Anomaly Detection in Network Traffic using Jensen-Shannon Divergence

Osman Salem[1] and Farid Naït-Abdesselam[1] and Ahmed Mehaoua[1,2]
[1]LIPADE Laboratory, University Paris Descartes, France
[2]Division of IT Convergence Engineering, POSTECH, Korea
{osman.salem, farid.nait-abdesselam, ahmed.mehaoua}@parisdescartes.fr

Abstract—**Anomaly detection in high speed networks is well known to be a challenging problem. It requires generally the analysis of a huge amount of data with high accuracy and low complexity. In this paper, we propose an anomaly detection mechanism against flooding attacks in high speed networks. The proposed mechanism is based on Jensen-Shannon divergence metric over sketch data structure. This sketch is used to reduce the required memory, while monitoring the traffic, by maintaining them into a predefined fixed size of hash tables. This sketch is also used to develop a probabilistic model. The Jensen-Shannon divergence is used for detecting deviations between previously established and current distributions of network traffic. We have implemented our approach and evaluated it using real Internet traffic traces, obtained from MAWI trans-Pacific wide transit link between USA and Japan. Our results show that the proposed approach is scalable and efficient in detecting anomalies without maintaining per-flow state information.**

I. Introduction

Security threats for computer network have increased significantly, which include viruses, worm-based attacks, Distributed Denial of Service (DDoS), etc. DDoS through TCP SYN flooding is able to make silent any web site, especially with the use of BOTNETs (roBOT NETworks) containing large number of slave machines (zombies). Therefore, flooding attack needs to be accurately detected in order to cope with ongoing attack as soon as possible.

Two existing approaches for intrusion detection system: misuse detection and anomaly detection. The misuse detection is based on signature database to detect attacks, by matching data level applications to predefined rules for malicious activities, and it is ineffective with unknown signature attacks (zero day attacks). Anomaly detection approaches are based on building statistical profile for normal traffic behavior during a training phase, and raise an alarm when observing abrupt changes in network traffic. Anomalies are defined as heavy deviation from the regular traffic pattern. Anomaly based approaches have the ability to detect zero-day attacks that induce traffic variations. Our approach to detect flooding is based on anomaly detection, since flooding attacks will change some statistical parameters describing traffic variation.

Recently, many web sites suffered from SYN flooding attack that aims to exhaust server resources and to deny access for legitimate users, e.g. twitter has been driven offline due to DDoS in 2009, and the list of victim web servers is very long (CNN, Yahoo, Amazon, ebay, DoubleClick, PayPal, etc.). With the distributed nature of this coordinated attacks, and its impact on the performance of the routers, the detection and reaction mechanisms must be pushed to the core network (Backbone), or near the sources of attack. However, with the growing complexity in analyzing huge amount of data traffic in backbone network, the analysis of each traffic flows is unscalable and computationally expensive. On the other hand, the aggregation the whole backbone traffic in one flow result in one time series with large values, and unnoticed deviations by flooding attack.

Sketch (or random aggregation counters) has been proposed and used in [1], [2] for more grained detection than aggregating the whole traffic in one time series. To detect anomaly using Sketch, time series forecasting methods (ARIMA, Holt-winters, residual number of SYN, Heavy Hitter, etc.) have been applied [2]. In general, when deviation in a time interval is larger than an adaptive threshold that depends on past values, an alarm is raised.

In this paper, we propose a sequential analysis technique to identify anomalies in network traffic. We use Jensen-Shannon Divergence (JSD [3]) over Sketch data structure. JSD measures the difference between 2 set of probability values, and detects the deviation between these sets in normal and under flooding conditions.

The Sketch data structure is an array of hash table, where each bucket contains a counter for monitored parameter (number of: packets, SYN, Bytes, etc.) resulted from randomly aggregating traffic with the same hash value of their destination IP address (DIP). The basic idea behind Sketch is to reduce the required amount of memory through random aggregation of flows rather than tracking per flow state information. In our framework, Sketch is also used to establish a probabilistic model by exploiting the counters of hash table. As the time is divided into discrete interval, the divergence (JSD) between previously established probabilistic model for normal traffic and the current model in the current interval is used to detect anomaly.

The rest of this paper is organized as follow. Section II briefly reviews related work. Section III presents the Sketch data structure and the Jensen-Shannon divergence. Section IV describes our approach for anomaly detection. Section V presents experimental results of the proposed approach. Finally, in Section VI we present the conclusion remarks.

978-1-4577-2053-6/12/$31.00 ©2012 IEEE

II. Related works

Several approaches have been proposed for network anomaly detection, and they are based on different techniques, such as Haar-wavelet analysis [4], [5], entropy based method [6], [7], sequential change point detection methods with the CUmulative SUM (CUSUM) algorithm [8], [9], [10], adaptive threshold analysis [11], Exponentially Weighted Moving Average (EWMA) [12], Holt-Winters seasonal forecasting based methods [13], [2], data reduction techniques with sketches [1], [14], SNMP MIB statistical data analysis [15], Principal Component Analysis (PCA) [16], [17], etc.

Malicious activity usually provokes an abrupt change in the statistical values of the parameters describing the traffic, such as the NetScan produced by worms outbreak, that send a large number of SYN from the same source IP, to scan the network before propagation phase. In [18], [19], a non-parametric version of CUSUM is used to detect deviations in the number of connection requests sent by given source to different destinations. In [20], the CUSUM algorithm is used to detect SYN flooding over one time series resulted from aggregation the whole traffic in one flow. In [8] a comparison between CUSUM and adaptive threshold for the detection of SYN flooding is presented.

When early approaches for network anomaly detection were focused on the definition of models able to represent the traffic pattern [20], [10], other advanced work aggregates the whole stream of packets in one time series, and applies a change point detection algorithm to detect the instant of anomaly occurrence [8], [9]. The latter have a good performance in terms of spatial and temporal complexities, but present the drawback of aggregating all traffic in one flow, especially in backbone network, where low intensity attacks cannot be detected, i.e. flooding attack with intensity 10^6 packet/s does not produce a noticeable deviation when the total number of packet is greater than 10^9. Furthermore, these methods use static threshold for detecting anomalies, which is not adequate with traffic variations, and may induce false alarm and miss detection. In this paper, we overcome these problems through the use of Sketch and dynamic threshold.

Sketch data structure uses the random aggregation for more grained detection than aggregating whole traffic in one time series. It has been used to summarize monitored traffic in a fixed memory, and to provide scalable input for time series analysis. Many type of counters have been used for detecting change in different traffic features, such as the number of: SYN, packets, flows, bytes, etc. For example, the number of SYN per destination IP address can be used to detect SYN flooding, since distributed attacks are directed toward unique victim.

The proposed method extends all these previous works, through the use of JSD over sketch, sliding window and dynamic threshold for anomaly detection. The method can be adjusted to detect any type of flooding (UDP, ICMP, SYN, ACK, etc.). Due to space restriction, we will present the method for SYN flooding attack detection. However, the same procedure can be applied to detect flooding through UDP and ICMP by adding 2 additional counters in each cell of sketch.

Under SYN flooding attack, the distribution of number of SYN toward a specific IP address will deviate from previously learned distribution under normal traffic condition. However, with the difficulty of finding a distribution probability that fits to traffic characteristics (self similarity, heavy-tailed, and long range dependence, etc.), we will use the divergence to detect deviations between the probability values resulted from the shared counters of Sketch in different time interval.

The use of sliding window during training phase is proposed to absorb normal traffic variations between two discrete interval. It improves the efficiency of the detection algorithm by reducing the false alarm rate.

III. Background

In this section, we briefly survey the Sketch data structure and Jensen-Shannon divergence used in our framework.

A. Sketch

Sketch is a multi-stage Bloom filter used to randomly aggregate large set of data into a fixed small memory. Let $S = s_1, s_2, \ldots, s_n$ denotes the set of input stream, where each item $s_i = (\kappa_i, \nu_i)$ is identified by a key $\kappa_i \in U$, drawn from a fixed universe of items U. $\nu_i \in \mathbb{R}$ is the value associated with each key. In our model, we use $\kappa_i = DIP$ and $\nu_i = \#SYN$, as our goal is to count the number of SYN received by destination. The sketch data structure is made up of d hash arrays. The arrival of a packet with key κ_i increments its associated counter in the j^{th} hash table by ν_i ($C_{j,h_j(\kappa_i)} += \nu_i$), as shown in algorithm 1 and in Figure 1. The update procedure is realized by d different hash functions, chosen from the set of 2-universal hash functions $h_j(\kappa_i) = \{((a_j\kappa_i + b_j) \bmod P_U) \bmod w\}$, to uniformly distribute κ_i over hash tables and to reduce collisions. The parameter P_U is a prime number larger than the maximum number in the universe, where Mersenne prime numbers of the form $2^i - 1$ are generally chosen for fast implementation. a_j and b_j are random integers smaller than P_U, with $a_j \neq 0$. Using d hash functions, the probability that two keys are aggregated in the same bucket over the d hash table is $(1/w)^d$.

Algorithm 1 Sketches Update procedure

```
for all TCP SYN segment received during $T$ do
    for $i = 0$ to $d - 1$ do
        $j = univ\_hash_i(DIP)$;
        $C[i][j].counter += \nu_i$;
    end for
end for
```

As the size of DIP (IPv4) is 32 bits, the hash functions reduce the dimension of monitored space (2^{32}) to a fixed size w (e.g. $w = 2^{10} = 1024$), through the random aggregation of multiple IP addresses in the same bucket when the value resulted from hashing the addresses are the same ($h(IP1) = h(IP2) = j$). The counter in each bucket is used to derive a probability as the ratio of counter value to the sum of whole counters in one hash table (eq. 5). Jensen-Shannon divergence is used to detect change in distributions defined by two sketches resulted from two discrete intervals.

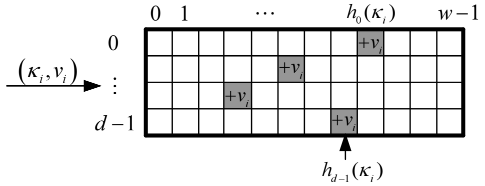


Fig. 1. Sketch data structure.

B. Jensen-Shannon Divergence

JSD [3] is used to measure the divergence between two sets of probability values. JSD is a smoothed version of Kullback-Leibler [3] divergence. For 2 discrete probability distributions $P = (p_1, p_2, \ldots, p_n)$ and $Q = (q_1, q_2, \ldots, q_n)$, with $p_i \geq 0$, $q_i \geq 0$ and $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i = 1$. The JSD divergence between P and Q is given by:

$$JSD(P, Q) = \frac{1}{2} KL(P, M) + \frac{1}{2} KL(Q, M) \tag{1}$$

Where $KL(P, Q)$ is the Kullback-Leibler distance between distributions P and Q:

$$KL(P, Q) = \sum_{i=1}^{n} p_i \log \frac{p_i}{q_i} \tag{2}$$

and M is the mean distribution of P and Q:

$$M = \frac{P + Q}{2} \quad \text{and} \quad m_i = \frac{p_i + q_i}{2} \tag{3}$$

JSD is the average of the KullBack-Leibler distances to the average distribution M. JSD can be expressed in the form:

$$JSD = \frac{1}{2} \left[\sum_{i=1}^{n} p_i \log \left(\frac{p_i}{m_i} \right) + \sum_{i=1}^{n} q_i \log \left(\frac{q_i}{m_i} \right) \right] \tag{4}$$

$JSD = 0$ iff P and Q are identical ($p_i = q_i$), and $JSD > 0$ when $P \neq Q$. It is a symmetric and bounded metric ($0 \leq JSD \leq \log(2)$) for orthogonal distributions ($p_i.q_i = 0$). As we aim to detect anomaly through the detection of deviations from normal traffic profile, the JSD determines the divergence between 2 probability distributions P and Q, which denote the distributions before and after the attack. JSD between P and Q must be near zero under normal traffic, with a large deviation (one spike) when distributions change occurs.

IV. Proposed approach

Our proposed approach to detect flooding attack over backbone networks (or high speed network), is based on the Sketch data structure and the Jensen-Shannon divergence. Firstly, the shared counters of the Sketch are continuously updated from the input data stream during a fixed time period T (e.g. $T = 1$ min). The key κ_i is the DIP address of packets and the reward $\nu_i = 1$ for SYN segment and zero for other traffic. At the end of each interval T, the probability $p_{i,j}$ is calculated as the ratio of each counter to the sum of all counters during this interval:

$$p_{i,j} = C_{i,j}.counter / \sum_{j=0}^{w-1} C_{i,j}.counter \tag{5}$$

Thus result in d distributions $(P_0, P_1, \ldots, P_{d-1})$, where the P_i is the set of probability $(p_{i,0}, p_{i,1}, \ldots, p_{i,w-1})$ resulted from the hash table i. The JSD between the current (Q_i) and a reference probability (P_i) measures is calculated for each line in the sketch. Deviation induces spike in $JSD_{i,k}$ values, and when more than L values of $JSD_{i,k}$ exceed a dynamic threshold, an alarm is raised.

To build a reference statistical model for the normal profile, we use the first few minutes as training phase, and a sliding window of fixed size of $N.T$ to update and calibrate the normal profile. To detect flooding attack, we use the probability set from previous window as reference distribution P_i, and the probability from current interval as Q_i distribution (as shown in figure 2). A large window size increases accuracy and the mean detection delay, and a tradeoff between accuracy and delay detection is required, as some attacks may be completely missed with large value of N, due to slow adaptation to traffic variations .

Sketch data structure holds the sliding window in each bucket of the 2D array. Each cell $C_{i,j}$ becomes a data structure, that contains: an array of fixed size for the past counters in each time slot in the sliding window ($W[N]$), two counters for current and sum of past values, and 2 float variables for holding the associated probabilities.

At the end of each interval $k.T$, d values of $JSD_{i,k}$ will be calculated (one per line). During malicious activity, $JSD_{i,k}$ incurs abrupt change, and when more than L value of $JSD_{i,k}$ exceed a dynamic threshold, we halt the updating procedure of sliding window until the end of attacks, to prevent the poisoning of statistical parameters in normal profile. If L values of $JSD_{i,k}$ exceeds the threshold for more than τ consecutive time slot, we trigger an alarm. This technique is used to avoid false alarm due to traffic variability (e.g. flash crowd), and the fact that DDoS attack last for more than 1 time slot to overload a server. When the attack stops, the divergence value will drift back near to zero, and the window continue to slide from the current time slot. Sliding window absorbs the normal variation of traffic, and performs better than two consecutive intervals.

In normal operation without anomaly, the first value of the array representing the sliding window will be dropped, other elements in the window are shifted one position ($C[i][j].W[k] \leftarrow C[i][j].W[k+1]$), and the current counter are pushed to the end of the window array. Afterward, probabilities $p_{i,j}$ are updated for each bucket.

To detect deviations in the time series of $J\hat{S}D_{i,k}$, we define a dynamic bound of $\mu_{k-1} + \alpha\sigma_{k-1}$. Significant deviation can be detected using the following equation:

$$JSD_{i,k} > \mu_{k-1} + \alpha\sigma_{k-1} \tag{6}$$

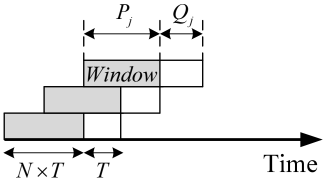


Fig. 2. Sliding window used for establishing normal profile.

where $JSD_{i,k}$ is the value of JSD in the current interval $k.T$ for line i, and μ_k & σ_k are the mean and the standard deviation respectively of $J\hat{S}D_{i,k}$ time series. $J\hat{S}D_{i,k}$ is the time series that contains only the value of $JSD_{i,k}$ that does not satisfy equation 6. μ_k and σ_k are updated dynamically using the Exponentially Weighted Moving Average (EWMA):

$$\mu_k = \beta\mu_{k-1} + (1-\beta)J\hat{S}D_{k-1} \tag{7}$$

$$\sigma_k^2 = \beta\sigma_{k-1}^2 + (1-\beta)(J\hat{S}D_{i,k} - \mu_k)^2 \tag{8}$$

The threshold is updated dynamically by adjusting the value of μ_k and σ_k as shown in equations 7 & 8. α is a parameter used for calibrating the sensitivity of the detection algorithm to variations, and to reduce the false alarm rate. Under normal traffic, divergence $JSD_{i,k}$ fall inside the bound of $\mu_{k-1} + 2\sigma_{k-1}$. When $JSD_{i,k}$ exceeds the dynamically updated threshold over L lines, an alarm is triggered. The decision function for alarms is given in equation 9. Another interesting approaches for estimating and adjusting dynamic threshold were proposed in [21], [22] for SIP INVITE flooding detection.

$$d(alarm_i) = \begin{cases} 1 & \textit{if } JSD_{i,k} > \mu_{k-1} + \alpha\sigma_{k-1} \\ 0 & \textit{Otherwise} \end{cases} \tag{9}$$

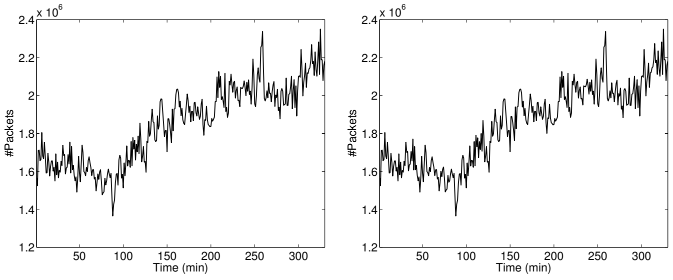
V. Experimental results

In this section, we evaluate the performance of JSD over Sketch for the detection of TCP SYN flooding attack. We use the real Internet MAWI [23] trans-Pacific traces, from 15/04/2010 07:30 to 12:45 as few hours in the life of the Internet, to test the efficiency of the proposed approach. IP addresses in the traces are scrambled by a modified version of tcpdpriv [23] tool, but correlation between addresses are conserved. We have analyzed this 5.5 hours of wide area network traces using the proposed approach, with a key of the sketch ($\kappa_i = DIP$), and a reward ($\nu_i = 1$) for SYN request only, and zero otherwise. Afterward, we inject real DDoS attacks with different intensity inside this trace to simulate distributed SYN flooding attacks.

For the used sketch, we set the width and number of hash functions to: $w = 1024$ and $d = 5$. The other parameters are set to: $N = 5$, $L = 4$, $\alpha = 3$, $\beta = 0.7$, $T = 1$. Finally, it is important to note that the processing of 5.5 hours of traces, takes a few minutes.

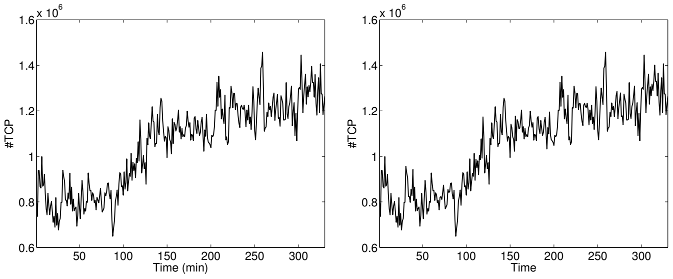
A. MAWI Traces

In order to proceed with test, we inject 8 real DDoS TCP SYN flooding attacks with different intensity in MAWI public traces (tcpdump files). These attacks are inserted each 30 minutes (at time t = 31, 71, 111, 151, etc.) and last for 10 minutes. The variations of the total number of packets before and after attacks are given in figure 3(a) & 3(b) respectively, with a time slot of 1 minute. Attacks are not visible when inspecting the variation of total number of packets due to aggregation with high number of packets. Also, the injected DDoS SYN flooding attacks don't induce a variation in the total number of TCP segments as shown in figure 4(a) & 4(b) respectively. In fact, the number of TCP segments is around 10^6 and attacks (shown in figure 6(a)) are of order 10^4 (insignificant variations). In figure 5(a), we present the variation of total number of SYN, where we can observe large deviations in the number of these requests. In spite of these heavy deviations, after inspection, we found these variations are legitimate, where SYN requests are not directed towards a specific destination (not the result of DDoS or flash crowd). Therefore, the analysis of the aggregation of whole SYN segments leads to false alarms, whereas the grained analysis with our proposed framework (1024 time series in each hash table) doesn't produces any alarm by JSD because the SYN request are directed to different destination IP addresses.



(a) Before SYN flooding attacks. (b) After SYN flooding attacks.

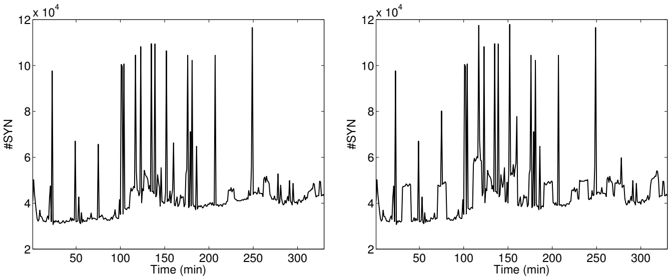
Fig. 3. Total number of packets before and after attacks.



(a) Before SYN flooding attacks. (b) After SYN flooding attacks.

Fig. 4. Total number of TCP segments.

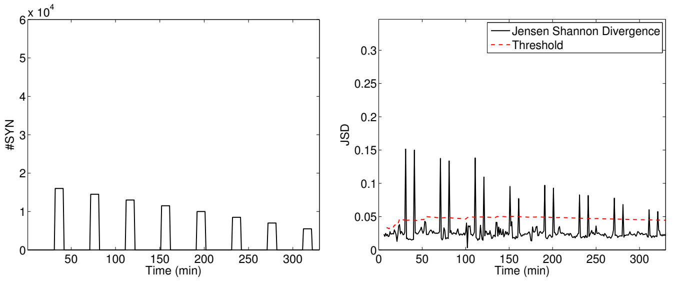
The variation of the number of SYN after the injection of 8 DDoS SYN flooding attacks with different intensity is given in figure 5(b), and the injected attacks are given in figure 6(a). Figure 6(b) shows the variation of JSD as well as the estimated threshold for the JSD. We get two spikes for each attack, the first is at the beginning of the attack and the second at the



(a) Before SYN flooding attacks.

(b) After SYN flooding attacks.

Fig. 5. Number of SYN.



(a) Attacks with different intensity.

(b) JSD.

Fig. 6. SYN flooding attacks and JSD.

end. When the current value of JSD is larger than dynamic threshold, an alarm is triggered.

VI. Conclusion and perspectives

In this paper, we propose a new sequential approach based on Sketch and Jensen-Shannon divergence for anomaly detection over high speed links. To evaluate the performance of the proposed approach, we conduct experiments over public traces. We proved that our approach is effective through implementation and testing on real traces with distributed SYN flooding attacks. The proposed framework is scalable and efficient for the detection of flooding attacks as proves the result of the conducted experiments.

Ongoing work will concern the extension of the proposed approach to automatically identify and pinpoint the malicious flows and the victim IP address. Furthermore, we want to investigate the performance of detection method using another divergence.

Most of the time, the Internet traffic is normal, and the reduction of exchanged monitoring data between Network Operation Center (NOC) and monitoring points is one of our interest for future work.

References

[1] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, "Sketch-Based Change Detection: Methods, Evaluation, and Applications," in *Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC'03)*, 2003, pp. 234–247.

[2] R. Schweller, Z. Li, Y. Chen, Y. Gao, A. Gupta, E. Parsons, Y. Zhang, P. Dinda, M.-Y. Kao, and G. Memik, "Reversible Sketches: Enabling Monitoring and Analysis Over High-Speed Data Streams," *IEEE/ACM Transactions on Networking*, vol. 15, no. 5, pp. 1059–1072, Oct. 2007.

[3] M. M. Deza and E. Deza, *Encyclopedia of Distances*. Springer, 2009.

[4] W. Lu and A. A. Ghorbani, "Network Anomaly Detection Based on Wavelet Analysis," *EURASIP Journal on Advances in Signal Processing*, pp. 1–16, 2009.

[5] S. Siripanadorn, W. Hattagam, and N. Teaumroong, "Anomaly Detection in Wireless Sensor Networks using Self-Organizing Map and Wavelets," *International Journal oF Communications*, vol. 4, no. 3, pp. 291–297, 2010.

[6] 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 (IMC'08)*, 2008, pp. 151–156.

[7] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies using Traffic Feature Distributions," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 4, pp. 217–228, 2005.

[8] V. A. Siris and F. Papagalou, "Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM'04)*, vol. 4, Dallas, USA, 2004, pp. 2050–2054.

[9] H. Wang, D. Zhang, and K. G. Shin, "SYN-dog: Sniffing SYN Flooding Sources," in *Proceedings of the 22th International Conference on Distributed Computing Systems (ICDCS'02)*. Washington, DC, USA: IEEE Computer Society, 2002, pp. 421–429.

[10] A. Tartakovsky, B. Rozovskii, R. Blazek, and H. Kim, "Detection of Intrusion in Information Systems by Sequential Chang-Point Methods," *Statistical Methodology*, vol. 3, no. 3, pp. 252–340, 2006.

[11] S. Bu, R. Wang, and H. Zhou, "Anomaly Network Traffic Detection Based on Auto-Adapted Parameters Method," in *Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM'08)*, 2008, pp. 601–607.

[12] N. Ye, S. Vilbert, and Q. Chen, "Computer Intrusion Detection through EWMA for Autocorrelated and Uncorrelated Data," *IEEE Transactions on Reliability*, vol. 51, no. 1, pp. 75– 82, March 2003.

[13] J. D. Brutlag, "Aberrant Behavior Detection in Time Series for Network Monitoring," in *Proceedings of the 14th USENIX conference on System administration (LISA '00)*, Berkeley, CA, USA, 2000, pp. 139–146.

[14] X. Li, F. Bian, M. Crovella, C. Diot, R. Govindan, G. Iannaccone, and A. Lakhina, "Detection and Identification of Network Anomalies using Sketch Subspaces," in *Proceedings of the 6th ACM SIGCOMM on Internet measurement (IMC'06)*, 2006, pp. 147–152.

[15] J. Yu, H. Lee, M.-S. Kim, and D. Park, "Traffic Flooding Attack Detection with SNMP MIB using SVM," *Computer Communications*, vol. 31, no. 17, pp. 4212–4219, 2008.

[16] A. Lakhina, M. Crovella, and C. Diot, "Diagnosing Network-Wide Traffic Anomalies," in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM'04)*, 2004, pp. 219–230.

[17] L. Huang, X. Nguyen, M. Garofalakis, and J. M. Hellerstein, "Communication-Efficient Online Detection of Network-Wide Anomalies," in *IEEE Conference on Computer Communications (INFOCOM'07)*, 2007, pp. 134–142.

[18] C. Bo, B.-X. Fang, and X.-C. Yun, "A New Approach for Early Detection of Internet Worms Based on Connection Degree," in *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, Guangzhou, China, 2005, pp. 2424–2430.

[19] O. Salem, S. Vaton, and A. Gravey, "A scalable, efficient and informative approach for anomaly-based Intrusion Detection Systems: theory and practice," *Int. J. Netw. Manag.*, vol. 20, no. 5, pp. 271–293, September 2010.

[20] H. Wang, D. Zhang, and F. Kang G. Shin, "Change-Point Monitoring for the Detection of DoS Attacks," *IEEE Trans. On Dependable and Secure Computing*, vol. 1, no. 4, pp. 1993–2004, 2004.

[21] H. Sengar, H. Wang, D. Wijesekera, and S. Jajodia, "Detecting VoIP Floods Using the Hellinger Distance," *IEEE Transactions on Parallel Distributed Systems*, vol. 19, no. 6, pp. 794–805, 2008.

[22] J. Tang, Y. Cheng, and C. Zhou, "Sketch-based SIP Flooding Detection using Hellinger Distance," in *Proceedings of the 28th IEEE conference on Global telecommunications (GLOBECOM'09)*, 2009, pp. 3380–3385.

[23] "MAWI working group traffic archive," http://mawi.wide.ad.jp/mawi/.