GS-EECS6414M Project Midterm Progress Report

Jia Ying Ou*
caryou@my.yorku.ca
York University
Toronto, Ontario

Yunge Hao*
hyggs@my.yorku.ca
York University
Toronto, Ontario

ACM Reference Format:
Jia Ying Ou and Yunge Hao. 2020. GS-EECS6414M Project Midterm Progress Report. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The Distributed Denial of Service (DDoS) attack is a well-known threat to network security. The DDoS attack is designed to prevent legitimate users from accessing network resources or a computer system. The first major DDoS kept Yahoo.com off the internet for about 2 hours, cost a potential loss of $500,000 [2] in the year 2000. According to the 2018 IDG report [3], there are 86% of people report experiencing one or more DDoS attacks and 70% are highly likely to consider changing their current solution to a higher effective solution. Therefore, analyzing complex network data to detect a DDoS attack is essential.

This project is based on two main studies. First, Sharafaldin et al. [12] addressed several shortcomings that appeared in the dataset of previous studies, such as the lack of finding a comprehensive dataset for detection model evaluation. A new dataset called CICDDoS2019 [12] is generated to remedy the limitation of previous datasets. It is generated by a testbed that simulates real-world network attacks. There are two parts to the dataset. One part is the training data, and one is the testing data. In the training data, it consists of benign traffics and 11 types of attacks, including NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, SYN and TFTP. In the testing data, it consists of benign traffics and 7 types of attacks, including PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag and SYN. There are 80 features extracted from the dataset using a tool called CICFlowMeter [4], such as source IP address, destination IP address, etc. They built their model using machine learning algorithms includes ID3, RF, Naïve Bayes and Logistic Regression for pattern capturing. However, the results are not robust enough.

We therefore apply an alternative and relatively novel approach from the second source [9] to analyze the dataset and detect DDoS attacks, Traffic Dispersion Graphs (TDG). The analysis is focused on the evolution of several graph metrics in TDG in time series to detect malicious activities and using the VF2 isomorphism algorithms to identify attack patterns in anomalous traffic.

The potential applications are in the domain of network security. One of the applications can be a server with a built-in attacker-detecting algorithm that helps it recognize certain requests to avoid overloading. Another application is that visualization identifies and characterizes problems to effectively increase operators' situation awareness, letting them detect and respond to malicious activities in a quicker manner. And the DDoS attack classifier shortens the time for the network operators to discover the pattern and serves as an aid when the operator is under the decision-making process for faster mitigation.

2 PROBLEM DEFINITION

The entire project has two tasks at hand. The first one is anomaly detection, and the second one is attack identification. They seem to be similar in nature, and they are indeed, though, the difference lies at how a network administrator treats them. The anomaly detector generates a warning message and reports it to the administrator whenever it detects an abnormal while keeps silent when there is no abnormal. While the attack identification part is responsible for finding out the cause for the anomaly [9]. It comes after detection, and solves a more complicated classification problem, i.e., a multi-label classification of various types of attacks. It not only distinguishes malicious attacks from benign network traffic but classifies different types of DDoS attacks, such as LDAP, Syn, UDP, DNS, etc.

3 RELATED WORK

There are many previous studies introduces graph mining for anomaly detection in the field of computer security. In 2003, Noble et al. [10] proposed an entropy-based anomaly detection scheme for the subdue graph as an algorithm for the identification of repetitive graph patterns. They present two methods. The first one is 'anomaly substructure detection' that looks for specific, unusual substructures within a graph. The second one is 'anomalous subgraph detection' with the graph partitioned into different sets of subgraphs, and compare each subgraph against the others for unusual patterns observation. This method analyzes the whole graph and to detect abnormal substructures and sub-graphs found in the entire graph. Each vertex and edge has a label to identify its type in this approach.

Kashima et al. [6] proposed a method using time sequence graphs for anomaly detection. The principal eigenvector of the dependency matrix is employed as a feature vector. For each time instance, the corresponding activity vectors are taken into account as a matrix, and the principal left singular vector is obtained to capture the

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

normal dependencies over time. They compute the angle between the activity vector of the new graph and the principal left singular vector obtained from the previous graphs to learn the anomaly score of the tested graph. Similarly, Sun et al. [13] introduce an anomaly detection approach based on the sequence of graphs. An approximation of the original matrix is obtained by carrying out Compact Matrix Decomposition(CMD) on the adjacency matrix for every graph, and approximation error is also computed to detect anomalies in the time-series of errors.

However, the above methods require a high computational cost. Therefore, it is not suitable for network attack anomaly detection. We will now review some papers that use the graph for network traffic anomaly detection.

Zhou et al. [14] introduced a multi-time series for a time-series graph for traffic anomaly detection. This method mines all the frequent patterns, which is conducive to detecting many kinds of abnormal traffic effectively. To detect anomalies in a large-scale network traffic flow, they considered the entropy of four attributes, which are Source IP Address, Destination IP Address, Source Port and Destination Port. However, this method creates an enormous size of network traffic graphs due to the inclusion of ports as nodes, which increases the computational complexity. To reduce the computational cost, we only used IP address to define nodes of the graph.

Iliofotou et al. [7] introduced a representation of network traffic with a series of related graph instances that change over time, a series of novel metrics that capture changes both in the graph structure and IP address of a TDG [8]. This is the first study that uses dynamically changing graphs to characterize and classify network traffic. This approach can be applied in traffic classification and the detection of polymorphic blending attack problems. dK-2 distance is used to quantify the change over time of TDGs in our approach.

Godiyal et al. [5] used the visualization of network traffic as directed graphs to apply algorithms to model attack patterns. Through graph representations of the traffic flows, they matched known subgraph patterns to recognize attacks. But their method is very time consuming since they consider the entire traffic flow. To resolve this limitation, we chose to separate the tasks of anomaly detection and pattern matching in order to speed up the time required for the matching algorithms with the size of the input graphs being smaller.

4 METHODOLOGY

4.1 General Model

The data analytic methods that we applied belong to graph mining. It has advantages over statistical and machine learning in not being subject to parameters adjusting; in the case of detecting network anomalies, the amount of false alarms generated by these methods is high [9]. Instead of using prediction models in ordinary machine learning algorithms and best features set, we adapt graph techniques with the goal of improvement in network analysis.

We used traffic dispersion graphs (TDGs) to model the network traffic from our dataset. Among the 80 features in our dataset, four of them were extracted to analyze, namely, source IP, destination IP, source port, destination port. In our network graph, each node is a distinct IP, could be either a client or a server, and each edge indicates a package transmission.

The network traffic was organized in continuous time series (exact to microsecond). Each interval is taken as a snapshot of a TDG graph by seconds, and each property of the graph is recorded correspondingly with its time interval, so that analysis as well as prediction can be done simultaneously. As new packets appear in the network and taken into account each second, new results is potentially on the fly.

4.2 Graph Metrics

Analysis of the TDGs consists of the following graph metrics.

4.2.1 total node degree.
This metric is the total degree of nodes in the network, counting both in-coming and out-going edges, averaged over a time interval of each second.

4.2.2 Vino, Vin, Vout [9].
Vin denotes the number of nodes with only incoming edges, i.e., the number of servers. Vout denotes the number of nodes with only outgoing edges, i.e., the number of clients. Vino denotes the number of nodes with both incoming and outgoing edges.

4.2.3 maximum degree (Kmax) [9].
The maximum degree of a node in the graph, it is also taken by each second and forms a series.

4.2.4 entropy of the degree distribution [9].
The formula for the entropy of the degree distribution is as following:

$$H(X) = \sum_{k=1,kmax} P(k)\log(P(k))$$

where P(k) is the probability that a node has degree k. Degree distribution describes the connectedness of the network, and the entropy measures the heterogeneity, the uncertainty of network connectedness.

4.2.5 graph edit distance [9].
Graph edit distance is the distance between to TDGs, where G_i and G_j is calculated from the minimum number of edit operations that is required to make graph G_i isomorphic to graph G_j using the formula:

$$d(G_i, G_j) = |V_i| + |V_j| - 2|V_i \cap V_j| + |E_i| + |E_j| - 2|E_i \cap E_j|$$

where V_i, E_i are the numbers of nodes and edges in graph G_i and G_j, respectively.

4.2.6 dK-2 distance metric [9].
Node degree, Vino, Vin, Vout, Kmax and entropy of the degree distribution are static metrics. Graph edit distance and dK-2 distance metric are dynamic metrics. This metric is used to define the similarity between the original graph and a synthetic graph. This dK-2 distance metric is built upon the dK-2 series[11].

Definition of dK-series:

- dK-0 is the average degree $\overline{k}$
- dK-1 is the node degree distribution P(k)
- dK-2 is the joint degree distribution (JDD) p(k_1, k_2)
- dK-d (d≥3) is the order-d distribution P_d (P_d describes how gropus of d-nodes with degree k_1, k_2, ...,k_d interrelated to each other)

The set of graphs having the same distribution P(k) is denoted as dK-1 graphs, which is a subset of the set of dK-0 graphs. The set of dK-2 graph is a subset of the set of dK-1 graphs, etc.

The dK-2 distance between two TDGs, G and G' is represented by the Euclidean distance between the corresponding joint degree distributions p(k_1, k_2) and p'(k_1, k_2) respectively.

4.3 VF2 Algorithm

To identify the attack, we first obtain the attack structure pattern that TDGs were generated. Then, we use the graph matching method for identification. In our approach, we use the VF2 algorithm. VF2 algorithm can be used for both graphs and sub-graph isomorphism to find out if one object is part of another object. We will be applying VF2 to identify attack patterns in abnormal TDGs for a faster approach since attack patterns are located in the abnormal traffic.

5 EXPERIMENTAL RESULTS

5.1 Anomaly Detection

5.1.1 total node degree.

Table 1 demonstrates when an attack happens, the total number of node degree is usually accompanied by a high degree of the network graph, high variance and high standard deviation of the degrees in series of the time interval. This data from Table 1 does make sense since a DDoS attack is flooding the bandwidth or resources of a targeted system by overwhelming it with data, which means there should be many connections between servers, therefore high node degrees.

Since each type of attack contains benign traffic in its dataset, the metric evaluated for benign traffic is considered separately. Table 2 is the node degree, variance and standard deviation for the benign traffic in each dataset. The magnitude of the metrics for benign traffic is significantly lower than those of attacks. Each attack has a much higher standard deviation and a higher mean of node degree than benign traffic. Since the mean can be largely affected by the dispersion of the set of values and we did not make assumptions on the distribution of node degree, combining standard deviation with the mean drew a more clear boundary between attacks and normal network and thus characterized the attacks more strongly than by the mean value alone.

Figure 1 is a histogram that illustrates the node degree of benign traffic and various attacks over time, where the horizontal axis indicates the number of occurrences of a total degree at each specific timestamp, and the vertical axis indicates the number of occurrences at each degree. It is clear to see that their large number of node degrees can distinguish attacks, but by their high degree of dispersion, the attack behaviour is even more apparent. As we can see from the left corner of Figure 1, benign traffic is condensed, the total number of degrees is small, as well as deviation.

type	node degree	variance	standard deviation
UDP-Lag	658	7898064	2810
TFTP	7534	348139363	18658
Syn	13229	5704713	2388
DrDos UDP	4145	30880770	5557
DrDoS SSDP	6326	26414932	5140
DrDoS SNMP	10189	63874372	7992
DrDoS NTP	970	3948043	1987
DrDoS NetBIOS	11225	33659824	5802
DrDoS MSSQL	13117	27189441	5214
DrDoS LDAP	7327	805219	897
DrDoS DNS	6520	67615473	8223
Average	7385	56011837	5879

Table 1: node degree for attacks

type	node degree	variance	standard deviation
UDP-Lag	46	9450	97
TFTP	37	6431	80
Syn	32	3087	56
DrDos UDP	27	5230	72
DrDoS SSDP	25	3608	60
DrDoS SNMP	18	4491	67
DrDoS NTP	36	5629	75
DrDoS NetBIOS	30	3146	56
DrDoS MSSQL	24	3626	60
DrDoS LDAP	8	395	20
DrDoS DNS	28	7628	87
Average	28	4793	66

Table 2: node degree for benign

5.1.2 Vino, Vin, Vout.

Due to the specialty of the way the network traffic is generated [12], in the dataset, the IP addresses of the servers and the clients are divided into two non-overlapping sets. A node with both incoming edges and outgoing edges does not exist in the graph. The analysis of this metric was solely on Vin and Vout.

Each node can send multiple packets via its ports for the degree of a node so that it contributes to an increase of edges and node degrees in a short time interval. Our case is different from that. We use 1 second as our time interval, Vin and Vout represent the number of nodes or hosts. The total count of the number of nodes and hosts is limited and does not vary so obviously. Thus Vin and Vout were counted minute-by-minute, instead of second.

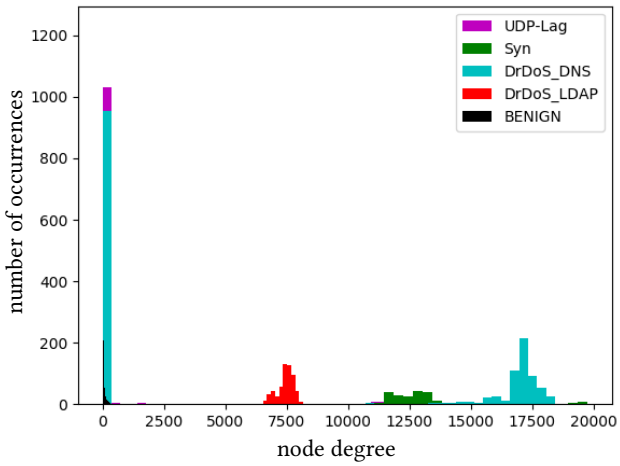

Figure 1: node degree dispersion

Figure 2 is the scatter of Vin versus Vout. Black dot indicates a benign and red triangle indicates an attack. Each spot is a pair of number of servers and number of clients over one minute. Overall, benign activities have a more scattered distribution of number of servers and clients ranges from 2 to 40 or so. When attacks happen, the number of clients were always restricted within 2 or 3, which is the opposite to the distribution for node degree, but each node has a significant impact on the node degree of the network graph. Figure 2 is the scatter of Vin versus Vout. Black dot indicates a

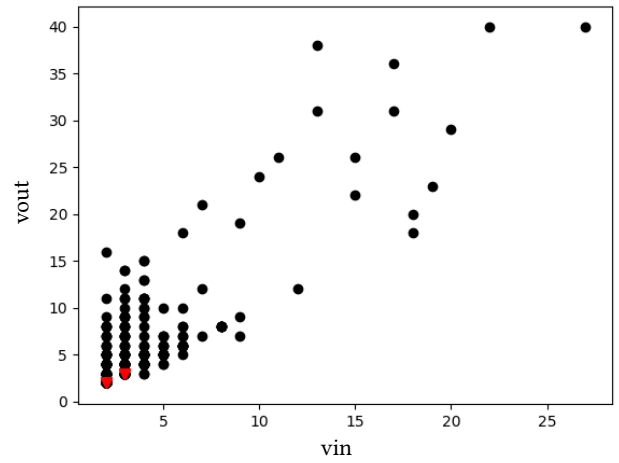

Figure 2: scatter Vin v.s. Vout

benign and red triangle indicates an attack. Each spot is a pair of number of servers and number of clients over one minute. Overall, benign activities have a more scattered distribution of number of servers and clients ranges from 2 to 40 or so. When attacks happen the number of clients were always restricted within 2 or 3, which is the opposite to the distribution for node degree, but each node has a great impact to the node degree of the network graph.

5.1.3 maximum degree (Kmax).

Kmax is calculated in the unit of seconds. The threshold for benign activity is loosely upper bounded by 100 in general. As an illustration, Figure 3 shows the maximum degree during UDP-Lag attack taken overtime in second. The red line indicates attack and the black line is the threshold for maximum degree, which is around 70.

It also demonstrates that attacks and benign traffic overlapped heavily, as it is the case for other attacks as well, and this is a thorny challenge when analyzing this dataset.

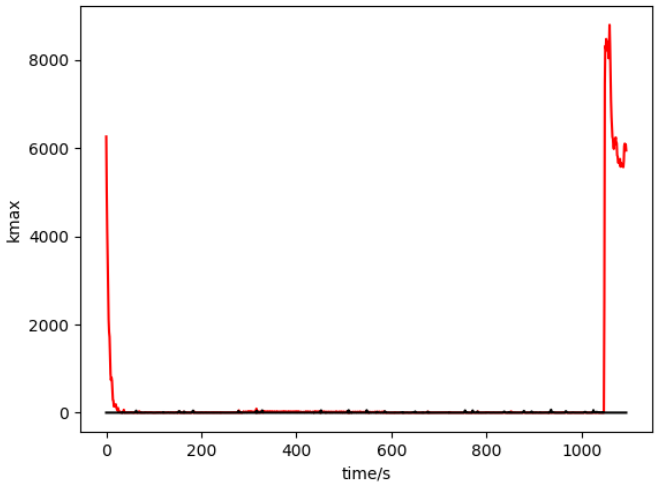

Figure 3: UDP-Lag Kmax

5.1.4 entropy of the degree distribution.

5.1.5 graph edit distance.

5.1.6 dK-2 distance metric.

5.2 Attack Identification

5.2.1 graph metrics.

Although figure 1 shows that, besides detecting abnormal from benign activities, some attacks that can be identified from each other by the distribution pattern of total node degree, as the case for DrDoS LDAP, Syn and DrDoS DNS, this is not the case for many others. Identification through a node degree is not recommended.

The following metrics will be used for the first part to make comparison of the result when utilizing different graph metrics.

- Maximum degree(Kmax)
- dK-2 distance
- the number of packets
- measured over time of trace of the dataset

5.2.2 VF2 classification.

In addition, the classification results from VF2 will be presented in a multi-label confusion matrix as a heatmap.

We will be validating our algorithm in the second part by using the testing dataset from the CICDDoS2019 dataset. The dataset CICDDoS2019 is generated in one day within 2 to 7 hours of duration [12], but our method and analysis allow it to be scaled for longer time interval of data. Since the testing dataset is smaller than the dataset that we use in the methodology part, thus, we will be scaling down during the evaluation process.

The three evaluation metrics are precision, recall and F-Measure.

- Precision, or positive predictive value, is the correct classification count of flows (TP) divided by total classifications (TP+FP).
$$Pr = \frac{TP}{(TP + FP)}$$
- Recall or sensitivity is the correct classification count (TP) divided by all generated flows (TP+FN).
$$Rc = \frac{TP}{(TP + FN)}$$
- F-Measure (F1) is a harmonic combination of precision and recall, ranging from 0 to 1, with 1 being the most desirable.
$$F1 = \frac{2}{(\frac{1}{Pr} + \frac{1}{Rc})}$$

Other datasets that can be used for evaluation. They are "CAIDA DDoS Attack 2007" dataset, and the DARPA dataset [1] mentioned as previous research. Also, from [9], there are traffic traces from POSTECH in July 2009 and CAIDA DDoS trace in 2007, which are the exact dataset used by [9].

REFERENCES

[1] 2000. DARPA Intrusion Detection Scenario Specific Datasets | MIT Lincoln Laboratory. https://www.ll.mit.edu/r-d/datasets/2000-darpa-intrusion-detection-scenario-specific-datasets

[2] 2000. Yahoo on Trail of Site Hackers | WIRED. https://www.wired.com/2000/02/yahoo-on-trail-of-site-hackers/

[3] 2018. IDG DDoS REPORT. https://www.a10networks.com/wp-content/uploads/a10-eb-14116-en-evolving-strategies-for-handling-todays-complex-costly-threat.pdf?mkt{_}tok=eyJpIjoiTkdZeU5UQXlZbUkxTmpGbCIsInQiOiJFM3JWNkx6UndMa2FkcXB6N\\k5NTSt3M0RNYTIxWEdobUwrY1NYd3FpNFVLK2UxMEtmQ1plWERQOE8wV2tn\\YStYTU5EQXVvYUJsa2lneGxcL0VobDY1Z1g0bGdcL2tkOHlMaHpwcXZWWlh\\jajUzVEZlN1VvRGpRUmlzS21YaG9PeTlpIn0{%}3D

[4] Lashkari Arash H., Zang Yongcan, Owhuo Gift, Mamun Mohammad S.I., and Gerard D. Gil. 2015. Network Traffic Flow analyzer. http://netflowmeter.ca/netflowmeter.html

[5] Apeksha Godiyal, Michael Garland, and John Hart. 2010. Enhancing Network Traffic Visualization by Graph Pattern Analysis. (01 2010).

[6] Tsuyoshi IDÉ and Hisashi KASHIMA. 2004. Eigenspace-Based Anomaly Detection in Computer Systems. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '04)*. Association for Computing Machinery, New York, NY, USA, 440–449. https://doi.org/10.1145/1014052.1014102

[7] Marios Iliofotou, Michalis Faloutsos, and Michael Mitzenmacher. 2009. Exploiting Dynamicity in Graph-Based Traffic Analysis: Techniques and Applications. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '09)*. Association for Computing Machinery, New York, NY, USA, 241–252. https://doi.org/10.1145/1658939.1658967

[8] Marios Iliofotou, Prashanth Pappu, Michalis Faloutsos, Michael Mitzenmacher, Sumeet Singh, and George Varghese. 2007. Network Monitoring Using Traffic Dispersion Graphs (Tdgs). In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement (IMC '07)*. Association for Computing Machinery, New York, NY, USA, 315–320. https://doi.org/10.1145/1298306.1298349

[9] Do Le, Taeyeol Jeong, H. Roman, and James Hong. 2011. Traffic dispersion graph based anomaly detection. *ACM International Conference Proceeding Series*, 36–41. https://doi.org/10.1145/2069216.2069227

[10] Caleb C. Noble and Diane J. Cook. 2003. Graph-Based Anomaly Detection. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '03)*. Association for Computing Machinery, New York, NY, USA, 631–636. https://doi.org/10.1145/956750.956831

[11] Alessandra Sala, Lili Cao, Christo Wilson, Robert Zablit, Haitao Zheng, and Ben Y Zhao. 2010. Measurement-calibrated graph models for social network experiments. In *Proceedings of the 19th international conference on World wide web*. 861–870.

[12] Iman Sharafaldin, Arash Habibi Lashkari, Saqib Hakak, and Ali Ghorbani. 2019. Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. 1–8. https://doi.org/10.1109/CCST.2019.8888419

[13] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. 2007. GraphScope: Parameter-Free Mining of Large Time-Evolving Graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*. Association for Computing Machinery, New York, NY, USA, 687–696. https://doi.org/10.1145/1281192.1281266

[14] Yingjie Zhou, Guangmin Hu, and Weisong He. 2009. Using graph to detect network traffic anomaly. In *2009 International Conference on Communications, Circuits and Systems*. 341–345. https://doi.org/10.1109/ICCCAS.2009.5250514