 514	130 240	926 754
	IBR5	718 127	795 168	129 718	924 886
	IBR6	184 679	206 924	65 456	272 380
	IBR7	1 760 361	1 757 821	340 319	2 098 140
11/17	IBR1	766 992	1 220 960	130 447	1 351 407
	IBR2	763 898	1 219 175	133 541	1 352 716
	IBR3	786 916	1 252 916	139 648	1 392 564
	IBR4	787 328	1 251 580	139 426	1 391 006
	IBR5	786 255	1 250 311	138 631	1 388 942
	IBR6	211 869	318 214	60 511	378 725
	IBR7	1 640 708	2 893 588	457 432	3 351 020
12/17	IBR1	1 122 320	1 171 603	229 087	1 400 690
	IBR2	1 123 470	1 155 283	229 246	1 384 529
	IBR3	1 154 379	1 224 109	238 185	1 462 294
	IBR4	1 153 283	1 210 041	237 723	1 447 764
	IBR5	1 152 105	1 211 396	236 837	1 448 233
	IBR6	292 465	283 987	86 260	370 247
	IBR7	2 465 537	2 636 704	885 483	3 522 187
01/18	IBR1	1 222 234	951 684	178 456	1 130 140
	IBR2	1 205 318	1 018 813	179 211	1 198 024
	IBR3	1 273 993	1 129 503	188 301	1 317 804
	IBR4	1 259 617	1 014 033	188 147	1 202 180
	IBR5	1 261 903	982 736	186 330	1 169 066
	IBR6	305 397	200 772	64 850	265 622
	IBR7	2 684 948	2 328 366	837 239	3 165 605
02/18	IBR1	969 620	913 053	160 520	1 073 573
	IBR2	1 037 664	877 166	160 360	1 037 526
	IBR3	1 150 017	929 834	167 787	1 097 621
	IBR4	1 035 114	903 667	167 066	1 070 733
	IBR5	1 003 235	918 990	165 831	1 084 821
	IBR6	214 368	153 044	51 254	204 298
	IBR7	2 320 905	2 062 389	844 700	2 907 089
03/18	IBR1	891 665	1 185 695	181 908	1 367 603
	IBR2	855 981	1 184 880	181 545	1 366 425
	IBR3	910 380	1 226 508	187 241	1 413 749
	IBR4	883 958	1 194 413	186 775	1 381 188
	IBR5	899 424	1 200 876	185 397	1 386 273
	IBR6	154 204	212 666	50 094	262 760
	IBR7	1 930 335	2 605 212	976 754	3 581 966

04/18	IBR1	1 126 983	970 082	240 620	1 210 702
	IBR2	1 126 847	932 705	239 578	1 172 283
	IBR3	1 168 901	978 649	244 848	1 223 497
	IBR4	1 135 329	968 092	245 859	1 213 951
	IBR5	1 142 181	974 047	244 092	1 218 139
	IBR6	194 802	185 937	67 958	253 895
	IBR7	2 465 819	2 097 983	1 116 147	3 214 130
05/18	IBR1	1 028 284	876 931	182 418	1 059 349
	IBR2	991 849	834 652	180 434	1 015 086
	IBR3	1 038 500	872 255	184 997	1 057 252
	IBR4	1 026 894	867 222	187 057	1 054 279
	IBR5	1 034 192	874 051	183 947	1 057 998
	IBR6	186 905	180 971	66 990	247 961
	IBR7	2 479 084	2 112 633	735 046	2 847 679
06/18	IBR1	898 242	831 163	161 107	992 270
	IBR2	855 229	802 318	159 857	962 175
	IBR3	892 447	843 495	164 805	1 008 300
	IBR4	888 326	832 611	165 953	998 564
	IBR5	894 723	827 027	163 275	990 302
	IBR6	180 771	220 516	67 190	287 706
	IBR7	2 352 326	2 029 177	495 353	2 524 530
07/18	IBR1	825 024	848 430	167 246	1 015 676
	IBR2	796 320	812 402	165 855	978 257
	IBR3	838 537	843 082	169 763	1 012 845
	IBR4	826 262	857 772	172 302	1 030 074
	IBR5	821 231	844 797	169 071	1 013 868
	IBR6	210 304	201 433	77 402	278 835
	IBR7	2 075 178	2 186 059	449 352	2 635 411
08/18	IBR1	837 270	909 581	178 406	1 087 987
	IBR2	800 656	902 416	177 601	1 080 017
	IBR3	831 756	909 108	181 089	1 090 197
	IBR4	847 184	899 275	182 890	1 082 165
	IBR5	834 450	895 148	179 418	1 074 566
	IBR6	197 499	175 154	81 336	256 490
	IBR7	2 121 035	2 414 989	514 376	2 929 365

D.2 Monthly Offset Comparison

This section details the seven graphs from the analysis performed in Section 5.3.

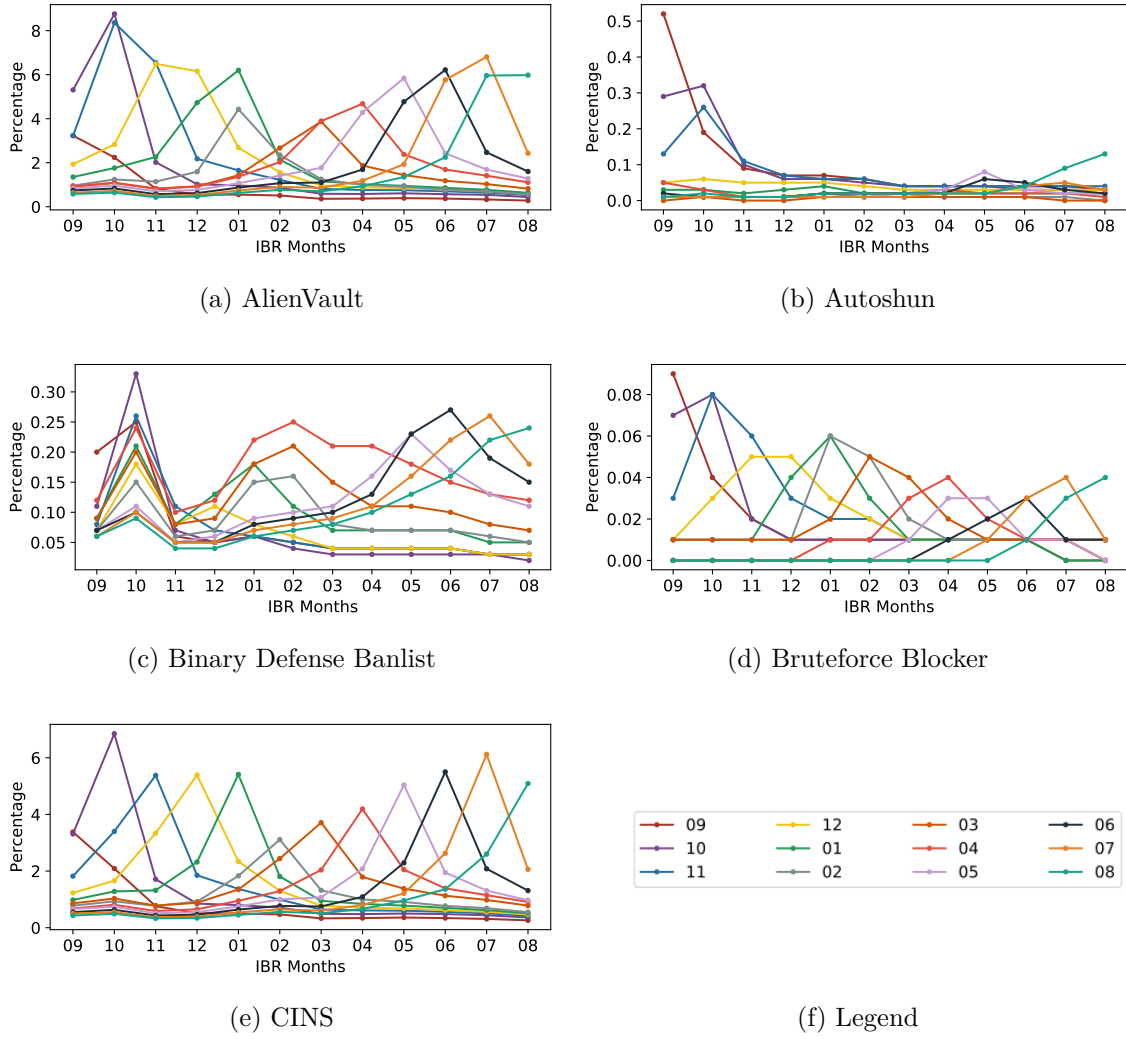
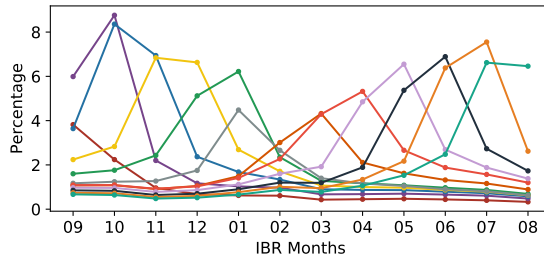
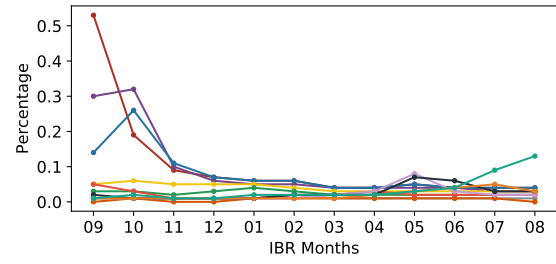


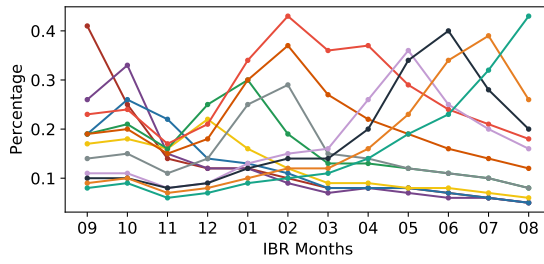
Figure D.1: IBR1 Monthly Offset Comparison



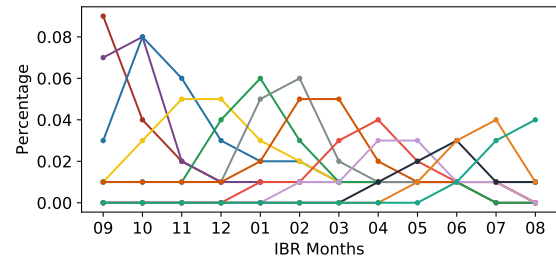
(a) AlienVault



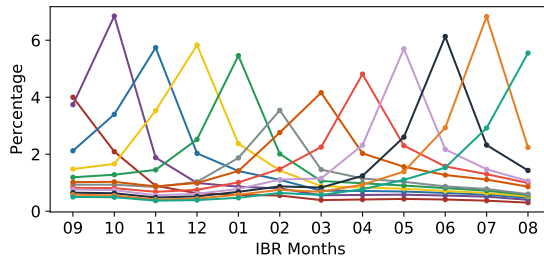
(b) Autoshun



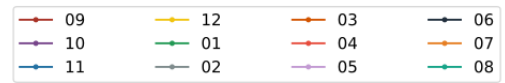
(c) Binary Defense Banlist



(d) Bruteforce Blocker

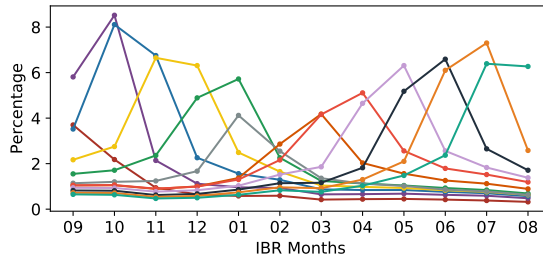


(e) CINS

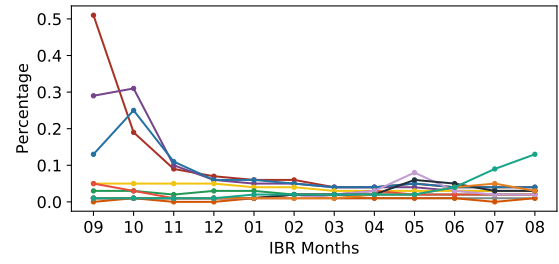


(f) Legend

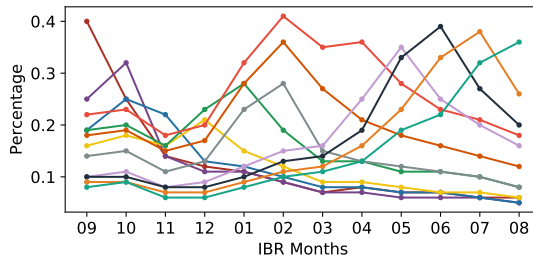
Figure D.2: IBR2 Monthly Offset Comparison



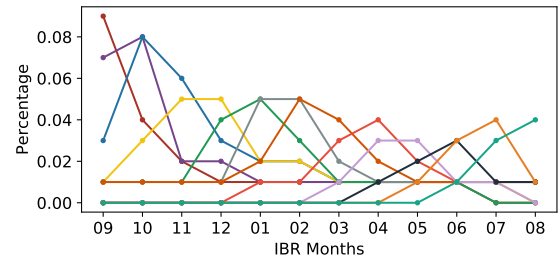
(a) AlienVault



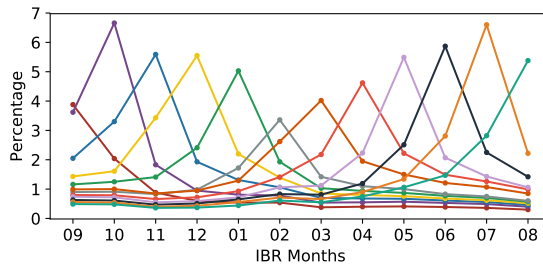
(b) Autoshun



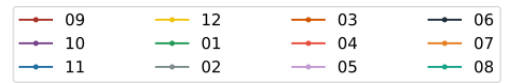
(c) Binary Defense Banlist



(d) Bruteforce Blocker

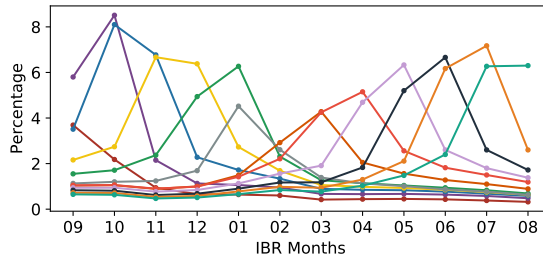


(e) CINS

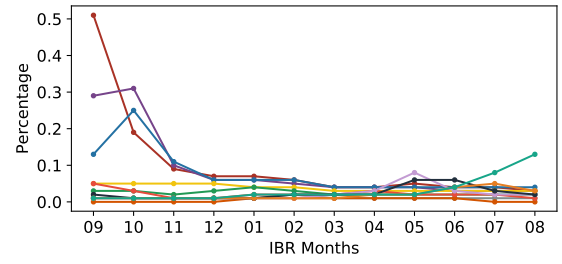


(f) Legend

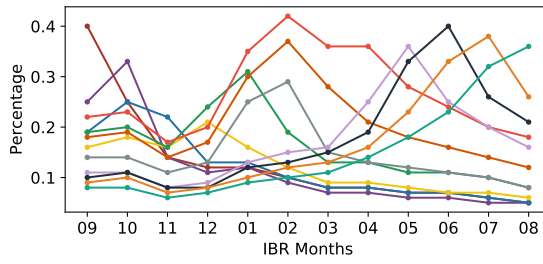
Figure D.3: IBR3 Monthly Offset Comparison



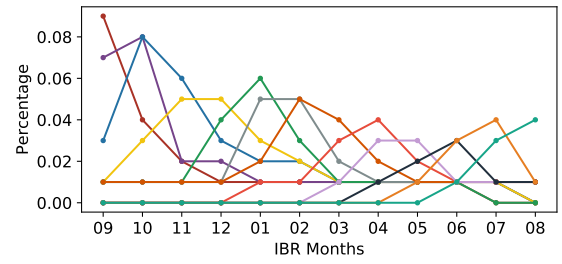
(a) AlienVault



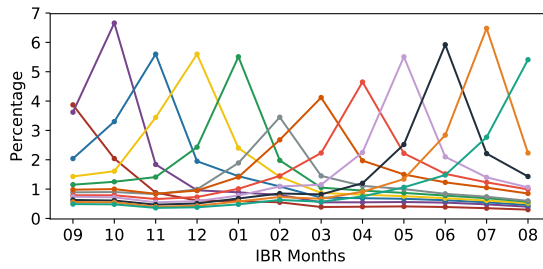
(b) Autoshun



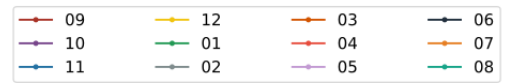
(c) Binary Defense Banlist



(d) Bruteforce Blocker

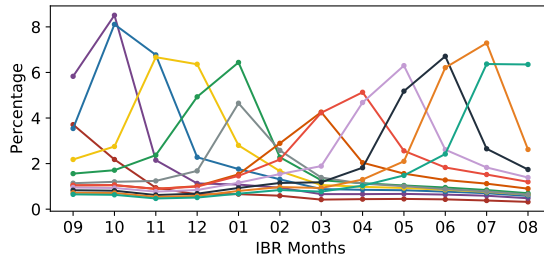


(e) CINS

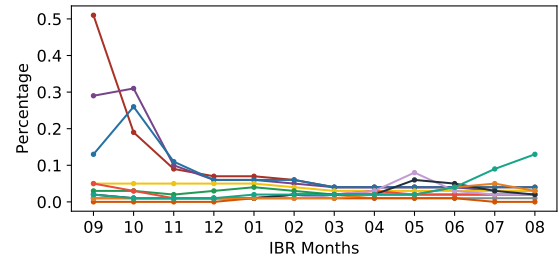


(f) Legend

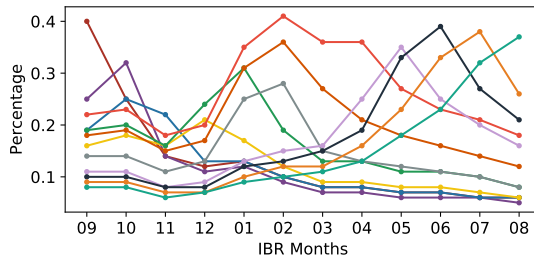
Figure D.4: IBR4 Monthly Offset Comparison



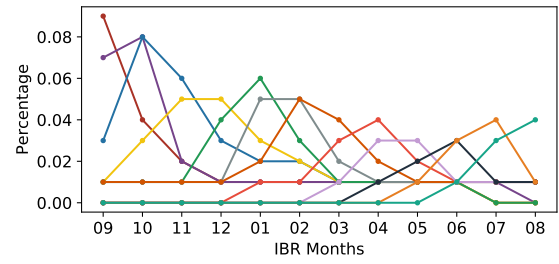
(a) AlienVault



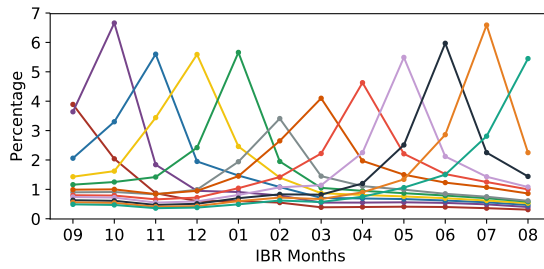
(b) Autoshun



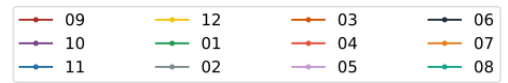
(c) Binary Defense Banlist



(d) Bruteforce Blocker

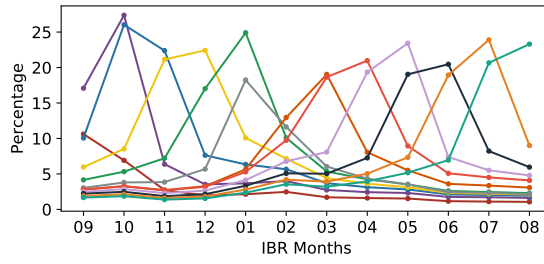


(e) CINS

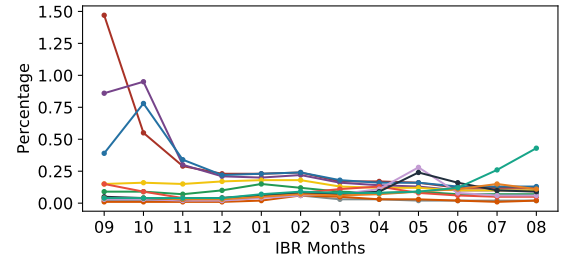


(f) Legend

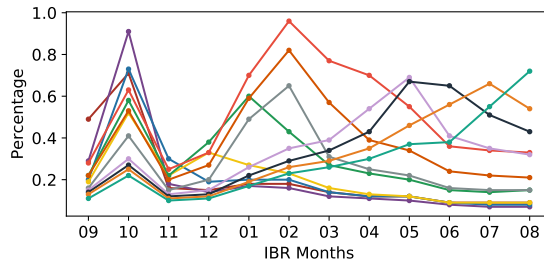
Figure D.5: IBR5 Monthly Offset Comparison



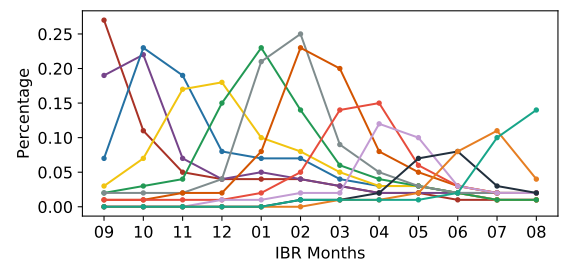
(a) AlienVault



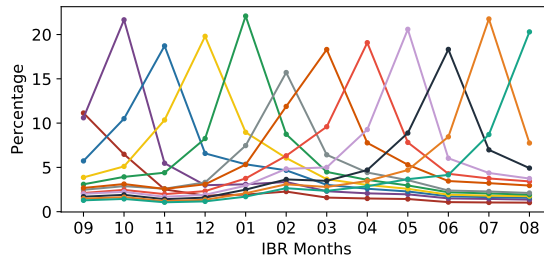
(b) Autoshun



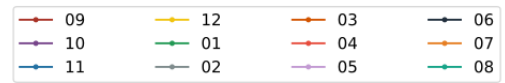
(c) Binary Defense Banlist



(d) Bruteforce Blocker

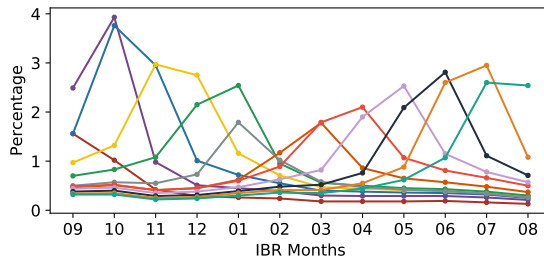


(e) CINS

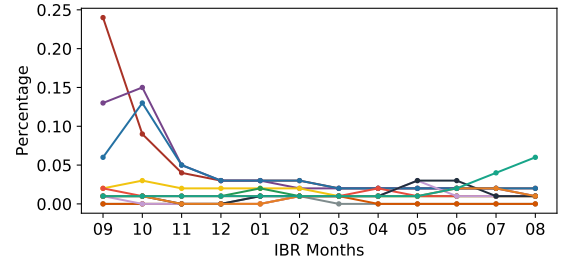


(f) Legend

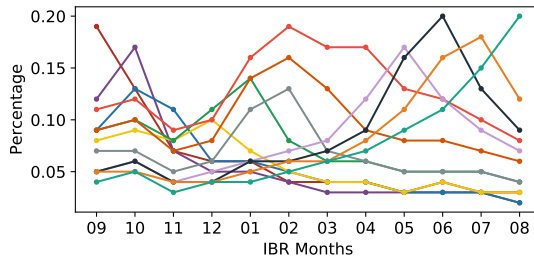
Figure D.6: IBR6 Monthly Offset Comparison



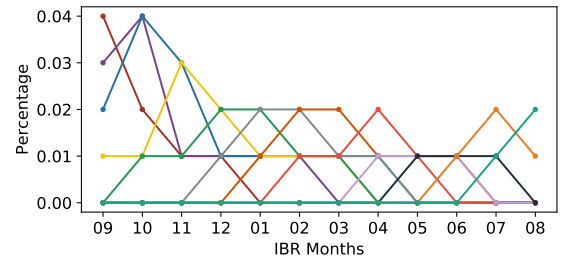
(a) AlienVault



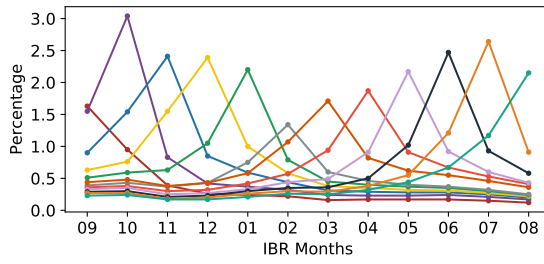
(b) Autoshun



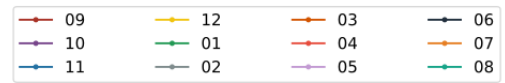
(c) Binary Defense Banlist



(d) Bruteforce Blocker



(e) CINS



(f) Legend

Figure D.7: IBR7 Monthly Offset Comparison

D.3 Persistent IP Addresses

The tables listed in this section contain the values used to plot Figure 5.7. Table D.3a contains the results for the five OSINT datasets chosen for this analysis, and Table D.3b contains the results for the seven IBR datasets.

Table D.3: Persistent IP Addresses

(a) OSINT Data

Window	AV	AS	BD	BB	CI
2	34 337	351	2 255	1 105	16 724
3	15 895	120	958	167	9 313
4	10 023	78	526	38	6 484
5	7 015	43	439	8	4 882
6	5 574	25	285	3	3 541
7	3 893	18	285	3	3 064
8	3 588	14	276	1	2 593
9	3 142	14	164	0	2 164
10	2 654	14	132	1 290	1 894
11	2 332	12	115	620	1 609
12	1 976	10	104	41	1 382

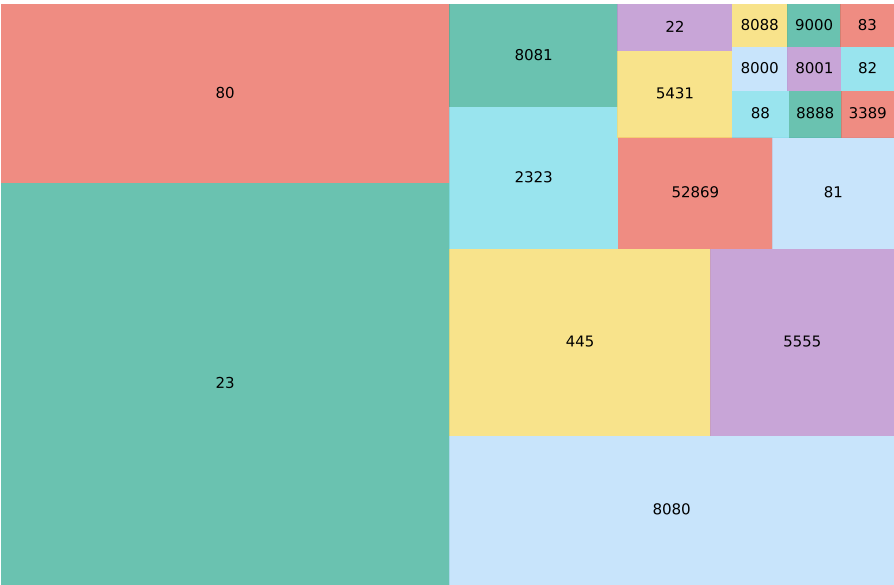
(b) IBR Data

Window	IBR1	IBR2	IBR3	IBR4	IBR5	IBR6	IBR7
2	123 133	123 411	130 665	130 240	129 718	65 456	340 319
3	52 463	53 718	58 928	58 745	58 570	32 478	148 940
4	31 983	33 047	37 384	37 331	37 191	21 062	92 156
5	23 213	23 944	27 717	27 869	27 533	15 664	64 506
6	17 831	18 123	20 805	20 933	20 758	12 523	47 040
7	14 724	15 266	17 296	17 472	17 235	10 539	38 974
8	12 012	12 478	14 207	14 361	14 120	8 571	31 115
9	10 172	10 494	12 051	12 117	11 933	7 417	25 774
10	8 665	8 919	10 371	10 417	10 223	6 329	21 882
11	7 542	7 788	9 037	9 115	8 974	5 587	19 130
12	6 535	6 718	7 827	7 914	7 811	4 818	16 949

Legend: AV: AlienVault; AS: Autoshun; BD: Binary Defense Banlist; BB: Bruteforce Blocker; CI: CINS

D.4 Highly Persistent IP Addresses

This section contains the treemap and values from the analysis results of Section 5.5.3. The five telescope datasets are illustrated in Figures D.8 to D.12. The two OSINT feeds (AlienVault and CINS) are illustrated in Figures D.13 and D.14. The five IBR real-world datasets are illustrated in Figures D.15 to D.19 and the two synthetic datasets are illustrated in Figures D.20 and D.21.

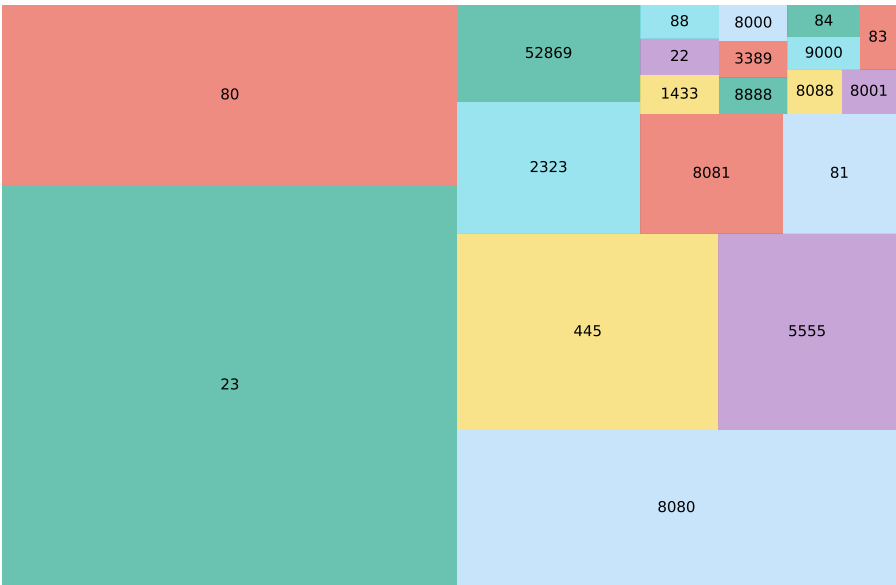


(a) Treemap

Port	Count
23/tcp	464 994
80/tcp	206 662
8080/tcp	170 692
445/tcp	125 928
5555/tcp	88 367
2323/tcp	61 539
8081/tcp	44 708
52869/tcp	44 582
81/tcp	34 875
5431/tcp	25 382
22/tcp	13 701
88/tcp	6 821
8888/tcp	6 343
3389/tcp	6 277
8000/tcp	6 247
8088/tcp	6 095
8001/tcp	6 063
82/tcp	5 890
9000/tcp	5 880
83/tcp	5 856

(b) Values

Figure D.8: Top 20 TCP Ports For RUSCOPE1

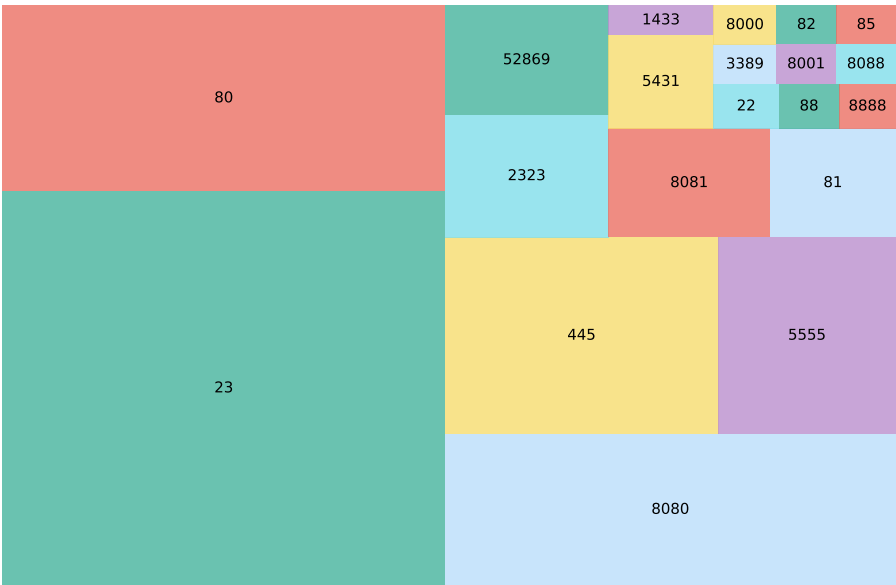


(a) Treemap

Port	Count
23/tcp	466 687
80/tcp	209 299
8080/tcp	174 642
445/tcp	131 828
5555/tcp	89 503
2323/tcp	61 662
52869/tcp	45 426
8081/tcp	43 958
81/tcp	34 679
UDP/18183	9 814
1433/tcp	7 800
22/tcp	7 425
88/tcp	6 690
8888/tcp	6 378
3389/tcp	6 371
8000/tcp	6 294
8088/tcp	6 169
8001/tcp	6 085
9000/tcp	5 963
84/tcp	5 934

(b) Values

Figure D.9: Top 20 TCP Ports For RUSCOPE2

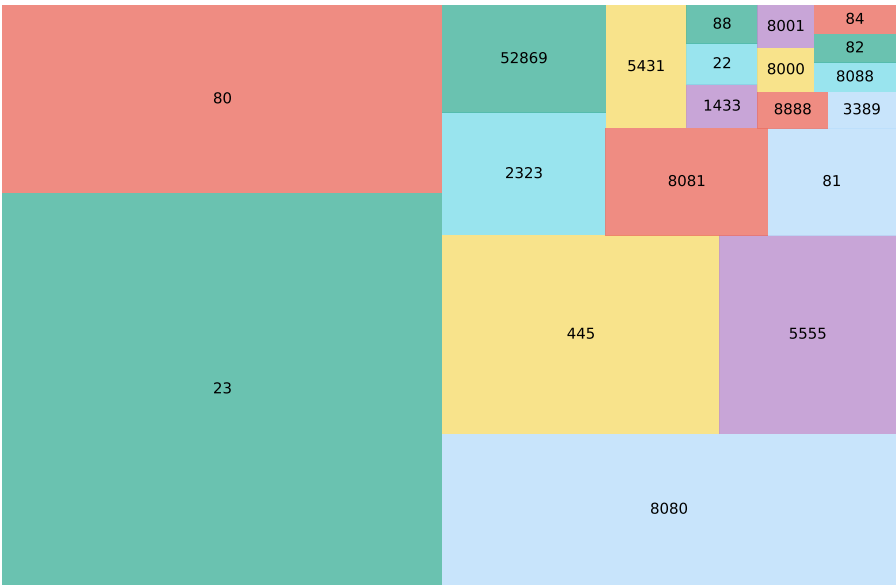


(a) Treemap

Port	Count
23/tcp	444 894
80/tcp	209 246
8080/tcp	174 378
445/tcp	136 904
5555/tcp	89 006
2323/tcp	51 100
52869/tcp	45 549
8081/tcp	44 834
81/tcp	34 888
5431/tcp	25 104
1433/tcp	7 934
22/tcp	7 395
88/tcp	6 847
8888/tcp	6 411
3389/tcp	6 314
8000/tcp	6 251
8001/tcp	6 066
8088/tcp	6 064
82/tcp	5 973
85/tcp	5 868

(b) Values

Figure D.10: Top 20 TCP Ports For RUSCOPE3

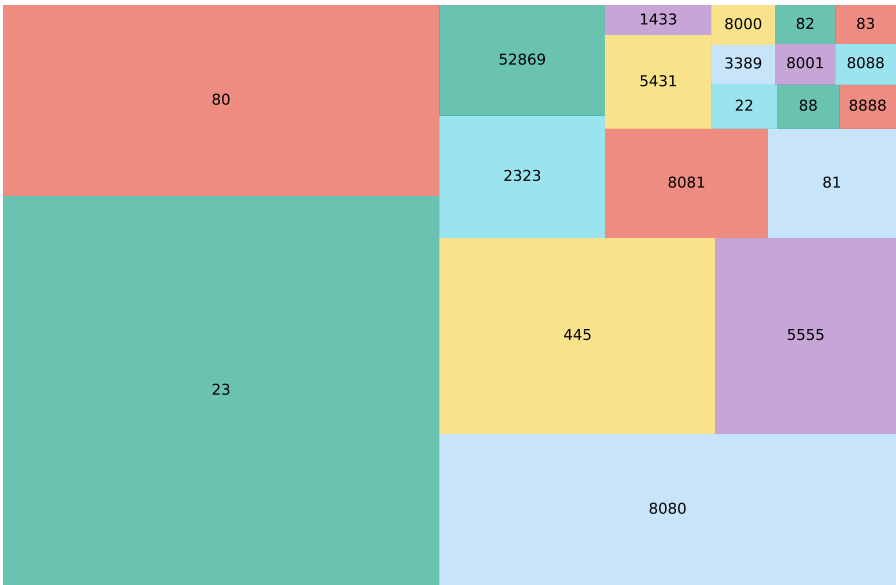


(a) Treemap

Port	Count
23/tcp	439 085
80/tcp	209 598
8080/tcp	175 294
445/tcp	139 779
5555/tcp	88 769
2323/tcp	51 101
52869/tcp	44 480
8081/tcp	44 199
81/tcp	34 863
5431/tcp	25 311
1433/tcp	7 900
22/tcp	7 417
88/tcp	6 877
8888/tcp	6 514
3389/tcp	6 300
8000/tcp	6 279
8001/tcp	6 134
8088/tcp	6 091
82/tcp	5 961
84/tcp	5 909

(b) Values

Figure D.11: Top 20 TCP Ports For RUSCOPE4

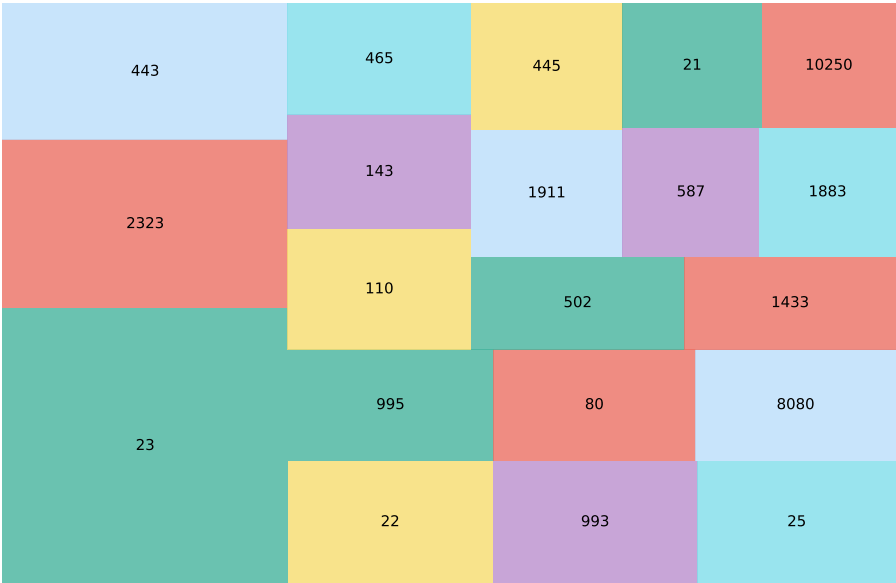


(a) Treemap

Port	Count
23/tcp	429 388
80/tcp	209 414
8080/tcp	174 776
445/tcp	136 249
5555/tcp	88 835
2323/tcp	51 186
52869/tcp	45 972
8081/tcp	44 750
81/tcp	35 151
5431/tcp	25 124
1433/tcp	7 976
22/tcp	7 392
88/tcp	7 085
8888/tcp	6 355
3389/tcp	6 339
8000/tcp	6 246
8001/tcp	6 131
8088/tcp	6 095
82/tcp	5 872
83/tcp	5 851

(b) Values

Figure D.12: Top 20 TCP Ports For RUSCOPE5

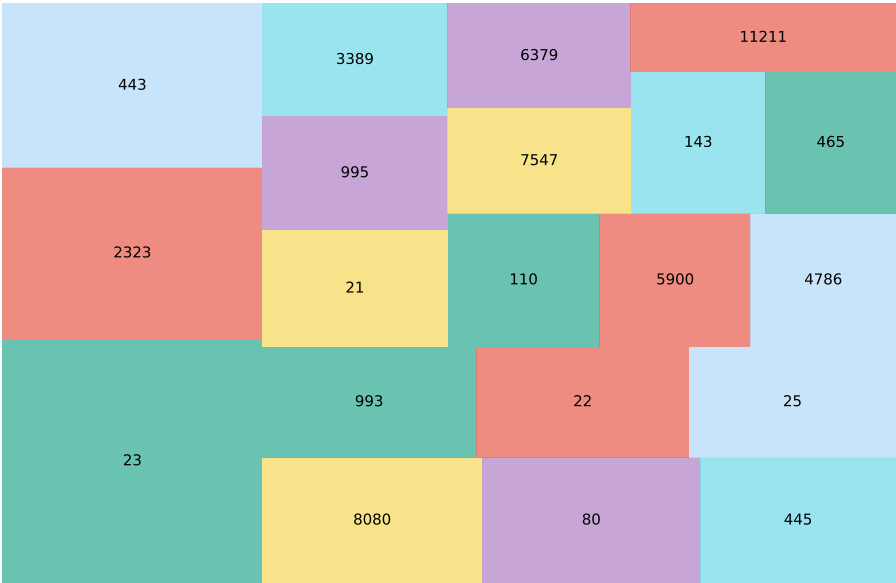


(a) Treemap

Port	Count
23/tcp	1218
2323/tcp	745
443/tcp	603
22 /tcp	391
993/tcp	391
25/tcp	378
995/tcp	355
80/tcp	349
8080/tcp	346
110/tcp	342
143/tcp	326
465/tcp	316
502/tcp	307
1433/tcp	304
1911/tcp	298
445/tcp	297
587/tcp	274
1883/tcp	273
21/tcp	269
10250/tcp	258

(b) Values

Figure D.13: Top 20 TCP Ports For AlienVault

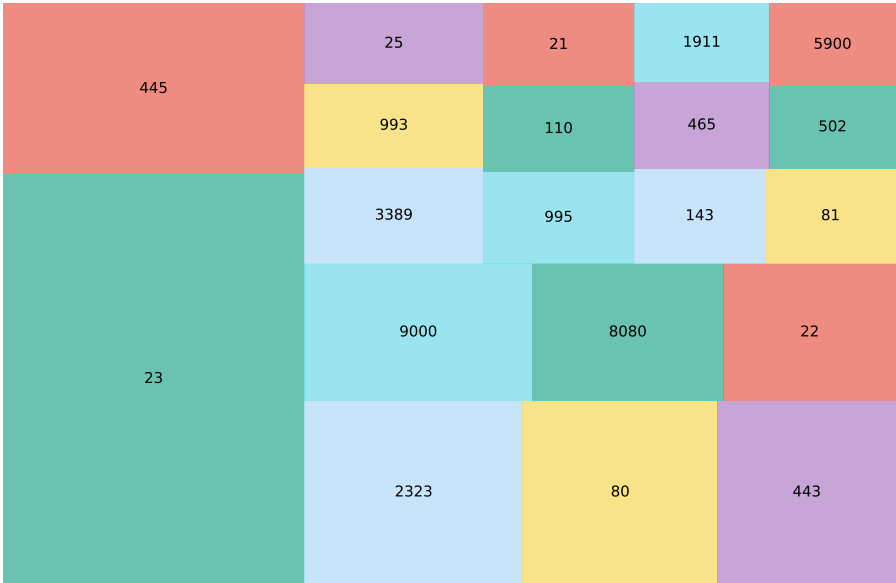


(a) Treemap

Port	Count
23/tcp	913
2323/tcp	638
443/tcp	613
8080/tcp	400
80/tcp	396
445/tcp	355
993/tcp	340
22/tcp	338
25/tcp	328
21/tcp	312
995/tcp	304
3389/tcp	301
110/tcp	291
5900/tcp	290
4786/tcp	278
7547/tcp	277
6379/tcp	276
143/tcp	275
465/tcp	265
11211/tcp	261

(b) Values

Figure D.14: Top 20 TCP Ports: CINS

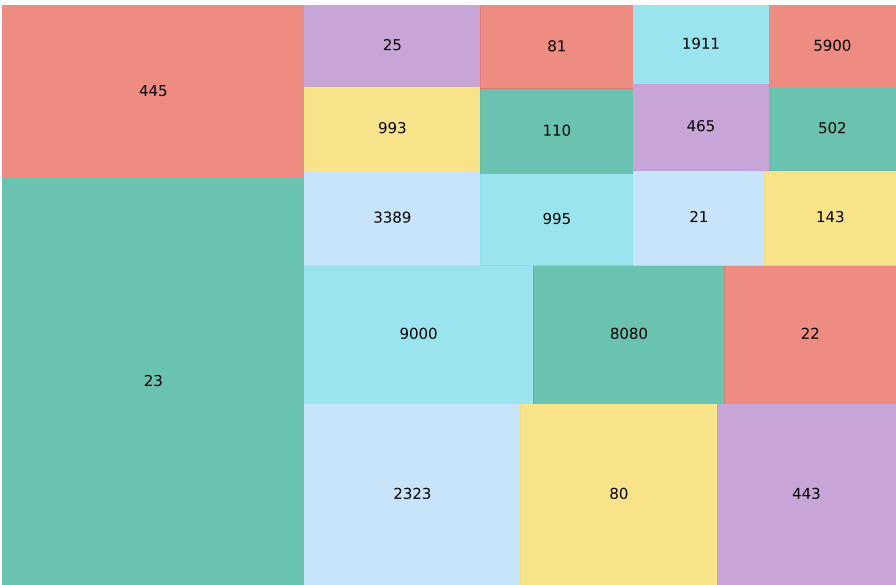


(a) Treemap

Port	Count
23/tcp	3791
445/tcp	1581
2323/tcp	1220
80/tcp	1091
443/tcp	996
9000/tcp	960
8080/tcp	809
22/tcp	727
3389/tcp	525
993/tcp	459
25/tcp	439
995/tcp	425
110/tcp	397
21/tcp	383
143/tcp	381
81/tcp	380
465/tcp	360
1911/tcp	325
502/tcp	321
5900/tcp	319

(b) Values

Figure D.15: Top 20 TCP Ports: IBR1

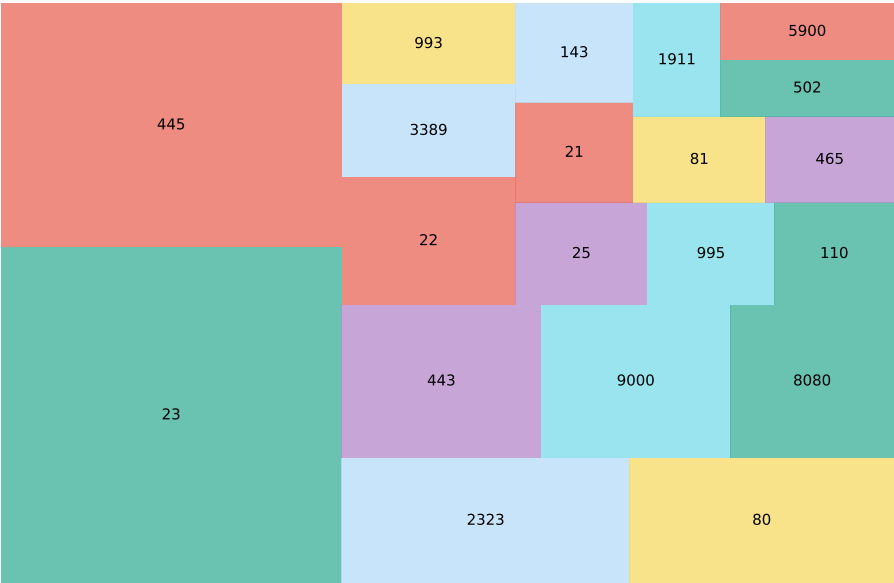


(a) Treemap

Port	Count
23/tcp	3752
445/tcp	1590
2323/tcp	1204
80/tcp	1090
443/tcp	993
9000/tcp	970
8080/tcp	810
22/tcp	724
3389/tcp	508
993/tcp	459
25/tcp	438
995/tcp	425
110/tcp	397
81/tcp	388
21/tcp	383
143/tcp	381
465/tcp	360
1911/tcp	325
502/tcp	321
5900/tcp	319

(b) Values

Figure D.16: Top 20 TCP Ports: IBR2

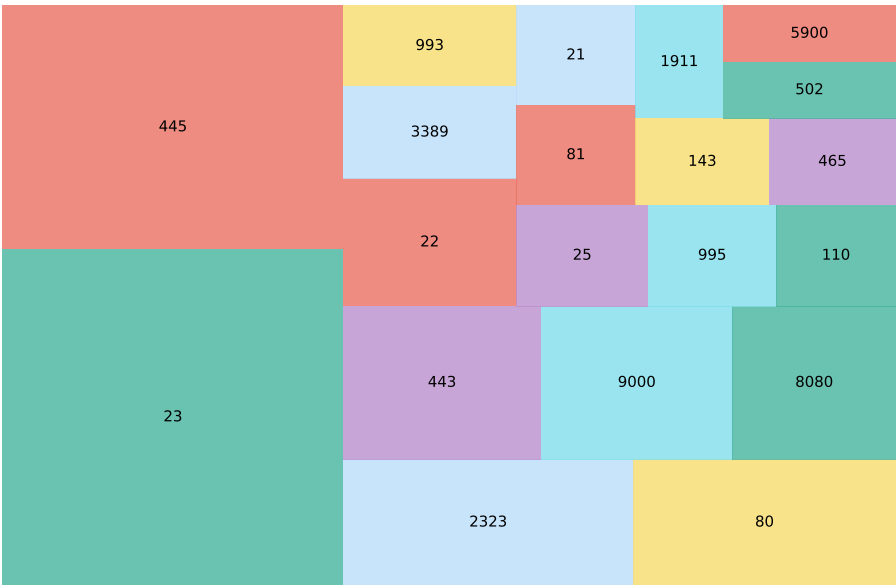


(a) Treemap

Port	Count
23/tcp	3747
445/tcp	2711
2323/tcp	1183
80/tcp	1085
443/tcp	995
9000/tcp	945
8080/tcp	815
22/tcp	725
3389/tcp	527
993/tcp	459
25/tcp	439
995/tcp	425
110/tcp	397
21/tcp	384
143/tcp	380
81/tcp	373
465/tcp	360
1911/tcp	325
502/tcp	321
5900/tcp	319

(b) Values

Figure D.17: Top 20 TCP Ports: IBR3

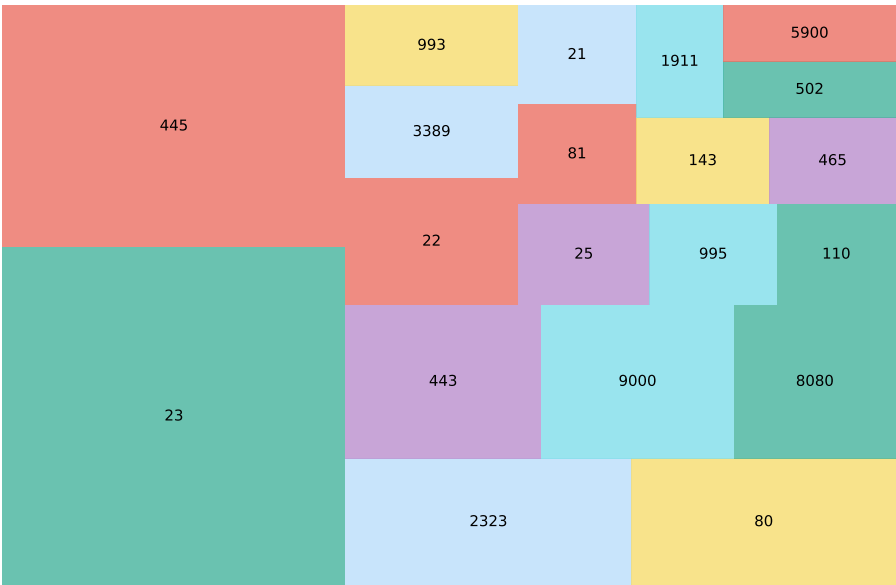


(a) Treemap

Port	Count
23/tcp	3763
445/tcp	2721
2323/tcp	1202
80/tcp	1088
443/tcp	994
9000/tcp	964
8080/tcp	820
22/tcp	725
3389/tcp	524
993/tcp	459
25/tcp	438
995/tcp	425
110/tcp	397
81/tcp	391
21/tcp	390
143/tcp	380
465/tcp	360
1911/tcp	325
502/tcp	321
5900/tcp	319

(b) Values

Figure D.18: Top 20 TCP Ports: IBR4

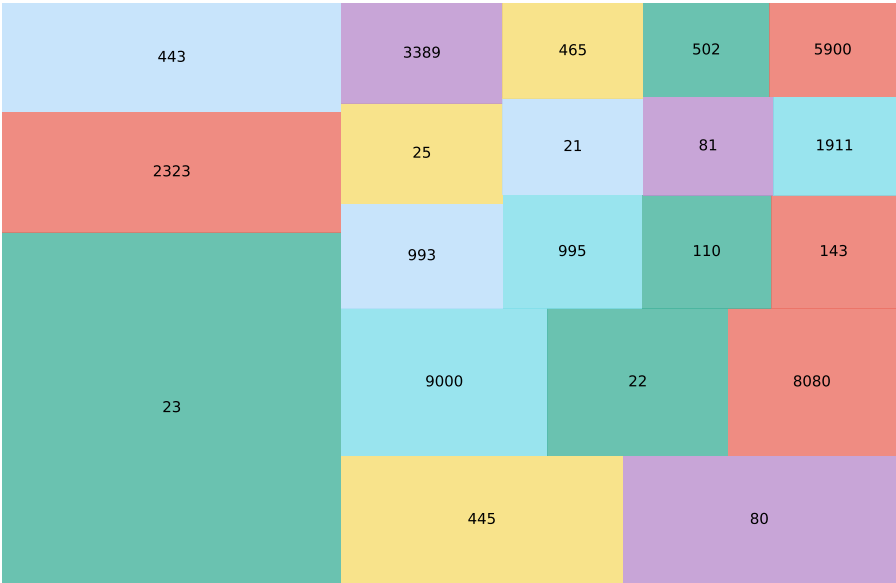


(a) Treemap

Port	Count
23/tcp	3833
445/tcp	2739
2323/tcp	1204
80/tcp	1110
443/tcp	995
9000/tcp	979
8080/tcp	822
22/tcp	725
3389/tcp	528
993/tcp	459
25/tcp	439
995/tcp	425
110/tcp	397
81/tcp	389
21/tcp	386
143/tcp	380
465/tcp	360
1911/tcp	325
502/tcp	321
5900/tcp	320

(b) Values

Figure D.19: Top 20 TCP Ports: IBR5

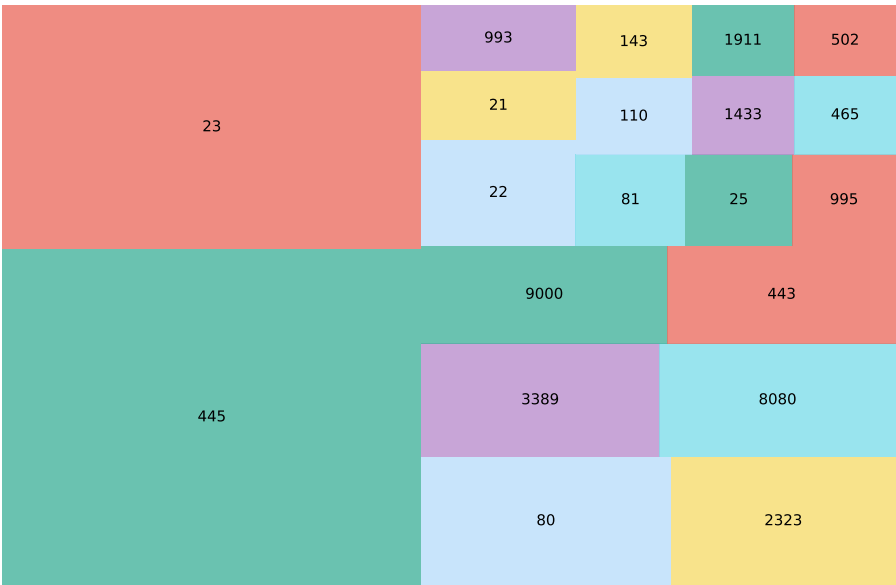


(a) Treemap

Port	Count
23/tcp	3209
2323/tcp	1102
443/tcp	991
445/tcp	972
80/tcp	941
9000/tcp	824
22/tcp	719
8080/tcp	669
993/tcp	459
25/tcp	438
3389/tcp	438
995/tcp	425
110/tcp	397
143/tcp	380
21/tcp	366
465/tcp	360
81/tcp	346
1911/tcp	325
502/tcp	321
5900/tcp	319

(b) Values

Figure D.20: Top 20 TCP Ports: IBR6



(a) Treemap

Port	Count
445/tcp	6348
23/tcp	4591
80/tcp	1452
2323/tcp	1307
3389/tcp	1215
8080/tcp	1205
9000/tcp	1089
443/tcp	1008
22/tcp	737
21/tcp	481
993/tcp	459
81/tcp	449
25/tcp	441
995/tcp	425
110/tcp	397
143/tcp	381
1433/tcp	365
465/tcp	360
1911/tcp	325
502/tcp	321

(b) Values

Figure D.21: Top 20 TCP Ports: IBR7

D.5 Additional Results

This section contains additional results obtain from various other tests performed on the data. The results in this section are not discussed in the main document but are available for the readers benefit. Table D.4 lists the top 20 unique TCP and UDP ports targeted by IP address on AlienVault. Table D.5 lists the top 20 unique TCP and UDP ports targeted by IP address on IBR6. Table D.6 lists a count of the top 20 TCP and UDP ports by IP addresses sending packets on a monthly basis. Table D.7 lists a count of the top 20 TCP and UDP ports by IP addresses sending packets on a monthly basis.

Table D.4: AlienVault Top 20 IP Addresses by Unique Ports Targeted

(a) Top 20 TCP				(b) Top 20 UDP			
Rank	Source IP	*TCP	UDP	Rank	Source IP	TCP	*UDP
1	5.188.10.241	1669	0	1	71.6.165.200	289	3122
2	60.191.38.77	1561	20	2	66.240.192.138	291	2994
3	5.188.10.103	940	0	3	66.240.236.119	289	2076
4	125.64.94.201	343	88	4	50.200.136.108	0	857
5	66.240.192.138	291	2994	5	212.178.154.174	0	787
6	71.6.165.200	289	3122	6	71.6.167.142	267	591
7	66.240.236.119	289	2076	7	203.74.132.232	2	517
8	125.212.217.214	268	52	8	71.6.146.130	267	478
9	71.6.135.131	268	52	9	59.120.11.180	2	101
1	71.6.146.186	268	52	1	125.64.94.201	343	88
11	89.248.167.131	268	52	11	212.252.132.121	3	55
12	94.102.49.190	268	52	12	125.212.217.214	268	52
13	71.6.167.142	267	591	13	71.6.135.131	268	52
14	71.6.146.130	267	478	14	71.6.146.186	268	52
15	125.212.217.215	267	52	15	89.248.167.131	268	52
16	198.20.70.114	267	52	16	94.102.49.190	268	52
17	71.6.146.185	267	52	17	125.212.217.215	267	52
18	80.82.77.139	267	52	18	198.20.70.114	267	52
19	80.82.77.33	267	52	19	71.6.146.185	267	52
20	82.221.105.6	267	52	20	80.82.77.139	267	52

* identifies the sorted column

Table D.5: IBR6 Top 20 IP Addresses by Unique Ports Targeted

(a) Top 20 TCP				(b) Top 20 UDP			
Rank	Source IP	*TCP	UDP	Rank	Source IP	TCP	*UDP
1	5.188.10.241	1669	0	1	185.163.109.66	249	5118
2	125.64.94.201	343	88	2	71.6.165.200	289	3122
3	66.240.192.138	291	2994	3	66.240.192.138	291	2994
4	71.6.165.200	289	3122	4	66.240.236.119	289	2076
5	66.240.236.119	289	2076	5	50.200.136.108	0	857
6	125.212.217.214	268	52	6	213.34.167.254	0	789
7	71.6.135.131	268	52	7	212.178.154.174	0	787
8	71.6.146.186	268	52	8	213.34.77.254	0	787
9	89.248.167.131	268	52	9	212.178.135.62	0	767
1	94.102.49.190	268	52	1	195.35.245.30	0	762
11	71.6.167.142	267	591	11	71.6.167.142	267	591
12	71.6.146.130	267	478	12	71.6.146.130	267	478
13	125.212.217.215	267	52	13	59.120.11.180	2	101
14	71.6.146.185	267	52	14	125.64.94.201	343	88
15	80.82.77.139	267	52	15	125.212.217.214	268	52
16	80.82.77.33	267	52	16	71.6.135.131	268	52
17	82.221.105.6	267	52	17	71.6.146.186	268	52
18	82.221.105.7	267	52	18	89.248.167.131	268	52
19	89.248.172.16	267	52	19	94.102.49.190	268	52
20	93.174.95.106	267	52	20	125.212.217.215	267	52

* identifies the sorted column

Table D.7: IBR6 Monthly Top 20 Port Request Count

(a) TCP

Rank	Port	09/17	10/17	11/17	12/17	01/18	02/18	03/18	04/18	05/18	06/18	07/18	08/18
1	23	3292	3303	3322	3297	3308	3274	3318	3286	3319	3286	3317	3303
2	2323	1554	1538	1547	1488	1495	1468	1491	1461	1440	1436	1441	1365
3	80	993	1014	1016	1022	1022	777	809	1048	1084	1232	1190	1054
4	443	978	976	997	1005	996	992	998	991	993	992	1000	998
5	445	938	900	932	929	1212	954	975	938	987	944	952	982
6	9000	871	863	872	866	873	863	863	862	865	863	860	858
7	22	449	198	442	431	730	470	570	685	710	671	712	745
8	25	509	465	542	490	546	506	512	520	530	486	505	511
9	993	453	466	497	486	494	478	477	474	488	497	492	490
1	3389	414	412	420	427	427	413	413	408	400	441	430	449
11	995	372	360	382	416	687	381	381	382	400	384	382	429
12	8080	152	146	168	152	183	142	421	382	703	776	821	769
13	81	288	308	318	326	334	339	434	738	368	430	529	368
14	110	380	375	385	404	434	382	417	404	389	402	383	400
15	143	362	389	386	404	430	367	383	417	388	402	416	384
16	21	183	402	410	371	411	366	410	397	381	416	391	378
17	465	360	328	384	359	402	383	381	348	387	349	383	382
18	502	604	334	334	358	353	307	340	357	341	342	341	324
19	1883	515	291	289	316	315	315	315	316	317	315	316	317
20	7547	327	331	327	314	344	294	332	348	326	313	309	293

(b) UDP

Rank	Port	09/17	10/17	11/17	12/17	01/18	02/18	03/18	04/18	05/18	06/18	07/18	08/18
1	53	388	449	433	474	474	457	476	429	444	443	459	474
2	123	380	377	359	375	376	375	377	379	378	377	377	376
3	47808	355	339	338	396	364	347	379	364	363	409	392	376
4	19	306	306	305	332	332	332	362	332	332	332	332	90
5	161	293	291	290	290	291	289	290	290	289	289	287	288
6	1900	167	199	167	193	209	180	186	203	157	171	187	171
7	137	92	91	89	119	116	115	116	115	124	115	116	116
8	1434	77	77	76	103	102	102	102	102	102	102	101	101
9	5353	76	76	75	102	101	101	101	101	101	101	102	100
1	389	91	63	63	91	88	88	89	88	89	104	89	88
11	623	63	63	62	89	88	88	91	88	88	88	88	88
12	5351	56	56	56	83	83	83	83	83	83	83	82	82
13	5060	63	63	62	89	88	54	33	89	88	88	88	87
14	520	49	49	48	75	74	74	74	74	74	74	74	75
15	17	45	44	45	71	71	71	86	71	71	71	71	71
16	111	46	44	45	71	71	71	71	71	71	71	72	73
17	17185	42	42	42	69	69	69	69	69	69	69	68	68
18	11211	0	0	0	0	0	34	107	36	90	90	91	90
19	500	37	50	51	50	49	34	34	34	43	34	35	43
20	69	36	36	35	35	34	34	34	33	33	49	35	35