Detecting Anomalies in Netflow Record Time Series by Using a Kernel Function

Cynthia Wagner and Thomas Engel

University of Luxembourg - SnT,
Campus Kircherg, L-1359 Luxembourg, Luxembourg
{cynthia.wagner,thomas.engel}@uni.lu
http://www.securityandtrust.lu

Abstract. This paper presents current work for the detection of anomalies in Netflow records by leveraging a kernel function method. Netflow records are spatially aggregated over time, such that the designed kernel function can capture topological and quantitative changes in network traffic time series.

Keywords: Netflow records, Aggregation, Kernel Function.

1 Introduction

Network operators are faced to new challenges on a daily base, while trying to keep their network in good health. Anomalies in networks can be triggered from various sources, ranging from simple network failures (e.g. hardware or software crash,...) to harmful attacks launched against the network (e.g. Worms, Denial-of-Service,...). To identify the anomalies, an available resource is Netflow records due to their compact format, but on ISP- level the storage of all Netflow records needs large storage capacities, since peak-rates of 60 000 flows/second are quite common. Therefore, the need for representing times series of Netflow records in an aggregated form is needed. In this paper, an accurate kernel method is introduced that captures topological and quantitative changes in aggregated network traffic with aim of detecting anomalies or attacks.

This paper is organized as follows: Section 2 describes the model. Section 3 presents experimental results and relevant work is presented in section 4. Conclusions and future work are presented in section 5.

2 The Model

2.1 Aggregation of Netflow Records

Spatial and temporal aggregation was first presented in [1], [5] for full packet captures. The aim of spatial aggregation is to extract host IP address (from source (src) or destination (dst)) and volume information from Netflow records and to spatially aggregated this data into tree-like profiles. A profile holds nodes,

R. Sadre et al. (Eds.): AIMS 2012, LNCS 7279, pp. 122–125, 2012.

which represent IP subnets or IP addresses. An IP address is split into two parts, $IP_{src,dst}$=(prefix, prefixlength). By this, the tree-like structures respect the IP address hierarchy space. The processing task analyses a new Netflow record by matching source/destination IP addresses with the most similar node, sharing the longest common IP prefix. If there is a matching node, the volume part ($vol_{src,dst}(i)$) is updated. When no match is found, then a new node and a branching point in the parent node are generated. This step is done by using Patricia trees [10] with a fixed tree size (N_{MAX} nodes).

For a spatially aggregated tree T, traffic profiles can be defined as, N a set of nodes for source or destination, where $T = \{n_1, ..., n_N\}$ with $n_i =< prefix_i, prefix\_length_i, vol_i >$ and a relation $child : T \rightarrow \mathcal{P}(T)$, providing a set of child nodes for a given node. Temporal aggregated profiles are time series of profiles, defined as:
$\{< T_1^{src,byt}, T_1^{dst,byt}, T_1^{src,pkt}, T_1^{dst,pkt} >, \ldots, < T_M^{src,byt}, T_M^{dst,byt}, T_M^{src,pkt}, T_M^{dst,pkt} >\}$ where $T_i^{src,byt}$ is a profile in a time window i for source IP address information and $T_i^{dst,byt}$ for destination information. Both use also volume in terms of bytes.

2.2 The Kernel Function Model

Kernel functions are accurate mathematical tools for evaluating complex data. In [13], [12], a kernel function can be defined as a mapping from an input space X, s.t. $K : X \times X \rightarrow [0, \infty[$, towards a similarity score $K(x, y) = \sum_i \phi_i(x)\phi_i(y) = \phi(x) \cdot \phi(y)$, with a feature vector $\phi_i(x)$ over x. For tracking quantitative and topological pattern changes in profiles, a new kernel function is introduced,

$$K_{src,dst}(T_n, T_m) = \sum_{i \in T_n^{src,dst}, j \in T_m^{src,dst}} s_{src,dst}(i, j) \times v_{src,dst}(i, j) \tag{1}$$

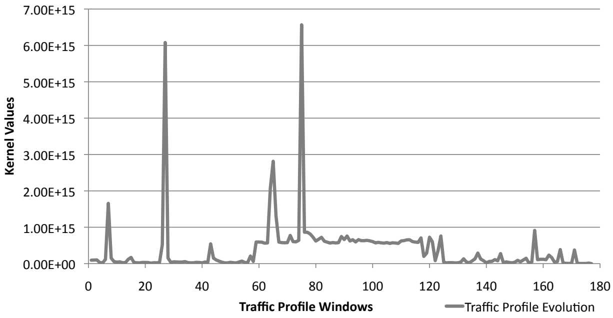
The first part $s_{src,dst}(i, j)$ is used for modeling changes in the network topology by analyzing node suffix lengths.

$$s_{src,dst}(i, j) = \begin{cases} \frac{2^{prefixlength_j}}{2^{prefixlength_i}} \texttt{ if } prefix_i \texttt{ prefix of } prefix_j \\ \frac{2^{prefixlength_i}}{2^{prefixlength_j}} \texttt{ if } prefix_j \texttt{ prefix of } prefix_i \\ 0 \texttt{ otherwise} \end{cases} \tag{2}$$

The second part $v_{src,dst}(i, j)$ is a Gaussian kernel treating traffic volume changes in tree nodes, defined as $v_{src,dst}(i, j) = exp\left(-\frac{|vol(src,dst)_i - vol(src,dst)_j|^2}{\sigma^2}\right)$. A more comprehensive version of the kernel function is presented in [14]. The kernel function takes as input successive traffic profiles and determines the similarity between these profiles and the higher the K-value, the more similar are the successive trees.

3 Experimental Results

For the experiments a small data set (of 5 minutes duration) provided by a local ISP from Luxembourg has been used (see Fig.1). The aggregation method has



# Flows	3 907 859
# IP Addresses	Source: 250 314 Destination: 235 120
# bytes Avg. bytes/Flow	24.1 GB 6 8829
# Packets Avg. Packets/Flow	38 132 130 9.76

Fig. 1. Attack Detection by Evaluating Netflow Records with the Kernel Function and the used Data Set

been applied to the data set to generate traffic profiles. The kernel function is applied to the traffic profiles for detecting similarities between these profiles. In Fig.1, the evaluation of applying the designed kernel function to aggregated Netflow records is presented. It can be seen that between profile 60 and 120 there was an attack. By a manual investigation of the data set for these time periods, it was observed that there was a UDP-flooding attack.

4 Related Work

Nowadays network architectures are monitored, such that an available source of information are IP flow records. In a work by [7], Netflow records are used to quantify the IPv6 deployment. Since large quantities of Netflow records are available, flow sampling is a common approach to reduce data [11], [3], but a major problem is to find good sampling rates. Another possibility to handle large quantity of data is to use aggregation, as used in this paper. In [1], [5], full packet captures are spatially aggregated into tree-like profiles. With aid of these generated traffic profiles, the authors were able to identify denial-of-service and flooding attacks. Besides the monitoring itself, the evaluation of such data is essential and a lot of techniques based on statistical evaluations exist, e.g. [8], [6]. More recent evaluation techniques refer to Machine Learning techniques, e.g. [9]. Kernel methods for example, a sub-domain of Machine Learning, are often applied as evaluation tools on large data sets to analyze this data on common pattern. In [2], parsed and pre-processed sentences are decomposed into tree-like structures and a kernel function is applied in order to detect similarities between these trees. In computer security, kernel methods are mainly used for intrusion detection and anomaly detection [4]. In [4], Support Vector Machines (SVMs) based on a tree kernel functions are used for classifying sequences of data.

5 Conclusion

In this paper, a new method for evaluating large quantities of IP flows has been presented. The contribution of the presented approach is twofold. First, the approach is based on a spatial aggregation technique that summarizes Netflow records into tree-like profiles. Second, with the design of an new kernel function, anomalies respectively attacks in Netflow data can be detected. Since IP address information is sensitive data, by evaluating data with the kernel function, the initial records is not needed anymore for further processing (e.g. SVMs), as it has been shown that the kernel function is able to highlight incidents. An optimization of the spatial-temporal aggregation and the kernel approach for traffic profiles is planned. Another future step is to integrate the kernel function within an online classication algorithm to run real-time network traffic analysis.

Acknowledgments. We address special thanks to RESTENA Luxembourg for their support.

References

1. Cho, K., Kaizaki, R., Kato, A.: Aguri: An Aggregation-Based Traffic Profiler. In: Smirnov, M., Crowcroft, J., Roberts, J., Boavida, F. (eds.) QofIS 2001. LNCS, vol. 2156, pp. 222–242. Springer, Heidelberg (2001)
2. Culotta, A., Sorensen, J.: Dependency Tree Kernels for Relation Extraction. In: 42nd Ann. Meet. on Association for Computational Linguistics, Spain (2004)
3. Estan, C.: Building a better NetFlow. In: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 245–256 (2004)
4. Kahn, L., Awad, M., Thuraisungham, B.: A new intrusion detection system using support vector machines and hierarchical clustering. The VLDB Journal 16(4), 507–521 (2007)
5. Kaizaki, R., Nakamura, O., Murai, J.: Characteristics of Denial of Service Attacks on Internet Using Aguri. In: Kahng, H.-K. (ed.) ICOIN 2003. LNCS, vol. 2662, pp. 849–857. Springer, Heidelberg (2003)
6. Karagiannis, T., Papagiannaki, K., Faloutsos, M.: BLINC: Multilevel Traffic Classification in the Dark. In: ACM SIGCOMM 2005, Pennsylvania, USA (2005)
7. Karpilovsky, E., Gerber, A., Pei, D., Rexford, J., Shaikh, A.: Quantifying the Extent of IPv6 Deployment. In: Moon, S.B., Teixeira, R., Uhlig, S. (eds.) PAM 2009. LNCS, vol. 5448, pp. 13–22. Springer, Heidelberg (2009)
8. Lakhina, A., Crovella, M., Diot, C.: Mining Anomalies Using Traffic Feature Distributions. In: ACM SIGCOMM 2005, Philadelphia, Pennsylvania, USA (2005)
9. McGregor, A., Hall, M., Lorier, P., Brunskill, J.: Flow Clustering Using Machine Learning Techniques. In: Barakat, C., Pratt, I. (eds.) PAM 2004. LNCS, vol. 3015, pp. 205–214. Springer, Heidelberg (2004)
10. Morrison, D.R.: PATRICIA- - Practical Algorithm To Retrieve Infromation Coded in Alphanumeric. ACM Journal 15(4), 514–534 (1968)
11. Paredes-Oliva, I., Barlet-Ros, P., Solé-Pareta, J.: Portscan Detection with Sampled NetFlow. In: Papadopouli, M., Owezarski, P., Pras, A. (eds.) TMA 2009. LNCS, vol. 5537, pp. 26–33. Springer, Heidelberg (2009)
12. Schoelkopf, B., Smola, J.: Learning with kernels, ch. 1-3. MIT Press (2002)
13. Vapnik, V.: Statistical Learning Theory. Wiley (1998)
14. Wagner, C., François, J., State, R., Engel, T.: Machine Learning Approach for IP-Flow Record Anomaly Detection. In: Domingo-Pascual, J., Manzoni, P., Palazzo, S., Pont, A., Scoglio, C. (eds.) NETWORKING 2011, Part I. LNCS, vol. 6640, pp. 28–39. Springer, Heidelberg (2011)