Traffic Dispersion Graph Based Anomaly Detection

Do Quoc Le, Taeyoel Jeong, H. Eduardo Roman, James Won-Ki Hong
Division of IT Convergence Engineering
Pohang University of Science and Technology (POSTECH), Korea
{lequocdo, dreamerty, roman, jwkhong}@postech.ac.kr

ABSTRACT

Detecting and diagnosing anomalous traffic are important aspects of managing IP networks. In this paper, we propose a novel approach to detect anomalous network traffic based on graph theory concepts such as degree distribution, maximum degree and dK-2 distance. In this approach, we have used the traffic dispersion graphs (TDG) to model network traffic over time. We analyze differences of TDG graphs in time series to detect anomalies and introduce a method to identify attack patterns in anomalous traffic. The approach has been validated by using network traces from POSTECH and CAIDA.

Keywords

Anomalous Traffic Detection, Network Monitoring and Analysis, DDoS Attacks, Traffic Dispersion Graphs, Network Security.

1. INTRODUCTION

As the Internet continues to grow in size and complexity, security has become a critical issue due to the occurrence of traffic anomalies. The latter may be due to several causes such as denial of services (DoS) attacks, flash crowds, port scans and the spreading of worms, which can compromise the proper functioning of the network. Anomalous traffic detection has thus become an indispensable component of any network security infrastructure. Detecting and identifying these risks is thus particularly important in network management but very difficult to achieve in actual operations. Methods for detecting traffic anomalies are typically based on machine learning, data mining or the statistical analysis of network models. However, these techniques often generate a huge number of false alarms, and as a result, further work is necessary in order to improve detection accuracy and performance.

During the last decade, complex networks concepts have found applications in a variety of domains ranging from computer science, sociology, to arts and biology. Consequently, a deeper insight has been gained in understanding their main structural characteristics. The main goal of this work is to employ similar concepts to improve on anomalous traffic detection reliability.

In this paper, anomalies are detected by using the traffic dispersion graphs method (TDG) to model network traffic [1] from which we consider selected quantities characterizing the graph. Our approach is constituted by two parts: one regards static properties of the graph and the second, a dynamical aspect describing the change of TDGs as a function of time. Graph matching algorithms are used to determine the causes of anomalies. The attacked graph pattern that we find from the DDoS CAIDA trace [2] is shown to be also present in abnormal traffic of a POSTECH trace, suggesting the presence of a similar anomalous traffic behavior.

The rest of this paper is organized as follows. In the next section, we briefly review previously published related work. In section 3, we describe background in term of TDG-based network traffic modeling, the graph metrics which we use to detect traffic anomalies and the graph matching to identify attacks. In section 4, we provide a detailed description of the implemented system to detect anomalies and identify attacks if present. Section 5 presents the experimental results and finally section 6 concludes the paper with some final remarks.

2. RELATED WORK

Algorithms have been suggested for detecting anomalies in three types of graph changes including label modification, vertex/edge insertion and deletion [3]. The minimum description length principle is used in each of the algorithms to determine the normal reference pattern.

Two techniques for graph-based anomaly detection were introduced in [4]. The first, called ‘anomalous substructure detection’, searches for specific, unusual substructures within a graph, while the second, denoted as ‘anomalous sub-graph detection’, partitions the graph into distinct sets of vertices (sub-graphs), which are tested against each other in the search of unusual patterns. These techniques are based on the SUBDUE application [5] to examine an entire graph and to detect abnormal substructures and sub-graphs contained within it. In their approach, each vertex and edge has a label to identify its type.

Kashima [6] proposed the spectral technique to detect anomalies in a time series of graphs. In this method, a feature vector is first extracted from the principal component of the adjacency matrix that is represented in a separate graph at every sampling time. The time-series of the activity vectors, describing the behavior of the network, is treated as a matrix. To capture the normal time variations in the traffic data, the principal left singular vector is obtained. The angle between activity vector of a new graph and the principal left singular vector obtained from the previous graphs is used to compute the anomaly score of the new graph.

Sun *et al.* [7] introduced an anomaly detection technique based on the sequence of graphs. Compact matrix decomposition (CMD) is performed on the adjacency matrix for each graph to obtain an approximation of the original matrix. The approximation error between the original adjacency matrix and the approximate matrix is computed over time to detect anomalies.

The methods mentioned above have typically high computational complexity [8]. Therefore, they cannot effectively be applied to analyze realistically large graphs used to model the Internet

SoICT 2011, October 13–14, 2011, Hanoi, Vietnam.

traffic. To reduce computational complexity, we thus analyze unlabeled graphs and just concentrate on their nodes.

Zhou *et al.* [9] proposed a network traffic anomaly method based on graph mining. They described the relationships among multi-time series graphs. In order to detect anomalies in the large-scale network traffic flow, they considered the entropy of four attributes: source IP address, destination IP address, source port and destination port. The drawback of this method is that it creates network traffic graphs which have enormous size, because of the inclusion of ports also as nodes, thus significantly increasing the computational complexity. In contrast, in our approach we use TDGs to monitor and analyze network traffic. A TDG is a graph in which each node represents an IP address and each edge a connection between two IP addresses.

Iliofotou *et al.* [10] represented network traffic by means of a series of related graphs that change over time, using several graph metrics. They applied their methods to the problem of traffic classification, suggesting a possible application to anomaly detection. Here, we introduce a new metric, denoted as dK-2 distance [13], to quantify the change over time of TDGs.

Godiyal *et al.* [11] used a graph matching method to identify attacks, however their method turns out to be very time consuming as they consider the whole traffic flow. In our approach, we detect abnormal TDGs first and then identify the attack patterns inside them. In this way, we reduce the computational complexity of the graph matching method.

3. BACKGROUND

In this section, we provide brief explanations of the concepts we use in our anomaly detection approach.

3.1 Graph-based Network Traffic Modeling

In this paper, we use traffic dispersion graph (TDG) to model network traffic [1]. TDG is a novel way to analyze network traffic with a powerful visualization. Iliofotou *et al.* [1] defined a traffic dispersion graph as a graphical representation of the various interactions ("who talks to whom") of a group of nodes. In IP networks, a node of the TDG corresponds to an entity with a distinct IP address and the graph captures the exchange of packets between various sender and destination nodes. Figure 1 illustrates an example of a TDG visualization of DNS service established from an analyzed 60 seconds long flow trace from the Internet junction at POSTECH.

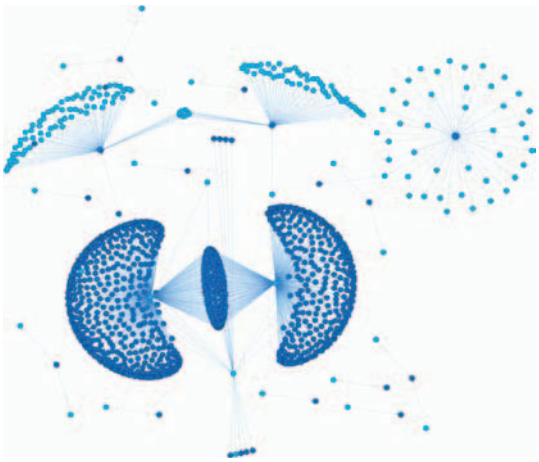

Figure 1. Example of a TDG (DNS) visualization at POSTECH.

The network traffic at the flow level, using the standard 5-tuple {srcIP, dstIP, srcPort, dstPort and protocol}, was analyzed. A TDG is defined as a directed graph G = (E,V) where V is a collection of nodes (IP addresses) in the traffic flows and E is a set of edges (flows) that connect pairs of nodes.

In our approach, bidirectional flows from network traffic were generated. A TCP flow begins with the first SYN packet and the ACK-flag not set, in order to identify the sender and the receiver of the flow. In UDP flows, the first packet of the flow defines the direction of it. To analyze TDGs, a set of graph metrics is used, as discussed next.

3.2 Graph Metrics

Graph metrics are used to represent and distinguish graphs. In our work, we focus on the metrics which describe structural properties of graphs. We also introduce a new graph metric to analyze graphs, like the dK-2 distance metric. The graph metrics are classified in two groups: static and dynamic metrics. Static metrics describe the properties of individual graphs, while dynamic metrics measure the differences between them.

3.2.1 Static Metrics

The static metrics introduced here can be found in [1].

3.2.1.1 Node degree

The degree of a node represents the number of edges connected to it. The degree is the most widely used metric for characterizing the graph. Because TDGs are directed graphs, the number of incoming and outgoing edges need to be considered separately, and are denoted as in-degree and out-degree, respectively. A high in-degree reflects the diversity of a host, whereas a high out-degree reflects a large number of destinations of a client.

For illustration, we report in Figure 2 the node degree distribution function obtained from POSTECH traffic at one minute, averaged over a period of one hour. The distribution shows an approximate power-law decay (scale-free) over at least one decade, with a power-law exponent of about 3.

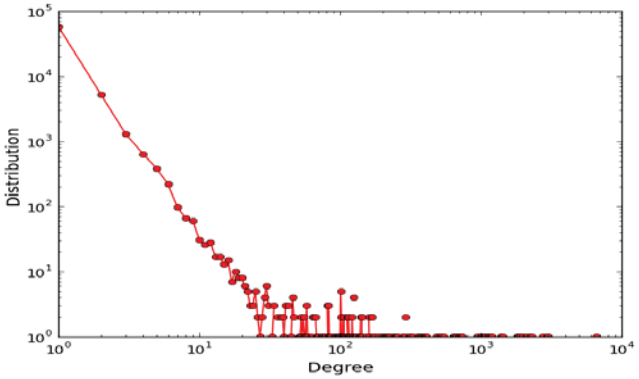

Figure 2. One-minute degree distributions of POSTECH traffic, obtained within one hour.

3.2.1.2 Vino, Vin, Vout

Vin, Vout and Vino denote the number of nodes that have just incoming edges, outgoing edges and both incoming and outgoing edges, respectively. The Vino metric reflects the existence of P2P applications in TDGs [1].

3.2.1.3 *Maximum degree (Kmax)*

The maximum node degree, denoted as Kmax, is a powerful metric to detect DDoS attack in a network as we will see in the following.

3.2.1.4 *Entropy of the degree distribution*

The entropy of the degree distribution, H(X), is defined as

$$H(X) = -\sum_{k=1,k_{max}} P(k)\, log\big(P(k)\big), \quad (1)$$

where $P(k)$ is the probability that a node has degree k.

This metric is used to quantify heterogeneity of the network.

3.2.2 *Dynamic Metrics*

To measure the diversity of TDGs over time, we used two dynamic metrics, the graph edit distance and the dK2 distance.

3.2.2.1 *Graph edit distance*

The graph edit distance between two TDGs, G_i and G_j, is calculated from the minimum number of edit operations required to make graph G_i isomorphic to graph G_j using the formula [10]:

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

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

A number of graph edit distance measurements were applied to a time series of TDGs to investigate network behavior over time. The graph edit metric was also used to detect graph anomalies [12].

3.2.2.2 *dK-2 distance metric*

The dK-2 distance metric introduced here is based on the graph similarity metric which was built upon the so-called dK-2 series in [13]. In the latter, this metric is used to quantify the similarity between the original graph and a synthetic one. The dK-series were introduced to quantify the amount of correlation present in real-world graphs [14]. In the dK-series definition, dK-0 represents the average degree $\bar{k}$, dK-1 the node degree distribution $P(k)$, dK-2 the joint degree distribution (JDD) $p(k_1, k_2)$ and dK-d (d$\geq$3) is the order-d distribution P_d (P_d describes how groups of d-nodes with degree $k_1, k_2, \ldots, k_d$ interrelated to each other). In addition, the set of graphs having the same distribution $P(k)$ (denoted as dK-1 graphs) is a subset of the set of dK-0 graphs, the set of dK-2 graphs is a subset of the set of dK-1 graphs and the set of dK-d graphs is a subset of the set of dK-(d-1) graphs [14].

The reasons for using here dK-2 series are that dK-d is able to capture increasingly complex graph properties with increasing d (the dK-series can capture more graph structure details than other metrics) and dK-2 requires lower computational complexity than dK-d for d $\geq$ 3.

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

3.3 Graph Matching

Graph matching indicates the process of finding the structural similarity of two graphs. Many methods for graph matching have been proposed in recent years [15]. Early approaches for graph matching were applied to finding isomorphism between two graphs. A graph isomorphism between graphs G and G' is a bijective mapping from the nodes of G to the nodes of G' that preserves all labels and the structure of the edges [16]. Similarly, a sub-graph isomorphism between G and G' is an isomorphism from G to a sub-graph of G' [16].

In our approach, we use VF2 algorithm [17] for graph matching to identify attack patterns in network traffic. The VF2 algorithm that can be used for both graph and sub-graph isomorphism has been developed in [17]. This algorithm can be described by means of the state space representation (SSR). In each state a partial mapping solution is maintained and only consistent states are kept. These states are generated using feasibility rules that remove pairs of nodes that cannot be isomorphic.

In VF2 algorithm, a matching process between two graphs, G1 and G2, consists in the determination of a mapping M which associates nodes of G1 to nodes of G2, and vice versa. Generally, the mapping M is expressed as the set of pairs (n,m) (with n $\in$ G1 and m $\in$ G2) each representing the mapping of a node n of G1 with a node m of G2. A mapping M $\subset$ N1×N2 is said to be an isomorphism if M is a bijective function that preserves the branch structure of the two graphs. A mapping M $\subset$ N1×N2 is said to be a graph-subgraph isomorphism if M is an isomorphism between G2 and a sub-graph of G1 [17].

Sub-graph isomorphism is useful concept to find out if one object is part of another object. In this paper, we use this concept to recognize attack patterns in network traffic. Attack patterns were also TDGs which were generated from attack traffics. The advantage of describing an attack pattern using TDGs instead of vectors is that TDGs allow for a more powerful representation of structural relations of attacks.

4. ANOMALY DETECTION AND ATTACK IDENTIFICATION

In this section, we propose a method to detect abnormal network traffic using graph metrics. In this paper, we define abnormal network traffic as the traffic cause by a malicious purpose including the traffic by DoS/DDoS attacks, Internet worms and scanning.

The overall process consists of two parts: anomaly detection and attack identification as illustrated in Fig. 3. The anomaly detection module receives flows from monitoring systems and then obtains TDGs from these flows. After that the TDGs are analyzed overtime to detect anomalies. The attack identification module is used to explore causes of anomalies after detecting abnormal traffic. An alarm is emitted if an attack is identified in the abnormal traffic.

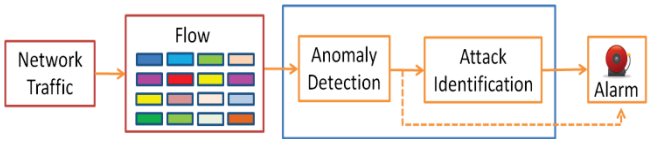

Figure 3. Overall detection process.

4.1 Anomaly Detection

In anomaly detection process, network traffic was sampled at regular points in time t_1, t_2, …, t_i, … and a time series of corresponding TDGs sequences G_1, G_2, …, G_i, … was obtained. Graph metrics, which is presented in the section 3.2, were used to analyze the obtained TDGs in time series to detect abnormal TDGs. There are several steps in the anomaly detection process.

Step 1: Sampling network traffic and generating network flows.

Step 2: Creating TDG (Dot format) from network flows in time sampling intervals.

Step 3: Calculating adjacency matrices of the TDG and calculating graph metrics of the TDG.

Step 4: Comparing values of graph metrics of the TDG with their threshold value. If a value of one of graph metrics exceeds its threshold value then the TDG is an abnormal TDG, otherwise the TDG is a normal one.

The NG-MON system [18] has been used to capture and convert network traffic to flows and to store them in the database. From the flows stored in the database, TDGs were generated in DOT format [19]. Adjacency matrices of TDGs are calculated from DOT files to extract the metrics. If one of the metrics exceeds its threshold value then the corresponding TDG is classified as an abnormal TDG. Then, the anomalies can be reported to the network administrator as illustrated in Figure 4.

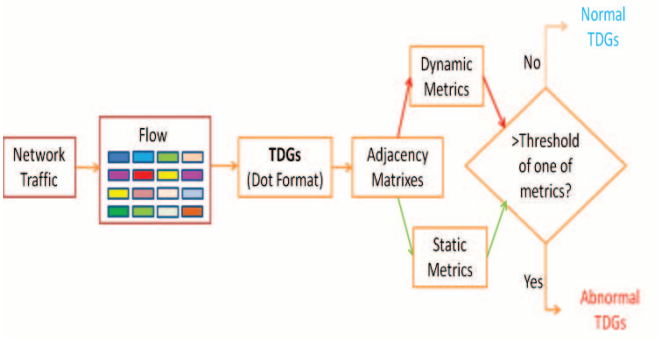

Figure 4. Detailed anomaly detection process.

To determine threshold values, we studied long network traffic records from the POSTECH Internet junction. In this work, we use static threshold values, which are determined from the comparison of normal and anomalous traffics. The threshold values need to be determined in each case according to the particular network environments.

To measure the proposed metrics (dK-2 distance metric) of consecutive TDGs, G_i and G_{i+1}, first the joint degree distribution algorithm, used in [20], was applied to the input adjacency matrix of two TDGs. Then the Euclidean distance between the two distributions JDD(G_i) and JDD(G_{i+1}) was calculated.

From the DOT format traffic graph, Graphviz library [21] is used to visualize the network traffic plot, thus making it easier to recognize and infer anomalous patterns. Several methods exist to visualize the layout in Graphviz such as neato, dot, twopi, circo, fdp and sfdp [21]. However, sfdp provides the best quality and low execution time. In the case of POSTECH traffic, it takes few seconds with a standard 2.4 GHz Core 2 Duo and 3GB of RAM computer to generate traffic figures.

4.2 Attack identification

Abnormalities occur in the network traffic by many reasons such as DoS/DDoS and Internet worm. The purpose of attack identification process is finding out causes of anomalies after detecting abnormal traffic.

4.2.1 Attack pattern

To obtain an attack structure pattern, the attack TDGs were generated, as discussed in section 4.1, from the attack traffic trace (Fig. 5). In this paper, DDoS attack pattern was obtained from DDoS CAIDA trace [2] (Fig. 6). Several attack structure patterns were also introduced in [11]. In the latter, attacks were identified using graph matching methods.

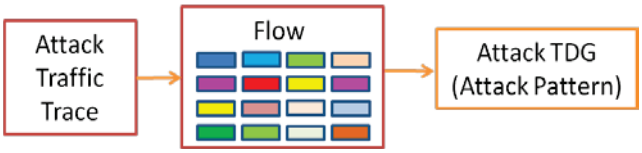

Figure 5. Attack pattern generation process.

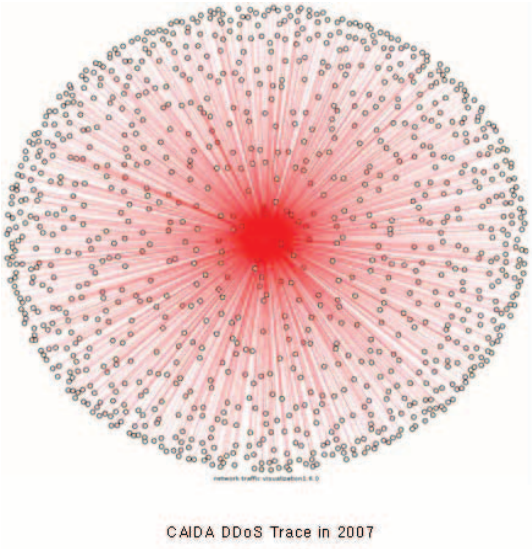

Figure 6. TDG visualization of DDoS attack pattern.

4.2.2 Attack detection

In our work, attack patterns contained in the abnormal graphs (TDGs) are identified by checking graph-sub-graph isomorphism property between the set of attack pattern graphs and the abnormal graphs (Fig. 7). To verify graph-sub-graph isomorphism, VF2 algorithm was employed because of VF2 is the most efficient among different isomorphism algorithms [22], [23].

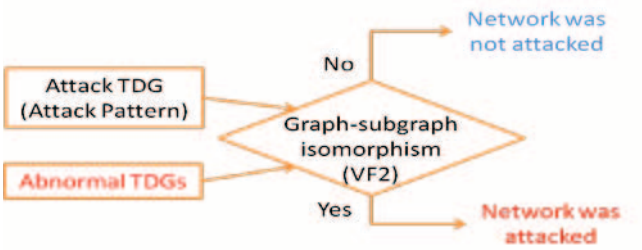

Figure 7. Attack identification process.

In [11], VF2 was also used to directly identify attack patterns in networks. However, here the graph matching method VF2 is used in a different way. We apply VF2 to identify attack patterns in abnormal TDGs because attack patterns are located in the abnormal traffic, thus making the algorithm much faster.

5. VALIDATION

To validate the correct working of our approach in anomaly detection and attack identification, we used two traffic traces: POSTECH's trace of July 7 2009 and CAIDA DDoS trace in 2007 [2].

The POSTECH trace contains traffic of a famous DDoS attack on July 7, 2009. This trace was captured during one hour activity. During July 2009, major government and commercial websites in South Korea had been subjected to heavy DDoS attacks which, according to intelligence agencies in South Korea and the United States, were probably launched by a special cyber warfare unit belonging to North Korean Army. At that time, many computers in POSTECH's network campus were zombies.

POSTECH's trace with one minute cycle is also sampled and a TDG is generated every minute. In this section, we just focus on two metrics, Kmax (static metric) and the new metric that we propose, i.e. the dK-2 distance (dynamic metric) to show the effectiveness of dK-2 distance in anomaly detection. Other metrics were analyzed and measured in [10].

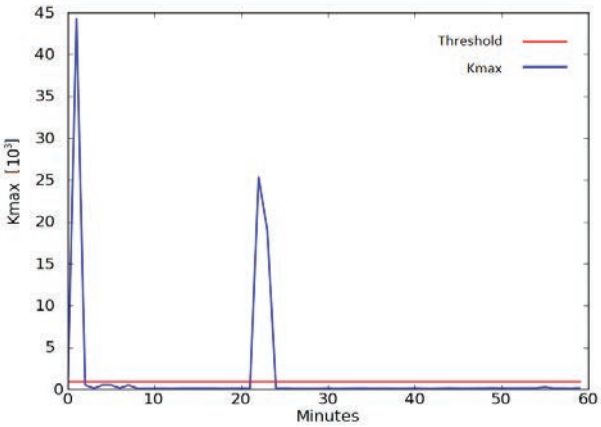

Figure 8. Kmax value over time of POSTECH's trace on 2009.7.9.

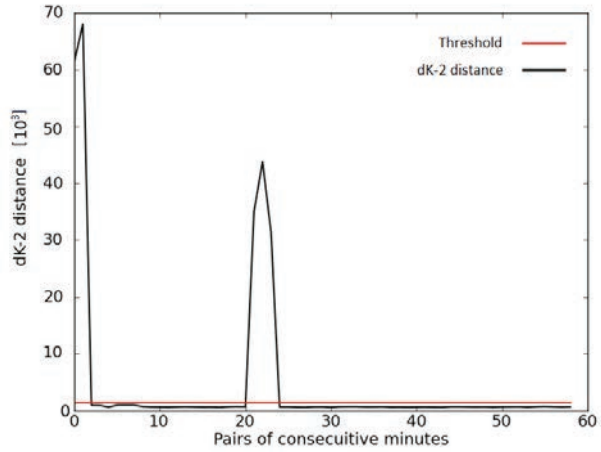

Figure 9. dK-2 distance value over time of POSTECH's trace on 2009.7.9.

Kmax shown in Figure 8 and dK-2 distance shown in Figure 9 metrics change dramatically in the 2nd, 22nd and 23rd minute of the trace. We can conclude that at these times, an anomaly occurred in the network traffic of POSTECH.

Using simple statistics such as counting the number of packets as illustrated in Figure 10, the number of flows and the number of bytes to analyze POSTECH's trace is difficult to detect anomalies. However, the anomalies are recognized easily by using the Kmax and dK-2 distance metrics to decompose the trace.

In the visualization of the 22nd minute and 23rd minute of POSTECH's trace as illustrated in Figure 11, the difference between the abnormal and normal graphs of the trace can be easily recognized.

By using graph matching, the DDoS attack pattern was identified at 2nd, 22nd and 23rd minute of the POSTECH trace. From flows data of that pattern, we discovered a botnet that communicated with one server over TCP port 6667 which is the standard port used for Internet Relay Chat (IRC) traffic [24]. We conclude that the attack in the POSTECH trace is an IRC-based botnet DDoS attack.

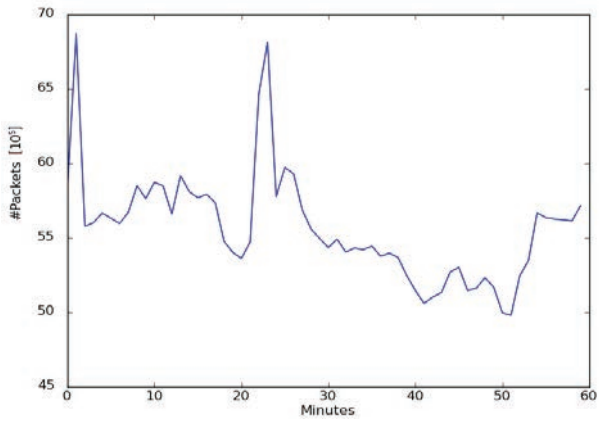

Figure 10. The number of packets over time of POSTECH's trace in 2009.7.9

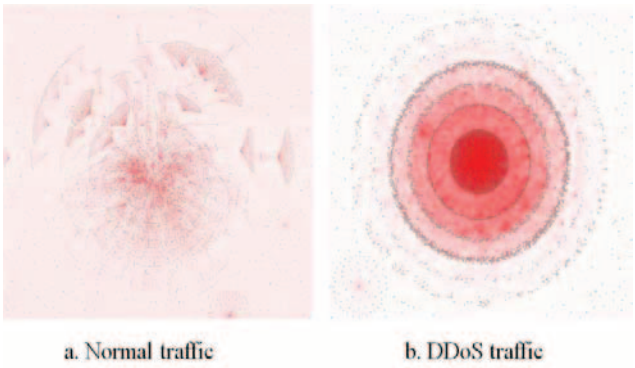

Figure 11. TDG visualization of POSTECH's trace on 2009.7.9.

6. CONCLUDING REMARKS

In this paper, we have investigated the problem of traffic anomaly detection using graph theory. We have proposed a new approach for traffic anomaly detection in which network traffic is modeled as a graph by using the TDG method, and the graph TDGs are studied and explored over time. We have validated the approach on a time series of traffic taken at POSTECH in 2009 using a high performance measurement system NG-MON. Our dataset includes actual anomalies (DDoS attack in 2009). In terms of accuracy, our approach detected anomalies in the POSTECH trace with 100% accuracy. The dK-2 distance metric was shown to be effective in detecting anomalies. By using DDoS CAIDA trace from 2007, the approach can be used to identify similar DDoS attack patterns observed in anomalies of POSTECH trace in 2009.

In future work, we plan to investigate other anomalous traffic traces to explore other attack patterns. We also plan to further validate our approach with other traces and compare the performance of our method with those of other methods.

Beside botnet DDoS attack, our approach can detect other types of DoS/DDoS attack such as host scanning, DNS amplification attack and UDP flooding. However, the approach cannot detect port scanning attack because a TDG describes the connections among hosts and TDG does not describe end-to-end connection. Therefore, we also plan to implement combining our approach with another method to detect port scanning attack.

REFERENCES

[1] Iliofotou, M., Pappu, P., Faloutsos, M., Mitzenmacher, M. Singh, S. and Varghese, G. Network monitoring using traffic dispersion graphs (tdgs). In Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (IMC '07). ACM, New York, NY, USA, 2007, 315-320. DOI= http://doi.acm.org/10.1145/1298306.1298349

[2] Hick, P., Aben, E., Claffy, K. and Polterock, J.2007.The CAIDA DDoS Attack 2007 Dataset. http://www.caida.org/data/passive/ddos-20070804dataset.xml (accessed on 2011-05-10).

[3] Eberle, W. and Holder, L. 2007. Anomaly detection in data represented as graphs. Intelligent Data Analysis, 2007, vol. 11, pp. 663–689.

[4] Noble, C.C. and Cook, D. J. 2003. Graph-based anomaly detection. In Proceedings of the 9th ACMSIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, 2003, 631–636.

[5] Cook, D.J. and Holder, L.B. 2000. Graph-based data mining. IEEE Intelligent Systems, 2000, 15(2), pages 32-41.

[6] Ide, T. and Kashima, H. 2004. Eigenspace-based anomaly detection in computer systems. Conference on Knowledge Discovery in Data: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 2004, 22(25):440–449.

[7] Sun, J., Faloutsos, C., Papadimitriou, S. and Yu, P.S. 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, 2007, pages 687–696.

[8] Chandola, V., Banerjee, A. and Kumar, V. 2009. Anomaly detection: A survey. *ACM Comput. Surv,* (July 2009), Article 15, 58 pages. DOI= http://doi.acm.org/10.1145/1541880.1541882

[9] Zhou, Y., Hu, G., He, W. 2009. Using graph to detect network traffic anomaly. Conference on Communications Circuits and Systems, 2009. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5250514

[10] Iliofotou, M., Faloutsos, M. and Mitzenmacher, M. 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). ACM, New York, NY, USA, 2009, 241-252. DOI= http://doi.acm.org/10.1145/1658939.1658967

[11] Godiyal, A., Garland, M. and John, C.H. 2010. Enhancing network traffic visualization by graph pattern analysis. https://agora.cs.illinois.edu/download/attachments/18744303/NetflowPatternGraph.pdf?version=1&modificationDate=1238354953000.

[12] Papadimitriou, P., Dasdan, A. and Garcia-Molina, H. 2008. Web graph similarity for anomaly detection. Technical Report, Stanford University, 2008. http://dbpubs.stanford.edu/pub/2008-1.pdf

[13] Sala, A., Cao, L., Wilson, C., Zablit, R., Zheng, H. and Zhao, B. 2010. Measurement-calibrated graph models for social network experiments. In WWW, 2010, pages 861—870.

[14] Mahadevan, P., Hubble, C., Krioukov, D., Huffaker, B., Vahdat, A. 2007. Orbis: rescaling degree correlations to generate annotated Internet topologies. SIGCOMM Computer Communications Review, 2007, 325–336.

[15] Conte, D., Foggia, P., Sansone, C., Vento, M. 2004. Thirty years of graph matching in pattern recognition. International Journal of Pattern Recognition and Artificial Intelligence, 2004, 18 (3) 265–298.

[16] Bunke, H. 2000. Recent developments in graph matching. In Proc. 15th International Conference on Pattern Recognition, 2000, pages 117-124.

[17] Cordella, L.P., Foggia, P., Sansone, C. and Vento, M. 2004. A (Sub)Graph isomorphism algorithm for matching large graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* 26, 10 (October 2004), 1367-1372. DOI= http://dx.doi.org/10.1109/TPAMI.2004.75

[18] Hong, J.W. 2004. Internet traffic monitoring and analysis using NG-MON. POSTECH, Advanced Communication Technology. The 6th International Conference, 2004, Volume: 1, page(s): 100- 120. http://ieeexplore.ieee.org/iel5/9073/28786/01292840.pdf

[19] Gansner, E., Koutsofios, E. and North, S.Drawing graphs with dot. http://www.graphviz.org/Documentation/dotguide.pdf

[20] Whitney, D. 2008. Basic Network Metrics. Lecture note. http://ocw.mit.edu/courses/engineering-systems-division/esd-342-network-representations-of-complex-engineering-systems-spring-2010/readings/MITESD_342S10_ntwk_metrics.pdf

[21] Graphviz – graph visualization software. http://www.graphviz.org/

[22] Foggia, P., Sansone, C. and Vento, M. 2001. A Performance Comparison of Five Algorithms for Graph Isomorphism. Proc. 3rd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition, 2001.

[23] Voss, S. and Subhlok, J. 2010. Performance of general graph isomorphism algorithms. Technical Report UH-CS-09-07, University of Houston, 2010.

[24] Kristoff, J. 2004. Botnets. 32nd Meeting of the North American Network Operators Group, October, 2004