

Constructing a Network Science Toolchain for Analyzing Network Traffic

*Submitted in partial fulfillment of the requirements for
the degree of
Masters in Information Security
in
Information Networking Institute*

Adam Y. Tse

M.S., Information Security, Carnegie Mellon University
B.S., Computer Science, Arizona State University

Carnegie Mellon University
Pittsburgh, PA

May 2018

© Adam Y. Tse, 2018

All Rights Reserved

Acknowledgements

In order to complete this large, ambitious work, I required the support of many people. I would like to take the time to thank these special individuals.

First, I would like to thank my excellent supervisors Kathleen Carley and Tim Shimeall. Their expansive knowledge in their fields, encouragement, guidance, and support were crucial in completing the research of this thesis. The scheduled weekly meetings and discussions were instrumental in my growth as a student and exponentially improved my knowledge in the field.

This thesis would not have been possible without the company and the individuals within it who allowed me to study and write about their data and contributed to discussions about the project. Though I am not allowed to disclose their identities, I wanted to take the time to thank them for their support throughout the project. I appreciate the time these individuals allocated to me from their busy schedules. Without the discussions with these individuals, the experiments in the paper would not have been conducted as thoroughly.

I would like to thank CyberCorps for financing my graduate degree under the Scholarship for Service program. Without the financial assistance that they provided, I would not have been able to complete this thesis or my graduate education.

Moreover, I am indebted to the rest of the colleagues at CASOS for providing me with feedback during our group meetings. I would like to thank Michael Kowalchuck for writing scripts to automate processes conducted on ORA processes. Without his support, many of the experiments would have been painful, manual processes. Additionally, I would like to thank Ian Cruickshank and Gian Maria Campedelli for discussing many different directions to take the work. Their strong backgrounds in telecommunication networks and statistics improved the quality of my work significantly. Lastly, I would like to thank Sumeet Kumar for reviewing earlier drafts of the paper, Geoff Dobson for providing the foundation of the work that this thesis expands upon, Sienna Watkins for making these individual and group meetings with Dr. Carley happen, and Rick Carley for continuously providing me updated versions of the ORA software.

I am grateful for my friends who attended my defense. Thank you, Emory, Ash, Mark, Caitlin, and

track means a lot to me because of its emphasis on self-learning, discovery, and innovation. With this personalized thesis, I have knowledge on fields that none of my other peers in the program have. This is something that very little programs within Carnegie Mellon offer. I would like to thank Dena for creating this program, supporting the thesis option, and encouraging me to pursue learning through discovery and innovation. Additionally, I would like to thank Asia and Jessica for supporting me through the process, defining all my deliverables, and attending my thesis defense.

Lastly, I owe my deepest gratitude to my father Greg, my mother Rose, my sister Ariana, and my brother Arif for continuously encouraging me, giving me the courage to grow, providing me company during breaks between work, and allowing me to be a part of the best family a man can ask for.

The completion of this thesis project was self-funded in its entirety. The company only provided data and did not financially sponsor the project in anyway.

Abstract

For this thesis, a toolchain was designed that aimed to process network traffic to identify host and event behavior. Network traffic is difficult for network administrators to analyze because both the area of responsibility and distribution of external actors are very large when protecting an enterprise network. Having a process that converts streaming network data into actionable intelligence greatly improves the operational capability of network administrators.

The process consisted of three phases: Netflow Collection, Network Analysis, and Actionable Classification which were validated using a series of experiments. The following experiments were performed: a comparison between behavior of normal weeks and a flash crowd incident, a comparison of behavior among functional groups within the corporation, an analysis of hosts reported for abusive behavior, and a classification method for identifying hosts by behavior. The toolchain revolved around using network science methods to gather, process, and measure data. Even though network science is typically used to analyze social network data, the similarities in size and structure of social network data and netflow data make it viable for similar analysis.

Contents

Acknowledgements	iii
Abstract	v
Contents	vi
List of Tables	x
List of Figures	xii
1 Introduction	1
1.1 Network Events	1
1.2 Data Collection	2
1.3 Threat Detection	3
1.3.1 Signature-Based System	3
1.3.2 Volume-Based Anomaly Detection	3
1.3.3 Feature-Based Anomaly Detection	4
1.4 Current Host and Network Behavior Characterization Methods	4
1.5 Network Science	4
1.6 The Toolchain	5
2 Literature Review	7
2.1 Graph Methods for Feature-Based Anomaly Detection	7
2.2 Alternate Feature-based Anomaly Detection Methods	8
2.2.1 Entropy Based Methods	8
2.2.2 Machine Learning Based Methods	10
3 Methods	12

3.1	Tool Chain Process	12
3.1.1	Netflow Collection	12
3.1.2	Network Analysis	13
3.1.3	Actionable Classification	14
3.2	Netflow Collection	14
3.2.1	Netflow	14
3.2.2	YAF Yet Another Flowmeter	14
3.2.3	SiLK System for Internet-Level Knowledge	15
3.3	Network Analysis	17
3.3.1	Network Science Measurements	17
3.3.2	ORA Social Network Analyzer	20
	Importing Data	21
	Generating Chart Measures	23
	Reducing Networks	23
3.3.3	AbuseIPDB	24
3.3.4	Network Size Reduction	24
	Top Change in Total-Degree Host Selection	25
	Top Increase in Sending Decrease in Receiving Host Selection	25
	Functional Sub Groups	25
3.3.5	Two-Sample Kolmogorov Smirnov Test	26
3.3.6	Correlation Matrix Generation	26
3.4	Actionable Classification	27
3.4.1	Dynamic Time Warping K-nearest Neighbor Classification	27
3.5	Challenges	28
4	Results	29
4.1	Netflow Data Structure: Network Analysis	29
4.1.1	Previous Network Structure	29
	Overview	29
	Results	30
4.1.2	Flash Crowd Incident Description	31
	Overview	31
	Raw Data Analysis Results	32

Top 20% Total Degree Centrality Analysis Results	38
Increased Egress and Decreased Ingress Rate Analysis Results	42
4.1.3 Effects of the Flash Crowd on Functional Groups	46
Overview	46
Results	46
4.1.4 Network Science Measurement Evaluation	47
4.1.5 AbuseIPDB Distribution	49
Overview	49
Results	49
4.1.6 AbuseIPDB Ego networks	53
Overview	53
Results	54
4.2 Actionable Classification	61
4.2.1 Classifying Hosts By Division	61
Overview	61
Results	62
5 Discussion	65
5.1 Netflow Collection	65
5.1.1 SiLK Usage	65
5.2 Network Analysis	66
5.2.1 Protocol Usage	66
Overview	66
Implications for Toolchain	66
5.2.2 Flash Crowd Analysis	67
Overview	67
Implications for Toolchain	68
5.2.3 Data Reduction	69
Overview	69
Implications for Toolchain	69
5.2.4 Functional Group Analysis	70
Overview	70
Implications for Toolchain	70

5.2.5	AbuseIPDB Analysis	70
	Overview	70
	Implications for Toolchain	71
5.3	Actionable Classification	71
5.3.1	Classifying Hosts by Division	71
	Overview	71
	Implications for Toolchain	72
6	Conclusions	73
6.1	Limitations	74
6.2	Future Work	75
	Bibliography	77

List of Tables

1.1	Comparisons between threat detection techniques	3
4.1	Distribution of Protocol Data	31
4.2	Two-Sample Kolmogorov-Smirnov Test of TCP comparison of Normal Week and Incident Week and both normal weeks	34
4.3	Two-Sample Kolmogorov-Smirnov Test of UDP comparisons of the Incident Week and a Normal Week and the Normal Weeks	34
4.4	Two-Sample Kolmogorov-Smirnov Test of ICMP comparisons of the Incident Week and a Normal Week and the Normal Weeks	36
4.5	Two-Sample Kolmogorov-Smirnov Test of ESP comparisons of the Incident Week and a Normal Week and the Normal Weeks	36
4.6	Two-Sample Kolmogorov-Smirnov Test of GRE comparisons of the Incident Week and a Normal Week and the Normal Weeks	38
4.7	Two-Sample Kolmogorov-Smirnov Test of the Top 20% Total-degree comparisons of TCP connections between the Incident Week and a Normal Week and between the Normal Weeks . . .	39
4.8	Two-Sample Kolmogorov-Smirnov Test of the Top 20% Total-degree comparisons of UDP connections between the Incident Week and a Normal Week and between the Normal Weeks . . .	41
4.9	Two-Sample Kolmogorov-Smirnov Test of the Top 20% Increased Egress and Decreased Ingress Hosts comparing TCP connections between the Incident Week and a Normal Week and between the Normal Weeks	44
4.10	Two-Sample Kolmogorov-Smirnov Test of the Top 20% Increased Egress and Decreased Ingress Hosts comparing UDP connections between the Incident Week and a Normal Week and between the Normal Weeks	44
4.11	Two-Sample Kolmogorov-Smirnov Test of server TCP connections between the Incident Week and a Normal Week and between Normal Weeks	47

4.12 Summary of the results of the two sample KS-test for all experiments	48
4.13 Daily two sample KS-test for normal weeks	48
4.14 Table of DTW/KNN results for all network science measurements individually calculated . . .	64

List of Figures

1.1	Social Network Visualization of TCP Traffic of Servers within Enterprise Network	5
3.1	Diagram of the toolchain process and the components that make up each phase	13
3.2	SILK Collector Setup	16
3.3	Visualization of ICMP Traffic of Enterprise Network during 4-hour Time Period	18
3.4	ORA Visualization of a 2 Radius Ego Network on a Company Host	21
3.5	Configuration for importing SiLK CSV files into ORA	22
3.6	Configuration for aggregating SiLK CSV files into ORA	22
3.7	Configuration for selecting measurements for chart measures in ORA	23
3.8	Report of a malicious host on abuseipdb.com	24
3.9	Graph depicting an example of the Kolmogorov-Smirnov test[11]	26
3.10	Graph visualizing distance measurements calculated in the dynamic time warping algorithm[51]	27
3.11	Visual of K-nearest neighbor algorithm where blue and red represent different classes, green represents a test point, and both circles represent different values of k. The inner radius would classify green as red and the outer radius would classify green as blue [3]	28
4.1	Time series of density of TCP connections during incident week; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, grey lines represent first level standard deviations of normal weeks	33
4.2	Time series of density of TCP connections during normal week	33
4.3	Time series of link count of ICMP connections during incident week; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, grey lines represent first level standard deviations of normal weeks	35
4.4	Time series of total-degree centrality of ESP connections during incident week; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, standard deviations were not included because the range was too large	37

4.5	Time series of total-degree centrality of ESP connections during a normal week	37
4.6	Time series of clustering coefficient of TCP connections during incident week after removing bottom 80% total-degree difference; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, grey lines represent first level standard deviations of normal weeks	39
4.7	Time series of clustering coefficient of TCP connections during normal week after removing bottom 80% total-degree difference	40
4.8	Time series of fragmentation of UDP connections during incident week after removing bottom 80% total-degree difference; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, grey lines represent first level standard deviations of normal weeks	41
4.9	Time series of fragmentation of UDP connections during normal weeks after removing bottom 80% total-degree difference	42
4.10	Time series of density of TCP connections during event week after selecting the top 20% increased sender and decreased receivers; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, standard deviations were not included because the values were too low	43
4.11	Time series of density of TCP connections during normal weeks after selecting the top 20% increased sender and decreased receivers	43
4.12	Time series of fragmentation of UDP connections during event week after selecting the top 20% increased sender and decreased receivers; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, standard deviations were not included because the range was too large	45
4.13	Time series of fragmentation of UDP connections during normal weeks after selecting the top 20% increased sender and decreased receivers	45
4.14	Time series of edge sum of server TCP connections within the company	47
4.15	Distribution of Countries for TCP IPs	50
4.16	Distribution of Countries for UDP IPs	51
4.17	Distribution of Countries for TCP Top 10% difference in total-degree centrality IPs	51
4.18	Distribution of Countries for UDP Top 10% difference in total-degree centrality IPs	52
4.19	Distribution of Countries for UDP Top 10% increased sending and decreased receiving rate	52
4.20	Distribution of Organizations over TCP	53

4.21	Time series of edge sum for 5 hosts in the Top 10% Total-degree centrality that are a part of Fortune 500 companies	55
4.22	Time series of edge sum for 5 hosts labelled abusive in the Top 10% Total-degree centrality . . .	56
4.23	Time series of edge count for 5 hosts labelled non-abusive in the 50% to 60% Total-degree centrality range	56
4.24	Time series of edge count for 5 hosts labelled abusive in the 50% to 60% Total-degree centrality range	57
4.25	Time series of node count for 5 hosts labelled non-abusive in the bottom 10% Total-degree centrality	58
4.26	Time series of node count for 5 hosts labelled abusive in the bottom 10% Total-degree centrality	58
4.27	Time series of standard deviation of edge count for company, non-abusive, and abusive categories in the top 10% Total-degree centrality	59
4.28	Time series of standard deviation of edge count for company, non-abusive, and abusive categories in the 50% to 60% Total-degree centrality range	60
4.29	Time series of standard deviation of edge count for company, non-abusive, and abusive categories in the bottom 10% Total-degree centrality	60
4.30	Time series plots of link sum for 6 hosts identified by division	62
4.31	Classification Report Heatmap of DTW/KNN calculated from link count	63
4.32	Classification Report Heatmap of DTW/KNN calculated from density	63

Chapter 1

Introduction

Network administrators have a large set of duties besides monitoring network activity. These duties typically include installation, management of hardware and software, the diagnosis and repairing of components, and directly working with users to solve problems they are facing. Additionally, the migration to cloud environments and installation of Internet of Things (IOT) devices within companies has led to an increased surfaced area that network administrators need to monitor and maintain. As company networks grow larger and more complex, it becomes even more difficult for staff to allocate resources into monitoring and responding to events without hiring more people into those roles. In the case where the company does not have the funds to gather more hands for the role, security becomes an afterthought. Therefore, it becomes more imperative to introduce techniques that can quickly identify network activity and host behavior without the direct supervision of a network administrator. This paper proposes and designs a toolchain or process that creates actionable intelligence out of telecommunication traffic data using network science.

The following sections will describe the current state of network events important to network administrators, data collection techniques, threat detection techniques, network behavior characterization, and the proposed toolchain using a network science approach.

1.1 Network Events

Network administrators must be aware of the possible threats that can hinder company operations. Lockheed Martin defined the cyber kill chain which defines the standard sequence of steps an attacker follows in offensive cyber operations[43]. Each step of the cyber kill chain can be directly monitored by a network administrator given the proper data and models. In the first phase, reconnaissance, malicious actors can perform port and network scans to find vulnerabilities and system/application information for individual

devices or the whole network topology[43]. Additionally, when this scanning like behavior is done from an internal device within the network, it can indicate an attacker already has access to a device within the enterprise network and is attempting to propagate his access deeper within the network. This phase is common in offensive cyber operations and is part of the weaponization and delivery phase of the cyber kill chain[43].

Alongside reconnaissance activity, network administrators must be aware of events that render systems and services unavailable. These attacks focus on flooding enterprise machines either through external connections or taking advantage of vulnerabilities caused by network protocols. One example for an external attack includes leasing botnets to flood the target with traffic. These attacks can also be caused by legitimate network traffic as seen in flash crowds and alpha flows. Some examples of network protocol attack are the Ping of death, the SYN flood, and the LAND attack. The Ping of death is an attack that impacts systems that do not have a safety check for malformed packets that are too large[37]. Ping of deaths are typically caused by sending large packets that when broken and reformed cause a buffer overflow within the system processing the traffic thereby disabling the system. SYN floods are when large amount of TCP sessions are half-created causing an allocation of resources that can ultimately hinder legitimate users from connecting to the network[40]. LAND attacks are an attack where the source and destination IP of a packet are the same causing a feedback loop of flooding the device with replies[42]. These are all examples of activities that network administrators should monitor for and they all have a distinct set of features and behaviors that can be used when tracking network traffic.

1.2 Data Collection

There are 2 main methods for gathering data in anomaly detection. The first method is packet-based inspection, the gathering of raw network traffic in packet form[69]. This can be performed using network tools such as Wireshark¹ and Tcpdump². Gathering data using packet-based inspection has the advantage of providing a comprehensive analysis because of the ability to analyze the payload of each packet. Information on data being exfiltrated, exploitive input, and virus data can be monitored using this approach. However, the main disadvantage is that this approach is not scalable due to the high amount of CPU processing, memory, and storage capacity required to actively monitor an enterprise network.

The other approach is flow-based inspection[62]. Flow-based inspection aggregates a set of packets by a common property defined by the flow protocol. Some of the most supported protocols among routers

¹Wireshark can be downloaded for free at <http://www.wireshark.org/>

²In most UNIX distributions, tcpdump is installed by default, however it can be downloaded for free at <http://www.tcpdump.org/>

Table 1.1: Comparisons between threat detection techniques

Feature	Signature-Based	Volume-Based	Feature-Based
Identifies network level anomalies	X	X	X
Identifies host level anomalies	X		X
Easy feature specifications	X	X	
Does not require packet data		X	X
Does not require data features for detection		X	X
Identifies zero-day threats		X	X
Fine-grained event detection			X

are Netflow[14] and IPFIX[15]. The packets are forwarded to a Flow Collector that is responsible for using the protocol to aggregate packet data into flows. The main advantages of flow-based inspection are that it requires significantly less data for storing and processing and that it minimizes privacy concerns. Though procedures are meant to protect the enterprise from external threats, many of the external hosts interacting within the network are customers to the enterprise. As a result, there are concerns when packet level data of customers is monitored. Flow-based inspection would remove this private data.

1.3 Threat Detection

There are 3 main approaches for threat detection in network traffic. Table 1.1 summarizes the pros and cons of each threat detection system. The following subsections will describe them.

1.3.1 Signature-Based System

Signature-based systems examine packet headers and payload for pre-defined information that indicate network activity of relevance to a network administrator[69]. These methods require companies to know the indicators of an attack beforehand, however this can be ineffective because of the ability of malware to obfuscate itself to avoid previously defined threat signatures. Network administrators predominantly use previously discovered threat signatures and the blacklisting of threatening domains or addresses. However, identifying blacklisted threats is not scalable and does not protect enterprise networks from zero-day intrusions [59][53]. Blacklisting also does not address security or performance issues that come from legitimate users.

1.3.2 Volume-Based Anomaly Detection

Volume-based anomaly detection focuses on establishing a threshold for normal traffic volume over the network[20][60][9]. When the threshold is broken, network administrators are alerted. Though this ap-

proach helps for network events that affect the traffic volume of the whole network such as DDoS and flash crowds, they do not help for targeted events such as worm propagation.

1.3.3 Feature-Based Anomaly Detection

Feature-based anomaly detection focuses on modeling the behavior of network events. Anomaly detection is difficult because of the difficulty in generating accurate models due to noise in network data [24][39]. Moreover, little research has been done on modelling the complex behavior resulting from network events of interest to network administrators [7][48][50]. The work described in the thesis uses feature-based anomaly detection. The Literature Review will describe the work done in this area in more detail.

1.4 Current Host and Network Behavior Characterization Methods

Current models have taken approaches focusing on in-degree and out-degree frequencies between various hosts and ports and size of flows within this graph structure [59][50]. Additionally, behavioral models are difficult to compute because of the noise resulting from the great size of data extracted from networks. In network science, these measurements and many more are considered when analyzing a network or graph structure. Network science provides a new dimension of measurements for quantifying network behavior in the form of structural measurements and grouping level measurements. Additionally, network science provides methods for extracting subsets of the data that can be used for more feature extraction. Rather than focusing on simple node to node interactions, a network science approach examines the interactions, relationships, and inter-connectedness between hosts within a network as a whole. This level of study is ideal because it requires limited metrics and reduces the data when compared to raw-packet data [59][53]. The quantified behavior can still be compared and modelled using traditional statistical approaches through chi-squared, entropy, and other classification techniques.

1.5 Network Science

Network science as a field had a predominant use for marketing and advertisement. However, the simple calculations that are used to measure and describe networks have found other applications in various fields. Network science focuses on quantifying the behavior of interactions between agents within a network. The main literature on network science can be found in literature by Stanley Wasserman and Katherine Faust[71]. Telecommunication networks are very large complex systems that can be made of hundreds of thousands of agents. Network science measurements are efficient to calculate making them an attractive approach to rapidly identify behavior when compared to manual approaches that require

2017-01-29 12:00:00

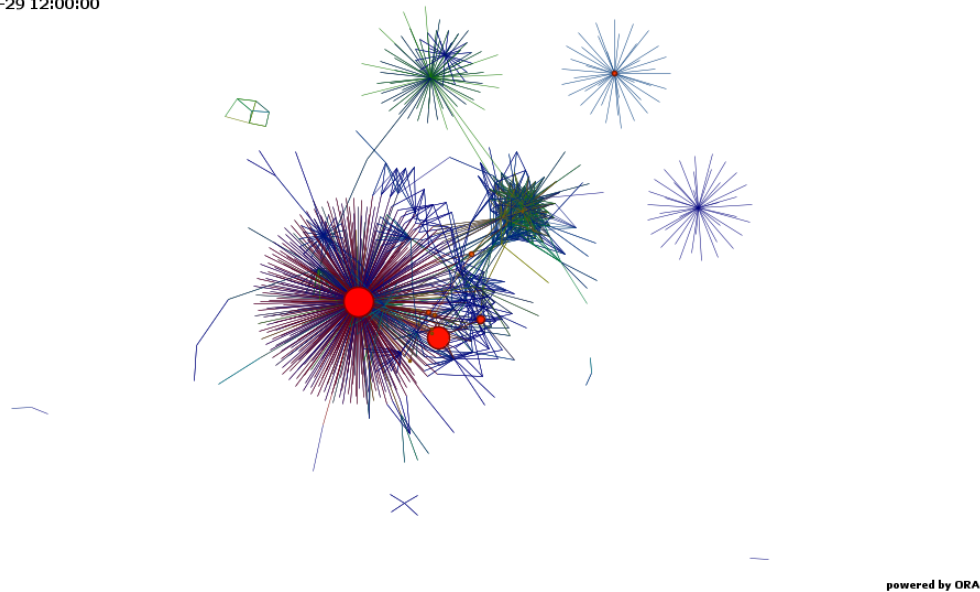


Figure 1.1: Social Network Visualization of TCP Traffic of Servers within Enterprise Network

searching for the cause of problems. Figure 1.1 shows a social network representation of a time range of TCP network server traffic over an enterprise network for a 4 hour time period. The bigger and the more red the node, the more connected the node is. This was calculated using total-degree centrality.

Network science has been applied to cyber-security in cyber-criminal network analysis, insider threat detection, and social attack graphs. The motivations behind cyber-criminal network analysis are to understand the behavior of dark web communities and the digital underground economy to better assess risk. The measurements used to analyze this medium include centrality, degree, topic modeling, and hub measurements [73][41]. The research on insider threat detection represented users/insiders, projects, data access, and data items as a network structure and used distance measurements to model normal behavior for insider threat detection [70]. Lastly, the research on cyber-attack graphs used closeness-centrality, degree centrality, agglomerative hierarchical clustering as a means for identifying groups of hosts involved in cyber-attacks from a graph of malicious traffic [64][21]. The approach described in this thesis focuses on applying network science to network communication within an enterprise.

1.6 The Toolchain

The end goal of the work described in this paper is a toolchain that enables network administrators to make sense of the events going on in their network despite the enormous amount of traffic flowing through it. A toolchain was developed because of the complexity of the procedure necessary to analyze

network traffic data from a network science perspective and the availability of already built tools that support many functions required to complete analysis. In their specific field, many of the techniques used in this analysis are not novel, however they have not been applied in the process described in this thesis. Implementing this process as components within a toolchain within a real enterprise network provides a quick method for validating the design and feasibility of implementing a similar tool and applying it to industry.

Summarizing network traffic into categorizations and models for different network events and host behavior would give network administrations improved cyber-situational awareness within their enterprise environment. Though a lot of work has examined steps of improving feature-based anomaly detection, little work has been done that describes the design of a complete toolchain used to analyze behavioral metrics of enterprise networks. As previously discussed, the problem is making sense of an overwhelming amount of data without manually conducting forensics at the outbreak of a problem. The proposed tool chain would consist of the following phases:

- **Netflow Collection:** reduces the size of packet data and preserves anonymity while preserving network structure
- **Network Analysis:** represents the amount of flows into measurements for host and aggregating netflows by time intervals
- **Actionable Classification:** classifies subset of streaming netflows into different classes of hosts by behavior and aggregate events

The thesis proposes a toolchain for monitoring networks efficiently using network science. This method was derived using experiments created from actual case studies on a real large-scale enterprise network. The paper will start with a comprehensive literature review on the current methods used in network monitoring highlighting the differences of the work done for this thesis. Next, the paper will define the toolchain and its components in detail in the Methods chapter. Then, the paper will describe the results of many experiments conducted to validate the methods. Then, the paper will discuss the results and its implications on the use of the toolchain. Finally, the paper will summarize the main findings on the use of the toolchain and discuss its limitations and recommendations for future work.

Chapter 2

Literature Review

As mentioned in the prior chapter, there has been a significant amount of work in network monitoring. Some work has focused on packet level data and others have focused on netflow. However, most of the work done in this paper focuses on a completely novel paradigm as many social network techniques have not been used for measuring network behavior. This chapter will focus on defining the technical work done using graph methods to characterize telecommunications networks and all other methods used in feature-based anomaly detection. The chapter will also compare each method with the work described in this thesis.

2.1 Graph Methods for Feature-Based Anomaly Detection

There have been many techniques used for anomaly detection, however few have used graph-based representations of host to host interactions to define features. Graph-based metrics were examined because absolutely no work was done in analyzing network traffic with all of the different measurements offered in the network science field. Graph theory was the closest field to network science in telecommunication network analysis.

One recent paper used Tsallis and Shannon entropy on a graph representation of a network to detect events such as DDoS, flash crowds, and port scans [2]. The graph structure in the Tsallis and Shannon entropy paper consisted of nodes of devices and links of connections over a time interval. The measurements used to quantify behavior included device out-degree, device in-degree, percent and distribution of packets sent, and top-k devices that received or sent the most data. The method in this paper was very similar, however rather than focusing on just in-degree and out-degree, the work of this thesis focuses on network density measurements and clustering measurements as well and various graph manipulations to highlight different changes within the data. The entropy part of the paper can be applied directly to the

same measurements defined in this thesis.

One paper used PageRank to reduce the size of data to highlight anomalous behavior [59]. The paper described a multi-staged process for filtering hosts to highlighting hosts and interactions of interesting. The first stage relied on identifying the dominant benign servers in the network using user defined thresholds for port, flow count, packet count, byte count, and in/out degree. The thresholds are set depending on if it is a mail server, DNS server, or web server. The second stage filters out all flows to and from the identified benign servers from the first stage. Last, a model was created based off the assumptions of the behavior of a C2 server and the authority and hub scores were used to filter hosts that most match that behavior. The implementation covered in this research is similar, however the research of the thesis adds network level measurements, grouping level measurement, and other methods of extracting hosts besides total-degree. This paper inspired many of techniques used in this paper for data reduction.

In a similar paper by Carnegie Mellon University, researchers used an entropy-based approach to detect manually created anomalies using IP addresses, ports, flow sizes, and degree distribution [50]. The models were created by injecting anomalies as a ground-truth dataset to create models using resulting metrics [50]. The current study expands on this approach by using a larger network, more network measurements, and different techniques to reduce the size of the data.

One paper used network science measurements to analyze the set of malicious traffic [64]. The paper used closeness centrality, a network science measurement to characterize a host's position within the network and used changes in these host level measurements to predict anomalies. The work has the limitations of only being ran with a small network structure. With the size of the network structure used in the thesis, closeness centrality may not provide as meaningful measurements without reducing the data. However, if the data is reduced, the methods described in the paper can be implemented into the work of this thesis.

2.2 Alternate Feature-based Anomaly Detection Methods

Though graph methods of characterizing network traffic are rare, feature-based anomaly detection is not novel. The main approaches used in feature-based anomaly detection methods are entropy and machine learning.

2.2.1 Entropy Based Methods

Entropy is the process of reducing network traffic into a single measurement and calculating the difference between the current period and a baseline period. One of the first works that used this approach was

[74]. The paper focused on using Shannon entropy on host to host data to detect 4 traffic patterns, Concentrated origin and concentrated destination, concentrated origin and dispersed destination, dispersed origin and concentrated destination, and dispersed origin and dispersed destination. Another approach was mentioned in the prior section that also used a graph-based approach of representing the network structure [50]. These methods are closer to a graph approach and rely on understanding the behavior of an attack and correlating it to how hosts communicate with each other in the network. The work of this thesis extends on this by using network science to define more complicated features of the interactions between hosts within the network.

Shannon entropy was expanded in [65]. The method they coined Traffic Entropy Spectrum aggregated traffic into bins of 5, 10, and 15 minutes, calculated Tsallis entropy values, and normalized the data using maximum and minimum entropy values to define dominating changes and whether the change should be normal or not. The method was tested using 3 DDoS attacks and 2 worm outbreaks. Again, network science measurements can be used alongside the methods of this paper to implement streaming systems that adjust the model for normal behavior.

Moreover, another method called dynamic entropy was performed in [35]. The dynamic approach made hosts keep track of the current degree of interactions they are receiving. This state constantly updates as connections change. The connection changes are monitored by tracking request and replies from hosts. When groups of hosts interact with each other, their activity and state changes are modelled into specific events. This approach can also be supplemental to the work described in this thesis.

Another more recent paper addresses the limitation entropy methods have on large networks using adjustable piecewise entropy[67]. Adjustable piecewise entropy divides the feature space into 2 parts before computation, high probability and low probability. This lowers the amount of computation that is required by removing parts in an intermediate calculation. The methods described in the thesis can be implemented with this method to improve performance.

Because entropy approaches look at host to host connections, entropy-based approaches and the proposed network science approach go hand to hand. The methods used in network science can be applied to entropy approaches for optimizing performance and implementing streaming systems. Network science provides new methods to quantitatively describe the relationship between hosts within the network during a given timeframe.

2.2.2 Machine Learning Based Methods

Most of work done on feature-based anomaly detection used machine learning approaches. Some of the techniques used included principle component analysis, graph representation of features, SVM, Markov Chains, and neural networks.

One paper used a technique called Principle Component Analysis to separate normal and anomalous behavior represented as a time series graph [39]. Principle component analysis focuses on

Though the following papers used a graph-based approach, they focused on creating graph of the features of individual netflows rather than a graph of host to host interactions. These papers tracked features of flows such as source IP, destination IP, and protocols within a graph as clusters to categorize anomalous activity [48][30]. The papers took entirely a big data approach and created blind models from a large amount of data. The work described in the thesis focused more on validating assumptions on telecommunication events based off the changes in network structure over time. Additionally, these papers used heavily labelled data from the 1999 KDD Cup network intrusion dataset. The level of detail of this data would not be feasible when applied to netflow collectors in a real enterprise network. However, this machine learning approach can still be applied using the measurements derived from a network science approach as features.

The following paper used distance sum-based support vector machines to classify network anomalies [27]. It focused on first using distance of features between netflows to reduce the feature set and following it with SVM to classify the sets of activity. The algorithm was applied to the 1999 KDD Cup network intrusion dataset. Again, the use of this data provides serious limitations on a live setting due to the amount of labelled data given. Additionally, it is very old and network structures have changed significantly since then.

Moreover another paper used Markov Chain to model network events and whether if it resulted in a success or not [58]. The paper used a series of attack data provided by DARPA. Their method focused on classifying sequences of system calls on a cloud server to determine if a malicious event is occurring. Markov models can be directly applied to the network science approach because of the temporal representation of network structures in the traffic data. Sequence or functions can be trained if ground truth data was provided. Unfortunately, this is very difficult to apply on live data that is unlabeled.

A recent paper used neural networks and fuzzy categories to create a semi-supervised approach of identifying network anomalies [6]. Fuzzy categories represent bins of uncertainty that are used to decide which connections can be fed back into the model to improve its performance. The paper used a neural network to classify anomalies using an NSL-KDD dataset. Like the feature graph paper, it had the lim-

itation of requiring heavily labelled data which is difficult to obtain in a live networking environment. Additionally, models change depending on the network's topology.

Overall, most of the research on classifying network events used a similar approach and looked at a set of labelled netflows and attempted to characterize what they are used for by its feature space. All of this research did not seek to understand and define the behavior of each network event but used a large amount of previous defined data in order to predict what a netflow is used for. However, these models have the limitation of being difficult to obtain data for and they would not be very resilient to evolving attack patterns and obfuscation techniques.

The attack landscape constantly changes within an enterprise network as new machines are added, new applications are used, and cloud environments are integrated. Additionally, threats change as new attack strategies develop which can show completely different behavior than those interpreted by models that do not use interactions and relationships as features. Moreover, getting the data labelled for future classifications is difficult because it requires seeing new attacks enough times to be able to characterize it. It can take years to get enough attack data

Chapter 3

Methods

The following chapter will discuss the whole proposed toolchain process and components within the toolchain. The chapter first introduces the toolchain in step-by-step detail. Next, the paper describes the components used to implement for each step of the toolchain. Finally, hypothesized challenges with the toolchain are discussed in the last section.

3.1 Tool Chain Process

As discussed in the introduction, the toolchain was divided into three phases. For this thesis, the toolchain was implemented in parts using a range of techniques. These parts were then tested to prove its feasibility. Figure 3.1 shows a diagram of the toolchain and the components that it is made of. The following sections and subsections describes the implementation of each part of the toolchain grouped into the three phases.

3.1.1 Netflow Collection

The Netflow Collector was defined in order to preserve anonymity while reducing the size of the data. Though netflow does abstract details of the interactions between hosts, network science only needs interactions to define network structures and highlight a host's role within the network. The Software Engineering Institute at Carnegie Mellon University's System for internet-Level Knowledge (SiLK) was the netflow collector used to represent this stage of the toolchain. SiLK has a very flexible method for querying netflows and includes many metrics for describing interactions between hosts. The output of SiLK is just simple CSV files that can be parsed into formats for any other tool in processes further along the toolchain. Alternative netflow collectors that can be used include ntopng¹ and nProbe², NFDUMP³,

¹<https://www.ntop.org/products/traffic-analysis/ntop/>

²<https://www.ntop.org/products/netflow/nprobe/>

³<https://github.com/phaag/nfdump>

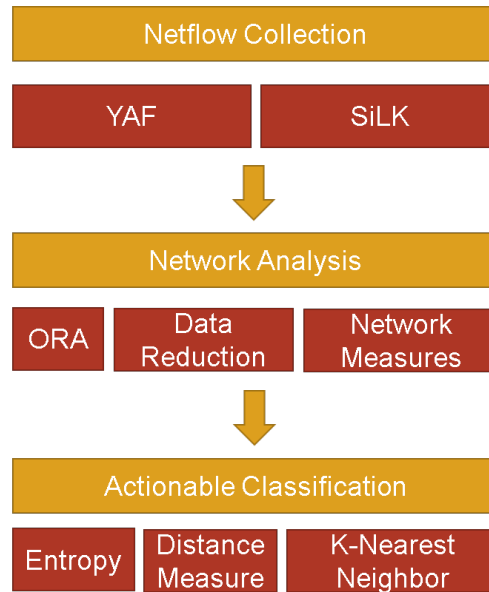


Figure 3.1: Diagram of the toolchain process and the components that make up each phase

and EHNT⁴. All of the listed tools are open source, free tools that require installation into a router and another machine used for storing netflow data. SiLK was used because it was already installed in the company's environment.

3.1.2 Network Analysis

Network science provides a mean to quantitatively describe the network. This phase of the toolchain centers around manipulating the network time intervals, forming subsets and ego networks out of key components within the network, and calculating network science level measurements over the intervals. In the experiments carried out in the thesis, ORA and Python were used. Python was used for network manipulation operations that ORA did not support. This step is necessary for creating quantitative data that can be used for classification. Though the metrics can be difficult to interpret because of the volume of data, it can still state some interesting things about the network. Other alternative social network analysis tools include UCINET⁵, Pajek⁶, and Gephi⁷. ORA was used because it supports a higher number of nodes and links than any other network science analysis tool, it supports the largest library of computational measurements, it offers temporal analytics, and offers a very robust easy-to-use interface for visualizing network structures.

⁴<http://ehnt.sourceforge.net/>

⁵<https://sites.google.com/site/ucinetsoftware/home>

⁶<http://mrvar.fdv.uni-lj.si/pajek/>

⁷<https://gephi.org/>

3.1.3 Actionable Classification

Lastly, the data used to describe network structures over time is processed into terms that a network administrator can respond to. Event period measurements and host behavior measurements can be bulked into models to predict their presence from a streaming set of netflow data. This final step completes the process that prunes an enormous amount of packet level data into actionable intelligence that a network administrator can use to make decisions. Actionable Classification was carried out using a combination of Python, R scripts, and machine learning algorithms.

3.2 Netflow Collection

The following section will describe the components implemented and integrated for the Netflow Collection phase.

3.2.1 Netflow

Network flows (or netflow as it is abbreviated) are simply logs of aggregated packet data throughout a network. According to the literature, netflow can include a flexible amount of information including source and destination IP addresses and port numbers, packet contents, and meta-information [32]. Flow exporting is cost efficient because passive collectors simply listen to activity without affecting observed traffic flows and only snapshots of aggregated packet captures are taken rather than the whole packets themselves.

The reason behind the popularity of netflow analysis is the privacy of the data when compared to packet level data and its significantly smaller size [59]. Thus, processing netflow data is a viable method for network administrators to conduct real-time analysis of network traffic and identify events of interests. Even though netflow reduces the size of network data by multiple orders of magnitude, it still has problems for enterprise networks because of the large number of hosts and activity going through them. Thus, it is still very hard to pinpoint anomalous behavior within the noise of normal behavior from other sets of hosts.

3.2.2 YAF Yet Another Flowmeter

The data was gathered using a tool developed by the Software Engineering Institute at Carnegie Mellon University called Yet Another Flowmeter (YAF)[34]. YAF takes network packet data and converts it into RFC-standard flow format, IP Flow Information Export (IPFIX)[68]. The format was a standard defined by

Cisco Systems for adding the functionality of gathering and analyzing aggregate packets using netflow. Under YAF's configurations, there are three conditions in which a netflow is created:

- A TCP session between two hosts is complete
- There is an idle time of 30 seconds between connections between two hosts
- The max flow duration 30 minutes expires

YAF was installed into the company network to generate netflow data. YAF is available for free by the Software Engineering Institute at Carnegie Mellon University⁸.

3.2.3 SiLK System for Internet-Level Knowledge

The data was queried using a tool developed by the Software Engineering Institute at Carnegie Mellon University called System for internet-Level Knowledge (SiLK). SiLK packs collected flow from YAF into a compact representation, stores it, then facilitates retrieval and analysis of stored flows. SiLK is compatible with many different flow formats, however IPFIX was the format used in the SiLK installation. The YAF/SiLK installation for this study uses a virtual machine deployed as a network boundary router to inspect traffic, convert packets into netflows, and store flow records including the source/destination address, source/destination port, transport protocol, flow size, and duration [66]. SiLK is available for free by the Software Engineering Institute at Carnegie Mellon University⁹. An illustration of YAF and SiLK integrated into a network topology is shown in Figure 3.2.

After gathering data, the data was extracted through queries from the SiLK Database. The SiLK queries used to gather the data in this research used a combination of time ranges and protocols. Each netflow included only the source IP address, destination IP address, and time the flow started.

The data was gathered on a large-scale enterprise network with between 1,000 and 5,000 employees and approximately 3,000 network connected devices. The network is configured so that any machine can connect to any external IP address on the internet through their own internal DNS routers. A week (Sunday through Saturday) of data was gathered from a flash crowd incident and two normal weeks without any reported incidents. The flash crowd was a result of the disclosure of an event that triggered a large public outburst causing a flash crowd and the crashing of a few machines within the network. In this paper, the disclosure will be termed the event and the resulting flash crowd will be termed the flash crowd.

⁸YAF installation instructions, download link, and user guide can be found the following website: <https://tools.netsa.cert.org/yaf/>

⁹SiLK installation instructions, download link, and user guide can be found on the following website: <https://tools.netsa.cert.org/silk/>

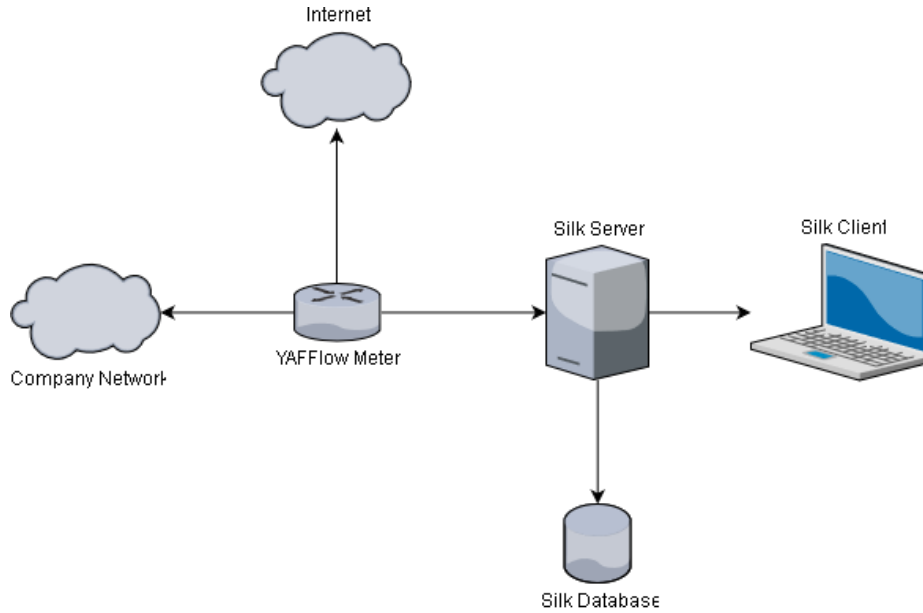


Figure 3.2: SiLK Collector Setup

After gathering data, the data was extracted through queries from the SiLK Database. The only metrics used for queries were time ranges and protocols. Each flow record gathered by SiLK included (among its 30 data elements [66]) a source IP address, a destination IP address, and the flow start time. There were 3 weeklong periods (Event, normal week 1, normal week 2), for each of the protocols examined.

The following command was used to gather the data in SiLK:

```
rwfilter -start=<STARTDATE> -end=<ENDDATE> -type=all -bytes=1- -proto=<PROTOCOLNUMBER>
-pass=stdout | rwcute -fields=1,2,9 -delimited=',' > <FILENAME>.csv
```

The start and end flags indicate the time interval of flows to extract within the csv file. The type defines if the extracted traffic is internal to internal, internal to external, or external to external. For the thesis, all traffic was extracted. Bytes determines the range of bytes the netflows should be when queried. For the thesis, any netflow with at least 1 byte was extracted. The proto flag indicates which protocol to extract flows from. For the thesis, protocols 17 (UDP), 6 (TCP), 50 (ESP), 47 (GRE), and 1 (ICMP) were extracted individually to their own CSV files. The pass flag indicates what output buffer would you want to send the query. The RWCut commands takes the output from a RWFilter command and converts it to a specified format. For the experiment, fields 1, 2, and 9 were taken. These fields pertained to the source host, destination host, and timestamp. Each field was delimited by a comma to create a CSV format and the output was written into a CSV file. Space was limited on the machine conducting SiLK queries so sometimes the whole week of data was divided into days or parts of days and aggregated at the very end using the Organization Risk Analyzer (ORA).

3.3 Network Analysis

The following section will describe the components implemented and integrated for the Network Analysis phase.

3.3.1 Network Science Measurements

In the network science perspective, this research examined methods for detecting changes within a network and the degree of change. Previous work by McCulloh and Carley has used statistical process control to detect behavioral changes within a network [45]. However, periodicity within networks caused noise that interfered with change detection.

In the network, each address represented an Agent node and each connection between addresses represented a link. This data was processed and analyzed using ORA, a dynamic meta-network assessment and analysis tool developed by CASOS at Carnegie Mellon University. All charts and network measurements were generated using this software. The data included roughly 42 networks each binned into 4-hour periods per week period. The rationality of the 4-hour period is that it encompasses most of the work day while creating a larger network structure. Having a larger time period can lower the granularity of the data when monitoring events over time, however having a larger network structure has the advantages of being able to calculate a greater variety of metrics.

Telecommunication network traffic gathered from a private network have a unique structure. Figure 3.3 shows a visualization generated by the Organization Risk Analyzer (ORA), a software used to analyze network data for the thesis, of a raw network structure of ICMP for a 4 hour period. ICMP was used because TCP visualizations required too much computational power to produce. For the visualization, the bottom 10% total-degree nodes were removed making the visualization a lot more readable. Without removing these nodes, typically each internal host was a star shaped network with many connections unique to each host. The connected hosts would be made up of any hosts the machine has connected to during that time period including any websites or resources the websites use, diagnosis/update checks for any software installed, the sending or accessing of emails, or any other internet function. The current visualization preserves the interactions with the most common external hosts in the network. Additionally, the nodes were sized and colored by their total-degree centrality to highlight the most active hosts within the telecommunication network. The biggest and reddest nodes are typically servers.

Typically, they are very sparse due to the variety of external hosts internal hosts connect to and the lack of visibility of external to external interactions. As a result, the network structures show largely clustered activity for internal hosts within the network amongst each other and to various external nodes. There

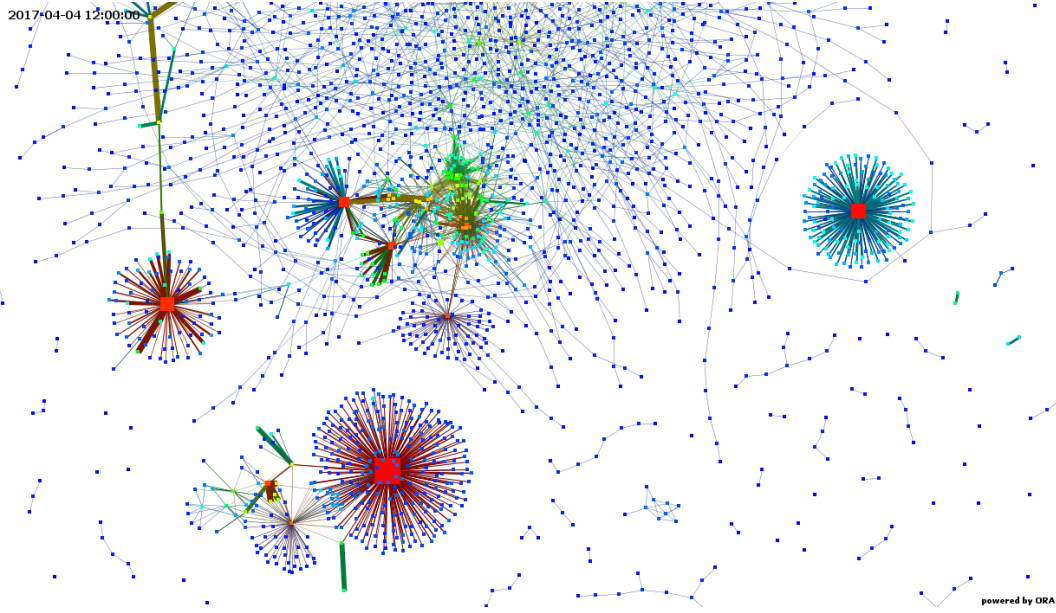


Figure 3.3: Visualization of ICMP Traffic of Enterprise Network during 4-hour Time Period

are groups of external nodes that have many interactions and this typically indicates external software that the enterprise uses such as email clients and cloud infrastructure. These differences in structure pose some limitations on calculating more complex measurements, however there are still many measurements that can help characterize the behavior of an enterprise network.

This paper uses techniques and measurements defined by Newman to analyze the enterprise network [47]. To analyze the enterprise network, a time series was created for the following network measurements:

- **Density:** The ratio of the number of links versus the maximum possible links for a network. In the context of telecommunications networks, it represents the overall connectedness of interactions between machines for a given time interval.

$$A = \text{binaryInputNetwork}$$

$$m = \text{numberRows}$$

$$n = \text{numberColumns}$$

$$\frac{\sum(A)}{m * n} \quad (3.1)$$

- **Fragmentation:** The proportion of nodes in a network that are disconnected. In the context of telecommunications networks, it is an indicator of separate interactional groups during the time

period.

$$\begin{aligned}
 S_k &= kthWeakNodes \\
 N &= numberNodes \\
 1 - \frac{\sum S_k(S_k - 1)}{N(N - 1)} & \quad (3.2)
 \end{aligned}$$

- **Clustering Coefficient:** The average density of each node's ego network. In the context of telecommunications network, it represents the degree to which IP addresses can be clustered into interactional groups during the given time period.

$$\begin{aligned}
 T &= numberTriangles \\
 C &= numberConnectedTriplets \\
 \frac{3 * T}{C} & \quad (3.3)
 \end{aligned}$$

- **Clique Count:** The number of distinct cliques to which each node belongs. A clique is defined as a group of three or more nodes that are all connected and that cannot be made larger by adding another node. In telecommunication networks, it represents the number of triads of interactions within the network during the time period.
- **Node Count:** The number of nodes within the network. In the telecommunication network, it represents the total number of machines interacting during the time period.

$$N \quad (3.4)$$

- **Link Count:** The number of links within the network. In the telecommunication network, it represents the total number of distinct interactions between machines during the time period.

$$L \quad (3.5)$$

- **Weighted Link Sum:** The number of weighted links within the network. In the telecommunication network, it represents the total number of interactions between machines during the time period.

$$L_w \quad (3.6)$$

- **Average Total-degree Centrality:** The average number of incoming and outgoing links out of all nodes. In the telecommunication network, it represents the average concentration level amongst IP addresses for the given time period. From now on, the equation will be represented as $tdc(A, i, V, N)$. The calculation for total degree is below[71]:

$$\begin{aligned} A &= network \\ i &= nodeToCalculate \\ V &= maxLinkValue \\ N &= numberNodes \\ \frac{\sum A(:,i) - A(i,i)}{2 * V(N - 1)} \end{aligned} \quad (3.7)$$

Finally, the average total-degree centrality of the network is calculated below

$$\frac{\sum_{i \in A} tdc(A, i, V, N)}{N} \quad (3.8)$$

Within the network, each node represented a host and each link represented a netflow. Figure 3.4 depicts an example ego network from the netflow data. Ego networks are generated from selecting a node and all its neighbors. Many of the network science measurements focus on characterizing the network's structure. Network science measurements have frequently been used as methods for measuring effects within network data [4]. This paper hopes to address if the network structure and host behavior changes during a flash crowd incident.

3.3.2 ORA Social Network Analyzer

ORA or the Organization Risk Analyzer was a tool developed by CASOS created for network analysis[12]. It was developed to help organizations and individuals with little statistical and technical background an-

IP-205.141.142.204 (205.141.142.204)

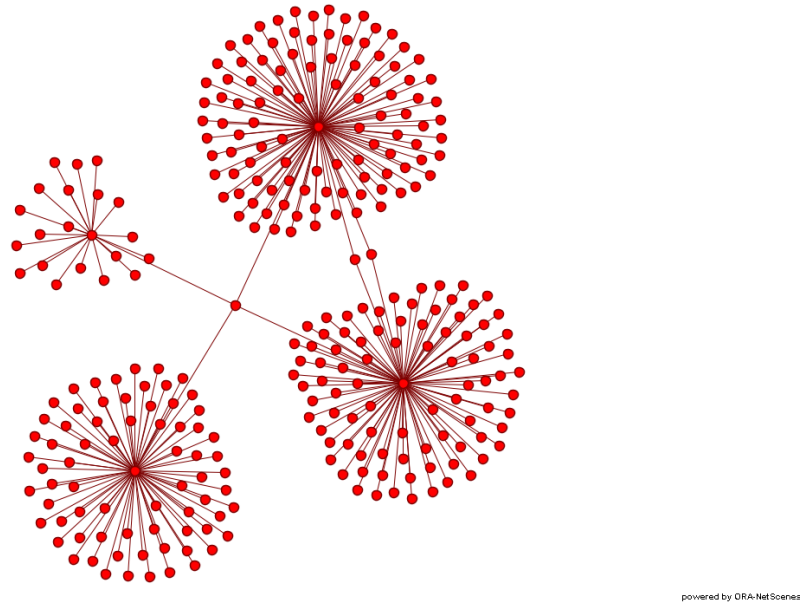


Figure 3.4: ORA Visualization of a 2 Radius Ego Network on a Company Host

alyze their network data. ORA can process data on many different formats and can calculate network-level and node-level metrics, clustering algorithms, correlation/regression reports, represent visualizations, in-editor network manipulations, and many other functions. ORA has the capability of processing a large amount of data and it proved very useful for the analysis of netflow data. ORA was one of the main tools used in calculating measurements, manipulating networks, and generating charts and reports. The operations that could not be done in ORA were done with Python or R. The user guide and the product page are included in footnotes below^{10,11}.

The following subsections will describe the process on importing data, calculating chart measures, generating ego networks, and reducing networks.

Importing Data

To import data, ORA's Data Import Wizard was used. This was accessed from File > Data Import Wizard. The configuration used was "Import Excel of text delimited files" > "Table of network links". CSV files in the following format were imported into the Data Import Wizard.

```
sip, dip, stime
<IP address>, <IP address>, <Start time>
...
```

¹⁰ORA's QuickStart Guide can be found on the following web page: <http://casos.cs.cmu.edu/projects/ora/ORA%20QuickStart%20-%20v2.pdf>

¹¹ORA-Lite can be downloaded from the following link: <http://casos.cs.cmu.edu/projects/ora/software.php>

Step 1: Select a file containing table data **with** column headers:
7 files selected Browse

Step 2: Check the columns that contain node names and enter the nodeset information:

SIP column contains:	DIP column contains:	STIME column contains:
Node names: Node names	Node names: Node names	Dates: Dates
Nodeset class: Agent	Nodeset class: Agent	Date pattern: yyyy/MM/dd'T'HH:mm:ss
Nodeset name: IP	Nodeset name: IP	
<input type="checkbox"/> Make repeated names unique	<input type="checkbox"/> Make repeated names unique	

Step 3: Define networks and attributes based on the columns:

Networks **Networks and Labels** **Networks combined names** **Attributes**

Source Node	Target Node	Link Value	Network	Network Column
sIP	dIP	sTime	IP x IP	

New Clear

Load configuration Save configuration

Cancel < Back Next > Finish

Figure 3.5: Configuration for importing SiLK CSV files into ORA

Step 1: Select a file containing table data **with** column headers:
7 files selected Browse

Step 2: Check the columns that contain node names and enter the nodeset information:

SIP column contains:	DIP column contains:	STIME column contains:
Node names: Node names	Node names: Node names	Dates: Dates
Nodeset class: Agent	Nodeset class: Agent	Date pattern: yyyy/MM/dd'T'HH:mm:ss
Nodeset name: IP	Nodeset name: IP	
<input type="checkbox"/> Make repeated names unique	<input type="checkbox"/> Make repeated names unique	

Step 3: Define networks and attributes based on the columns:

Networks **Networks and Labels** **Networks combined names** **Attributes**

Source Node	Target Node	Link Value	Network	Network Column
sIP	dIP	sTime	IP x IP	

New Clear

Load configuration Save configuration

Cancel < Back Next > Finish

Figure 3.6: Configuration for aggregating SiLK CSV files into ORA

These files were all generated from SiLK. Figure 3.5 shows the configurations that were set in the next window. These configurations mean that each source IP to destination IP constitute a link. The STIME column is used as a timestamp for the netflow link. In the next window, networks are configured to be aggregated by 4 hours within a dynamic meta-network. Figure 3.6 shows these configuration options.

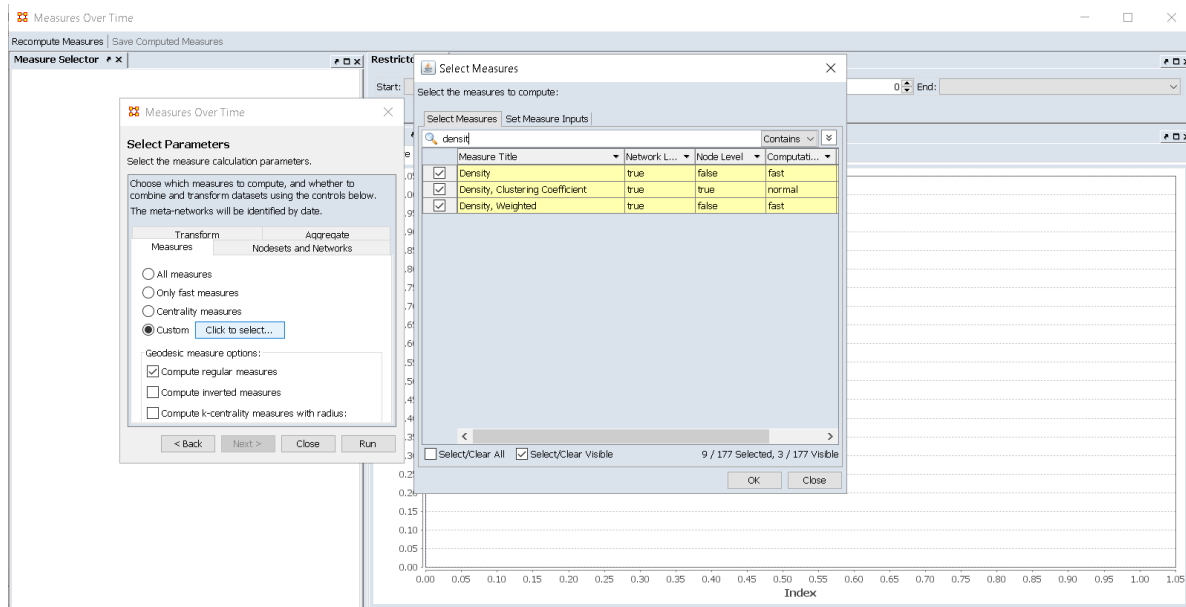


Figure 3.7: Configuration for selecting measurements for chart measures in ORA

Generating Chart Measures

Generating chart measures were very simple in ORA. To generate them, the dynamic meta-network was selected and the Measure Charts... button was pressed. From there, Custom Measurements were selected, and the 9 measurements used in the thesis were selected. Figure 3.7 shows this configuration menu.

Finally, after the charts were generated, the Save Computed Measures button was clicked and the data were saved into CSV files.

Reducing Networks

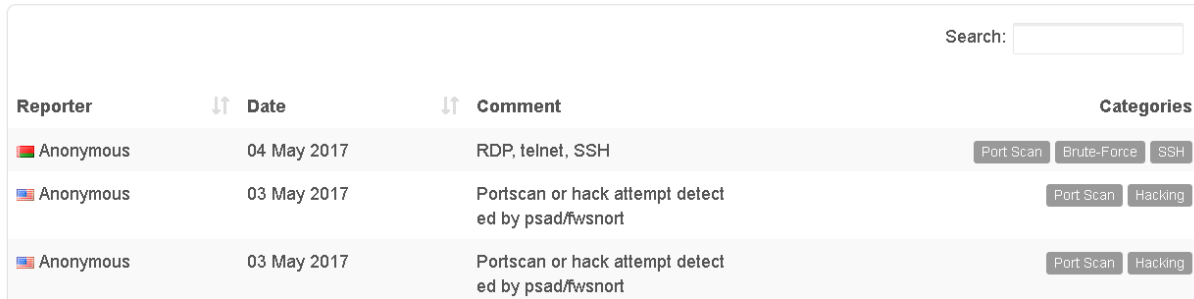
To reduce networks, the following procedure was taken. The dynamic meta-networks were copied and pasted within the work place. For each meta-network within the network, a subset of the network was generated. To do this, the network and agent list were selected. The navigation tab Nodes > Select nodes from file were pressed.

Then a line delimited file of IP addresses was passed as a parameter resulting in all the nodes whose IP address appeared in the file being selected. Finally, the navigation tabs Nodes > Keep only selected nodes was selected. This resulted in the inclusion of only the selected nodes within the network. This was repeated for every meta-network within the dynamic meta-network.

IP Abuse Reports for 183.60.48.25:

This IP address has been reported a total of **1260** times. 183.60.48.25 was first reported on 02 Jul 2013. The most recent report was **3 days ago**.

Recent Reports: We have received reports of abusive activity from this IP address within the last week. It is potentially still actively engaged in abusive activities.






Reporter	↑↓ Date	↑↓ Comment	Categories
 Anonymous	04 May 2017	RDP, telnet, SSH	Port Scan Brute-Force SSH
 Anonymous	03 May 2017	Portscan or hack attempt detected by psad/fwsnort	Port Scan Hacking
 Anonymous	03 May 2017	Portscan or hack attempt detected by psad/fwsnort	Port Scan Hacking

Figure 3.8: Report of a malicious host on abuseipdb.com

3.3.3 AbuseIPDB

AbuseIPDB is a project managed by Marathon Studios Inc¹². It is used by webmasters, system administrators, and network administrators to identify potentially malicious hosts. It is maintained by a network of administrators who can freely report IP addresses who were found conducting malicious behavior. Network administrators typically label the reason for reporting hosts. AbuseIPDB users have reported millions of IP addresses and its popularity among network administrators have been increasing exponentially since 2014. Figure 3.8 shows an example report of a host on abuseipdb.com.

For the project, AbuseIPDB was used as a source for ground truth malicious IP addresses. The hosts within the network examined were queried by the amount of reports they received from AbuseIPDB and those that exceeded a certain number of reports were considered malicious. Having a set of labelled malicious IPs creates the ability to characterize and possibly detect malicious behavior on a network by host. Because hosts can be freely reported by any user of AbuseIPDB, it can potentially produce inaccurate data. For example, number of Google and Facebook owned IPs have been reported and it is unclear if they should be considered malicious. However, if their behavior seemed intrusive enough to warrant hundreds of reports from network administrators, then its categorization may still prove useful.

3.3.4 Network Size Reduction

Because the netflow data encompasses the whole enterprise network and network events typically only affected select groups within the corporation, a few approaches were used to extract the groups affected. Previous research found that sampling can destabilize centrality measurements on a network so reduction

¹²AbuseIPDB can be accessed from: <https://www.abuseipdb.com/>

techniques that followed the reasoning of the hypothesized results were used [18][10]. These sampling techniques were used to eliminate noise within the data to help evaluate hypothesis on the effects of incidents to normal network behavior. Many network events influence a subset of hosts within the network and it is sometimes beneficial to analyze subsets of a network. The following approaches were only done on TCP and UDP.

Top Change in Total-Degree Host Selection

Two static networks were created by unionization (combining network bins into a single network where all link weights were summed together), one corresponding to the week of the event and the other to the normal work week. Then total-degree centralities were calculated for all IPs. Finally, the IPs within the top 20% difference were extracted. Previous dynamic analysis was repeated with only the extracted IPs. The rationality behind this approach was that the interactions of hosts who were affected by an incident should have had significantly different behavior metrics during the time of the incident. This technique is meant to be used to find hosts whose behavior changed as a result of an incident and generates a network structure with only those queried hosts. The top 20% was used because it seemed to generate a small enough sample to show quantitative difference in measurements while being big enough to keep the same network structure and no other values were experimented with.

Top Increase in Sending Decrease in Receiving Host Selection

After generating the two static networks per period, the top 20% hosts that had the greatest increase in out-degree and greatest decrease in in-degree were chosen between the event static network and the normal static network. This method essentially highlighted the hosts that transmitted more and received less during the event week than normal. Again, this method assumes that the event resulted in the decrease of interactions for some hosts and the increase of interactions from other hosts. It was hypothesized to have good performance at highlighting hosts who were either brought down or engaging in a higher amount of connections during denial of services.

Functional Sub Groups

The last network size reduction technique required ground truth data from the company itself. For this approach, groups of IP addresses by division within the company were placed into their own network alongside their interactions with other hosts. Examining the differences in interactions of individual division over time and during telecommunication events is beneficial to understanding how divisions use their machines and how they respond to telecommunication events. Quantifying the behavior of each

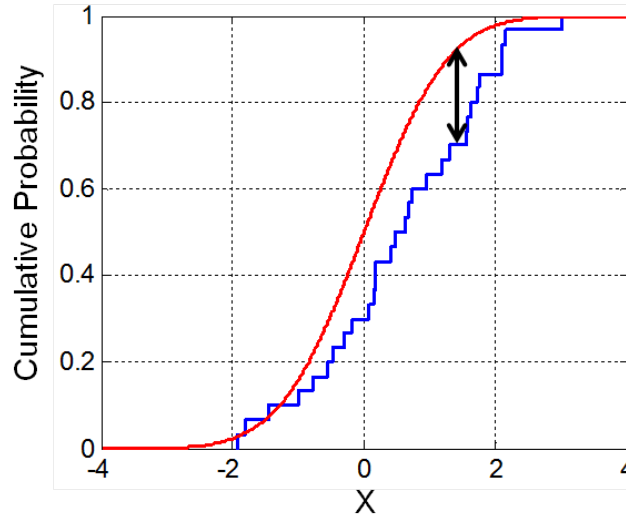


Figure 3.9: Graph depicting an example of the Kolmogorov-Smirnov test[11]

group can help in creating models of normalcy for each division as well. For the company being examined, 4 groups were analyzed. These groups included technical user machines, corporate management user machines, campus security user machines, and servers. Most of the groups had more than 100 IP addresses, however corporate management had the largest amount of IP addresses totaling about 1,000.

3.3.5 Two-Sample Kolmogorov Smirnov Test

The Two-Sample Kolmogorov Smirnov test (KS test) is a non-parametric method for comparing two distributions. KS tests work by calculating the difference between the empirical distribution function of a sample and the cumulative distribution function of a baseline sample[44]. Figure 3.9 depicts an example of the KS test where the red line represents a cumulative distribution function of the baseline sample and the blue line represents the empirical distribution function of the sample.

A number of literature has used the KS Test in order to test for the difference in behavior of computer networks over a period of time[46][28][26]. For the proposed toolchain, the KS test was a crucial component in initially testing if network science measurements differ between events and groups of hosts proving the feasibility of a classifier. KS Tests were calculated using R.

3.3.6 Correlation Matrix Generation

Correlation Matrices are typically used to represent the changes between two matrices. In the case of the dynamic metanetworks represented in the netflow, they are used to compare static intervals of two dynamic metanetworks. Correlation values use Pearson Correlation to compare two matrices. The Pearson

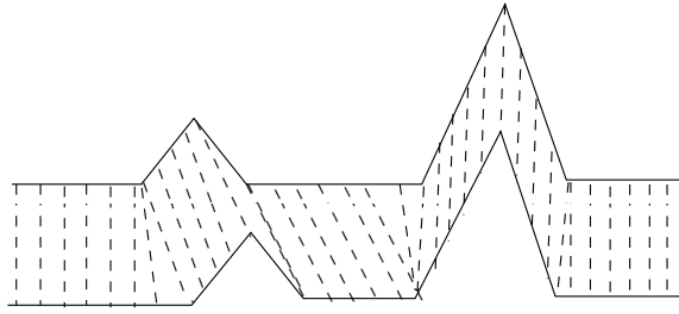


Figure 3.10: Graph visualizing distance measurements calculated in the dynamic time warping algorithm[51]

Correlation calculates the number of standard deviations both matrices are from their own respected mean and normalizes them with each other [63][33]. Previous work on netflow has used Pearson’s Correlation to quantify the differences between computer network models[13][22][49]. Correlation Matrices are used to compare the specific days as in Mondays or Fridays of two different dynamic metanetworks. The different metanetworks can be comprised of ego-networks composed of a different set of hosts or different events that can exhibit different behavior over time.

3.4 Actionable Classification

The following section will describe the components implemented and integrated for the Actionable Classification phase.

3.4.1 Dynamic Time Warping K-nearest Neighbor Classification

Dynamic time warping is a technique used to measure the differences between two time series. Dynamic time warping was created to address the gap of Euclidean Distance wherein same events that have different speeds and reaction times do not result in matches. Dynamic time warping works by calculating every combination of distances between points in each time series graph and selecting an optimal path that minimizes the distance between both time series[8]. Figure 3.11 visualizes the distance metrics calculated from the dynamic time warping algorithm.

After calculating a distance metric between two time series, classification algorithms such as K-nearest neighbor are used in defining a model for a class of data given a set of labelled data points. K-nearest neighbor or KNN works by storing a set of labelled points and evaluating the neighbors of an unlabeled point. The class is determined by the composition of classes among neighbors given a radius distance,

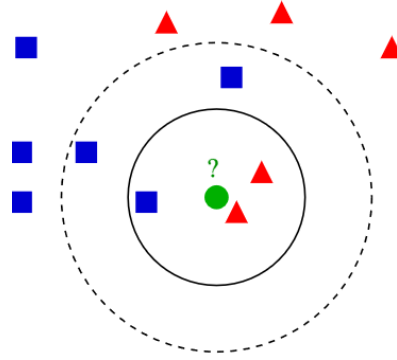


Figure 3.11: Visual of K-nearest neighbor algorithm where blue and red represent different classes, green represents a test point, and both circles represent different values of k . The inner radius would classify green as red and the outer radius would classify green as blue [3]

K[19]. The following literature have used either Euclidean distance or dynamic time warping as a distance measurement and K-nearest neighbor to classify netflow data[31][72][29][56][54].

3.5 Challenges

However, computer network analytics is a difficult problem because of the sheer size of data. Much of the data, especially in a large-scale enterprise network is noise. Users visit hundreds of websites a day and these websites connect to hundreds of advertisement distributors every day. All this activity will be captured in netflow and previous activity is easily influenced by current events and changing trends.

Additionally, care must be taken into not abstracting too much of the data. For instance, no matter how many packet records a collector gathers, if a network administrator even finds one pair of internal hosts communicating over a port that should not be in use, he already found an event of interest. Administrators have a simple set of things that they are interested in within their network, so an automatic detection tool chain must include these features.

Moreover, event detection algorithms may be hard to define because their effectiveness depends on the number of occurrences the model was trained on. It is difficult to integrate event data from other computer networks because topologies vary, and this can cause a very different set of reactions for another network when compared with the network of interest.

Lastly, host behavior can be very difficult to differentiate between. Previous network security methods used threat signatures and tracked a context free graph of actions that indicate malicious behavior. These methods require packet-level inspection. Only having link level behavioral interactions between hosts may result in ill-defined classes for detection algorithms.

Chapter 4

Results

The thesis was made up of a variety of experiments to implement parts of the toolchain, learn more about the data, find the meaning of various network measurements calculated from the data, and test methods of processing the data to highlight important information to network administrators. This chapter focuses on describing these experiments and their results. These experiments are effectively case studies of the toolchain in action split into smaller components. The experiments and their results highlight the best components to use in constructing a toolchain that improves a network administrator's capabilities.

4.1 Netflow Data Structure: Network Analysis

This section focuses on the experiments analyzing the whole netflow structure and how it changes depending on the environment.

4.1.1 Previous Network Structure

Overview

The network structure was created as a part of an experiment examining the change in network behavior after a flash crowd incident. The results of the experiment are in the following section.

The data was gathered on a large-scale enterprise network and started with a very simple structure. The network was dynamic meaning that many network structures were created representing a single time frame within a larger time frame. Each node represented a host and each link represented a unidirectional netflow from host to host. Each link has a weighted value representing the number of flows within the timeframe of the network.

Additionally, the data was placed in bins by the following protocol: TCP, UDP, ICMP, ESP, and GRE. These protocols were the top 5 protocols used within the network and each of these bins indicated different functional usages. The rational of this type of binning was that the behaviors of these protocols should vary. Protocol bins made the most sense for reducing some noise and is frequently used in other netflow analysis [39][50]. The functions of each protocol are as follows [57]:

- **TCP:** Main transport protocol of Internet protocol suite, majority of network traffic including the World Wide Web, email, remote administration, and file transfer. These activities are human-directed and task-oriented and may occur over durations of up to several seconds or minutes[25].
- **UDP:** Secondary transport protocol of Internet protocol suite, includes Domain Name System requests, SNMP, RIP, DHCP, voice and video traffic, and streaming content. These activities are human-directed and task-oriented, but most frequently occur over durations of a few seconds at most[55].
- **ICMP:** Supporting protocol of Internet protocol suite, used by network devices for error messages and debugging information. This traffic is not human-directed or task-oriented and occurs in brief bursts[17].
- **ESP:** Used in IPsec protocol suite for enforcing confidentiality, authentication, and integrity in IP packets. This traffic is typically ongoing activity in support of established connections and is human-directed but not task-oriented[38].
- **GRE:** Used with IPsec VPNs to enforce security of IP packets. This traffic is infrequent, occurring only at the start of these VPNs[23].

Results

The data included roughly 42 networks each binned into 4-hour periods per week period. There were a couple of holes, at most 24 hours long in some of the data because of technical issues with the SiLK Collector during those periods. There were 3 weeklong periods (Event, normal week 1, normal week 2), for each of the 5 protocols. This totaled to about 630 networks and roughly 300GBs of data. The distribution of the amount of network data is illustrated in Table 4.1. By protocol, the data was distributed as follows per 4-hour network:

Overall, the initial data structure was effective at representing the network. Periodicity with the work week was clear even after splitting the networks into 4-hour aggregate structures. Additionally, splitting

Table 4.1: Distribution of Protocol Data

Protocol	Number of Hosts	Number of Links	Size in MB	Percentage of Size
TCP	400,000	4,000,000	271,000	91.11%
UDP	35,000	400,000	23,900	8.03%
ICMP	40,000	30,000	1,830	0.62%
ESP	70	130	509	0.17%
GRE	13,000	25,000	215	0.07%

the network by protocol showed clear differences in behavior as well. However, it was clear that some protocols were unnecessary because of the lack of data.

One concern with this structure was the possibility of it abstracting important information from a network administrator. Netflow already abstracts a significant amount of information when analyzing telecommunication data, however aggregating the structure into 4-hour periods may obscure signs of an attack or exfiltration attempt. Moreover, a toolchain for monitoring an enterprise network using the techniques described in this paper will require streaming data. Therefore, smaller time intervals of network aggregation would be necessary to continuously update models.

4.1.2 Flash Crowd Incident Description

Overview

This experiment examined a case study of a flash crowd incident on a large, corporate-scale enterprise network. A flash crowd incident is a surge in traffic upon a machine from legitimate users that results in dramatic performance reduction or even crashing of the machine [5]. The particular flash crowd that this paper discusses resulted in the denial of service of a few machines. The flash crowd was caused by the disruption of a service the company offered that resulted in a sudden, large public outburst over the company network that disabled some machines that directly interacted with external customers. This paper compares the event with normal work weeks. As a part of the analysis, various network science techniques were used to reduce the dataset to highlight the impact of the flash crowd while removing extraneous noise.

Analyzing the differences in measurements of network incidents such as flash crowds and a normal work week makes it viable to create models to help predict them [7]. This prediction would allow network administrators to respond to events before they affect end users or reduce operational capability. A paper by Amaral et. al. shows the types of methods to which network science can be applied to detect network activities in real time [2]. Their method focused on applying Page Rank to reduce the size of data to highlight anomaly behavior. The method is similar to the total-degree measurement in the network science field and a similar data reduction method was used in the thesis.

A week (Sunday through Saturday) of data was gathered from a flash crowd incident and two normal weeks without any reported incidents. In this section, the disclosure of the service outage will be termed the event and the resulting flash crowd will be termed the flash crowd. In previous literature, flash crowds have shown a clear difference in behavior [7][5].

For the experiment, it was hypothesized that TCP and ICMP traffic behavior would be significantly different during the incident time when compared to the normal weeks and all normal weeks should show consistent behavior among their protocols. ESP and GRE should not be affected because it was assumed that the flash crowd would not affect user machines within the network. ESP and GRE focus on VPN and security protocols and these procedures will mostly be carried out by user machines within the network. UDP should not be affected because the majority of UDP traffic should be from the internal DNS servers. As a result, changes in internal host behavior are what would cause the most changes in DNS UDP traffic within the network as a whole.

For specific details on the scale of change, traffic over TCP and ICMP were predicted to increase overall during the time period. This was factored by the increased number of customer external machine interacting with the web server and other customer interaction portals all expressing concerns over the service outage. Therefore, link counts, link sums, and node counts should increase during these periods. Density measurements should decrease as a result of the flash crowd because it is directly correlated with the number of external hosts connecting to the network.

Raw Data Analysis Results

The initial measurements without any data reduction illustrated many differences between behaviors of different protocols, however it was not as successful at illustrating the effect of the event and flash crowd on the network.

For TCP, Density dropped after the disclosure of the incident to the public and skyrocketed a little before the end of the event. This indicates that immediately after the incident, more disconnected components of IPs were interacting within the network. This illustrates interactions between customers who are expressing their fears and concerns. The skyrocketing near the end may indicate that internally there was more communication about the status of the incident and attempts to resolve it. The time series of TCP density of the incident week and a normal week is in Figures 4.1 and 4.2 respectively.

The table 4.2 shows the result of a Two-Sample Kolmogorov-Smirnov Test. The test aimed to see if the normal week and incident week shared the same distributions for the measurements recorded.

The expected results of the KS-test were that the normal weeks should not show significant differences and that the incident week with the normal week should show significant differences. The KS-test for TCP

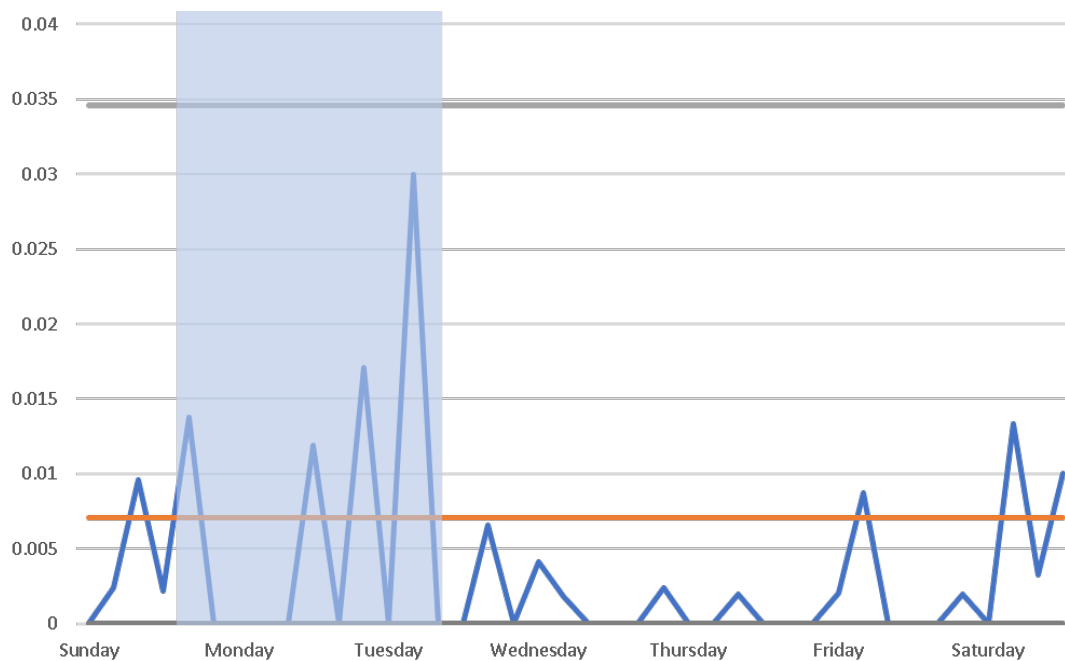


Figure 4.1: Time series of density of TCP connections during incident week; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, grey lines represent first level standard deviations of normal weeks

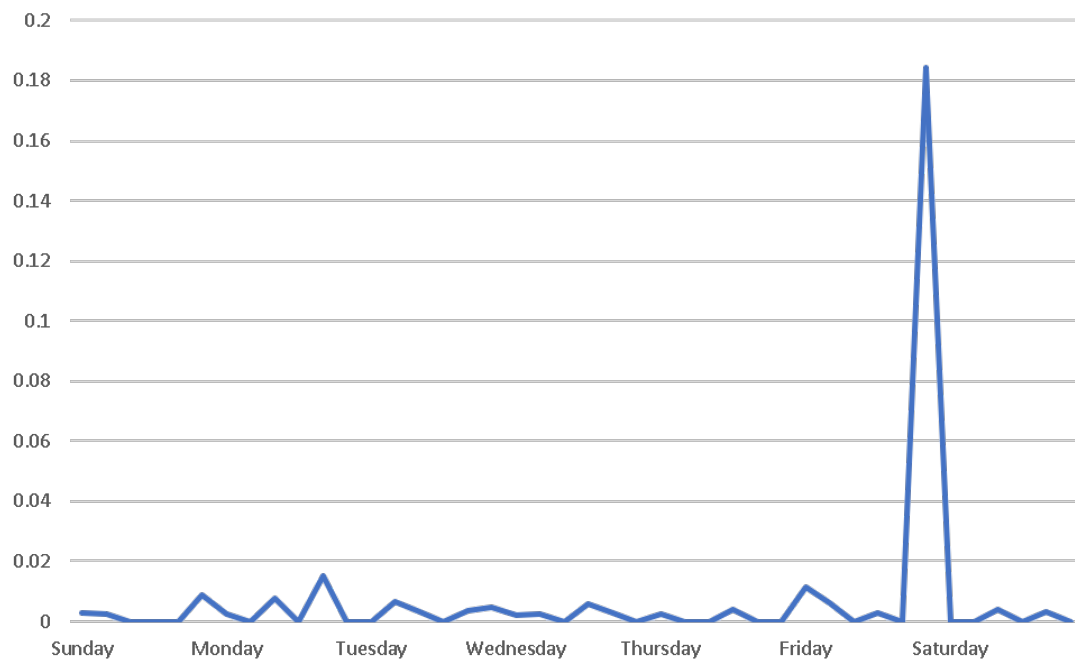


Figure 4.2: Time series of density of TCP connections during normal week

Table 4.2: Two-Sample Kolmogorov-Smirnov Test of TCP comparison of Normal Week and Incident Week and both normal weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	0.175	0.557	0.1667	0.6041
Weighted Density	0.1286	0.8872	0.1429	0.7848
Fragmentation	0.1310	0.8179	0.1905	0.4355
Edge Count	0.5036	2.965e-05	0.6667	3.781e-09
Edge Sum	0.9512	<2.2e-16	0.6429	5.793e-08
Node Count	0.2488	0.1278	1	<2.2e-16
Clique Count	0.6	7.846e-07	0.2619	0.1121
Clustering Coefficient	1	<2.2e-16	0.3333	0.0188

Table 4.3: Two-Sample Kolmogorov-Smirnov Test of UDP comparisons of the Incident Week and a Normal Week and the Normal Weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	0.1429	0.8037	0.1905	0.4313
Weighted Density	0.5897	1.556e-06	0.6191	2.046e-07
Fragmentation	0.1557	0.6451	0.2619	0.1123
Edge Count	0.2454	0.1433	0.1429	0.7912
Edge Sum	0.9762	<2.2e-16	0.5952	6.892e-07
Node Count	0.2088	0.2897	0.1905	0.4355
Clique Count	0.5952	1.196e-06	0.3333	0.0188
Clustering Coefficient	0.5476	1.08e-05	0.4286	0.0009

found that only clique count supported both assumptions. Density, weighted density, and fragmentation supported the hypothesis that normal weeks should show consistent behavior. Edge count, edge sum, and clustering coefficient supported the hypothesis that the incident week should show a different distribution of measurements than the normal weeks.

The results for UDP were difficult to interpret. There were no significant spikes or changes during the flash crowd time period and all time periods showed similar behavior for all measurements. Overall, this was expected. The UDP protocol is used mainly for DNS queries, network management, routing, and some voice and video traffic. However, the vast majority of traffic over UDP within the network in question is internal hosts to internal DNS servers. As a result, the flash crowd event should not result in significant changes in the composition of UDP network traffic. Tables 4.3 shows the results of the KS-test for UDP traffic.

Overall, UDP's results supported the hypothesis quite well. Density, fragmentation, edge count, and node count all satisfied the hypothesis in which behaviors should be consistent regardless if it was the flash crowd week or a normal week. Unfortunately, its results were not consistent with TCP's which may point to the theory that a different set of network science measurements are required to analyze network events depending on the protocol.

For ICMP, density dropped after the start of the event and increased before the end of the event.

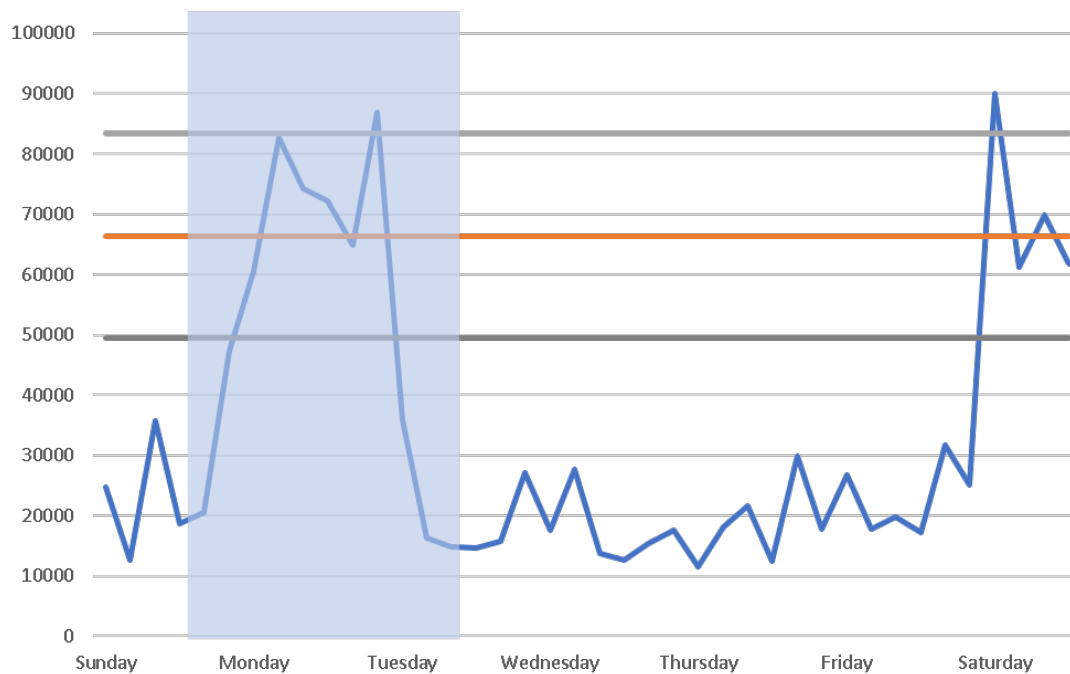


Figure 4.3: Time series of link count of ICMP connections during incident week; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, grey lines represent first level standard deviations of normal weeks

This indicates that immediately after the incident, more disconnected components of IPs were interacting within the network. The drop was probably caused because of the significant increase of new unique machines connecting to the network. The skyrocketing near the end may indicate that internally there were many errors or debug messages between machines within a majority of the network. Additionally, the number of unique machines connecting to the network decreased.

Fragmentation was low during the incident and rose at the same time the density rose. This indicates that internal communities were breaking apart because of the increased number of machines connecting to the network. The new unique machines connecting to the network had very sparse connections. The question is who are these machines? Are they third party network administrators? Possible attackers? Reserve backup machines? More analysis should be done on this phenomenon.

The link count and node count significantly increased. This indicates that traffic and the number of unique machines connecting increased significantly. This may indicate the outage. The time series of ICMP link count of the incident week is in Figure 4.3. Overall, the normal week was very cyclic so ICMP is probably a great protocol to monitor from a network science perspective to find anomalies. Table 4.11 shows the results of the Two-Sample KS Test on ICMP comparing the incident week and the normal weeks.

Table 4.4: Two-Sample Kolmogorov-Smirnov Test of ICMP comparisons of the Incident Week and a Normal Week and the Normal Weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	0.63571	1.286e-07	0.21429	0.2898
Weighted Density	0.22619	0.2453	0.2619	0.1121
Fragmentation	0.40357	0.001566	0.5476	3.935e-06
Edge Count	0.65833	6.847e-09	0.2857	0.0645
Edge Sum	0.88095	<2.2e-16	0.7381	2.213e-11
Node Count	0.68214	1.522e-09	0.2619	0.1123
Clique Count	0.05119	1	0.1191	0.9272
Clustering Coefficient	0.075	0.9998	0.1191	0.9272

Table 4.5: Two-Sample Kolmogorov-Smirnov Test of ESP comparisons of the Incident Week and a Normal Week and the Normal Weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	0.2798	0.0810	0.4523	0.0004
Weighted Density	0.1393	0.8216	0.95238	<2.2e-16
Fragmentation	0.16548	0.6288	0.3571	0.0094
Edge Count	0.2988	0.0515	0.3571	0.0094
Edge Sum	1	<2.2e-16	0.9762	<2.2e-16
Node Count	0.2083	0.3362	0.4762	0.0001
Clique Count	1.1102e-16	1	0	1
Clustering Coefficient	1.1102e-16	1	0	1

It was hypothesized that ICMP would have significant differences between the incident and the normal weeks and that the normal weeks should show consistent behavior. Both hypothesis were supported by the results of network measurements, density, edge count, and node count. Fragmentation and Edge Sum results supported the incident and normal week comparison and Weighted Density, Clique Count, and Clustering Coefficient supported the normal week comparison.

Both, ESP and GRE, did not show visible effects from the incident. ESP and GRE mostly consist of automatic traffic while using secure connections and VPNs. The data was very cyclic with the work day for all measures. The time series of ESP total-degree centrality of the incident week and normal week are in Figures 4.4 and 4.5 respectively. Table 4.5 shows the result of the KS test comparing the incident week with the normal week and both normal weeks for ESP.

For ESP, it was hypothesized that all weeks should show consistent behavior among each other. Both hypotheses were supported for Clique Count and Clustering Coefficient. The hypothesis of density, weighted density, fragmentation, edge count, and node count for the comparison between the incident week and normal week. Interestingly, there were significant differences between both normal weeks for ESP. It is unclear why these results could have occurred, however it may have to do with interference from an unknown network event. Table 4.6 shows the results of the KS test for GRE.

Like ESP, it was hypothesized that all weeks should show consistent behavior among each other for

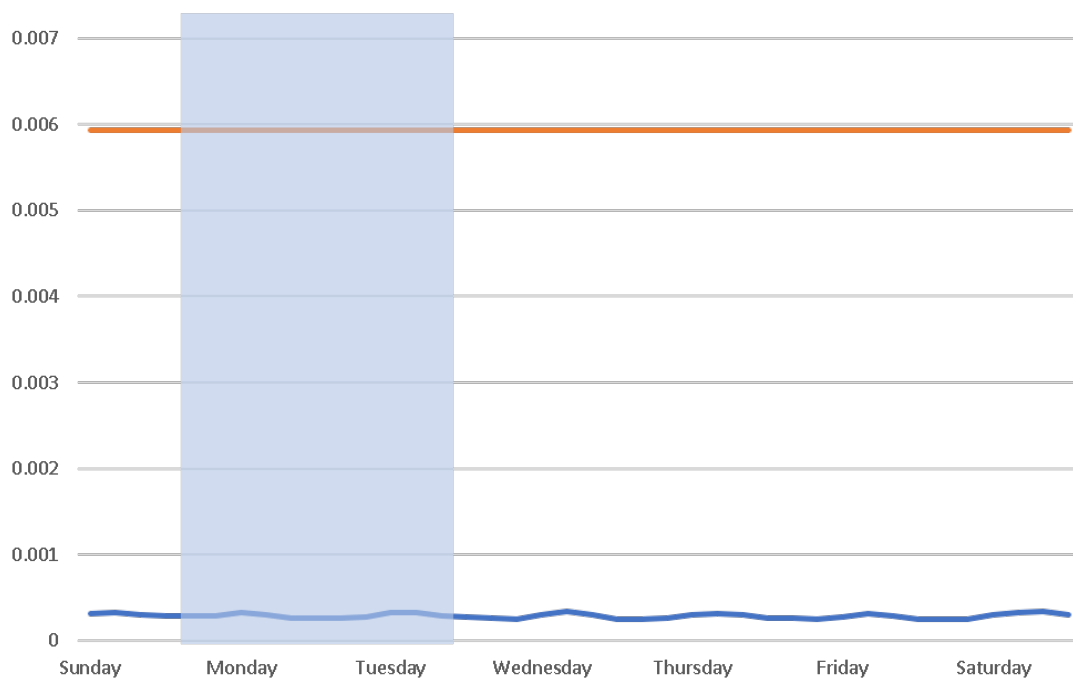


Figure 4.4: Time series of total-degree centrality of ESP connections during incident week; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, standard deviations were not included because the range was too large

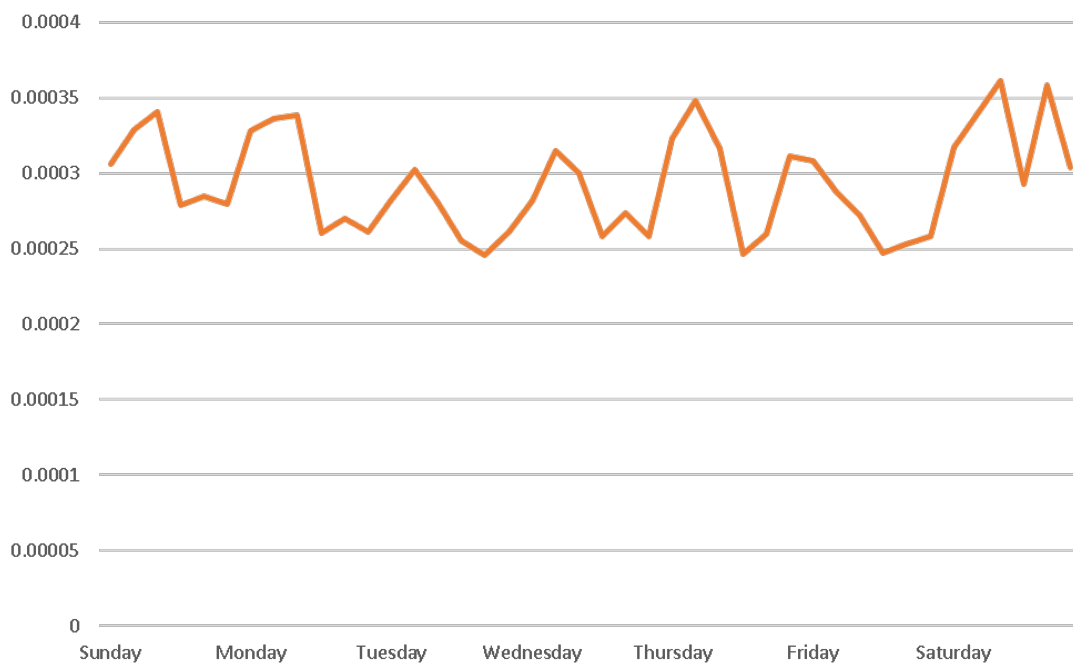


Figure 4.5: Time series of total-degree centrality of ESP connections during a normal week

Table 4.6: Two-Sample Kolmogorov-Smirnov Test of GRE comparisons of the Incident Week and a Normal Week and the Normal Weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	1	<2.2e-16	1	<2.2e-16
Weighted Density	0.8333	9.278e-12	0.7381	2.312e-10
Fragmentation	0.5378	1.613e-05	0.6905	7.455e-10
Edge Count	1	<2.2e-16	1	<2.2e-16
Edge Sum	1	<2.2e-16	0.54762	3.935e-06
Node Count	1	<2.2e-16	1	<2.2e-16
Clique Count	4.1633e-17	1	0	1
Clustering Coefficient	4.1633e-17	1	0	1

GRE. Overall, GRE had similar results to ESP. Only clique count and clustering coefficient supported both hypothesis. However, unlike ESP, no other hypotheses were supported individually comparing incident week with normal week or comparisons among both normal weeks. Like ESP, it is unclear why there are significant differences among weeks for GRE.

Overall, the time series generated from the netflow traffic indicated very different behaviors between the protocols. Automatic protocols like ESP and GRE showed persistent cyclic behavior regardless of infrastructure stress. TCP and ICMP behavior showed very clear behavioral changes during the event. However, the effect on the event on UDP traffic was difficult to define due to more erratic measures on both time periods.

Top 20% Total Degree Centrality Analysis Results

The first technique to reduce the size of the dataset did not work as expected. It seemed to highlight the cyclic behavior and there were no differences between the normal week and the incident week.

For TCP, density, link count, node count, average total-degree, and clique count exhibit more cyclic behavior coinciding with the normal work day schedule. Fragmentation and the clustering coefficient had significant changes.

After removing the bottom 80% difference in total-degree centrality, the fragmentation dropped during the incident. This might be a result of removing nodes of low total-degree centrality. Otherwise, the measurement indicates that during the incident there was more frequent communication between individuals within the network that normally do not communicate with each other.

Even though fragmentation dropped, the clustering coefficient did not increase as much as it did during the incident. Without reducing the data, the clustering coefficient increased from 5E-8 to 4.5E-7 during the incident. This was an increase of about 9 times. On the contrary, the clustering coefficient only increased from 1E-4 to 3E-4, an increase by 3 times after reducing the data. This may mean the web traffic during incident may have been sparser than originally expected or that the difference in total-degree may

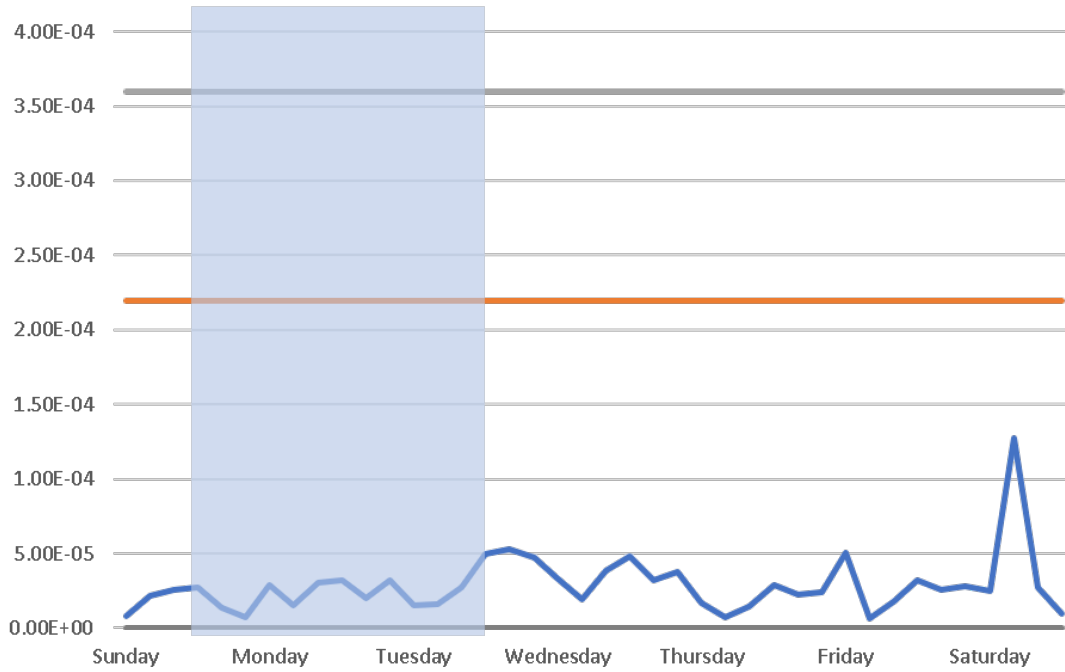


Figure 4.6: Time series of clustering coefficient of TCP connections during incident week after removing bottom 80% total-degree difference; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, grey lines represent first level standard deviations of normal weeks

Table 4.7: Two-Sample Kolmogorov-Smirnov Test of the Top 20% Total-degree comparisons of TCP connections between the Incident Week and a Normal Week and between the Normal Weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	0.1679	0.6107	0.2619	0.1121
Weighted Density	0.7369	4.341e-10	0.2381	0.1848
Fragmentation	0.9762	<2.2e-16	0.3095	0.0352
Edge Count	0.2155	0.2436	0.5238	1.279e-05
Edge Sum	0.625	4.807e-08	0.3095	0.0358
Node Count	0.1369	0.8372	0.7381	2.312e-10
Clique Count	0.3857	0.0045	0.3810	0.0045
Clustering Coefficient	0.975	<2.2e-16	0.3571	0.0094

not have captured an accurate representation of the IPs affected by the incident. Figures 4.6 and 4.7 show the clustering coefficient of TCP of the event week and the normal week. Table 4.7 shows the results of the KS test comparing the incident week with a normal week and normal weeks together of the top 20% different total-degree central hosts between the service outage and a normal time period for TCP traffic.

It was hypothesized that reducing the data would highlight a greater effect by narrowing down the network to hosts whose behavior changed during the service outage. Therefore, the incident week should show significantly different behavior and the normal weeks should show consistent behavior. However, the expected results did not take place. For the comparison between the incident week and the normal week, fragmentation, edge sum, clique count, and clustering coefficient showed significant differences.

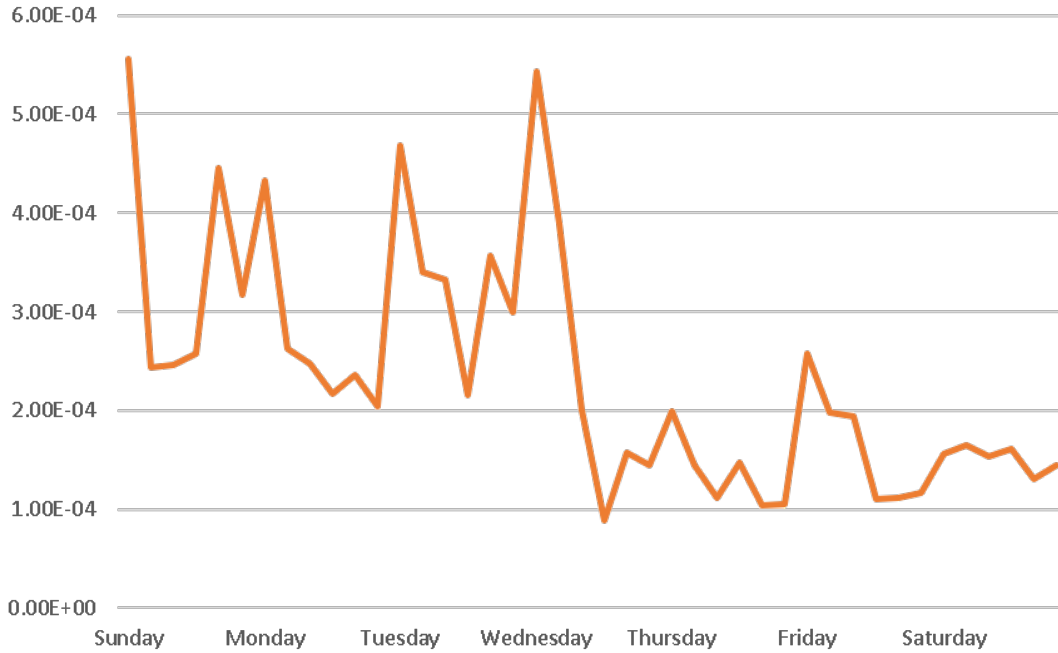


Figure 4.7: Time series of clustering coefficient of TCP connections during normal week after removing bottom 80% total-degree difference

For the comparison between normal weeks, density and weighted density were the only measures that were consistent.

It is unclear why the data reduction resulted in this way. The greatest possible explanation is that the method of highlighted hosts with different behavior was not correct. This is probably because total-degree centrality does not take link weights into account only the unique hosts the host is connected too. As a result, unique links will most change with the hosts that connect to the greatest range of external hosts. For these networks, these were the servers. This explains why networks extracted using this approach showed increased cyclic behavior. The proper method to gather the highlighted change would be to find the greatest difference of link weights for each node.

For UDP, density, link count, node count, clustering coefficient, and clique count were almost identical. Fragmentation and average total-degree centrality were the only measurements with significant differences between the weeks.

Fragmentation and total-degree centrality had lower degrees of effect when compared to the raw data. The lower degrees may mean that the community extracted from total-degree was more cohesive but may be less interactive than the raw dataset. Additionally, both measurements exhibited cyclic behavior during the incident that was not seen in the normal weeks. This cyclic behavior means that the total-degree centrality difference sampling strategy selected actual work machines at the company's campus or selected

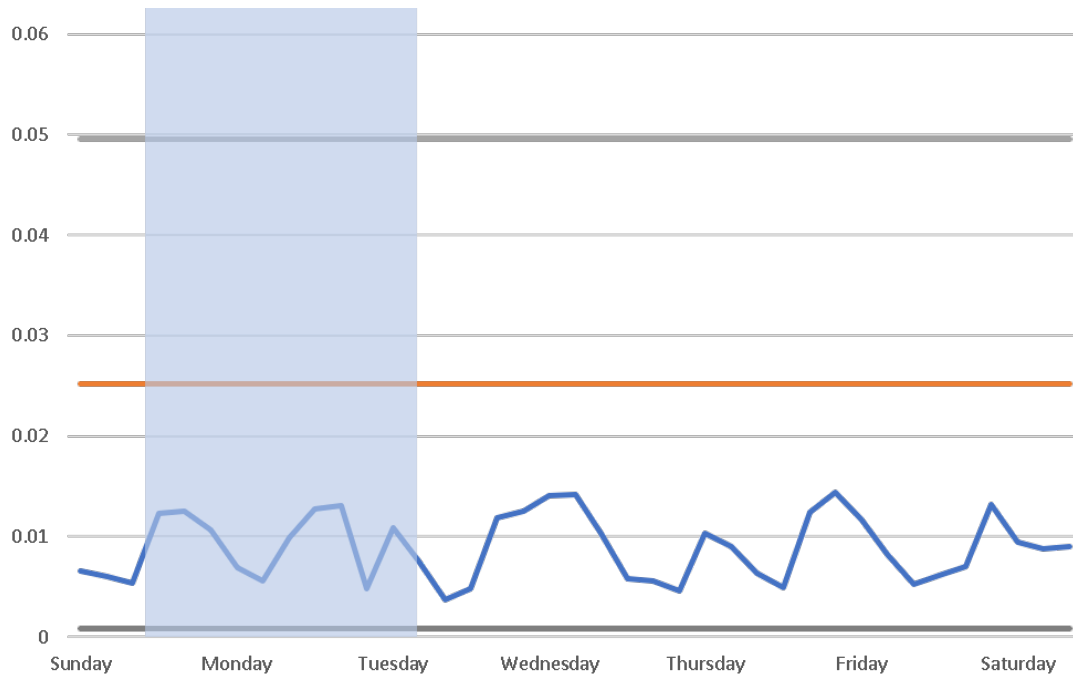


Figure 4.8: Time series of fragmentation of UDP connections during incident week after removing bottom 80% total-degree difference; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, grey lines represent first level standard deviations of normal weeks

Table 4.8: Two-Sample Kolmogorov-Smirnov Test of the Top 20% Total-degree comparisons of UDP connections between the Incident Week and a Normal Week and between the Normal Weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	0.2106	0.331	0.3095	0.0358
Weighted Density	0.6978	5.601e-09	0.4286	0.0009
Fragmentation	0.1685	0.5383	1	<2.2e-16
Edge Count	0.1612	0.5994	0.2857	0.0645
Edge Sum	0.9762	<2.2e-16	0.2143	0.2924
Node Count	0.1447	0.7164	0.3095	0.0358
Clique Count	0.1319	0.8734	0.2857	0.0649
Clustering Coefficient	0.1319	0.8734	0.4048	0.0021

autonomic traffic centered around the company's time zone. Figures 4.8 and 4.9 show fragmentation of UDP of the event week and the normal week. Table 4.8 shows the results of the KS test comparing the incident week with a normal week and normal weeks together of the top 20% different total-degree central hosts between the service outage and a normal time period for UDP traffic.

It was hypothesized that all weeks should show consistent behavior for UDP. The hypothesis was validated for Edge Count and Clique Count. The hypothesis of the comparison between incident week and normal week was validated for density, node count, and clustering coefficient. The hypothesis of the comparison between both normal weeks was validated by edge sum. Interestingly, like many of the previous results, the normal week comparison showed significant differences between measurements

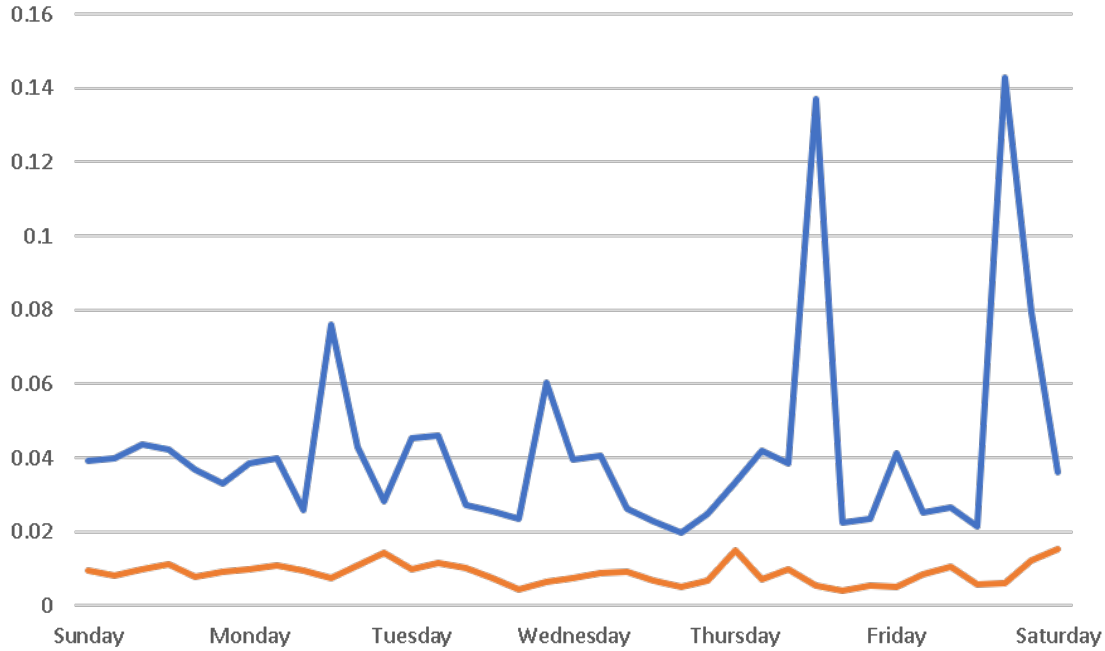


Figure 4.9: Time series of fragmentation of UDP connections during normal weeks after removing bottom 80% total-degree difference

between normal weeks. These results shared the same limitations as the TCP top 20% difference in total-degree.

Increased Egress and Decreased Ingress Rate Analysis Results

The second technique to reduce the size of the dataset did not work as expected. It seemed to have highlight the cyclic behavior and there were no differences between the normal week and the incident week.

For TCP, much of the noise for all measurements was removed and as a result, changes that were seen in the normal set were intensified. Additionally, cyclic behavior with the normal work day was found in density, link count, node count, and total-degree centralization. Figures 4.10 and 4.11 show the density of TCP of the event week illustrating the highlighted cyclic behavior. The script to extract IPs of high sender/receiver rate on the normal week has been running for more than a week and still has not complete. Table 4.9 shows the results of the KS-test for the top 20% increased egress and decreased ingress hosts over TCP.

It was hypothesized that the incident week would have significant differences when compared to the normal week and the normal weeks should show no significant differences. Only edge count and clique count validated both hypothesis. Density, weighted density, fragmentation, edge sum, node count,

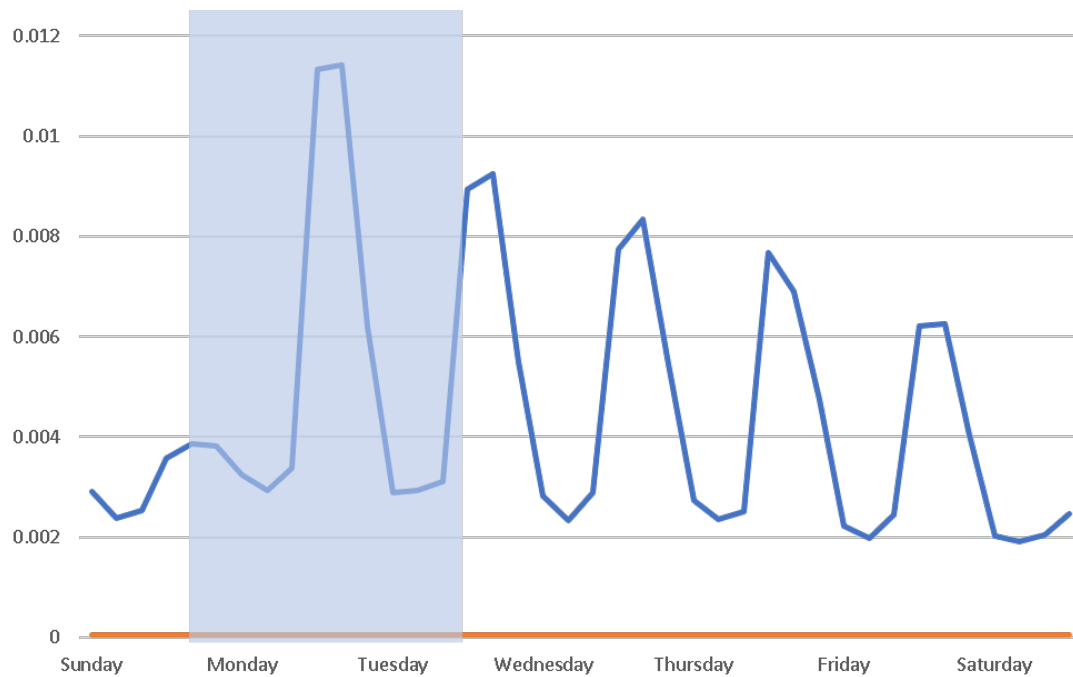


Figure 4.10: Time series of density of TCP connections during event week after selecting the top 20% increased sender and decreased receivers; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, standard deviations were not included because the values were too low

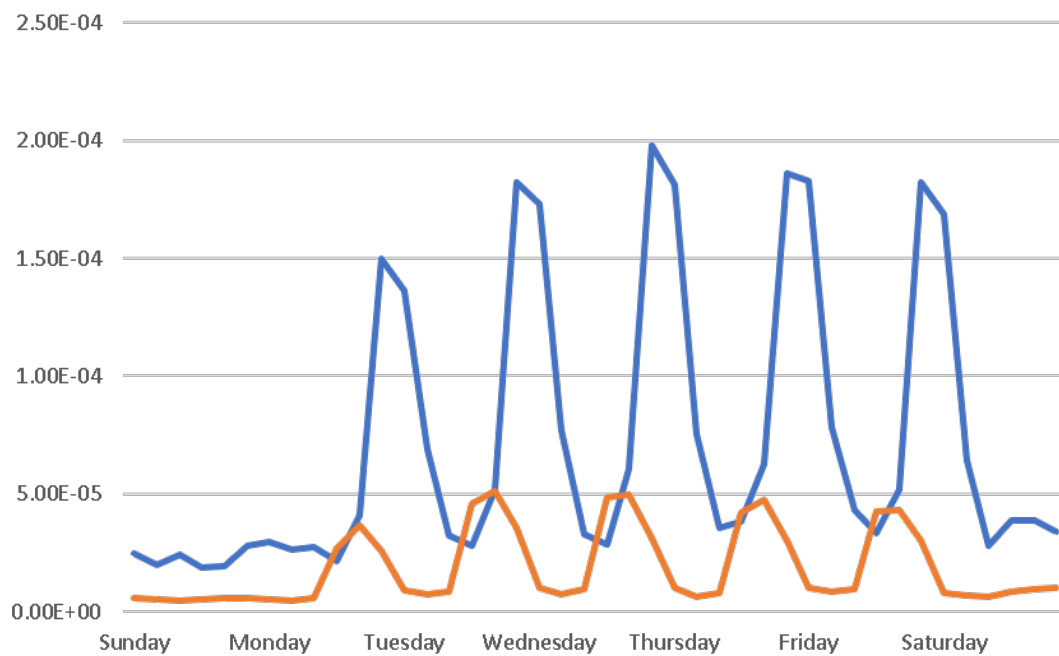


Figure 4.11: Time series of density of TCP connections during normal weeks after selecting the top 20% increased sender and decreased receivers

Table 4.9: Two-Sample Kolmogorov-Smirnov Test of the Top 20% Increased Egress and Decreased Ingress Hosts comparing TCP connections between the Incident Week and a Normal Week and between the Normal Weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	1	<2.2e-16	0.6429	5.793e-08
Weighted Density	1	<2.2e-16	0.7143	9.879e-10
Fragmentation	1	<2.2e-16	0.8810	<2.2e-16
Edge Count	0.6429	1.064e-08	0.2143	0.2924
Edge Sum	0.5714	7.828e-07	0.5714	1.133e-06
Node Count	1	<2.2e-16	1	<2.2e-16
Clique Count	0.5893	1.323e-06	0.1905	0.4313
Clustering Coefficient	0.7560	1.354e-10	0.3571	0.0094

Table 4.10: Two-Sample Kolmogorov-Smirnov Test of the Top 20% Increased Egress and Decreased Ingress Hosts comparing UDP connections between the Incident Week and a Normal Week and between the Normal Weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	0.4451	0.0004	1	<2.2e-16
Weighted Density	0.4359	0.0009	0.6191	2.046e-07
Fragmentation	0.2180	0.2449	1	<2.2e-16
Edge Count	0.4689	0.0003	0.5238	1.279e-05
Edge Sum	0.8480	3.331e-15	0.4762	0.0001
Node Count	0.4689	0.0001	0.9286	<2.2e-16
Clique Count	0.6429	1.102e-07	0.3333	0.0188
Clustering Coefficient	0.6191	3.714e-07	0.3333	0.0188

and clustering coefficient validated the incident hypothesis. Overall, the reduction had great results at highlighting the differences between the incident week and a normal week, however it performed poorly at finding consistent behavior among the normal weeks.

For UDP, the most change happened in grouping measurements such as fragmentation, clustering coefficient, and clique count. All measures showed an increase in fragmented smaller groups during the incident. This indicates that a greater variety of IPs were interacting in the network during the incident. This can lead to the theory that internet background radiation traffic (ongoing activity on the internet not related to any business mission) increased during the incident which may imply that malicious users were aware of the incident and conducted more active operations upon the network. An interesting change in density showed that work day cycles were broken up. This might be because of the greater proportion of nodes from other countries that run in different work day cycles. Figure 4.12 and 4.13 shows the fragmentation of UDP of the event week and the normal week.

For UDP, it was hypothesized that behavior should be consistent among all weeks regardless of a flash crowd. No measurement supported the hypothesis. It is very difficult to determine why this behavior was done and it is unclear what hosts were really extracted using this method because this method shares the same issues of the difference in total-degree reduction method. This method showed no consistency

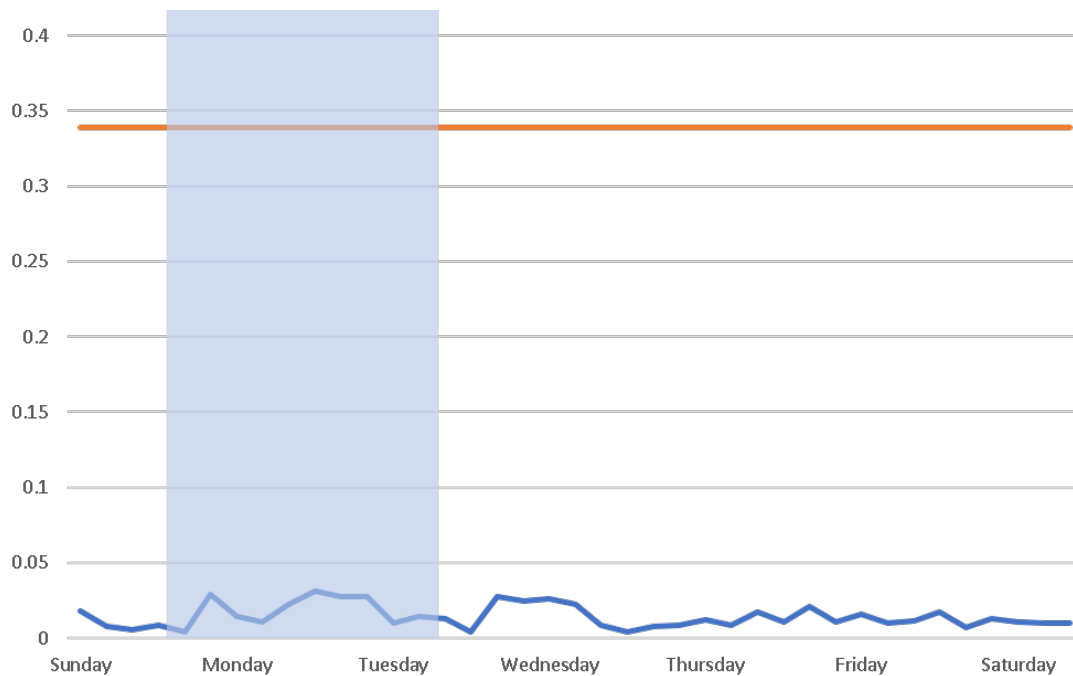


Figure 4.12: Time series of fragmentation of UDP connections during event week after selecting the top 20% increased sender and decreased receivers; shaded region indicates the disclosure of the event, orange line represents the mean of normal weeks, standard deviations were not included because the range was too large

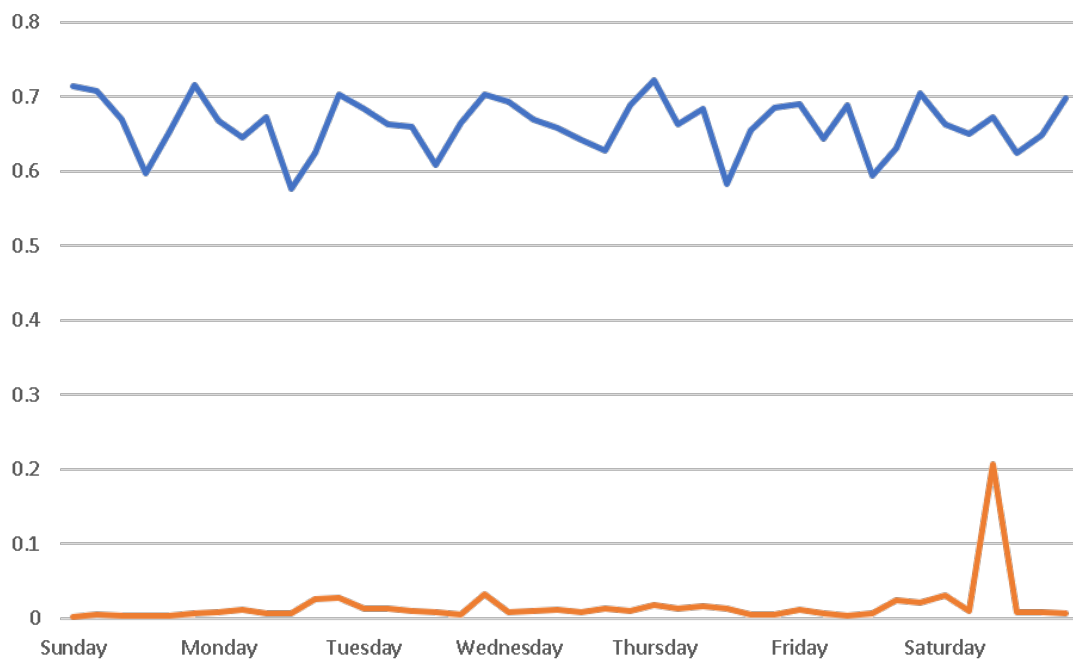


Figure 4.13: Time series of fragmentation of UDP connections during normal weeks after selecting the top 20% increased sender and decreased receivers

among normal weeks as well.

4.1.3 Effects of the Flash Crowd on Functional Groups

Overview

Because of the inconclusive results of the flash crowd, the analysis was repeated but only upon select internal groups within the corporation. The corporation is made up of a variety of offices each requiring different technological abilities and higher levels of access in managing infrastructure. It was hypothesized that these groups would show differences in behavior during normal weeks and the flash crowd weeks. Examining the changes in behaviors of offices would help a corporation understand the appearance of normal behavior and abnormal behavior caused by an incident similar to a flash crowd. By understanding these differences in behaviors, it becomes more feasible to create models that can automatically detect deviations from the norm.

Results

The only division network that was created was the server network. It was hypothesized that the servers within the company would have very cyclic behavior with the work day but should maintain a consistent rate of activity. Additionally, the flash crowd was expected to not have that much of an effect on the network because server activity is automatic, and it should only really affect a few select servers. The presence of other servers will normalize the effect and obfuscate smaller changes from the event. Figure 4.14 shows the edge sum during the incident and normal weeks.

The visual analysis supported the hypothesis by showing cyclic behavior that peaks a little after noon and a consistent baseline during nights and weekends. However, it deviated from the hypothesis in the degree of changes and baseline. Like the results on the raw analysis, the normal weeks had significantly greater values than the incident week. This was most likely an issue with the SiLK installation. However, this does not discount the changes in cyclic behavior for normal week 2. Each peak increased by more than 4 times its baseline value for normal week 2 and the rest of the weeks doubled on each peak. It is unclear why normal week 2 resulted in this way because no anomalous behavior was reported during the time.

None of the results validated the hypothesis. All of the chart measures showed similar results as Figure 4.14 and there were significant differences between all 3 of the weeks. There should not have been significant differences between all the weeks because of the nature of the machines selected. Research on more data must be conducted to explain this phenomenon.

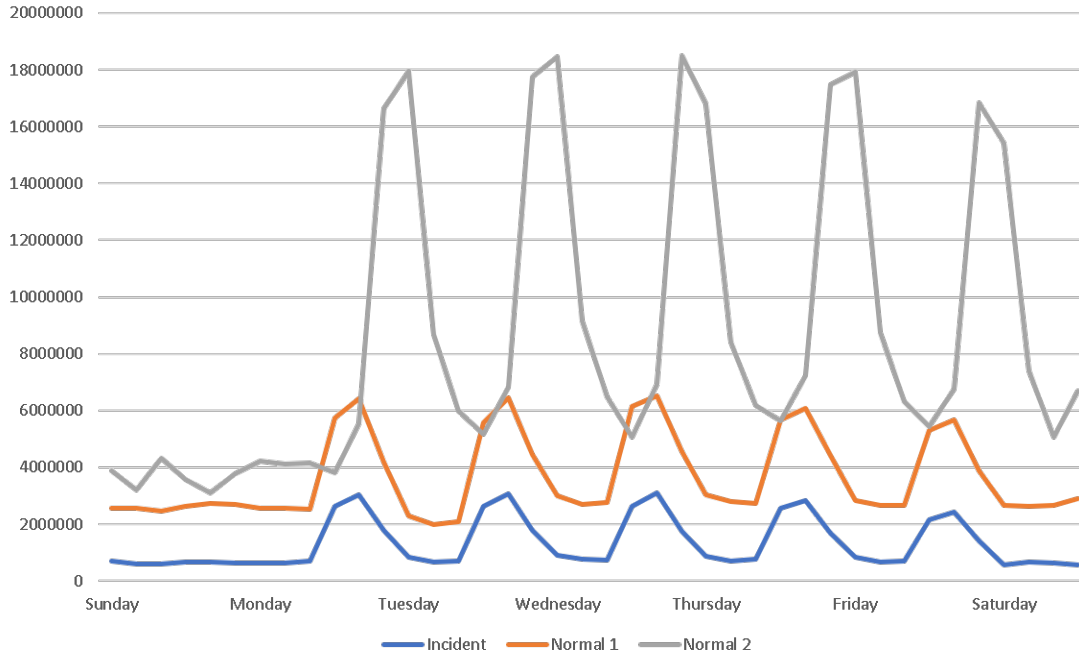


Figure 4.14: Time series of edge sum of server TCP connections within the company

Table 4.11: Two-Sample Kolmogorov-Smirnov Test of server TCP connections between the Incident Week and a Normal Week and between Normal Weeks

Measurement	Incident Difference	Incident P-Value	Normal Difference	Normal P-Value
Density	0.6512	3.726E-08	0.9767	<2.2e-16
Weighted Density	0.9535	<2.2e-16	0.9767	<2.2e-16
Fragmentation	0.3466	0.0129	0.5349	9.082e-06
Edge Count	0.7850	1.165e-11	0.9535	<2.2e-16
Edge Sum	0.9767	<2.2e-16	0.6279	8.674e-08
Node Count	0.9767	<2.2e-16	0.9767	<2.2e-16
Clique Count	0.9767	<2.2e-16	0.3954	0.0024
Clustering Coefficient	0.9036	2.665e-15	0.6977	1.626e-09

4.1.4 Network Science Measurement Evaluation

The KS-tests conducted in the previous sections offer a look on the effectiveness of each network science measurement assuming the hypothesis accurately represented the expected behavior. The following section will examine the results of the KS-Test for each network science measurement and analyze the effectiveness of each network science measurement and offer possible explanations for the results. Table 4.12 shows the aggregated results for all of the KS-test completed in the previous experiments.

Overall, clique count and edge count were the network science measures that best validated the assumptions made about the data. It is important to note that a limitation of this approach is that detailed information about the enterprise network was inaccessible. Topology and event information during these

Table 4.12: Summary of the results of the two sample KS-test for all experiments

Measurement	Raw Data Analysis	Top 20% Total-degree	Top 20% Egress/Ingress	Total
Density	2/5	0/2	0/2	2/9
Weighted Density	0/5	0/2	0/2	0/9
Fragmentation	1/5	0/2	0/2	1/9
Edge Count	2/5	1/2	1/2	4/9
Edge Sum	0/5	0/2	0/2	0/9
Node Count	2/5	0/2	0/2	2/9
Clique Count	3/5	1/2	1/2	5/9
Clustering Coefficient	2/5	0/2	0/2	2/9

Table 4.13: Daily two sample KS-test for normal weeks

Measurement	UDP	ICMP	ESP	GRE
Density	6/7	3/7	0/7	0/6
Weighted Density	0/7	4/7	0/7	0/6
Fragmentation	4/7	1/7	2/7	0/6
Edge Count	5/7	2/7	0/7	0/6
Edge Sum	1/7	0/7	0/7	2/6
Node Count	6/7	2/7	0/7	0/6
Clique Count	4/7	7/7	7/7	6/6
Clustering Coefficient	2/7	7/7	7/7	6/6

time periods may lead to results contrary to the assumptions made. Regardless, these results are significant because there were a standard set of measurements that followed the hypothesis. Moreover, it still showed consistency despite the limitations of the data reduction methods.

This procedure was also limited because of the lack of data points used in comparing network science measurements. Only three weeks were used so there was a limited amount to compare. As a result, the following table was created comparing how many times the hypothesis was validated daily. These networks were reduced from 4-hour bins to 1-hour bins. As a result, each daily time series had 24 points and 7 time series plots were compared for each combination of available weeks. The total amount of datapoints these KS-test have is 105 not counting the degree reduction techniques. These were not included because of the limitations they had.

To combat this limitation, the network was split from 4-hour bins into 1-hour bins. Time series chart measures were calculated for each day so 24 metanetworks were used for each test. Then, each day of the first normal week was compared with the corresponding day of the second normal week. Table 4.13 shows the results these tests. TCP was not included because of the amount of time required to calculate measurements in 1-hour bins.

Overall, the results were very consistent with the week-long time series comparisons. UDP performed very well at identifying consistent behavior, ICMP performed decently well, and ESP and GRE performed poorly though they showed consistent patterns. For UDP, non-weighted measurements performed the

best per day, ICMP showed better performance in structural measurements, and all performed well with clustering measurements. Interestingly, weekends tended to deviate the most between both weeks. This makes sense because of cyclic work day patterns. ESP and GRE showed consisted behavior in terms of cyclic structure from the visual analysis, however performed the worst with KS Test. It can be included that weighting should be adjusted in KS test or another statistical test should be used.

4.1.5 AbuseIPDB Distribution

Overview

During the analysis, an interesting phenomenon was discovered. A large number of IPs from non-US countries had a high network presence even though there is no reason for these hosts to be communicating with the network. Many of these IPs have been found on AbuseIPDB, a database where webmasters and system administrators can report and query IP addresses associated with malicious activity. Some of the high total-degree external IPs had over 1,000 reports for port scanning on various rarely used ports and other reconnaissance activity. Other IPs have been reported for being a part of distributed denial of service (DDoS) attacks. The greedy reconnaissance behavior of some of these bot machines can be attributed to nation states crawling the whole internet looking for potential vulnerable targets[61].

Previous research coined the term Internet Background Radiation for these types of interactions[52]. These types of interactions make up a significant portion of traffic within the internet. Previous research has attempted to characterize and detect this traffic behavior[52][1][36].

For this study, dynamic ego networks were created using the most reported IPs for both UDP and TCP. Ego networks are currently used as a sampling strategy to eliminate noise within large networks[16]. Then network measurements were calculated and their time series were compared to those with the whole network. Only second-degree neighbors were used because these machines typically had a very high total-degree and a higher degree would create a noisier sample.

Results

AbuseIPDB revealed many interesting aspects of the data. With the addition of reporting data, AbuseIPDB used WhoIs queries to get information about the organization, ISP, hostname, country, and city of the IP address. After the dynamic network analysis, distributions were calculated on the country and organization of the IP addresses both before and after pruning. These distributions show what kind of IP addresses had a high difference in total-degree or high sending rate and low receiving rate during the incident.

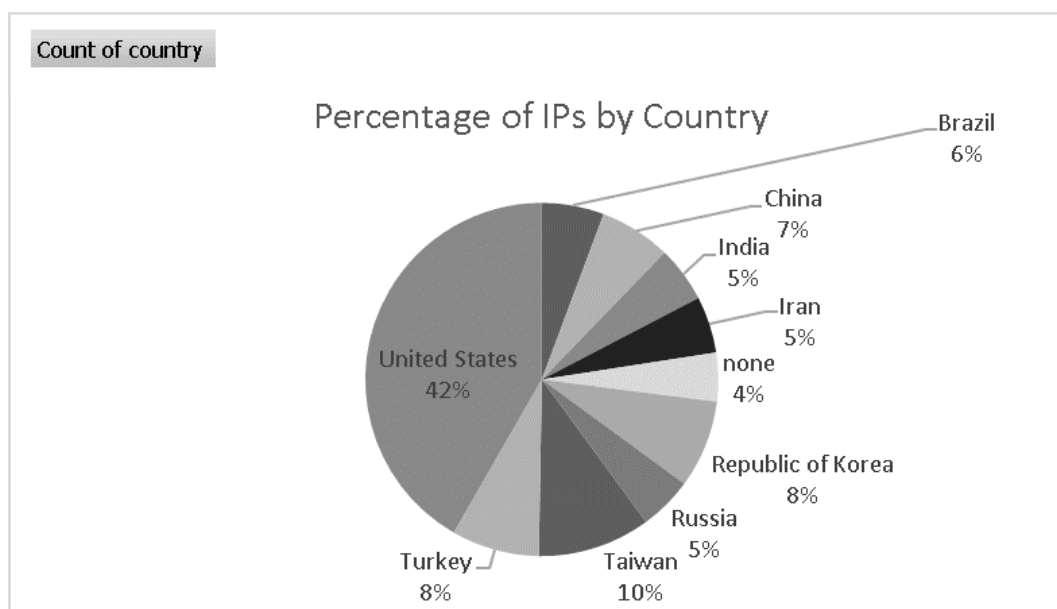


Figure 4.15: Distribution of Countries for TCP IPs

The distribution of IPs by country over TCP without filtering is below in Figure 4.15. It is interesting because more than 50% IPs within the company network are from foreign countries even though the company should only have to conduct business with parties in the United States. Some possible reasons for this can be because of varying views of what is ethical use of the internet, organized crime groups or malicious state actors conducting reconnaissance, or non-native groups moving through foreign countries to complicate attribution.

The distribution of IPs by country over UDP is in Figure 4.16. There is a significantly greater distribution of IPs within the United States using UDP on the network. This makes sense because of the more autonomic nature of UDP traffic. Additionally, it is good that the network has a reasonably low interaction from other countries because that may indicate data exfiltration attempts.

The country distribution of the top 10% difference in total-degree centrality over TCP and UDP after the event is in Figure 4.17 and 4.18 respectively. While TCP showed an increase in United States traffic, UDP showed a decrease. Though TCP's results may be a good sign, UDP's results is a bad sign and can indicate foreign actors having a very high total-degree within the network. This indicates that the foreign machines are conducting greedy reconnaissance behavior.

The country distribution of the top 10% increased sending and decreased receiving rate over UDP is on Figure 4.19. The TCP analysis is still pending due to missing attribute data. This pruning strategy found the least amount of United States IPs and found a significant amount of unknown and Korean IPs. It is interesting that Russia and China were not higher in this list.

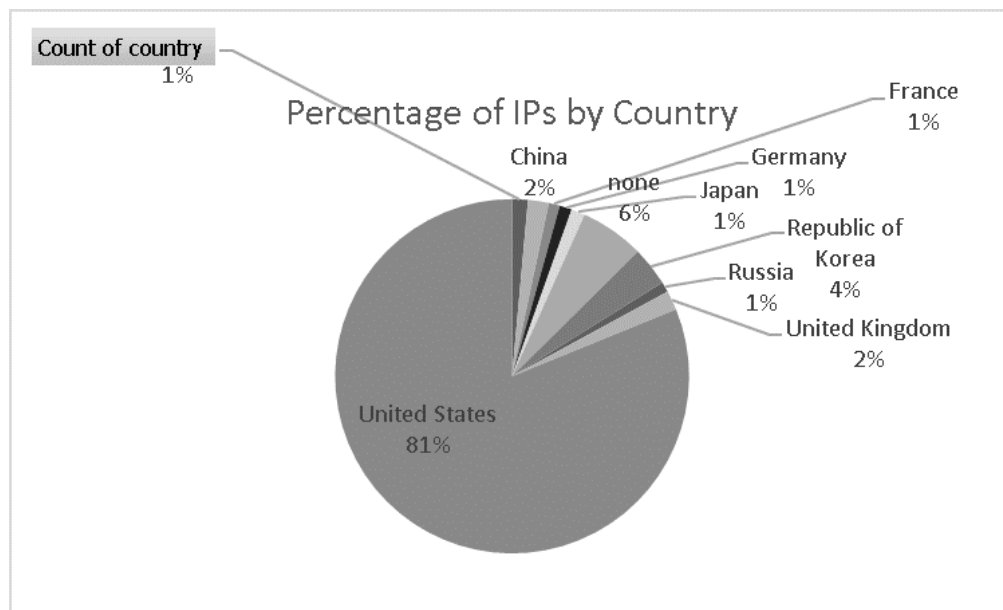


Figure 4.16: Distribution of Countries for UDP IPs

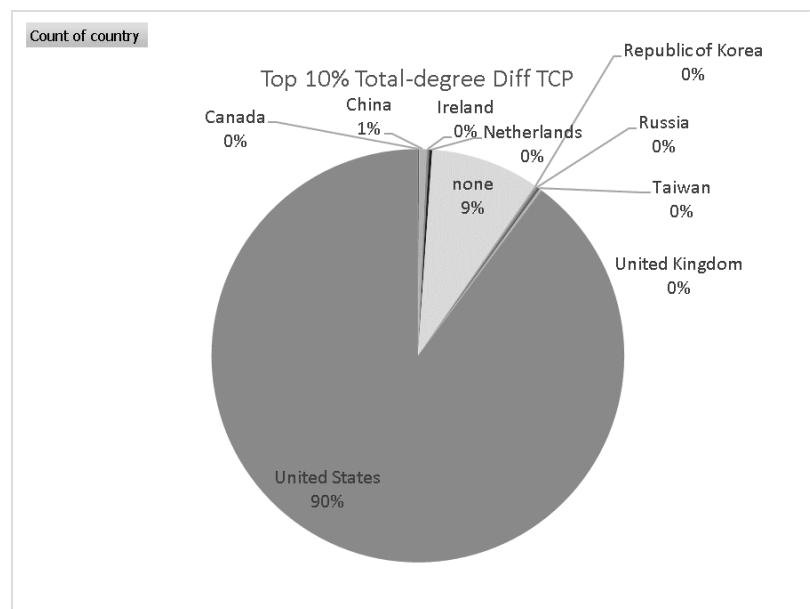


Figure 4.17: Distribution of Countries for TCP Top 10% difference in total-degree centrality IPs

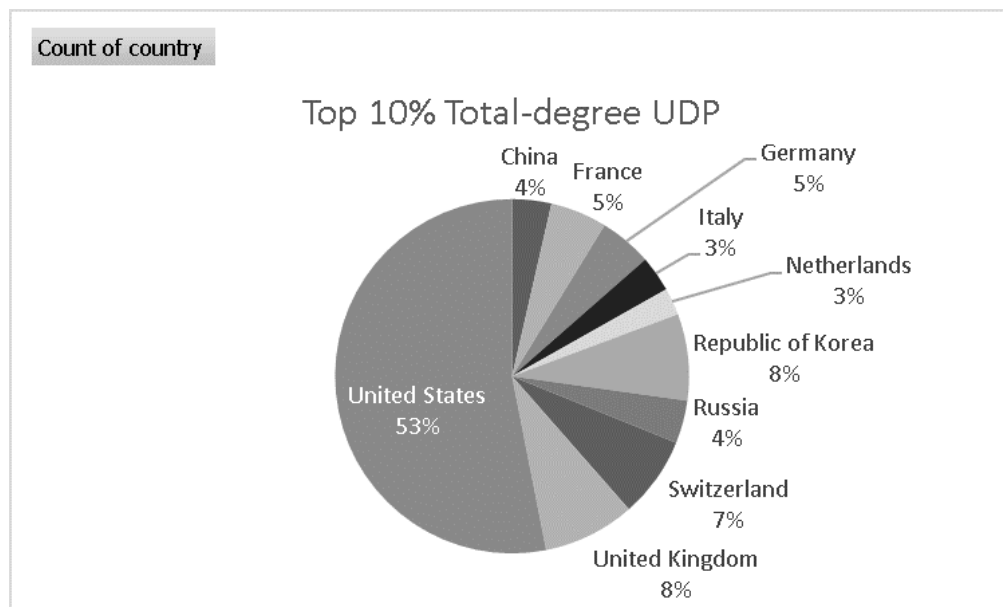


Figure 4.18: Distribution of Countries for UDP Top 10% difference in total-degree centrality IPs

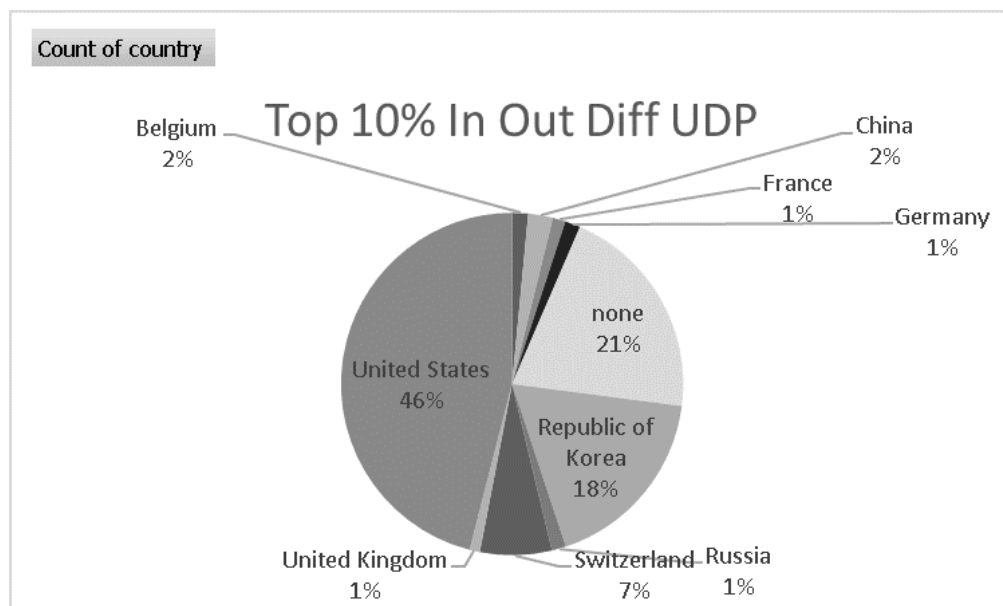


Figure 4.19: Distribution of Countries for UDP Top 10% increased sending and decreased receiving rate

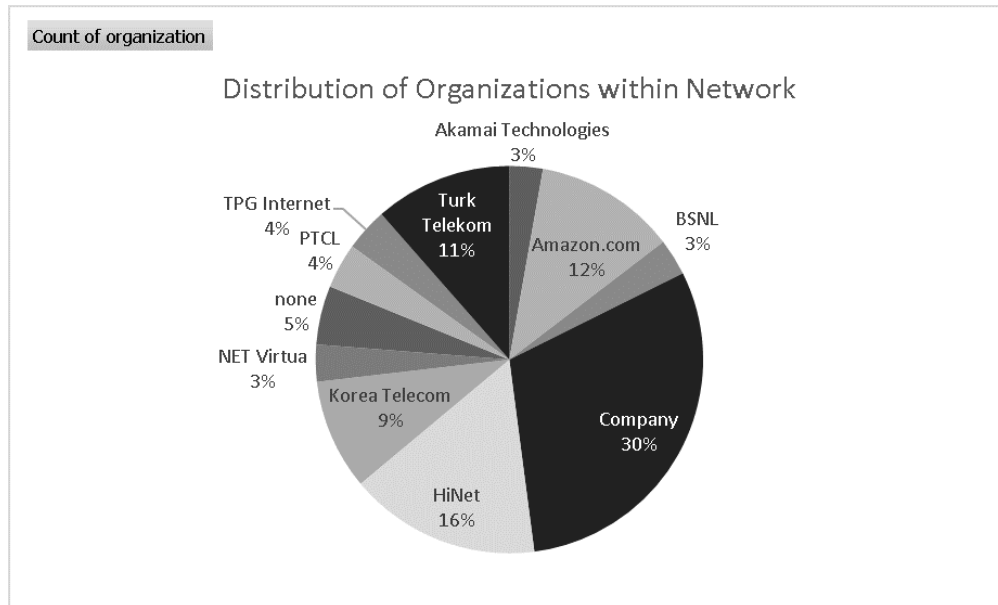


Figure 4.20: Distribution of Organizations over TCP

The organization distribution of IPs within the network using TCP is on Figure 4.20. The “None” organization represents either none or unknown. “Company” is the name of the company being analyzed. Many of the foreign organizations are ISPs who have no control over what activity is conducted on their domains. However, it is important to note that some countries have more control over their ISPs than others such as China compared to India.

4.1.6 AbuseIPDB Ego networks

Overview

AbuseIPDB provides a method to label potentially abusive hosts within the network. The question the experiment detailed in this section sought to answer is if there are any differences in behavior between hosts labelled abusive and hosts not labelled abusive. By proving abusive hosts exhibit different behavior detected through network science measurements than their non-abusive counterparts, creating a classifier for detecting abusive hosts becomes more feasible.

To complete this experiment, hosts were divided in bins separated by total-degree centrality. Total-degree centrality shows the activeness of a host within the network. This division helps focus on whether the abusive label results in differences in behavior and removes biases associated with the activeness of the host. The bins used for this experiment were the top 10% total-degree central hosts, 50 to 60% total-degree central hosts, and the bottom 10% total-degree central hosts.

From within these bins, 5 hosts were chosen that were non-abusive and 5 hosts were chosen that were labelled abusive. The non-abusive hosts were hosts from Fortune 500 Companies for the top 10% total-degree central hosts or random hosts that were not reported and not part of the organization for the other bins. The company of the IP address was found through reverse DNS lookup and publicized IP ranges of companies. Abusive hosts were random hosts with the top 10% highest number of reports on AbuseIPDB. The experiment aimed to see if the behavior defined by network science measurements had some correlation to the number of reports made on the host through AbuseIPDB. Internal hosts were not examined. This experiment has the limitation of not establishing a ground truth for non-abusive hosts. The Fortune 500 Companies method is more accurate than the 0 reports approach, however it was not feasible with the composition of hosts in the other categories. Another limitation is the small sample size used for each category. This experiment was meant to be a preliminary study to see if there were any differences in behavior and explanations for their differences. To test the differences for a large subset of labelled hosts, a classification method was created. The effectiveness of the classification was used to determine the differences in behavior between abusive and non-abusive hosts. The experiment is described in a later section.

Ego-networks were then created for each of these hosts where the ego-network represents the chosen host, its neighbors, and its neighbors' connections amongst each other. Network science measurements, link count, link sum, node count, density, and weighted density are then calculated for each of these networks and plotted over time. These measurements were chosen because the network size got reduced to such a small size where other measurements lost meaning.

Results

For, the top 10% total-degree central hosts, there were many differences in behavior between abusive and non-abusive Fortune 500 hosts. Companies showed very cyclic behavior that matched the work day of the corporation. This points to the assumption that many of these hosts were external tools used by the corporation such as cloud infrastructure, email providers, human resource tools, or content distribution networks. Figure 4.21 shows the cyclic behavior of Fortune 500 hosts within the network for edge sum. This can lead to the possibility of using alignment with company periodicity as a feature for detecting normal behavior amongst external host interaction.

On the other hand, the abusive hosts showed much more erratic behavior. These hosts typically showed an enormous amount of activity in burst. This caused the data to have a lot of holes which can hinder the ability to create classifiers for the abusive category. Because these hosts are both highly prevalent in the network and they have a large amount of reports on AbuseIPDB, it can be safely assumed that these

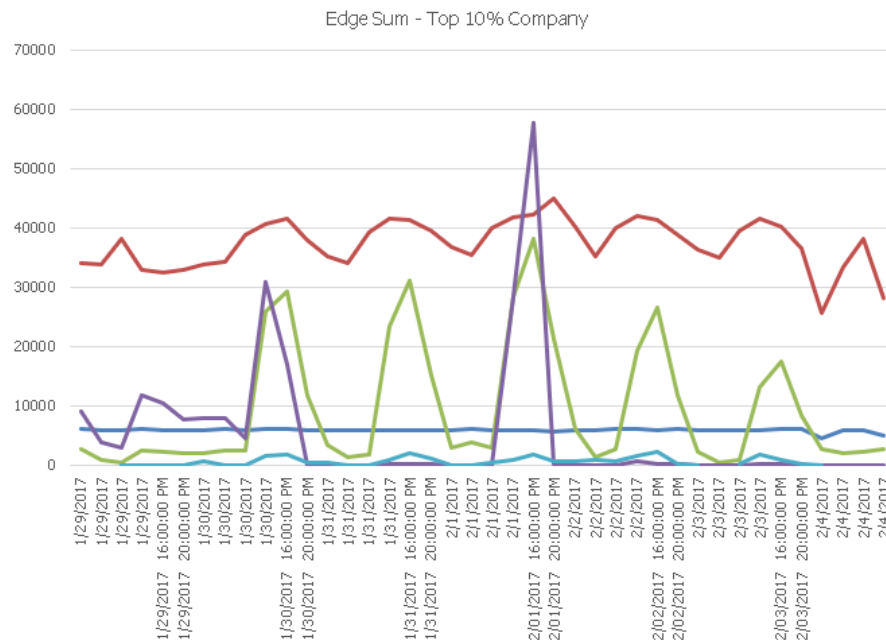


Figure 4.21: Time series of edge sum for 5 hosts in the Top 10% Total-degree centrality that are a part of Fortune 500 companies

hosts engage in malicious behavior. Figure 4.22 shows the erratic behavior of the abusive top 10% total-degree central hosts. However, the behavior of these hosts is not homogenous which also makes creating a classifier from this category of hosts more difficult.

The 50-60% total-degree central hosts had significantly different results than the top 10% total-degree central hosts. First, the data was much more complete. There was a low number of holes in both the non-abusive and abusive categories. Within the sample of 5, the non-abusive category was more tightly clustered, and the abusive category was more erratic. The abusive category had on average a higher number of unique connections than the non-abusive categories as well. Figures 4.23 and 4.24 respectively show the edge-count of non-abusive and abusive hosts within the network. Because not much is known about the non-abusive hosts, these findings may not mean as much. The IP addresses of these hosts were still from countries that have no affiliation with the corporation being examined. Analysis on a larger set of hosts using the classification method will help substantiate results.

The bottom 10% had difficult results to interpret. Both bins were fraught with holes in data and this was expected considering these hosts had some of the least involvement with the network. It was interesting that the hosts reported by AbuseIPDB had less holes within the network though. Figures 4.25 and 4.26 visualize the differences in node count between abusive and non-abusive hosts. The lesser number of holes may indicate that abusive hosts tend to have a lingering amount of activity within the

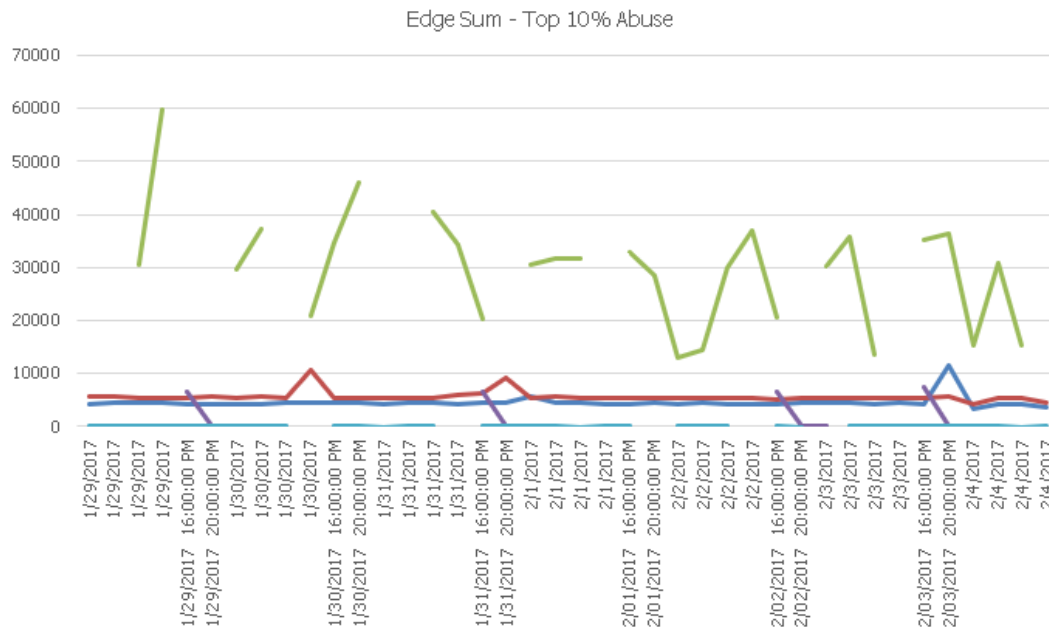


Figure 4.22: Time series of edge sum for 5 hosts labelled abusive in the Top 10% Total-degree centrality

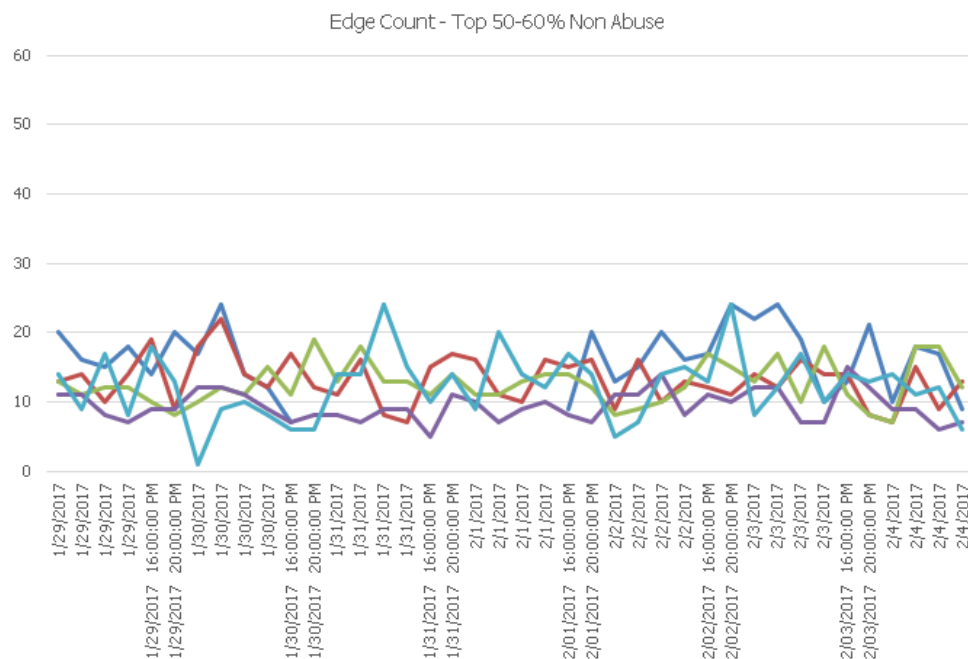


Figure 4.23: Time series of edge count for 5 hosts labelled non-abusive in the 50% to 60% Total-degree centrality range

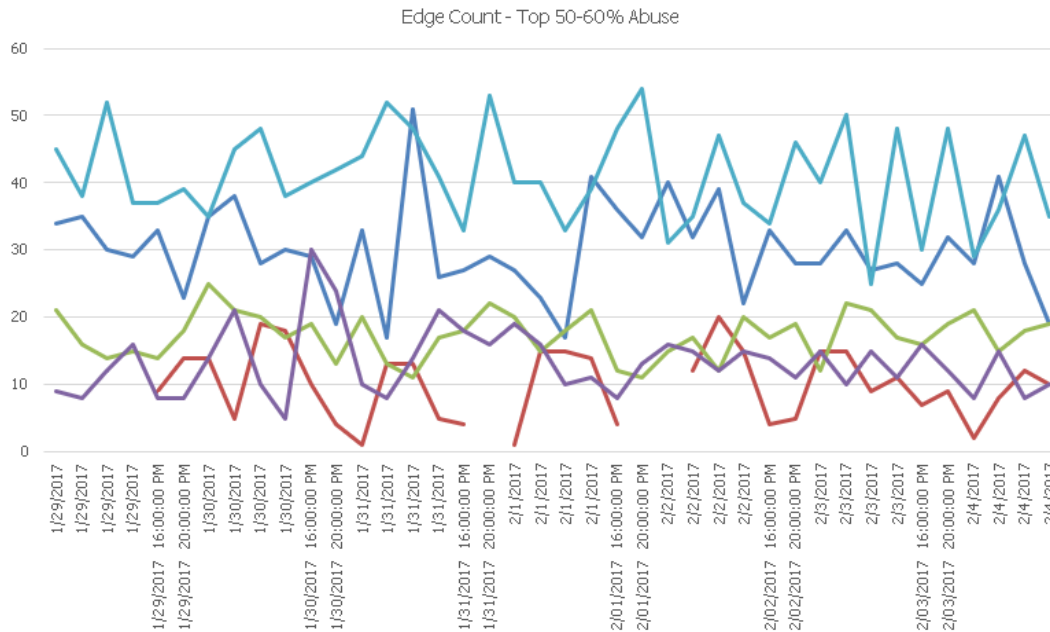


Figure 4.24: Time series of edge count for 5 hosts labelled abusive in the 50% to 60% Total-degree centrality range

network. This can resemble malicious activities like a port scan or exfiltration obfuscated by limiting network activity. However, the analysis with a larger subset of hosts would substantiate these claims.

Lastly, graphs of the standard deviations for each total-degree centrality bin were used to visualize the differences in variation amongst the host for each bin. The previous graphs were able to show some of these differences; however, this analysis was meant to quantify them in greater detail. Standard deviation proved to show a large amount of variation between abusive and non-abusive hosts, however it is difficult to prove with only a sample of 5 hosts.

Figure 4.27 shows 3 different categories and their standard deviations of edge counts among hosts. This figure shows an unused category called non-abuse which consisted of many local hosts within the network. This category was taken out because the netflow collector gathers outgoing connections from these hosts making the network structure significantly different when compared to external hosts. The Fortune 500 hosts exhibited cyclic behavior that match those of the internal hosts, however the frequencies are a lot lower than the internal hosts. This difference is again explained by the netflow collector and the outgoing information captured about internal hosts within the network. On the other hand, abusive hosts exhibit very burst-like behavior with high degree. The internal and abusive hosts showed a lot more variation in edge count measurements than the Fortune 500 hosts and this provides evidence that making classifiers based off corporate external tools is more feasible.

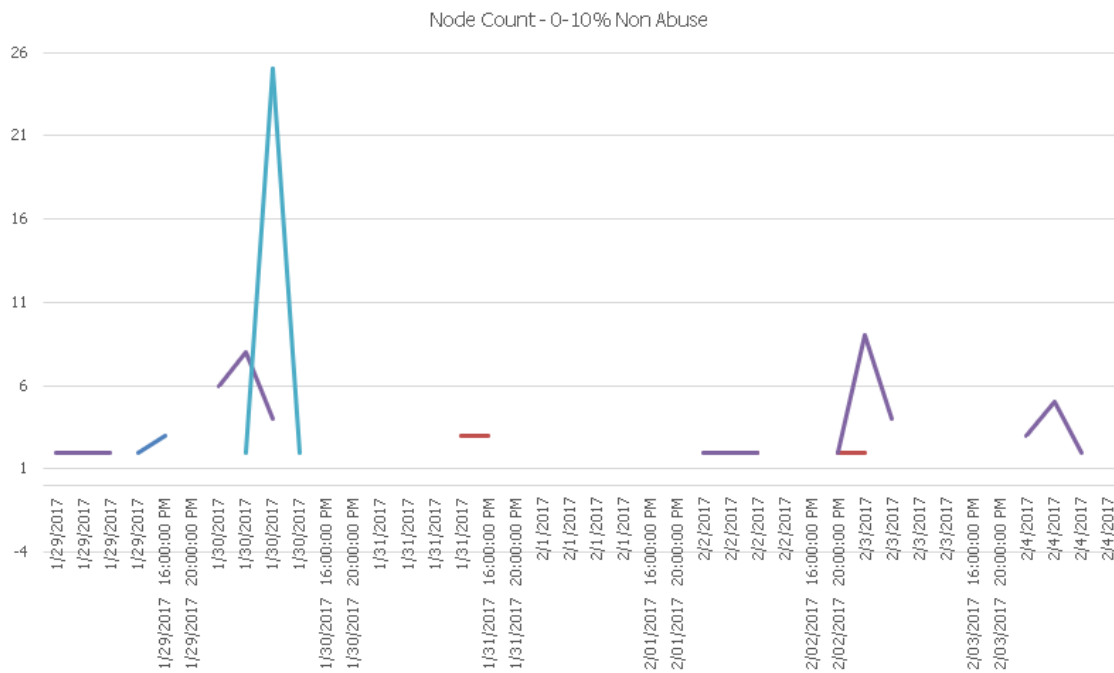


Figure 4.25: Time series of node count for 5 hosts labelled non-abusive in the bottom 10% Total-degree centrality

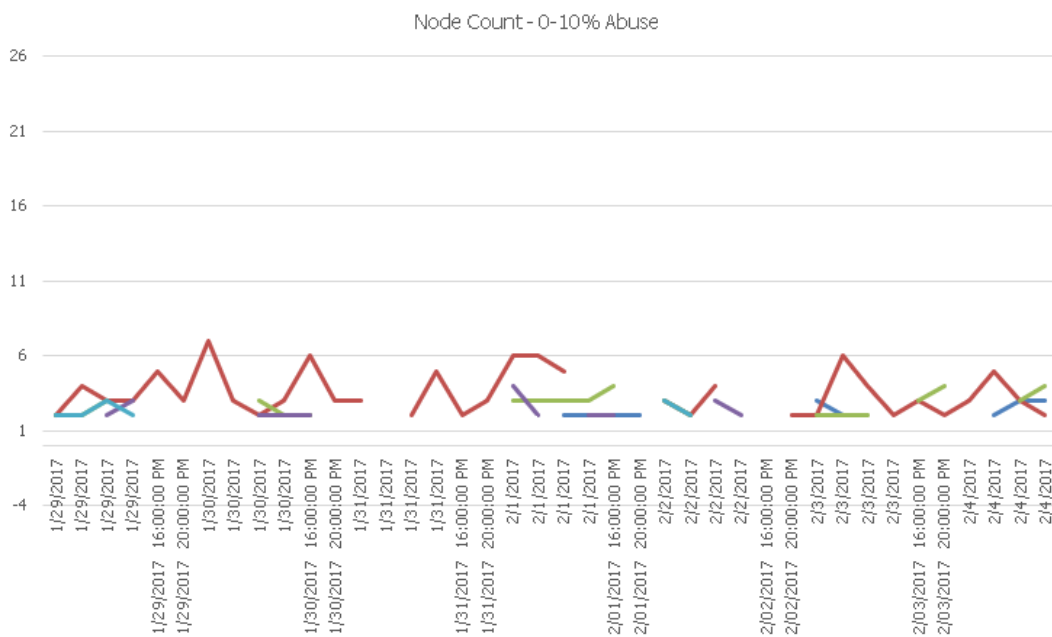


Figure 4.26: Time series of node count for 5 hosts labelled abusive in the bottom 10% Total-degree centrality

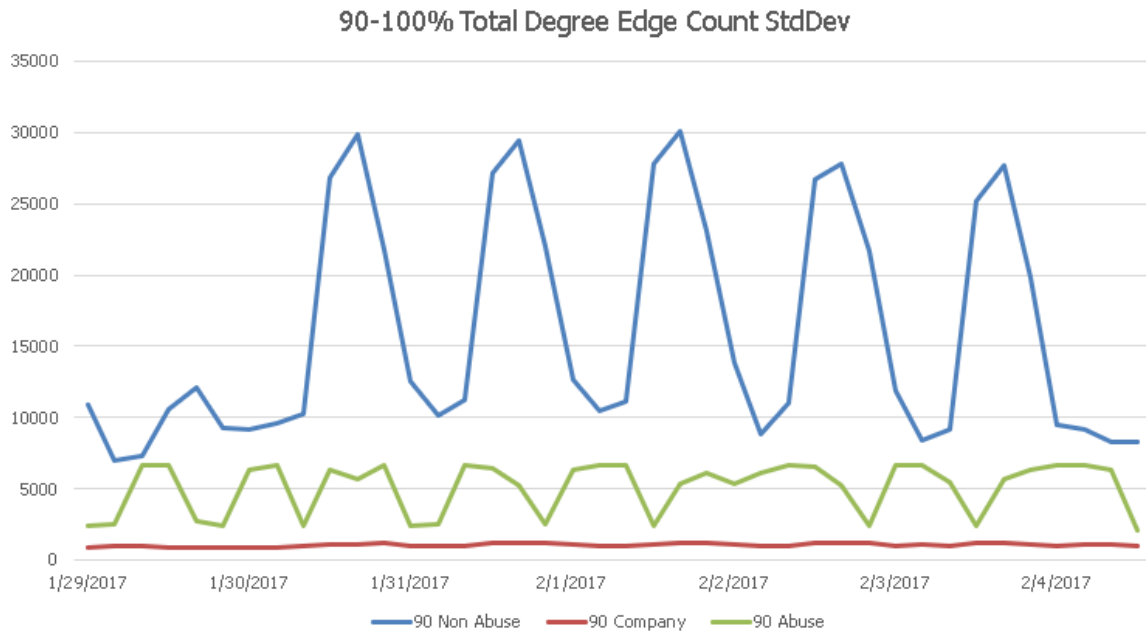


Figure 4.27: Time series of standard deviation of edge count for company, non-abusive, and abusive categories in the top 10% Total-degree centrality

Figure 4.28 shows the standard deviation of edge count for the 50 to 60% total-degree central hosts for abusive and non-abusive hosts. The 50 to 60% bin showed clear differences of standard deviation between abusive and non-abusive hosts. The curves for both bins did not intersect with each other and abusive hosts showed greater variation and higher variations in standard deviations than the non-abusive hosts. Like the previous results, this substantiates the claim that non-abusive hosts show more standard behavior than those reported for abuse on AbuseIPDB.

Lastly, figure 4.29 shows the standard deviation of edge count for the bottom 10% total-degree central hosts for abusive and non-abusive hosts. The 0 to 10% results were not as clear. The non-abusive hosts rested at 0 for a majority of the week and the abusive hosts showed erratic behavior between 0 and 4. This was a result of the large number of holes within the data. The 0 to 10% total-degree central hosts will not be examined in future experiments.

Overall, the study provided findings that non-abusive hosts are more varied and can provide easier models to generate than tracking abusive behavior. This study will be confirmed in the Actionable Classification sections of this thesis.

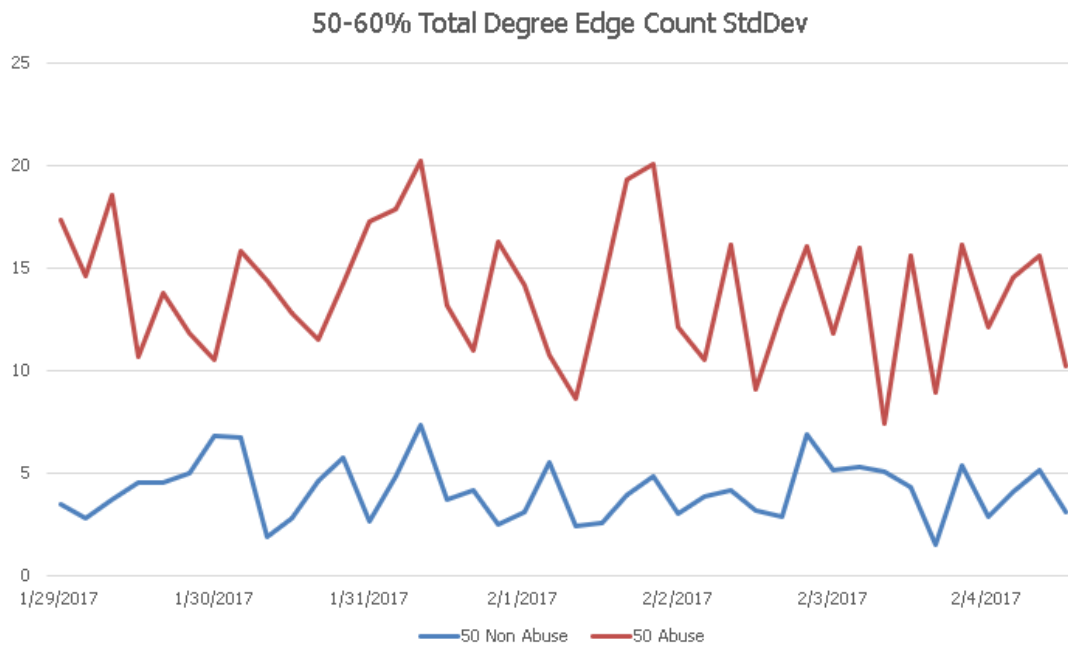


Figure 4.28: Time series of standard deviation of edge count for company, non-abusive, and abusive categories in the 50% to 60% Total-degree centrality range

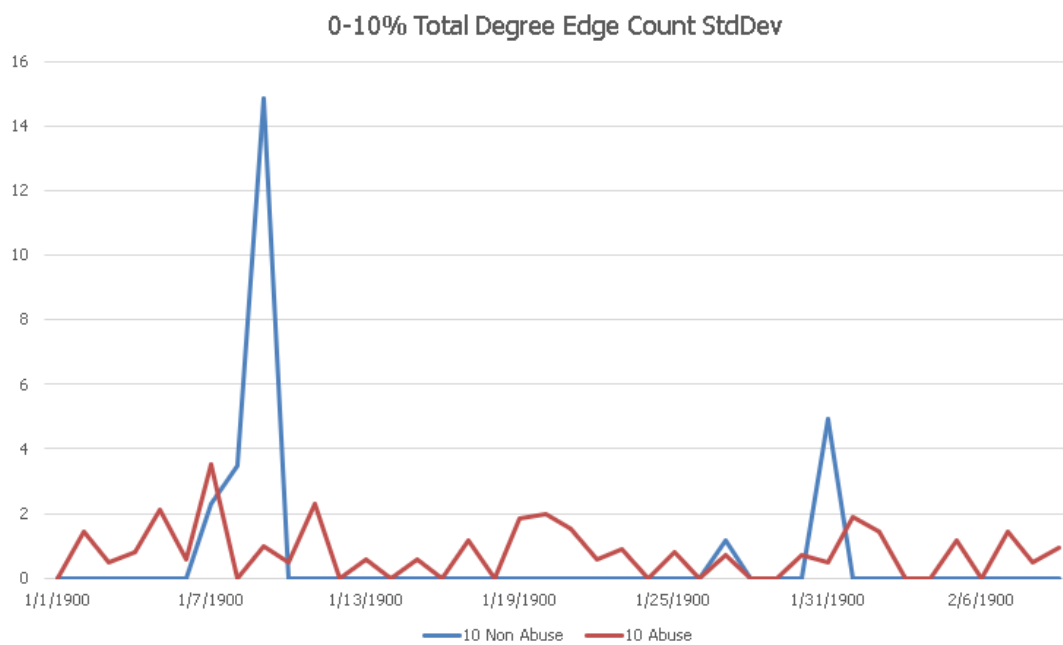


Figure 4.29: Time series of standard deviation of edge count for company, non-abusive, and abusive categories in the bottom 10% Total-degree centrality

4.2 Actionable Classification

This section focuses on the experiments testing classification of hosts. First, hosts are classified as abusive using data provided by AbuseIPDB. Last, hosts are classified by company division. Only hosts were classified because there was not enough data to create classifiers for normal behavior and incident behavior.

4.2.1 Classifying Hosts By Division

Overview

The previous section described methods used to see if there were any differences of network behavior between different divisions of user machines within the enterprise network. The following experiment would test if there are similarities in network structure among divisions within the company. This information would provide value for network administrators by defining normal behavior for each division of internal user machines. For example, if the behavior of an IT user machine starts changing from the model of IT user machines to one matching a server, it may indicate the machine is being used in a data exfiltration attempt or as a backdoor in a cyber-attack.

For the experiment ego-networks for every internal host were created and grouped by IP range that represented their division. The experiment examined 4 divisions within the company: servers, IT user machines, campus security, and management. The sampling distribution was comprised of a 20% testing and 80% training sample for each division and it totaled approximately 950 hosts for training and 220 hosts for testing. Figure 4.30 shows a visual of some of the ego network measurements by division.

After the samples were created, a distance matrix was calculated among all the hosts through dynamic time warping. The measurements that were used were density, weighted density, node count, link count, and link sum. The reason these measurements were used was that reducing the network into ego-networks with only first-degree neighbors made the more complicated measurements irrelevant. Then, K-nearest neighbor was applied on the test sample using the distance matrix calculated through dynamic time warping and the training sample. Finally, a classification report was generated to measure the accuracy of the model generated from the test sample. The whole procedure was completed using Python, Scipy, Matplotlib, and Numpy.

The experiment was completed the first time by calculating the distance matrix of network science measurements individually. However, this approach has the limitations of only comparing one measure of the network and does not give a complete representation of the network. As a result, another approach was done by adding the distance of a combination of network science measures in calculating the distance

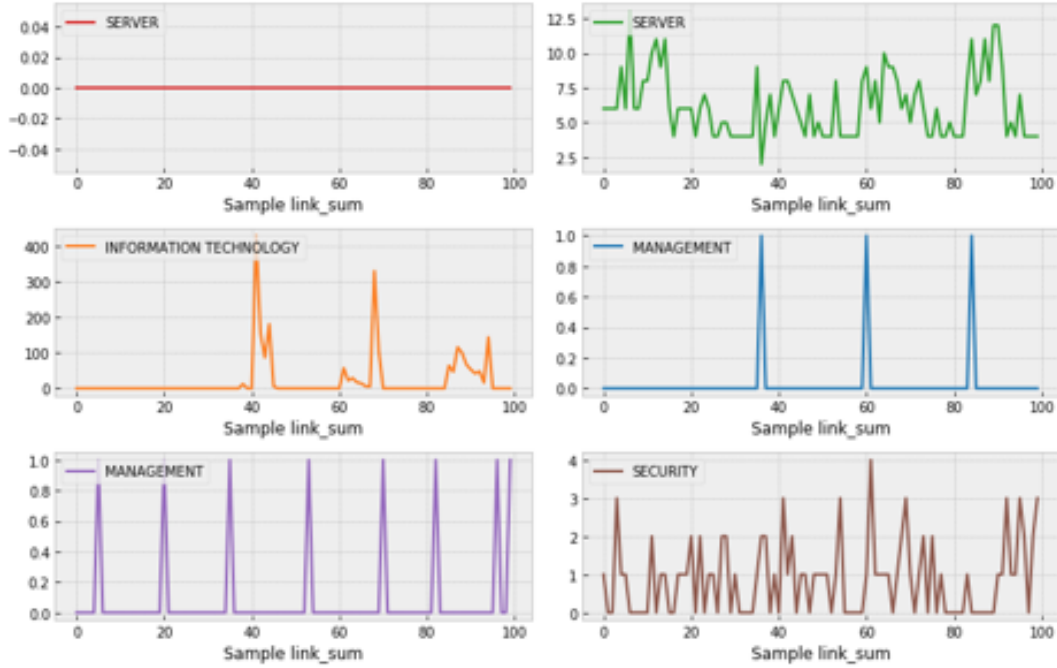


Figure 4.30: Time series plots of link sum for 6 hosts identified by division

matrix. To do this, the other difference measurements calculated through dynamic time warping were put into a multidimensional plane in order calculate a new distance between host ego networks.

Results

Overall, the classifier with single measurements did not show great performance. Figure 4.31 and 4.32 show the results of the best and worst measurements for the classifier respectively. All of the other heatmaps showed a similar distribution among divisions. The management division was the easiest to classify and the rest of the categories had poor results. This may indicate that all hosts show similar behavior and management hosts have the highest chance of being classified because of the large number of hosts in the category.

The rest of the results are shown in table 4.14 which show the precision and recall for each of the network science measurements among each division. The management division had the most accurate results but had the largest sample size so it could have been the most accessible group hosts can classify themselves as. The most accurate measurements for classification ended up being weighted density and link count which showed similar results.

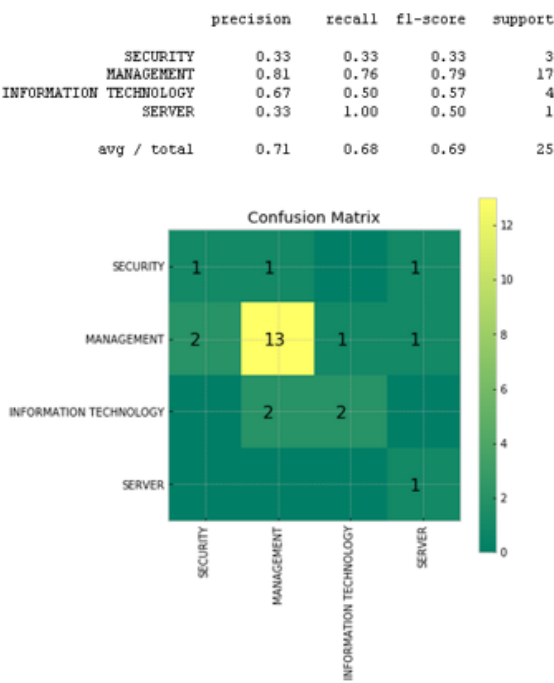


Figure 4.31: Classification Report Heatmap of DTW/KNN calculated from link count

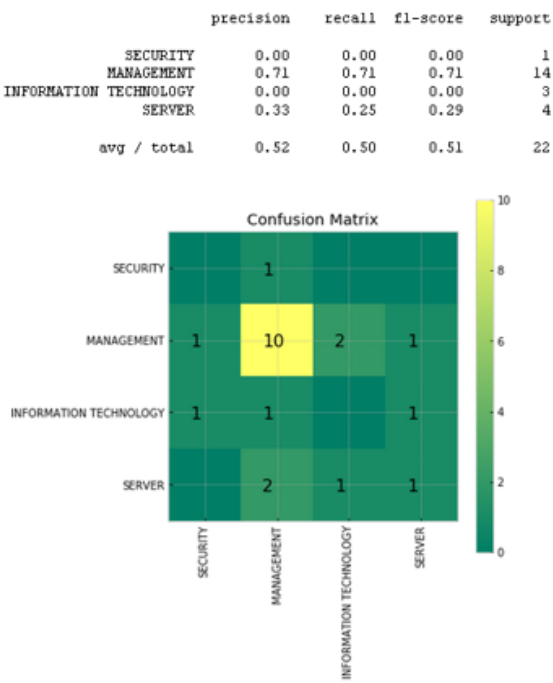


Figure 4.32: Classification Report Heatmap of DTW/KNN calculated from density

Table 4.14: Table of DTW/KNN results for all network science measurements individually calculated

Measurement	Precision	Recall	F1-score	Support
Density	0.52	0.5	0.51	22
Weighted Density	0.75	0.67	0.7	24
Node Count	0.48	0.55	0.51	22
Link Count	0.71	0.68	0.69	25
Link Sum	0.62	0.58	0.60	24

Chapter 5

Discussion

The discussion summarizes the experiments and their implications for the toolchain. Each experiment was an implementation of a component of the toolchain. Effectively, the whole set of experiments show the complete execution of the whole toolchain. Therefore, the experiments validate the effectiveness of possible tools that can be used within the toolchain. The chapter is divided by the phases of the toolchain and the experiments performed in each phase.

5.1 Netflow Collection

Overall, there was little work that needed to be done at the netflow collection phase. Network flows are commonly used in industry when analyzing network traffic and the work has been well cited. As long as the data being examined has been reduced in size, no packet data has been included, and if common network activity should be consistent after reduction of the data, then the netflow collection component has done its job.

5.1.1 SiLK Usage

Overall, SiLK and YAF completed its job. A large amount of data was extracted and described in the Result chapter. Additionally, differences in behavior for protocol and the work week cycles were apparent. Moreover, SiLK offered a lot of fields that were not explored for the experiments that can provide helpful measurements for future work.

One issue that arose with using SiLK and YAF was the amount of time it took to resolve issues when the installation had problems. The reason why this process was so difficult was that the company did not have a specific person in charge of managing the installation. Because of this, it was impossible to gather more data near the end of the project.

Most of these limitations should not impact a company with their own netflow collection installation. As a result, SiLK and YAF are components that would not need changes if this toolchain was to be implemented on a real enterprise network.

5.2 Network Analysis

The network analysis was the focus of most of the experiments conducted in this thesis. The experiments conducted for this section included analyzing the difference of network science measurements between network protocols, the differences of network structures between a flash crowd week and normal weeks, data reduction strategies using degree, data reduction strategies using company divisions, and AbuseIPDB analysis.

5.2.1 Protocol Usage

Overview

The networks were binned by protocol because it was assumed that each protocol would show drastically different behavior. Overall, this assumption was validated. Specifically, automatic protocols, ESP and GRE showed more consistent behavior. Traffic caused by protocols controlled by users such as TCP and ICMP showed more predictable temporal behavior that changed with the flash crowd and the work week cycle. UDP showed more erratic behavior that was more difficult to correlate with the information known about the network.

Implications for Toolchain

It is important to know the type of events that network administrators look for. Every protocol has their own set of anomalies that a network administrator must track to ensure no malicious activity is happening with their network. The most common attack vector is TCP and as a result there is a larger variety of attack models that must be tracked. Additionally, attack vectors are chosen by how easy it is to obfuscate the attack within regular network traffic. For commonly used protocols like TCP, UDP, and ICMP, this is easy. However, if any small change in ESP and GRE are seen, then it is easy to notice because of the lack of traffic performed using these protocols.

Though the effects from the flash crowd were not significant, there was a significant difference in behavior between different protocols. The ESP and GRE clearly exhibited different behavior due to its autonomous nature. TCP and ICMP had clear differences though it is difficult to attribute the cause of

the changes. As a result, splitting data by protocol may be a viable method for cutting network data to reduce noise and highlight behaviors of interest.

Overall, binning these measurements helps narrow down some of this activity. For example, changes in ICMP activity during the Flash Crowd indicated troubleshooting. These were expected results, but if this similar pattern of behavior is unexpected, it can indicate network reconnaissance from an attacker who has access to a backdoor within the network. Typically, ICMP traffic is a precursor before a machine connects to the internet so examining its use would improve cyber-situational awareness.

Additionally, binning by ESP and GRE would require lower threshold models because of how much they reduce the model. Identifying anomalies within these protocols is easy because of how consistent their behavior is. Though it is rare for attacks to happen using these protocols, as soon as some behavior changes and the network structure changes, it is easy to automatically identify it.

Separating the protocols offer the biggest advantage of separating activity from the protocols with the highest activity, TCP and UDP. If all this activity was aggregated into one network bin, then a lot of these behaviors would not be noticeable.

5.2.2 Flash Crowd Analysis

Overview

Overall, the study has found that there is some differentiation in network behavior between the flash crowd and a normal work week, however not a significant amount. The majority of changed behavior seemed to be in differences in degrees of clustering and fragmentation when analyzing a network as a whole. Average total-degree, link, and node count seemed to consistently show cyclic behavior that show little change from the event.

The results of the KS-test found significant differences in distribution for all protocols when comparing the incident week and normal week. However, the main issue was that comparing normal weeks resulted in significant differences as well. The KS-test found that the measures edge count and clique count validated the highest amount of hypothesis created from knowledge of the protocols and network events. The deviations from the hypothesis may be because of the increased volume of data, incorrectly defined hypothesis due to network noise, and interference from unknown anomalies. Without more data or a reliable method of reducing the data, it is inconclusive if network science measurements help characterize flash crowds because of the inconsistency in measured network behavior for normal weeks.

The greatest limitation is the low amount of comparisons. For the experiments, only 3 weeks were compared among each other. Table 4.12 clearly showed this limitation with the low sample size of 9

comparisons. Additionally, more information on the topology and network events during each time period may change the expected results. Therefore, the results could be correct, however the wrong assumptions are being made.

Another limitation was the large scope of data without a ground truth means of classifying it. The examined data contained all netflow records of every host interacting on the company network internally and externally. The scope of a flash crowd incident would impact a smaller subset of internal hosts. If the hosts can be labelled by web servers, email servers, and external clients, then the impact of the flash crowd may be clearer. The methods implemented in the research hoped to extract the effect based off what was expected to happen during the event.

Overall, the methods produced okay results that found significant differences in behavior between different network protocols and the incident and a normal week. However, the results were severely impacted because of the inability to find consistency among normal weeks. Finding consistency of normal weeks is crucial for anomaly detection because of the difficulty in getting a sample size large enough to model network events.

Examining the behavior of high degree central hosts raised the concern of possible persistent threats conducting reconnaissance behavior on the network. Thus, even if resources are allocated to solving network issues, defensive measures should still be practiced. These potentially abusive high degree centrality hosts can be subject for future research.

Implications for Toolchain

The size of enterprise networks is large and methods must be used to remove and analyze it in parts to remove noise. Analyzing the IT network from a network science perspective is useful because all that is required is the Source IP, Target IP, and time. This reduces the size exponentially while only requiring storing IP information, thereby, enforcing privacy for end users. Network science provides a new set of metrics to reduce the size of data, cluster data, and in real time, measure behavior.

Overall, the KS-test found edge count and clique count were the best measurements used to validate the hypothesis. The next set of measurements that validated results were density, node count, and clustering coefficient. As a result, these measurements should provide focus on future implementations of the toolchain. The measurements that did not fare well were weighted and represented the amount of connections that took place during the period. Conducting KS-test on weighted measures with greater amount of variation may result in inaccurate results. Some parameter may need to be applied to KS-test that makes the test less strict with finding similarities in distribution for weighted measurements.

Because the results of hypothesis were not able to find consistent normal behavior, there is not enough evidence to support that network science measures provide an accurate representation. Being able to identify normal behavior is very important because it is difficult to create models on incident data. Incident data is significantly more rare and difficult to simulate within an enterprise network. More data and continued experiments should be done to prove if network science measurements can be used to identify normal behavior and network events.

5.2.3 Data Reduction

Overview

The sampling techniques using degree centrality failed to highlight possible effects from the flash crowd incident. Surprisingly, sampling techniques using total-degree centrality seemed to remove noise from the data and highlighted cyclic behavior and more study must be done finding how to pinpoint affected areas during infrastructure stress. This is mainly because total-degree was the incorrect measurement used to extract the hosts with the highest change in behavior.

Difference in total-degree highlights the hosts that changed variation in hosts that the select host connects to. As a result, this selected mostly servers who have the largest total-degree. External hosts were typically not selected because they connected to a relatively smaller number of servers within the internal enterprise network. The proper measurement to use would have been link sum within the host's ego network. Regardless, labelled data from the organization would greatly improve the ability to pinpoint the hosts affected by the flash crowd incident.

Implications for Toolchain

Instead of using total-degree which targets variation of hosts the select host is connecting to, using link sum may improve the results. Total-degree was the incorrect measurement to use and a different network science measurement should be used to highlight hosts that had the highest change in behavior when comparing an incident with a normal week. Another approach is split the network into subset of hosts by functional group within the organization. This information is significantly easier to obtain if the toolchain is being developed from within the company.

5.2.4 Functional Group Analysis

Overview

The functional group analysis was limited because of the incompleteness of the campus security, management, and IT user machines. The only results completed were the server hosts. The server hosts exhibited the same behavior as expected individually. They showed cyclic behavior that peaked early afternoon of the work day and exhibited a constant baseline during the nights and weekends. However, all 3 weeks differed tremendously from each other regardless of normal week. It was hypothesized that there should be consistency between all 3 weeks because the sample includes all servers and the flash crowd event would only show.

Implications for Toolchain

The main change that these results indicate for the toolchain is that more research needs to be completed on normal weeks. It is safe to discount observations of the data until it can be observed in future data because of how inconsistent these results are from knowledge of the events themselves conceptually and from the perspective of the company. For now, the inconsistencies of the data can be dismissed as a SiLK collection issue until recent data can confirm these observations. However, the cyclic nature of server machines makes servers seem like a viable category to classify hosts. Later sections will confirm the performance of classifying the hosts.

5.2.5 AbuseIPDB Analysis

Overview

Overall, the small sample of abusive hosts and non-abusive hosts highlighted that there is more variation in hosts that were tagged for abusive behavior. This implies that making classification models by non-abusive hosts is easier. However, variation was not the only difference and abusive hosts typically had the higher range of network interactions when compared to non-abusive hosts.

After cross correlating hosts with high total-degree centrality within the network with Abuse IP Database, a database of user reported potentially malicious hosts, many highly-reported hosts were prevalent in the network during both the incident week and the normal weeks. Potential threats are persistent at all times, even during infrastructure stress; defensive measures should not be disregarded even during these periods.

However, this initial experiment had many limitations. The first limitation was that only 5 hosts made

the sample size. The sample of 5 was only used for visual analysis, however conclusions cannot be made on the data as whole with this sample. Additionally, identifying hosts as non-abusive was difficult. Only the top 10% total-degree central hosts non-abusive category was accurate because of the use of Fortune 500 companies to identify these hosts. However, the other results only searched for hosts with 0 reports on AbuseIPDB. These hosts may still be conducting malicious behavior, but they just may not have been reported by a network administrator.

Despite these limitations, abusive hosts are easy to identify in AbuseIPDB and make up fairly confident ground truth of the behavior of these hosts. Thus, abusive hosts may still provide a sufficient set for classification. Future tests on a larger set of abusive hosts should be performed to confirm this assumption. If classification algorithms perform well on AbuseIPDB abusive hosts, then network science measurements provide a great method for quantifying network behavior for hosts and identifying potentially abusive hosts.

Implications for Toolchain

Overall, creating ego-networks using AbuseIPDB showed good results. It is common for firewalls and intrusion detection systems to use signatures and IP address lists of hosts compromised by botnets from other sources. AbuseIPDB provides the same role for the toolchain described in this thesis. Though finding hosts prevalent in the network that have been reported was easy, there is not enough evidence to support the assumption that their network science measurements differ between non-abusive hosts. These assumptions will be validated in the actionable classification phase when identifying AbuseIPDB reported hosts from host behavior.

5.3 Actionable Classification

The last phase was validated by creating models for individual host activity by class. These experiments consisted of classifying hosts by their individual traffic activity represented by a network. Two approaches were tested, identifying abusive hosts and hosts by internal divisions. Both of these classification strategies would equip network administrators with knowledge on how to respond to anomalous host activity.

5.3.1 Classifying Hosts by Division

Overview

The individual network science measurements showed adequate results. It identified hosts within the management division around 70% of the time. However, the other categories scored between 30% and

60%. Regardless of measurement, the results were fairly consistent among each division and the division that performed the best had the highest sample size. This indicates that either the management was the easiest to classify or that there were many false classifications to the management division from the other hosts and host behavior was consistent regardless of division. However, because the recall was still around 70% for the management division than this may not that big of an issue.

Classifying with measurements individually produces the shortcomings of only using one feature of the network structure. The goal of the thesis was to attempt to quantify the whole network structure and this requires many measurements to define. As a result, more work should be done on using multiple network science measurements to calculate distance measurements for k-nearest neighbor. Additionally, there are more advanced classification techniques that can be applied such as neural networks and Markov chains to classify time series. These techniques may offer higher quality results and should be examined as well.

An additional limitation is that dynamic time warping makes the assumption that temporal features have less weight when determining the differences between time series. It can be argued that temporal features are very important for telecommunication events because timing of traffic flows determine the difference between DDOS attacks and flash crowds. Other attack types might require timing differentiation as well. On the contrary, an entirely Euclidean approach should not be used because network behavior will always differ for hosts between different time periods. There must be a mediation between strict difference measurements and more relaxed difference measurements and different situations may require different approaches. The current experiment only used dynamic time warping to calculate measurements and it was lenient towards temporal differences.

Implications for Toolchain

For individual measure classification, the best measures were weighted density and link count. Both had a range of 50% to 70% precision and 50% to 70% recall. These results are not accurate enough to apply to real network operations and should not be implemented in its state. Methods should be done to improve it which include adding multi-dimensional distance measurements using other network science measures, applying transformations before calculating distances of time series plots, and applying other machine learning techniques. If other methods do not work, then representing hosts as individual networks may not provide a structure robust enough to characterize host behavior.

Chapter 6

Conclusions

The thesis describes the design and implementation of many possible components of a network science toolchain used to convert network traffic data into actionable intelligence for network administrators. The design followed a procedure that is similar to how other feature-based anomaly detection system were designed focusing on reducing the data, representing the data, and then modelling the data. The approach described in this thesis is novel because of its application of network science to the field.

It was hypothesized that network science would provide very good results due to the nature of its scale of analysis. Rather than examine metrics associated to individual hosts, network science focuses on using a large set of metrics to describe the whole network structure. Effectively, a combination of network science measures becomes a profile for the network traffic during a period of time and its representation over time is a series of time plots. These time plots can then be used to generate models to events and host profiles that network administrators can respond to.

The work completed in this thesis marks a preliminary study on the capabilities and potential techniques that can be used when analyzing this data using network science as a field of study. Within the work, a series of implementations and experiments on various aspects of the toolchain were carried out and validated using assumptions made from the network topology and network events. Though the results did not align clearly with the hypothesis for many experiments, the capability of network science as a means should not be discounted. The work contributed to the field by providing a method of representing network traffic through graph structures and time series plots describing the network structure, identified differences in behavior of traffic over network protocol, examined the quantified differences between a flash crowd and normal weeks, used IP ranges as divisions for network structures for pinpointing changes in network behavior, compared IP addresses labelled as abusive with those that were non-abusive, and tested a classifier for hosts using host behavior as a parameter.

The work described in this research is novel and seeks to quantitatively represent network traffic using network structures and its associated measures. The closest previous work only investigated graph measurements which focused on node level measurements. Even though the results were not as conclusive, the research elevates the field by providing a new field of work for telecommunication feature-based anomaly detection.

The following sections will conclude with the limitations of the thesis and directions to take future work.

6.1 Limitations

The limitations of the research done in this thesis is the lack of longer series of data, the difficulty in getting ground truth information about the network and events from the company, and the difficulty of working with live streaming data.

One of the greatest limitations of the work was the low amount of data points used for comparisons. Event data was compared by weeks and only 3 weeks were provided for comparisons. Having more weeks of data would help validate measurement comparisons used in the paper. Additionally, this can be solved by changing the time limits being compared. This was shown from a previous experiment that used daily data as parameters for the KS tests instead of weekly data. Alternative methods that take a similar approach and increase the number of datapoints compared in statistical tests should be done.

Much of the network, network events, and data was proprietary and there was a strict bureaucracy of getting information from the company. This made getting information to better explain network science measurements difficult. Even by the end of the research, it is unclear what hosts were most affected by the flash crowd incident and these are all details that the company should possess. Some information was given such as the timeline of the service outage and flash crowd incident and IP ranges of company divisions, however a lot of other information was not given. If the network topology complete with IP ranges were given, then it would make data reduction significantly easier for forensics on any network event.

Working with live data has advantages because its unique and real. Most research in the same field use simulated data or published data. However, the data used in this research was collected live while network operations were taken place. Not only was the data live, but it was on a very large enterprise network. This data represents exactly how the toolchain would perform in a real enterprise network. However, working with the data is difficult. The SiLK installation had issues that resulted in holes in the data after the instance was brought down. Some of these holes were as small as a few hours and others

were as large as a whole day. Solving these issues was also difficult as someone within the company had to work to resolve them and a lot of times, the netflow collector issues took less priority than other tasks for the company administrators.

6.2 Future Work

The first goal of future work is to be able to define normal behavior using the network science measurements described in the thesis. If normal behavior can be defined clearly, then finding differences with network events should be simpler. To solve this, more data on normal weeks can be collected or an alternative data source can be used. The new normal data can be compared and clustered till a baseline model can be established using previous classification techniques.

Moreover, more incident data can be gathered or simulated to create models of events themselves. When an incident occurs, the same comparison and reduction experiments can be used to characterize the differences of that week on the normal model. It may be unfeasible to rely on actual events happening within the network to create classification models, however it may be possible to simulate event data and inject it into the network.

Additionally, the statistical tests currently focus on individual network science measurements. However, individual metrics do not quantify the network structure as a whole. To capture the whole network, combinations of network science measurements should be used to characterize the behavior of the network. This multi-dimensional plane then should be applied to comparison and classification processes to characterize changes in behavior of network traffic.

Another area for future work would be to reexamine the flash crowd event using different data reduction strategies and with more topology information from the company. The current methods were making inferences on the effects and network structure. The results from the experiments clearly lead to the fact that some information is missing about the incident and the network. One simple approach for reducing the data would be to filter netflows by packet size. The size of netflows and number of packets within the flow tell a significant amount of information about the network communication. Network administrators can tell if a streaming service is running or simple heart beat messages are being sent to hosts. By binning netflows by size, the models can focus on the network communication that network administrators are more interested about.

Another approach to reducing the data would be to use more information about the network to bin different network structures. Having complete information about the network and incidents is crucial to making correct hypothesis about the experiments to validate processes within the toolchain. Additionally,

this information is available for the security experiments who would be responsible for implementing the toolchain in a real, enterprise situation. For future work, either more information should be gathered on the current network or simulated data should be defined to help validate the assumptions of the process.

The last area for future work centers on the Actionable Classification stage. There are many methods for analyzing and classifying time series data. One approach is to expand on the K-nearest neighbor approach and use different distance measurements besides dynamic time warping, transform the time series data before calculating distance, and using combinations of network science measurements to calculate distances. Another approach is to apply different classification machine learning algorithms to time series data. Previous research centered on using machine learning methods on well labelled network traffic and have exceptional results. These same methods can be applied using the features generated from network science.

Bibliography

- [1] M. Abu Rajab, J. Zarfoss, F. Monroe, and A. Terzis, "A multifaceted approach to understanding the botnet phenomenon," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. ACM, 2006, pp. 41–52. 49
- [2] A. A. Amaral, L. de Souza Mendes, B. B. Zarpelão, and M. L. P. Junior, "Deep ip flow inspection to detect beyond network anomalies," *Computer Communications*, vol. 98, pp. 80–96, 2017. 7, 31
- [3] A. A. AnAj. (2007) Knnclassification. [Online]. Available: <https://upload.wikimedia.org/wikipedia/commons/e/e7/KnnClassification.svg>. [Accessed 03-15-18]. xii, 28
- [4] B. S. Anderson, C. Butts, and K. Carley, "The interaction of size and density with graph-level indices," *Social networks*, vol. 21, no. 3, pp. 239–267, 1999. 20
- [5] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. Long, "Managing flash crowds on the internet," in *Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on*. IEEE, 2003, pp. 246–249. 31, 32
- [6] R. A. R. Ashfaq, X.-Z. Wang, J. Z. Huang, H. Abbas, and Y.-L. He, "Fuzziness based semi-supervised learning approach for intrusion detection system," *Information Sciences*, vol. 378, pp. 484–497, 2017. 10
- [7] P. Barford and D. Plonka, "Characteristics of network traffic flow anomalies," in *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*. ACM, 2001, pp. 69–73. 4, 31, 32
- [8] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370. 27
- [9] R. B. Blazek, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of denial-of-service attacks via adaptive sequential and batch-sequential change-point detection methods," in

- Proceedings of IEEE systems, man and cybernetics information assurance workshop*. Citeseer, 2001, pp. 220–226. 3
- [10] S. P. Borgatti, K. M. Carley, and D. Krackhardt, “On the robustness of centrality measures under conditions of imperfect data,” *Social networks*, vol. 28, no. 2, pp. 124–136, 2006. 25
- [11] Bscan. (2013) Ks example. [Online]. Available: https://commons.wikimedia.org/wiki/File:KS_Example.png. [Accessed 03-15-18]. xii, 26
- [12] K. M. Carley, “Ora: A toolkit for dynamic network analysis and visualization,” in *Encyclopedia of social network analysis and mining*. Springer, 2014, pp. 1219–1228. 20
- [13] S. Chakravarty, M. V. Barbera, G. Portokalidis, M. Polychronakis, and A. D. Keromytis, “On the effectiveness of traffic analysis against anonymity networks using flow records,” in *International conference on passive and active network measurement*. Springer, 2014, pp. 247–257. 27
- [14] B. Claise, “Cisco systems netflow services export version 9,” 2004. 3
- [15] —, “Specification of the ip flow information export (ipfix) protocol for the exchange of ip traffic flow information,” 2008. 3
- [16] A. Clauset, M. E. Newman, and C. Moore, “Finding community structure in very large networks,” *Physical review E*, vol. 70, no. 6, p. 066111, 2004. 49
- [17] A. Conta and M. Gupta, “Internet control message protocol (icmpv6) for the internet protocol version 6 (ipv6) specification,” 2006. 30
- [18] E. Costenbader and T. W. Valente, “The stability of centrality measures when networks are sampled,” *Social networks*, vol. 25, no. 4, pp. 283–307, 2003. 25
- [19] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967. 28
- [20] A. Dainotti, A. Pescapé, and G. Ventre, “Nis04-1: Wavelet-based detection of dos attacks,” in *Global Telecommunications Conference, 2006. GLOBECOM’06. IEEE*. IEEE, 2006, pp. 1–6. 3
- [21] H. Du and S. J. Yang, “Discovering collaborative cyber attack patterns using social network analysis,” in *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer, 2011, pp. 129–136. 5

- [22] M. Evangelou and N. M. Adams, "Predictability of netflow data," in *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on*. IEEE, 2016, pp. 67–72. 27
- [23] D. Farinacci, P. Traina, S. Hanks, and T. Li, "Generic routing encapsulation (gre)," 1994. 30
- [24] F. Feather, D. Siewiorek, and R. Maxion, "Fault detection in an ethernet network using anomaly signature matching," in *ACM SIGCOMM Computer Communication Review*, vol. 23, no. 4. ACM, 1993, pp. 279–288. 4
- [25] B. A. Forouzan and S. C. Fegan, *TCP/IP protocol suite*. McGraw-Hill Higher Education, 2002. 30
- [26] F. Givehki and A. Nicknafs, "Mobile control and management of computer networks using sms services," *Telematics and Informatics*, vol. 27, no. 3, pp. 341–349, 2010. 26
- [27] C. Guo, Y. Zhou, Y. Ping, Z. Zhang, G. Liu, and Y. Yang, "A distance sum-based hybrid method for intrusion detection," *Applied intelligence*, vol. 40, no. 1, pp. 178–188, 2014. 10
- [28] H. Haddadi, R. Landa, A. W. Moore, S. Bhatti, M. Rio, and X. Che, "Revisiting the issues on netflow sample and export performance," in *Communications and Networking in China, 2008. ChinaCom 2008. Third International Conference on*. IEEE, 2008, pp. 442–446. 26
- [29] L. Hao, C. G. Healey, and S. E. Hutchinson, "Ensemble visualization for cyber situation awareness of network security data," in *Visualization for Cyber Security (VizSec), 2015 IEEE Symposium on*. IEEE, 2015, pp. 1–8. 28
- [30] W. He, G. Hu, and Y. Zhou, "Large-scale ip network behavior anomaly detection and identification using substructure-based approach and multivariate time series mining," *Telecommunication Systems*, vol. 50, no. 1, pp. 1–13, 2012. 10
- [31] Z. Hejun and Z. Liehuang, "Encrypted network behaviors identification based on dynamic time warping and k-nearest neighbor," *Cluster Computing*, pp. 1–10, 2017. 28
- [32] R. Hofstede, P. Čeleda, B. Trammell, I. Drago, R. Sadre, A. Sperotto, and A. Pras, "Flow monitoring explained: From packet capture to data analysis with netflow and ipfix," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2037–2064, 2014. 14
- [33] L. Hubert and J. Schultz, "Quadratic assignment as a general data analysis strategy," *British journal of mathematical and statistical psychology*, vol. 29, no. 2, pp. 190–241, 1976. 27
- [34] C. M. Inacio and B. Trammell, "Yaf: yet another flowmeter," in *Proceedings of LISA 2010: 24th Large Installation System Administration Conference*, 2010, p. 107. 14

- [35] Z. Jian-Qi, F. Feng, Y. Ke-Xin, and L. Yan-Heng, "Dynamic entropy based dos attack detection method," *Computers & Electrical Engineering*, vol. 39, no. 7, pp. 2243–2251, 2013. 9
- [36] A. Karasaridis, B. Rexroad, D. A. Hoeflin *et al.*, "Wide-scale botnet detection and characterization." *HotBots*, vol. 7, pp. 7–7, 2007. 49
- [37] M. Kenney, "Ping of death," *Insecure.org*, 1996. 2
- [38] S. Kent, "Ip encapsulating security payload (esp)," 2005. 30
- [39] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing network-wide traffic anomalies," in *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4. ACM, 2004, pp. 219–230. 4, 10, 30
- [40] J. Lemon *et al.*, "Resisting syn flood dos attacks with a syn cache." in *BSDCon*, vol. 2002, 2002, pp. 89–97. 2
- [41] G. L'huillier, H. Alvarez, S. A. Ríos, and F. Aguilera, "Topic-based social network analysis for virtual communities of interests in the dark web," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, pp. 66–73, 2011. 5
- [42] G. A. Marin, "Network security basics," *IEEE security & privacy*, vol. 3, no. 6, pp. 68–72, 2005. 2
- [43] L. Martin. (2014) Cyber kill chain. [Online]. Available: http://cyber.lockheedmartin.com/hubfs/Gaining_the_Advantage_Cyber_Kill_Chain.pdf. [Accessed 04-20-18]. 1, 2
- [44] F. J. Massey Jr, "The kolmogorov-smirnov test for goodness of fit," *Journal of the American statistical Association*, vol. 46, no. 253, pp. 68–78, 1951. 26
- [45] I. McCulloh and K. M. Carley, "Detecting change in longitudinal social networks," Military Academy West Point NY Network Science Center (NSC), Tech. Rep., 2011. 17
- [46] S. Moghaddam and A. Helmy, "Interest-based mining and modeling of big mobile networks," in *Big Data Computing Service and Applications (BigDataService)*, 2015 IEEE First International Conference on. IEEE, 2015, pp. 1–6. 26
- [47] M. E. Newman, "The structure and function of complex networks," *SIAM review*, vol. 45, no. 2, pp. 167–256, 2003. 18
- [48] C. C. Noble and D. J. Cook, "Graph-based anomaly detection," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 631–636. 4, 10

- [49] J. Noble and N. M. Adams, "Correlation-based streaming anomaly detection in cyber-security," in *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 311–318. 27
- [50] G. Nychis, V. Sekar, D. G. Andersen, H. Kim, and H. Zhang, "An empirical evaluation of entropy-based traffic anomaly detection," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*. ACM, 2008, pp. 151–156. 4, 8, 9, 30
- [51] OGREBot. (2015) Dynamic time warping. [Online]. Available: https://commons.wikimedia.org/wiki/File:Dynamic_time_warping.png. [Accessed 03-15-18]. xii, 27
- [52] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of internet background radiation," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM, 2004, pp. 27–40. 49
- [53] A. Patcha and J.-M. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer networks*, vol. 51, no. 12, pp. 3448–3470, 2007. 3, 4
- [54] N. Patwari, A. O. Hero III, and A. Pacholski, "Manifold learning visualization of network traffic data," in *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*. ACM, 2005, pp. 191–196. 28
- [55] J. Postel, "User datagram protocol," Tech. Rep., 1980. 30
- [56] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for qos: a statistical signature-based approach to ip traffic classification," in *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*. ACM, 2004, pp. 135–148. 28
- [57] K. Seo and S. Kent, "Security architecture for the internet protocol," 2005. 30
- [58] W. Sha, Y. Zhu, M. Chen, and T. Huang, "Statistical learning for anomaly detection in cloud server systems: a multi-order markov chain framework," *IEEE transactions on cloud computing*, 2015. 10
- [59] L. Singh and A. Cheng, "Distilling command and control network intrusions from network flow meta-data using temporal pagerank," in *Telecommunication Networks and Applications Conference (ITNAC), 2016 26th International*. IEEE, 2016, pp. 107–114. 3, 4, 8, 14
- [60] V. A. Siris and F. Papagalou, "Application of anomaly detection algorithms for detecting syn flooding attacks," in *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, vol. 4. IEEE, 2004, pp. 2050–2054. 3

- [61] B. Smith, W. Yurcik, and D. Doss, "Ethical hacking: the security justification redux," in *Technology and Society, 2002.(ISTAS'02). 2002 International Symposium on.* IEEE, 2002, pp. 374–379. 49
- [62] A. Sperotto, G. Schaffrath, R. Sadre, C. Morariu, A. Pras, and B. Stiller, "An overview of ip flow-based intrusion detection." *IEEE Communications Surveys and Tutorials*, vol. 12, no. 3, pp. 343–356, 2010. 2
- [63] J. H. Steiger, "Tests for comparing elements of a correlation matrix." *Psychological bulletin*, vol. 87, no. 2, p. 245, 1980. 27
- [64] S. Strapp and S. J. Yang, "Segmenting large-scale cyber attacks for online behavior model generation," in *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction.* Springer, 2014, pp. 169–177. 5, 8
- [65] B. Tellenbach, M. Burkhart, D. Sornette, and T. Maillart, "Beyond shannon: Characterizing internet traffic with generalized entropy metrics," in *International Conference on Passive and Active Network Measurement.* Springer, 2009, pp. 239–248. 9
- [66] M. Thomas, L. Metcalf, J. Spring, P. Krystosek, and K. Prevost, "Silk: A tool suite for unsampled network flow analysis at scale," in *Big Data (BigData Congress), 2014 IEEE International Congress on.* IEEE, 2014, pp. 184–191. 15, 16
- [67] G. Tian, Z. Wang, X. Yin, Z. Li, X. Shi, Z. Lu, C. Zhou, Y. Yu, and Y. Guo, "Mining network traffic anomaly based on adjustable piecewise entropy," in *Quality of Service (IWQoS), 2015 IEEE 23rd International Symposium on.* IEEE, 2015, pp. 299–308. 9
- [68] B. Trammell, E. Boschi, L. Mark, T. Zseby, and A. Wagner, "Specification of the ip flow information export (ipfix) file format," Tech. Rep., 2009. 14
- [69] V. Vaidya, "Dynamic signature inspection-based network intrusion detection," Aug. 21 2001, uS Patent 6,279,113. 2, 3
- [70] K. Viet, B. Panda, and Y. Hu, "Detecting collaborative insider attacks in information systems," in *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on.* IEEE, 2012, pp. 502–507. 5
- [71] S. Wasserman and K. Faust, *Social network analysis: Methods and applications.* Cambridge university press, 1994, vol. 8. 4, 20

- [72] C. V. Wright, F. Monrose, and G. M. Masson, "Using visual motifs to classify encrypted traffic," in *Proceedings of the 3rd international workshop on Visualization for computer security*. ACM, 2006, pp. 41–50. 28
- [73] M. Yip, N. Shadbolt, T. Tiropanis, and C. Webber, "The digital underground economy: A social network approach to understanding cybercrime," 2012. 5
- [74] A. Ziviani, A. T. A. Gomes, M. L. Monsoro, and P. S. Rodrigues, "Network anomaly detection using nonextensive entropy," *IEEE Communications Letters*, vol. 11, no. 12, 2007. 9