Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks

Vasilios A. Siris and Fotini Papagalou

Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH)
P.O. Box 1385, GR 711 10 Heraklion, Crete, Greece
Tel. +30 2810 391726, fax: +30 2810 391601, email: vsiris@ics.forth.gr

***Abstract*— We investigate statistical anomaly detection algorithms for detecting SYN flooding, which is the most common type of Denial of Service (DoS) attack. The two algorithms considered are an adaptive threshold algorithm and a particular application of the cumulative sum (CUSUM) algorithm for change point detection. The performance is investigated in terms of the detection probability, the false alarm ratio, and the detection delay. Particular emphasis is on investigating the tradeoffs among these metrics and how they are affected by the parameters of the algorithm and the characteristics of the attacks. Such an investigation can provide guidelines to effectively tune the parameters of the detection algorithm to achieve specific performance requirements in terms of the above metrics.**

I. Introduction

Over the past few years many sites on the Internet have been the target of denial of service (DoS) attacks, among which TCP SYN flooding is the most prevalent [1]. Indeed, recent studies[1] have shown an increase of such attacks, which can result in disruption of services that costs from several millions to billions of dollars.

The aim of denial of service attacks are to consume a large amount of resources, thus preventing legitimate users from receiving service with some minimum performance. TCP SYN flooding exploits TCP's three-way handshake procedure, and specifically its limitation in maintaining half-open connections. A TCP connection starts with the client sending a SYN message to the server, indicating the client's intention to establish a TCP connection. The server replies with a SYN/ACK message to acknowledge that it has received the initial SYN message, and at the same time reserves an entry in its connection table and buffer space. After this exchange, the TCP connection is considered to be half open. To complete the TCP connection establishment, the client must reply with an ACK message. In a TCP SYN flooding attack, an attacker, from a large number of compromised clients in the case of distributed DoS attacks, sends a very large number of SYN messages, with fictitious (spoofed) IP addresses, to a single server (victim). Although the server replies with SYN/ACK messages, these messages are never acknowledged by the client. As a result, many half-open connections exist on the server, consuming its resources. This continues until the server has consumed all its resources, hence can no longer accept new TCP connection requests.

In this paper we present and evaluate two anomaly detection algorithms for detecting TCP SYN attacks: an adaptive threshold algorithm and a particular application of the cumulative sum (CUSUM) algorithm for change point detection. Our focus is on investigating the tradeoffs between the detection probability, the false alarm ratio, and the detection delay, and how these tradeoffs are affected by the parameters of the detection algorithm and the characteristics of the attacks. Such an investigation can assist in tuning the parameters of the detection algorithm to satisfy specific performance requirements. Our results show that although simple and straightforward algorithms, such as the adaptive threshold algorithm, can exhibit good performance for high intensity attacks, their performance deteriorates for low intensity attacks. On the other hand, algorithms based on a strong theoretical foundation can exhibit robust performance over various attack types, and without necessarily being complex or costly to implement. Detection of low intensity attacks is particularly important since this would enable the early detection of attacks whose intensity slowly increases, and the detection of attacks close to the sources, either in routers or monitoring stations, thus facilitating the identification of compromised hosts that are participating in distributed DoS attacks [2].

Next we present a brief overview of related work. The authors of [3] investigate predictive detection of anomalies for a web server, analysing time series measurements of the number of http operations per second. The proposed statistical model considers both seasonal and trend components, which are modelled using a Holt-Winters algorithm, and time correlations which are modelled using a second order autoregressive model. After removing the above non-stationarities from the time series measurements, anomalies are detected using a generalized likelihood ratio (GLR) algorithm. A similar approach is used in [4], which considers measurements collected in MIB (Management Information Base) variables. The authors of [5] model the seasonal and trend components similar to [3]. A problem is detected when the actual measured value deviates from the predicted value (estimated using a moving average procedure) by some number of standard deviations.

This work was supported in part by the EC funded project SCAMPI (IST-2001-32404).

The authors are also with the Dept. of Computer Science, Univ. of Crete.

[1] 2002 and 2003 CSI/FBI Cybercrime Survey Report. The 2003 report indicates that DoS attacks alone were responsible for a loss of $65 million.

The author of [6] considers a similar approach for modelling the seasonal and trend component, and detects an anomaly when the measured variable falls outside a confidence band, which is estimated from previous differences of the measured variable and its predicted value.

The authors of [2] propose an approach for detecting SYN flooding attacks using a CUSUM-type algorithm, which is applied to the time series measurements of the difference of the number of SYN packets and the corresponding number of FIN packets in a time interval. Our work also considers a CUSUM-type algorithm, however the specific form, hence the corresponding equations, differ; moreover, we apply it to measurements of the number of SYN packets, while avoiding the need to explicitly take into account the seasonality and trend by considering an exponential weighted moving average for obtaining a recent estimate of the mean rate of SYN packets. Finally, the authors of [7] also consider a CUSUM-type algorithm, combined with a χ^2 goodness-to-fit test.

In addition to the specific algorithms we investigate, our work differs from the above in that we emphasize on investigating the performance of the detection algorithms in terms of three metrics: detection probability, false alarm ratio, and detection delay. Moreover, our experiments investigate how the tradeoff between these metrics is affected by the parameters of the detection algorithm and the characteristics of attacks.

The rest of the paper is organized as follows. In Section II we present the two anomaly detection algorithms that we investigate. In Section III we present and discuss the results investigating the performance of the algorithms, in terms of detection probability, false alarm ratio, and detection delay, and how the performance is affected by the parameters of the algorithm and the characteristics of the attacks. Finally, in Section IV we present some concluding remarks and identify related ongoing work.

II. Anomaly detection algorithms

In this section we present the two statistical anomaly detection algorithms that we apply for detecting SYN flooding attacks. The first, which we will refer to as adaptive threshold algorithm, is a rather straightforward and simple algorithm that detects anomalies based on violations of a threshold that is adaptively set based on recent traffic measurements. The second is an application of the cumulative sum (CUSUM) algorithm, which is a widely used anomaly detection algorithm that has its foundations in change point detection theory. Our selection of these two algorithms is twofold: First, based on the numerical experiments presented in Section III, we wish to demonstrate that a simple and naive algorithm can exhibit satisfactory performance for some types of attacks, such as high intensity attacks, but can have very bad performance for other types of attacks, such as low intensity attacks. Second, we wish to demonstrate that algorithms based on a strong statistical foundation can exhibit robust performance over various attack types, without necessarily being complex or costly to implement.

A. Adaptive threshold algorithm

This algorithm relies on testing whether the traffic measurement, number of SYN packets in our case, over a given interval exceeds a particular threshold. In order to account for seasonal (daily and weekly) variations and trends, the value of the threshold is set adaptively based on an estimate of the mean number of SYN packets.

If x_n is the number of SYN packets in the n-th time interval, and $\bar{\mu}_{n-1}$ is the mean rate estimated from measurements prior to n, then the alarm condition is

$$\text{If} \quad x_n \geq (\alpha+1)\bar{\mu}_{n-1} \quad \text{then ALARM signalled at time } n\,,$$

where $\alpha > 0$ is a parameter that indicates the percentage above the mean value that we consider to be an indication of anomalous behaviour. The mean μ_n can be computed over some past time window or using an exponential weighted moving average (EWMA) of previous measurements

$$\bar{\mu}_n = \beta\bar{\mu}_{n-1} + (1-\beta)x_n\,, \tag{1}$$

where β is the EWMA factor.

Direct application of the above algorithm would yield a high number of false alarms (false positives). A simple modification that can improve its performance is to signal an alarm after a minimum number of consecutive violations of the threshold.

$$\text{If} \quad \sum_{i=n-k+1}^{n} 1_{\{x_i \geq (\alpha+1)\bar{\mu}_{i-1}\}} \geq k \quad \text{then ALARM at time } n\,, \tag{2}$$

where $k > 1$ is a parameter that indicates the number of consecutive intervals the threshold must be violated for an alarm to be raised.

The tuning parameters of the above algorithm are the amplitude factor α for computing the alarm threshold, the number of successive threshold violations k before signalling an alarm, the EWMA factor β, and the length of the time interval over which traffic measurements (number of SYN packets) are taken.

B. CUSUM (Cumulative SUM) algorithm

The CUSUM algorithm belongs to the family of change point detection algorithms that are based on hypothesis testing, and was developed for independent and identically distributed random variables $\{y_i\}$. According to the approach, there are two hypothesis θ_0 and θ_1, with probabilities p_{θ_0} and p_{θ_1}, where the first corresponds to the statistical distribution prior to a change and the second to the distribution after a change. The test for signalling a change is based on the log-likelihood ratio S_n given by

$$S_n = \sum_{i=1}^{n} s_i\,, \qquad \text{where} \qquad s_i = \ln\frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)}\,.$$

The typical behaviour of the log-likelihood ratio S_n includes a negative drift before a change and a positive drift after the change. Therefore, the relevant information for detecting a

change lies in the difference between the value of the log-likelihood ratio and its current minimum value [8]. Hence the alarm condition for the CUSUM algorithm is

$$\text{If} \quad g_n \geq h \quad \text{then ALARM signalled at time } n\,, \qquad (3)$$

where

$$g_n = S_n - m_n \quad \text{and} \quad m_n = \min_{1 \leq j \leq n} S_j\,. \qquad (4)$$

The parameter h is a threshold parameter.

Assume that $\{y_i\}$ are independent Gaussian random variables with known variance σ^2, which we assume remains the same after the change, and μ_0 and μ_1 the mean before and after the change. After some calculations [8], (4) reduces to

$$g_n = \left[g_{n-1} + \frac{\mu_1 - \mu_0}{\sigma^2} \left(y_n - \frac{\mu_1 + \mu_0}{2} \right) \right]^+ . \qquad (5)$$

Above we have assumed that $\{y_n\}$ are independent Gaussian random variables. Of course this is not true for network traffic measurements, such as the number of SYN packets, due to seasonality (weekly and daily variations), trends, and time correlations. Such non-stationary behaviour should be removed before applying the CUSUM algorithm. One approach for achieving this is proposed in [3], where seasonality and trend is removed using the Holt-Winters algorithm and time correlations are removed using an autoregressive algorithm. In addition to leading to complex and time-consuming calculations, experiments we have conducted showed that the above approach, applied to the problem of detecting SYN flooding attacks, leads to minor gains compared to simpler approaches. For this reason we consider the following simple approach: We apply the CUSUM algorithm to $\tilde{x}_n$, with

$$\tilde{x}_n = x_n - \bar{\mu}_{n-1}\,,$$

where x_n is the number of SYN packets in the n-th time interval, and $\bar{\mu}_n$ is an estimate of the mean rate at time n, which is computed using an exponential weighted moving average, as in (1). The mean value of $\tilde{x}_n$ prior to a change is zero, hence the mean in (5) is $\mu_0 = 0$. A remaining issue that needs to be addressed is the value of μ_1, i.e. the mean traffic rate after the change. This cannot be known beforehand, hence we approximate it with $\alpha\bar{\mu}_n$, were as in the adaptive threshold algorithm the average $\bar{\mu}_n$ is updated using an exponential weighted moving average, and α is an amplitude percentage parameter, which intuitively corresponds to the most probable percentage of increase of the mean rate after a change (attack) has occurred. Hence, (5) becomes

$$g_n = \left[g_{n-1} + \frac{\alpha\bar{\mu}_{n-1}}{\sigma^2} \left(x_n - \bar{\mu}_{n-1} - \frac{\alpha\bar{\mu}_{n-1}}{2} \right) \right]^+ . \qquad (6)$$

It is interesting to contrast the above approach with that in [2], where daily variations are addressed by dividing the difference of the number of SYN packets and the number of FIN packets in a time interval, with the average number of FIN packets, hence is based on detecting changes when the number of SYN packets exceeds the number of FIN packets. Our approach is more general, since it can be applied to attacks other than SYN flooding. Indeed, an interesting application would be to use the algorithm for early detection of QoS (such as maximum delay) violations; such an approach can be justified by the fact that a large number of QoS violations are due to anomalies (including DoS attacks), hence anomaly detection techniques can warn for potential QoS violations before they occur.

The tuning parameters of the CUSUM algorithm are the amplitude percentage parameter α, the alarm threshold h, the EWMA factor β, and the length of the time interval over which traffic measurements are taken. These parameters are identical to the ones for the adaptive threshold algorithm, except for h which is the alarm threshold in the CUSUM algorithm.

III. Performance Evaluation

In this section we investigate the performance of the two algorithms presented in the previous section for detecting TCP SYN flooding attacks. The performance metrics considered include the detection probability, the false alarm rate, and the detection delay. Additional experiments investigating how different parameters of the detection algorithm and the characteristics of the attack affect the performance appear an the extended version of this paper [9].

Our experiments used actual network traffic taken from the MIT Lincoln Laboratory[2]. We used trace data taken during two days, with the trace from each day containing 11 hours of collected packets (08.00-19.00). The first investigations that we present considered SYN packet measurements in 10 second intervals. In some experiments, we also used a 14.5 hour trace taken from the link connecting the University of Crete's network to the Greek Research and Technology Network (GRNET).

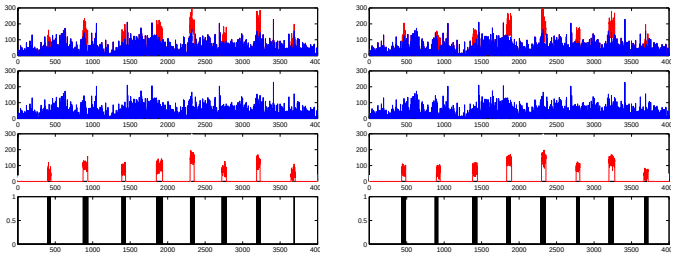
The attacks were generated synthetically; this allowed us to control the characteristics of the attacks, hence to investigate the performance of the detection algorithms for different attack types. The duration of one attack was normally distributed with mean 60 time intervals (10 minutes assuming 10 second intervals) and variance 10 time intervals. The inter-arrival time between consecutive attacks was exponentially distributed, with mean value 460 time intervals (approximately 77 minutes assuming 10 second intervals); this results in approximately 8 attacks in an 11 hour period.

The detection probability is the percentage of attacks for which an alarm was raised, and the false alarm ratio (FAR) is the percentage of alarms that did not correspond to an actual attack. Unless otherwise noted, the parameters we considered for the adaptive threshold algorithm were $\alpha = 0.5$, $k = 4$, and $\beta = 0.98$, and the parameters for the CUSUM algorithm were $\alpha = 0.5$, $h = 5$, and $\beta = 0.98$.

A. High intensity attacks

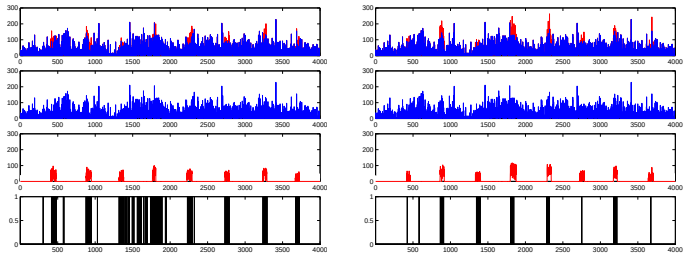
Our first experiment considered high intensity attacks, whose mean amplitude was 250% higher than the mean traffic rate, which was approximately 31.64 SYN packets in one time interval; the length of the time interval was 10 seconds.

[2]DARPA intrusion detection evaluation: `http://www.ll.mit.edu/IST/ideval`



(a) Adaptive threshold (b) CUSUM

Fig. 1. High intensity attacks. Both the adaptive threshold and the CUSUM algorithm have very good performance.



(a) Adaptive threshold (b) CUSUM

Fig. 2. Low intensity attacks. The performance of the adaptive threshold algorithm has deteriorated significantly. compared to its performance for high intensity attacks. On the other hand, the performance of the CUSUM algorithm remains very good.

Figures 1(a) and 1(b) show the results for the adaptive threshold and the CUSUM algorithm, respectively. The horizontal axis in these figures is the time interval, with 0 and 4000 corresponding approximately to 8:00 and 19:00, respectively. In each figure, from top to bottom, we have the traffic trace with attacks, the original traffic trace without attacks, the attacks only, and finally the bottom graph shows the time intervals where an alarm was raised. The figures show that both the adaptive threshold and the CUSUM algorithm have excellent performance for high intensity attacks, since they both yielded a detection probability of 100% and a false alarm ratio (FAR) of 0%. The detection delay was very close: 3.01 and 2.75 time intervals, respectively.

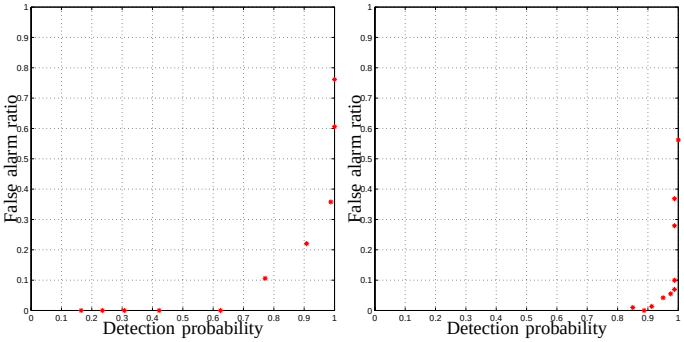
B. Low intensity attacks

Next we investigate the performance of the attack detection algorithms in the case of low intensity attacks, whose mean amplitude is 50% of the traffic's actual mean rate. Detection of low intensity attacks is important for two reasons: First, early detection of DoS attacks with increasing intensity would enable defensive actions to be taken earlier. Second, detection of low intensity attacks would enable the detection of attacks close to the sources, since such a placement of detectors can facilitate the identification of stations that are participating in a distributed DoS attack.

Figure 2(a) shows that for low intensity attacks the performance of the adaptive threshold algorithm has deteriorated significantly, giving a very high FAR equal to 32%. On the other hand, Figure 2(b) shows that the performance of the CUSUM algorithm remains close to its performance in the case of high intensity attacks, namely the FAR was less than 9%. Nevertheless, the detection delay of the CUSUM algorithm has increased to 10.25 time intervals, from only 2.75 time intervals in the case of high intensity attacks. Note that the detection probability for both algorithms was 100%.

The difference in the performance of the adaptive threshold and the CUSUM algorithms lies in the way each maintains memory: the adaptive threshold algorithm has memory of whether the threshold was violated or not in the previous $k-1$ time intervals. On the other hand, the CUSUM algorithm maintains finer information on the amount of data exceeding the amount expected based on some estimated mean rate, (6).

1) Tradeoff between detection probability and false alarm ratio: The above results were for specific values of the parameters of the two detection algorithms. Figures 3(a) and 3(b) show the tradeoff between the detection probability and the false alarm ratio (FAR) for different values of k for the adaptive threshold algorithm (2), and h for the CUSUM algorithm (3). Each point in the graph corresponds to a different value of the tuning parameter, k or h, in the interval $[1, 10]$. The data for each point was the average of 50 runs. Observe that the CUSUM algorithm exhibits better performance, supporting our observation in the previous section.



(a) Adaptive threshold (b) CUSUM

Fig. 3. Detection probability and false alarm ratio for low intensity attacks. The CUSUM algorithm has better performance than the adaptive threshold algorithm (better performance corresponds to points towards the lower-right).

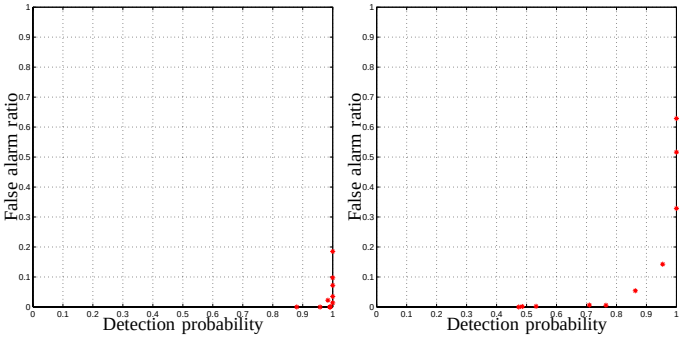
Figures 4(a) and 4(b) shows the performance of the CUSUM and of the algorithm in [2], for traces from the University of Crete (for which h obtains values in the interval $[10, 100]$). The algorithm of [2] is given by

$$g_n = [g_{n-1} + (X_n - a')]^+ ,$$

where X_n is the (# of SYN pkts - # of FIN pkts)/(average # FIN pkts). The graph in Figure 4(b) for the algorithm of [2] was obtained for an alarm threshold $h = 9$, and for a' in the interval $[1, 10]$. Observe that the CUSUM algorithm discussed in this paper has better performance than the algorithm in [2].

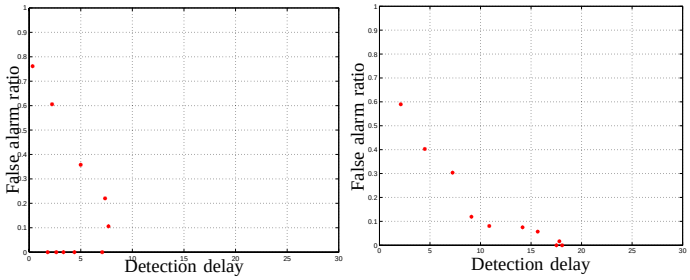
Graphs such as those in Figures 3 and 4 can assist in tuning the parameters of the detection algorithm. Indeed, note that the alarm threshold h is different for different traces, and controls the sensitivity of the attack detection.

2) Tradeoff between false alarm ratio and detection delay: Next we investigate the tradeoff between the false alarm ratio and the detection delay. Figures 5(a) and 5(b) show the results in the case of low intensity attacks for the adaptive threshold



(a) CUSUM (b) algorithm in [2]

Fig. 4. False alarm ratio and detection probability for the CUSUM algorithm proposed in this paper and the algorithm in [2].



(a) Adaptive threshold (b) CUSUM

Fig. 5. False alarm ratio and detection delay for the adaptive threshold and the CUSUM algorithms for low intensity attacks (better performance corresponds to points towards the lower-left).

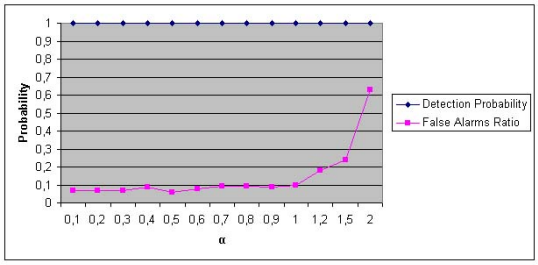
and the CUSUM algorithm, respectively. Each point in the graph corresponds to a different value of the tuning parameter, k or h. Note that in Figure 5(a), which is for the adaptive threshold algorithm, the values on the lower-left correspond to low detection delay, but have a small detection probability.

3) Effect of the amplitude factor α: Figure 6(a) shows the effect of the amplitude factor α for the CUSUM algorithm, when the threshold parameter h was adjusted in order to achieve a 100% detection probability. The graph was obtained by taking the average of 10 runs, which yielded a 95% confidence interval of ± 0.045. The figure shows that the performance of the CUSUM algorithm was indifferent to the factor α, for a large range of its values, approximately $[0.1, 1]$.

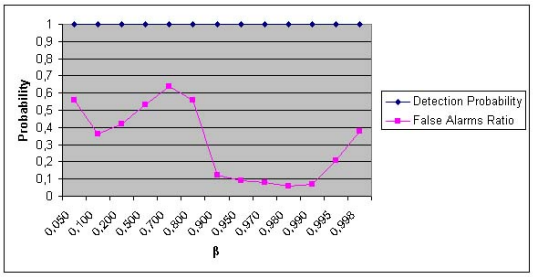
4) Effect of the EWMA factor β: Figure 6(b) shows the effect of the EWMA factor β for the CUSUM algorithm, when the threshold parameter h was adjusted in order to achieve a 100% detection probability. As before, the graph was obtained by taking the average of 10 runs, which yielded a 95% confidence interval of ± 0.045. The figure shows that the best performance of the CUSUM algorithm was for values of β in the interval $[0.95, 0.99]$.

IV. Conclusions

We described and investigated two anomaly detection algorithms for detecting SYN flooding attacks, namely an adaptive threshold algorithm and an algorithm based on the CUSUM change point detection scheme. Our investigations considered the tradeoff between the attack detection probability, the false alarm ratio, and the detection delay, and how these are affected by the parameters of the anomaly detection algorithm.



(a) Amplitude factor α



(b) EWMA factor β

Fig. 6. Effect of amplitude factor α and EWMA factor β for the CUSUM algorithm.

Our results illustrate that although a simple straightforward algorithm such as the adaptive threshold algorithm can have satisfactory performance for high intensity attacks, its performance deteriorates for low intensity attacks. On the other hand, an algorithm based on change point detection, such as the CUSUM algorithm, can exhibit robust performance over a range of different types of attacks, without being more complex.

Ongoing work focuses on the application of the algorithms to an actual production network, for both the incoming and the outgoing traffic, the combination of the algorithms with defensive mechanisms, and the application of the algorithms for early detection of QoS, such as maximum delay, violations.

References

[1] D. Moore, G. Voelker, and S. Savage, "Inferring Internet denial of service activity," in *Proc. of USENIX Security Symposium*, 2001.

[2] H. Wang, D. Zhang, and K. G. Shin, "Detecting SYN flooding attacks," in *Proc. of IEEE INFOCOM'02*, 2002.

[3] J. Hellerstein, F. Zhang, and P. Shahabuddin, "A statistical approach to predictive detection," *Computer Networks*, vol. 35, pp. 77–95, 2001.

[4] M. Thottan and C. Ji, "Adaptive thresholding for proactive problem detection," in *Proc. of IEEE Int'l Workshop on Syst. Manag.*, 1998.

[5] P. Hoogenboom and J. Lepreau, "Computer system performance problem detection using time series models," in *Proc. of USENIX Summer 1993 Technical Conference*, June 1993.

[6] J. Brutlag, "Aberrant behavior detection in time series for network monitoring," in *Proc. of LISA XIV*, December 2000.

[7] R. B. Blazek, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of denial-of-service attacks via adaptive sequential and batch sequential change-point detection methods," in *Proc. of IEEE Workshop on Syst., Man, and Cybern. Informat. Assurance*, June 2001.

[8] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes: Theory and Applications*. Prentice Hall, 1993.

[9] V. A. Siris and F. Papagalou, "Application of anomaly detection algorithms for detecting SYN flooding attacks," ICS-FORTH, Tech. Rep. No. 330, December 2003.