FlowSummary: Summarizing Network Flows for Communication Periodicity Detection

Neminath Hubballi, Deepanshu Goyal

Department of Computer Science & Engineering, Indian Institute of Technology Guwahati,
Assam 781039, INDIA {neminath,g.deepanshu}@iitg.ernet.in

Abstract. Data summarization is an important technique to understand large datasets and discover useful patterns. In this paper we formulate a problem of summarizing network flow data to discover periodic communication behavior. An efficient implementation method for discovering periodic patterns is described in this paper and it has successfully discovered such patterns in a simulated and real application.

1 Introduction

Data summarization is an important technique for analyzing large dataset. It represents a large set of data points in a compressed format. There are many domain specific data summarization techniques proposed in the literature [10, 9]. Similarly there are summarization techniques proposed for cyber attack detection [3] in the literature. There are limitations to existing techniques such as information loss and have limited scope of detecting a specific subset of cyber attacks. However in this paper we propose a method which is lossless and can be useful in detecting a range of cyber attacks. The summarization technique detects periodic communication behaviors of a host. It is shown that a large set of applications (both good and bad) exhibit periodic communication behaviors [2]. These are mostly automated communications and not driven by user actions. For example, an application may be polling the server for possible new updates. A backup database may be synchronizing the data with a main server at periodic intervals. On the other hand these automated communications may be due to a malicious application running in the host exporting sensitive data. From a system administrator's perspective identifying such communications is important as it will have implications from security and privacy. The proposed summarization technique is useful for detecting network anomalies thus the works in this category are briefly reviewed. There are many anomaly detection techniques such as only packet header based [8, 6] flow based [5, 7, 3] and protocol anomaly detection [4]. Header based anomaly detection techniques derive some statistics based on fields available in packet header. These statistics forms a feature vector and using a suitable data mining algorithm, abnormal behaviors can be detected. Flow based techniques on the other hand rate the connection between two communication end points. A suitable rating function is used to differentiate abnormal connections w.r.t normal connections. Protocol anomaly detection techniques detect abnormalities against the specified protocol behavior.

2 Periodicity Detection

The proposed technique is based on the fact that, periodic communications exhibit very low variance and standard deviation considering their inter time differences.

Table 1. Transactions of Flows

Flow	Source IP	Destination IP	Time
F_1^1	123.123.123.123	208.208.208.208	01:02:01,01-01-2013
F_2^1	123.123.123.123	208.208.208.208	01:12:01,01-01-2013
F_1^2	123.127.123.127	208.109.208.109	06:12:01,01-01-2013
F_3^1	123.123.123.123	208.208.208.208	01:22:01,01-01-2013
F_2^2	123.127.123.127	208.109.208.109	06:32:01,01-01-2013
F_3^2	123.127.123.127	208.109.208.109	06:52:01,01-01-2013

On the other hand communication flows of random nature exhibit very high variance. Thus we use the standard deviation of communications established between end points (identified by IP addresses) as a measure to decide whether they have periodic communication or random communication. Standard deviation is given by the Equation 1.

$$SD = \sqrt{\Big(\frac{1}{M-1}\sum_{i=1}^{M}(X_i - \mu)^2\Big)} \tag{1}$$

In Equation 1, for any random variable X, X_i is the value taken by that random variable in one observation and μ is the mean of all the values taken by that random variable.

Table 2. Transactions with $DiffTime$

Flow	Source IP	Destination IP	$DiffTime$
F_1^1	123.123.123.123	208.208.208.208	NA
F_2^1	123.123.123.123	208.208.208.208	10
F_1^2	123.127.123.127	208.109.208.109	NA
F_3^1	123.123.123.123	208.208.208.208	10
F_2^2	123.127.123.127	208.109.208.109	20
F_3^2	123.127.123.127	208.109.208.109	20

We define the communication end points as two hosts identified by two distinct IP addresses which have exchanged at least one packet between them. We say a flow has started whenever there is an exchange of packets between two hosts and this communication is represented in the form of $SrcIP$-$DestiIP$ pairs indicating source and destination IP addresses of hosts involved. We maintain four parameters along with these pairs and subsequently show that these parameters are sufficient to describe about the nature of interaction between the two hosts. Let the host at which network traffic is collected be $\mathcal{H}$. This can be done using a suitable software like tcpdump [1]. Let P_1, P_2, ... ,P_N be a series of packets exchanged with $\mathcal{H}$ representing several communications over a period of time and $IP_1, \ldots IP_I$ ($1 \leq I \leq N$) be the total number of peer hosts involved in communication with $\mathcal{H}$. For each peer host identified by an IP address IP_K ($1 \leq K \leq I$) there may be one or more flows and let these flows be represented as

$F_1^{IP_K}, F_2^{IP_K}, \ldots, F_M^{IP_K}$ ($M \leq N - I + 1$). Let $t_1^{IP_K}$, $t_2^{IP_K}$, ... ,$t_M^{IP_K}$ be the corresponding timestamps of these flows (identified by the time-stamp of first packet in the flow). A set of flows can be tabulated as in Table 1 and we refer this as transaction data.

A flow indicates the beginning of a new communication and our aim is to identify the nature of such interactions in this network traffic. To achieve this we derive a parameter called $DiffTime$ - which is a time-stamp difference between two successive flows i.e., the difference between $t_l^{IP_K}$ and $t_{l+1}^{IP_K}$ in the time series. With $DiffTime$ calculated for each flow, the transactions in Table 1 can be shown as in Table 2. In Equation 1 $DiffTime$ of individual flows is considered for calculating the standard deviation. Standard deviation calculation requires all the $DiffTime$ values for the entire period. Normally to detect periodic communications a hosts traffic need to be collected for a longer duration. Network traffic is bulky for storage and processing, thus an efficient technique is provided for standard deviation calculation. This technique is motivated by a similar approach for summarization [11]. To describe the nature of communication we summarize the flows between two hosts as shown in Equation 2. And call it as a $FlowSummarry$

$$SrcIP, DstIP, LS, SS, M, t_l \tag{2}$$

where
$SrcIP$ - Source IP address (sender of the first packet in a flow)
$DstIP$ - Destination IP address (recipient of the first packet in a flow)
LS - Linear sum of $DiffTime$ i.e., $\sum_{l=1}^{M}(t_{l+1}^{IP_K} - t_l^{IP_K})$
SS - Squared sum of $DiffTime$ i.e., $\sum_{l=1}^{M}(t_{l+1}^{IP_K} - t_l^{IP_K})^2$
M - Number of flows seen between the two hosts during the period
t_l - Timestamp of the last flow between two hosts
Using the information in $FlowSummarry$, it is possible to calculate standard deviation of inter time difference $DiffTime$. Mean μ is calculated as shown in Equation 3.

$$Mean = \mu = \frac{LS}{M} \tag{3}$$

Similarly the calculation of variance can be deduced to Equation 7 from Equation 1

$$SD = \sqrt{\Big(\frac{1}{M-1}\sum_{i=1}^{M}(X_i - \mu)^2\Big)} = \sqrt{\Big(\frac{1}{M-1}\Big[\sum_{i=1}^{M}(X_i^2 + \mu^2 - 2X_i\mu)\Big]\Big)} \tag{4}$$

$$SD = \sqrt{\Big(\frac{1}{M-1}\Big[\sum_{i=1}^{M}X_i^2 + \sum_{i=1}^{M}\mu^2 - 2\sum_{i=1}^{M}X_i\mu\Big]\Big)} \tag{5}$$

$$SD = \sqrt{\Big(\frac{1}{M-1}\Big[\sum_{i=1}^{M}X_i^2 + M*\mu^2 - 2\mu\sum_{i=1}^{M}X_i\Big]\Big)} \tag{6}$$

$$SD = \sqrt{\Big(\frac{1}{M-1}\{SS + M*(\frac{LS}{M})^2 - 2*LS*\frac{LS}{M}\}\Big)} \tag{7}$$

With respect to a host $\mathcal{H}$, all the flows originating from the host will have $\mathcal{H}$'s IP address as source IP address and different destination addresses. On the other hand all the flows intended towards $\mathcal{H}$ will have different source addresses and $\mathcal{H}$'s IP address as destination address. This allows us to represent the interaction of $\mathcal{H}$, in the form of a graph.

3 Experimental Results

There are two types of experiments done, first is on a simualted application behavior and the other is with a real application exhibiting periodic behavior. For both the experiments network traffic was logged using tcpdump on a host connecting to ISP via a DSL broadband.

Table 3. Statistics for Artificial Dataset

Source IP	Destination IP	Flow Count	Std Devtn
192.XX.XX.4	202.XX.XX.6	4025	2.90
202.XX.XX.6	192.XX.XX.4	4024	2.86
192.XX.XX.4	216.XX.XX.148	12	0.88
184.XX.XX.29	192.XX.XX.4	12	7.02
192.XX.XX.4	184.XX.XX.29	13	6.97
174.XX.XX.114	192.XX.XX.4	14	3.97
192.XX.XX.4	174.XX.XX.114	14	3.99
173.XX.XX.51	192.XX.XX.3	14	5.05
192.XX.XX.4	74.XX.XX.54	14	5.74
74.XX.XX.54	192.XX.XX.4	14	5.85
192.XX.XX.4	115.XX.XX.38	15	1.54
202.XX.XX.246	192.XX.XX.4	15	1.83
192.XX.XX.4	119.XX.XX.10	16	0.25
119.XX.XX.10	192.XX.XX.4	16	0.25
209.XX.XX.191	192.XX.XX.4	18	0.24
74.XX.XX.162	192.XX.XX.4	20	4.56
192.XX.XX.4	74.XX.XX.162	20	4.58
174.XX.XX.139	192.XX.XX.4	21	1.41
115.XX.XX.38	192.XX.XX.4	21	1.53
174.XX.XX.139	192.XX.XX.4	21	1.74
192.XX.XX.4	50.XX.XX.165	26	6.49
50.XX.XX.165	192.XX.XX.4	26	7.50

In the first experiment, periodic communication behaviour was simulated with a Perl script generating requests for a web page for every 100 seconds. To make it close to real world traffic a small random number (between 1 to 10) was added to the base period so that there is a slight variation in the initiated communication. This expriment was run for 5 days along with other normal user usage. For the second experiment network traffic for real application having periodic behavior for 7 hours along with other normal browsing activity of end user was collected. A sport web page which shows live score card during a cricket match and also live text commentary was chosen as a real application. Normally such pages refresh the scorecard automatically using a html refresh directive (with a timer) thus the communication between web site and the local computer is periodic in nature and this exactly suit our experiments.

1)Artificially Generated Dataset: By analyzing simulated traffic we generated $FlowSummarry$ as described in previous section. For this experiment we assume a standard deviation of less than 10 to be a periodic communication. Table 3. shows list of periodic communications identified by $FlowSmmary$. First and second column represent the source and destination IP addresses of flows and third column shows the number of flows between these end hosts and last column shows the standard deviation. First two rows in Table 3. represent two genuine periodic communications with their standard deviations. The first entry shows communication from the host to the web server and second entry shows the connection in other direction. We can notice that, standard deviation for both the communications is around 2.9. with total number of flows being around 4000, in each direction.

Table 4. $FlowSummary$ Groups

Instances	Flow Count	SD Range	FP
337	001 - 002	NA	NA
220	003 - 005	000.47 - 10366.00	NA
178	006 - 010	000.30 - 08640.00	NA
089	011 - 015	002.32 - 05519.60	10
044	016 - 020	002.32 - 05212.53	05
030	021 - 025	001.41 - 04513.13	03
026	026 - 030	006.49 - 01044.70	02
012	031 - 035	005.72 - 02482.44	00
011	036 - 040	200.69 - 02415.39	00
008	041 - 045	011.17 - 00801.18	00
005	046 - 050	010.53 - 00085.00	00
013	051 - 100	031.90 - 01886.23	00
002	>100	002.86 - 00002.90	00
975	Total		

Table 4. shows the summary of communications during the period of monitoring. We observed a total of 975 distinct TCP flows i.e 975 distinct IP addresses being contacted by the host monitored (there are several flows for each of these IP addresses) in the traffic. This table shows the groups of flows based on the number of flows seen for a particular end host pair. Column 1 shows number of distinct end host pairs and column 2 shows the range for number of flows for a particular pair. For example row 1 is interpreted as follows. There are 337 distinct $SrcIP$-$DstIP$ pairs each having either one or two flows between them. Column 3 in this table shows the standard deviation range for the corresponding $FlowSummarry$ group. Column 4 shows the number of false positives in that group. These are the flows identified to be periodic by the algorithm and essentially the same flows shown in Table 3. except the first 2 entries. To qualify for periodic communication we use minimum threshold for number of flows to be at least 10. As we notice from the table nearly 60% of these $FlowSummarry$s are having a flow count of less than 5 (majority of them with count of 1). A closer observation, revealed that majority of these is communications originated from searches. This is because when the user is searching for particular information, within a short period of time it is likely that she searches for it repeatedly (till she finds useful information that she wanted). Typical user behavior is to open the search page and see if the information is what she wanted. Once the page is opened a connection is established with corresponding server.

Table 5. Statistics for Real Application Dataset

Source IP	Destination IP	Flow Count	Std Devitn
192.XX.XX.4	122.XX.XX.184	417	0.40
122.XX.XX.184	192.XX.XX.4	415	0.41
192.XX.XX.4	96.XX.XX.48	6	7.9
96.XX.XX.48	192.XX.XX.4	6	8.5
192.XX.XX.4	119.XX.XX.10	4	9.1
119.XX.XX.10	192.XX.XX.4	4	9.2
192.XX.XX.4	115.XX.XX.6	6	9.9
115.XX.XX.6	192.XX.XX.4	6	9.9

Out of 975 distinct communications a total of 20 false positives and there are 2 true positives. This is considered to be reasonably good performance even when we pick the threshold for standard deviation as high as 10 when the real periodic flows standard deviation is around 3. By reducing its value we can get a better performance. In other words this threshold governs the performance and we leave it to the wisdom of system administrator to fix an appropriate value for this threshold.

2) Real Application Dataset:

Similar to the last experiment; we tabulate the connection flows with standard deviation up to 10 in this case too. Table 5. shows periodic flows identified by the algorithm. First two entries in this case are two true positives corresponding to the sport web page and remaining are false positives. Although the number of false positives reduced compared to the previous case but this dataset represent less hours of activity in comparison to the previous one, thus reducing the traffic related to searches which were major contributors of false positives.

4 Conclusion

In this paper a data summarization problem is formulated for communication periodicity detection and an efficient implementation technique is described. Proposed method is a lossless summarization technique and is evauluated with traffic of simulated and real application having periodic communications.

References

1. http://www.tcpdump.org.
2. G. Bartlett, J. Heidemann, and C. Papadopoulos. Using low-rate flow periodicities for anomaly detection: Extended. Technical report, University of Southern California, 2009.
3. V. Chandola and V. Kumar. Summarization- compressing data into an informative representation. *Knowledge of Information Systems*, 12(3):355–378, 2007.
4. M. Collins. *A Protocol Graph Based Anomaly Detection System*. PhD thesis, School of Electrical and Computer Engineering, Carnegie Mellon University, 2008.
5. L. Ertz, E. Eilertson, A. Lazarevic, P. Tan, V. Kumar, and J. Srivastava. Chapter 3: The MINDS- Minnesota Intrusion Detection System, 2004.
6. N. Hubballi, S. Biswas, and S. Nandi. Towards reducing false alarms in network intrusion detection systems with data summarization technique. *Security and Communication Networks*, 6(3), 2013.
7. M. Kim, H. Kang, S. Hong, S. Chung, and J. Hong. A flow-based method for abnormal network traffic detection. In *IEEE/IFIP Proceedings of the Network Operations and management Symposium*, pages 217–228. Springer-Verlag, 2004.
8. M. V. Mahoney and P. K. Chan. PHAD: Packet Header Anomaly Detection for identifying hostile network traffic. Technical report, Florida Institute of Technology, 2001.
9. M. Mampaey and J. Vreeken. Summarizing categorical data by clustering attributes. *Data Mining and Knowledge Discovery*, 26(1):130–173, 2013.
10. T. Mielikainen. *Summarization Techniques for Pattern Collections in Data Mining*. PhD thesis, University of Helsinki, 2005.
11. T. Zhang, R. Ramakrishnan, and M. Livny. Birch: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.