An Effective Anomaly Detection Method in SMTP Traffic*

Hao Luo[1], Binxing Fang[2], Xiaochun Yu[2], and Zhigang Wu[1]
(Corresponding author: Hao Luo)

Research Center of Information Intelligent and Information Security[1]
Institute of Computing Technology, Chinese Academy of Sciences
No.6 Kexueyuan South Road Zhongguancun, Haidian District Beijing, 100080 China (Email: luohao@hit.edu.cn)
Department of Computer Science and Engineering, Harbin Institute of Technology[2]
No.92, West Da-Zhi Street, Harbin, Heilongjiang, 150001 China

(Selected paper from ITNG 2006)

Abstract

We investigate an effective and robust mechanism for detecting SMTP traffic anomaly. Our detection method cumulates the deviation of current delivering status from history behavior based on the leaky integrate-and-fire model to detect anomaly. The simplicity of our detection method is that the method requires neither the set of anomalies to be detected, nor the thresholds to be supplied by the user. Furthermore the proposed method need not store history profile and has low computation overhead, which makes the detection method itself immune to attacks. The performance evaluation results show that leaky integrate-and-fire method is quite effective at detecting constant intensity attacks and increasing intensity attacks in the SMTP traffic. Compared with other anomaly detection method, our detection method has better detecting performance.

Keywords: Anomaly detection, integrate-and-fire model, SMTP traffic

1 Introduction

Email is the most popular Internet service now [2], and it allows people to communicate by exchanging electronic messages globally. These messages can be delivered in a few seconds to a couple of hours. An added attraction is the relatively low cost of sending large messages. Combined, these benefits give users a convincing argument for accessing to email, and thus the connection of their systems to the Internet.

The SMTP [9] is used as the basis for most electronic mail, and SMTP is a simple protocol with only a few basic commands. There are several security threats that associated with these commands and the Denial-of-service attack is one of the most popular threats of SMTP. The Denial-of-service attacks based on SMTP aim at flooding a network or computer with massive email messages to prevent legitimate usage. In most cases a computer is affected because it cannot handle the load created by receiving large numbers of messages at the same time, or running out of storage space, or cannot handle large messages [4]. An example of Denial-of-service attacks of SMTP is error mails bouncing back attack [14], and a report shows on October 2003, at least two domains in the United States had been received hundreds of thousands of error mails from all over the Internet [6].

Another important threat of SMTP is email-based viruses. Email virus has become one of the major Internet security threats today. An email virus is a malicious program, which hides in an email attachment, and becomes active when the attachment is opened. A principal goal of email virus attacks such as Melissa [3] is to generate a large volume of email traffic over time, so that email servers and clients are eventually overwhelmed with this traffic, which effectively disrupting the usage of the email service. Modern email viruses are more damaging, taking actions such as creating hidden backdoors on the infected machines that can be used to commandeer these machines in the subsequent coordinated attack.

In this paper, we propose an effective and robust method for detecting SMTP traffic anomaly, which is complementary to alert the threats mentioned above. The effect of our detection method is that the method need not store history profile and has low computation overhead. Instead of monitoring the ongoing traffic at the front end or the victim server, our method checks the SMTP server's delivery log. The benefit of checking SMTP log to detect traffic anomaly is that we need not monitor raw traffic of the server exchanging and make computation

*A preliminary version of this work appeared in proceedings of international conference on Information Technology: New Generations (ITNG 2006).

overhead very low. The SMTP log also provides detail information about receiving and sending status. The key feature of our method is to utilize the leaky integrate-and-fire model (LIF) to cumulate the deviation of current delivering status from the history status. The leaky integrate-and-fire neuron model is a weighted sum model, and the newer input data will play a more important role in the result. The old data will be dropped from the result by a weighted factor. In this way, our method archives higher detection accuracy and shorter detection delay. The efficacy of our detection method is validated by simulating experiment with real background test data.

The remainder of this paper is organized as follows. The Section 2 shows the related works of network anomaly detection. In Section 3 we discuss the leaky integrate-and-fire model based SMTP traffic anomaly detection method. In Section 4, we evaluate our anomaly detection method and compare our method with a non-parametric Cumulative Sum (CUSUM) method. Finally, Section 5 presents our conclusions.

2 Related Works

It is possible to track the behavior of the network continuously by online learning and statistical approaches, and the statistical analysis method has been used to detect both anomalies corresponding to network failure [12], as well as network intrusions [13].

The most classical method of statistical analysis was trying to predict the normal behavior of network status, and the anomaly was detected when the actual measured value deviates from the predicted value by some number of standard deviations. The authors of [5] proposed a predictive detection method used in web server anomaly detection, by analyzing the time series measurements of the number of http operations per second. The statistical model considered both seasonal and trend components, which were modelled using a Holt-Winters algorithm. Time correlations were modelled using a second order auto-regressive model. After removing the non-stationarity from the time series measurements, anomalies are detected using a generalized likelihood ratio algorithm. This method need store history profile for future using.

The signal process method for detecting network anomaly was proposed in recent years. A wavelet approach was proposed and implemented by Paul and others [1], they used wavelet filter to process four classed of network traffic anomalies: outages, flash crowds, attacks and measurement failures. Their results showed that wavelet filters are quite effective at exposing the details of both ambient and anomalous traffic. However, the authors also mentioned that their signal analysis method could not detect anomalies in real time.

The most related researches to our approach are shown in papers [10, 13]. The authors proposed approaches for detecting SYN flooding attacks using CUSUM-type algorithm, and both of these two cases make use of the standard sequential change point detection approach. The approach of [13] applied the time series measurements of the difference of the number of SYN packets and the corresponding FIN packets in a time interval. And the deviation of the measurement from the upper bound of the measurement expectation was cumulated and compared with a predefined threshold. On the other hand, paper [10] applied the CUSUM algorithm to the measurements of the number of SYN packets. They calculated the deviation by an exponentially weighted moving average value method [10], and the deviation has also been cumulated and compared to detect the anomaly. The simulation results have shown that SYN flooding attacks could be detected with high accuracy by both of two CUSUM-type algorithms. Because our method has similar computing process with CUSUM-type algorithm, and we cumulate the deviation of actual measurement by a weighted sum method, we will take some comparisons in the following sections.

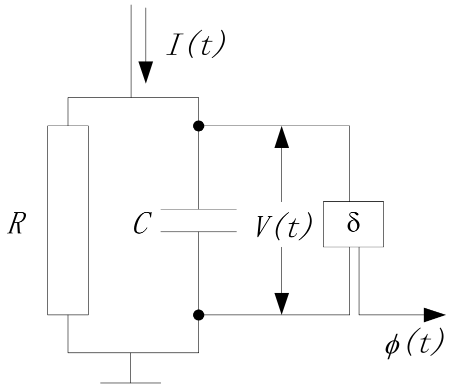


Figure 1: Schematic diagram of the leaky integrate-and-fire neuron model

3 SMTP Anomaly Detection

In this section, a real time statistical analysis method we developed using the theory of leaky integrate-and-fire model is discussed. Unlike the traditional network intrusion detection system that detects the anomaly directly by the deviation of current behavior from the profiled normal history behavior, our method cumulates the deviation in a period to detect the anomaly according to the method of integrate-and-fire model described. Compared with the CUSUM-type algorithms, the detection algorithm based on integrate-and-fire model is more sensitive to current network status.

Our method uses SMTP server's log to detect the anomaly. SMTP server log provides a mail server's receiving and sending information, and the log also includes delivery time of each mail. Since our work is detecting SMTP traffic anomalies, this data source is sufficient.

3.1 The SMTP Behavior Deviation Evaluation

Let $\{x_n, n = 0, 1, \cdots\}$ be the serial of mail numbers that a mail server received within one sampling period, and let $\{y_n, n = 0, 1, \cdots\}$ be the corresponding sent mail numbers in the same sampling period. We define $\{\Delta_n, n = 0, 1, \cdots\}$ be the number of received mails minus that of the corresponding sent mails collected within one sampling period.

In general, the mean of $\{\Delta_n\}$ is dependent on the accounts number of SMTP server, and it may also depend on the access patterns, for example, varying with time of the day and week. To make our detecting algorithm more general, we should eliminate these dependencies. Thus, $\{\Delta_n\}$ is normalized by the average number of $\overline{Y}_n$ of $\{y_n\}$. $\overline{Y}_n$ can be computed over some past time windows or using the exponentially weighted moving average (EWMA) of previous measurements. Here we use EWMA method and EWMA method can be described as:

$$\overline{Y}_n = \beta\overline{Y}_{n-1} + (1-\beta)y_n.$$

Where β is the EWMA factor that represents the memory in the estimation. Define $X_n = \Delta_n/\overline{Y}_n$, and $\{X_n\}$ is no longer dependent on the network size or time-of-day.

So we can define the deviation of SMTP behavior for a given interval n as:

$$D_n = X_n - \overline{X}_{n-1}.$$

Where X_n is the mean of X_n and estimated from measurements prior to n. The mean $\bar{X}_n$ is also computed by EWMA method. The deviation of SMTP behavior D_n is used as input data of our anomaly detection method.

3.2 The Leaky Integrate-and-fire Model

The leaky integrate-and-fire model has been proposed as a model of neurons for a long time. It can be used for processing time-varying signals [11] and also can be used in powerful computing systems [8]. The simplest form of integrate-and-fire model consists of a resistor R in parallel to a capacitor C driven by an external current $I(t)$. The voltage $V(t)$ across the capacitor C is compared to a threshold δ . If $V(t) = \delta$ at time t an output spike $\phi(t)$ is generated and $V(t)$ is reset to an initial voltage U_r. The schematic diagram of leaky integrate-and-fire model is shown in Figure 1.

Between spikes, the voltage of a leaky integrate-and-fire model is governed by:

$$\frac{dV(t)}{dt} = -\frac{V(t)}{RC} + \frac{I(t)}{C}.$$

Suppose that a spike has occurred at t_i. For $t > t_i$ the stimulating current is $I(t)$. The $V(t)$ can be expressed as:

$$V(t) = U_r\text{exp}(-\frac{t-t_i}{RC}) + \frac{1}{C}\int_0^{t-t_i} \text{exp}(-\frac{s}{RC})I(t)ds.$$

When leaky integrate-and-fire model is used to detect SMTP anomaly, the deviations of SMTP behavior in each interval of $t > t_i$ are inputted, and the $V(t)$ are tested as alarm condition. The detail of detection algorithm will be described in Section 3.3.

3.3 Anomaly Detection Approach

In our SMTP traffic anomaly detection approach, the SMTP health status is obtained by the output of leaky integrate-and-fire model. In the process of capacitor recharging, when the input current is constant, the earlier input current, the faster voltage raising. Therefore, in our detection method, the deviation of SMTP behavior D_n will be inputted into leaky integrate-and-fire model from the current interval to the last spike occurred interval. This means we input current D_n first, and than the one just before current, and so on. In this way, the current SMTP delivery status will play a more important role in the detection result. Because D_n is the discrete value, suppose that a spike has occurred at interval n_k, the output of leaky integrate-and-fire model at interval n can be gotten from (4) as:

$$\begin{aligned} V'(n) &= U_r exp(-\frac{n-n_k}{RC}) \\ &+ \frac{1}{C}\sum_{i=1}^{n-n_k} \text{exp}(-\frac{n-n_k-i+1}{RC})D_{n_k+i}. \end{aligned}$$

Let U_r=0, $L'(n) = CV'(n)$and $K = RC$, from (5) we get:

$$L'(n) = \sum_{i=1}^{n-n_k} \text{exp}(-\frac{n-n_k-i+1}{K})D_{n_k+i}.$$

So we have:

$$\begin{aligned} L'(n-1) &= exp(-\frac{n-n_k}{K})D_{n_k+1} \\ + &\quad \text{exp}(-\frac{n-n_k+1}{K})D_{n_k+2} \\ + &\quad ... + \text{exp}(-\frac{1}{K})D_{n-1}. \end{aligned}$$

Therefore

$$L'(n) = \text{exp}(-\frac{1}{K})(L'(n-1)) + D_n.$$

As the negative SMTP behavior deviation means no anomaly in our detection, according (8), here we let

$$\begin{cases} L'(n) = exp(-\frac{1}{K})(L'(n-1) + D_n)^+ \\ L(0) = 0, \end{cases}$$

be our network status function. Where $n > 0$ and x^+ is equal to x if $x > 0$ and x^+ is equal to 0 otherwise. We will use $L(n)$ in making detection decisions. Here we call K as cumulating factor.

Let H represents the anomaly threshold. At interval n, if $L(n) > H$, an alarm will be signaled at time n, otherwise the network status is normal. If the alarm is signaled at time n, $L(n)$ will be reset to 0.

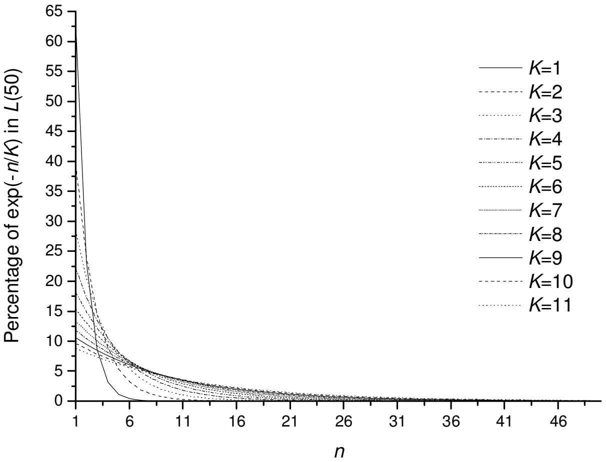


Figure 2: Contribution of exp(-n/K) with different K

The algorithm of LIF can be expressed as:

```
# Define EWMA factor $\beta$, Cumulating factor $K$ and
Threshold H DEFINE $\beta$, K, H;
GLOBAL Ln = 0, Yn = 0, Xn = 0;
Function AnomalyDetection (MR, MS)
# MR: Mail Received number in the given sample intelval
# MS: Mail Sent number in the given sample intelval
    Yn = $\beta$* Yn + (1-$\beta$) * MS;
    X = (MR - MS)/Yn;
    Xn =$\beta$* Xn + (1-$\beta$) * X;
    Dn = Xn - X;
    Ln = exp(-1/K)(Ln + Dn);
    IF (Ln ¿ H) THEN # Anomaly Detected
        Ln = 0;
        RAISE ALARM;
    ENDIF;
EndFunction
```

3.4 Parameter Specification

The tuning parameters of above algorithm are the cumulating factor K for computing the network health status, the alarm threshold H, and the EWMA factor β. In general, the EWMA factor β is chosen as 0.98 [10, 13], and here we also chose $\beta = 0.98$ as our EWMA factor in experiments. To implement our leaky integrate-and-fire anomaly detection algorithm, we still need to specify two tunable parameters: K and H. The cumulating factor K decides how we cumulate the SMTP status deviation to detect the anomaly, and the alarm threshold H depends on K. From (9) we can find that D_n has different contribution to $L(n)$ with different K. Figure 2 shows the percentage of $(exp(-n/K)D_n)$ in $L(50)$, where we set $D_n = 1, n = 1, 2, \cdots, 50$.

We can see clearly from Figure 2 that the smaller K, the more contribution $exp(-1/K)$ does, and the shorter history profile is referred. When $K = 1$, $exp(-1)$ contributes 63.21% to $L(50)$, and about 8 intervals are evidently referred in $L(50)$; when $K = 15$, $exp(-1/15)$ contributes 8.79% to $L(50)$, and all 50 intervals are referred in L(50). Here we can see when $K = 5$, $\sum_{n=1}^{3} exp(-n/5)$ contributes about 45% of integrate result, and $\sum_{n=1}^{10} exp(-n/5)$ contributes about 91% of result. This means when we chose $K = 5$, the calculating result not only emphasizes the first three inputs, but also refers enough history information. So in our detection algorithm, we chose $K = 5$ as our cumulating factor.

Suppose in the normal condition, $x_n = 1$ and $y_n = 1$, and we should raise an alarm when x_n increases to 1.6 times of normal value. As the typical attacking duration observed in the Internet is 10 minutes [13], when we decide cumulating factor K, we can calculate H by following algorithm:

```
Function GetThreshold(K)
    LET e=0;
    FOR I=1 TO 10 DO
        $e = e + exp(-I/K)$;
    ENDFOR
    RETURN e * 0.6;
EndFunction
```

When we choose K=5, we can get H=2.4 by above algorithm. Here we select this parameter pairs as our predefined parameters of LIF method.

4 Performance Evaluation

In this section, we firstly chose parameters for our method. In order to compare our method with a CUSUM-type algorithm described in [13], we also chose parameters for algorithm in [13]. The algorithm of [13] is given by

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

Where a is the upper bound of $E(X_n)$ and g_n is the anomaly detecting condition. In addition to the parameters choice, we evaluate how the parameters of our detection algorithm affect the detecting performance.

Secondly, we investigate the performance of our leaky integrate-and-fire method for detecting SMTP traffic anomaly. The performance metrics considered include the detection probability, the false alarm rate, and the detection delay. The detection probability is the percentage of attacks for which an alarm was raised, the false alarm ratio (FAR) is the percentage of alarms that did not correspond to an actual attack [10], and the detection time is the detection delay after the attack starts.

Our experiments uses actual SMTP server delivery log taken from our campus mail server as background data. We use mail server's log during 2.5 days and measure the SMTP deliveries in one minute. Our test set includes 120412 receiving mails information and 80358 sending mails information with average receiving speed 33.45 mails per minutes and sending speed 22.32 mails per minutes. This log does not include spam mails and virus mails

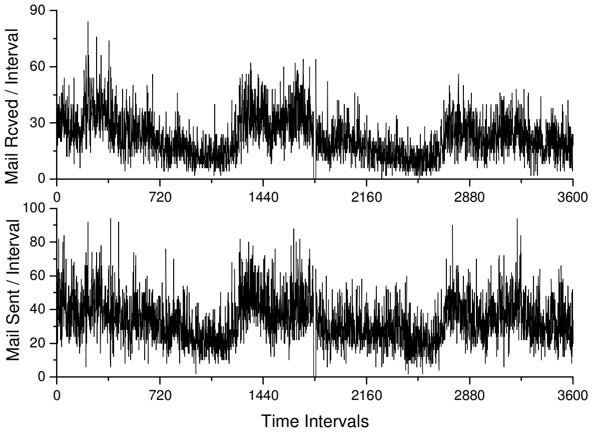


Figure 3: Trace of mail logs

information as there is a commercial mail filter protects our mail server, so this log is clean. The Figure 3 shows the trace of mail logs.

The attacks in our experiments are generated synthetically, and this allowed us to control the characteristics of the attacks, hence to investigate the performance of the detection algorithms for different attack intensities and types. The typical attacking duration observed in the Internet is 10 minutes [7], therefore the attacks are generated with mean duration 10 time intervals. The interarrival time between consecutive attacks was random distributed in 60-180 time intervals with mean values 120 intervals.

4.1 Parameters Selection

In order to select appropriate parameters for evaluating the detecting performance and the compare our method with CUSUM method, we enumerate each possible parameter of two anomaly detection methods.

For our method, we test threshold H from 2 to 4 with Step 0.1 and test cumulating factor K from 1 to 15 with Step 1. For CUSUM method, we test a from 0.6 to 1.6 with Step 0.05 and test threshold TH from 0.6 to 6 with Step 0.1. We reserve those parameter pairs that can archive average 100% of detection probability in 10 round tests, and the results of detection delay and false alarm ratio of these parameter pairs are shown in Figure 4. The test set is generated by overlapping constant intensity attacks with the duration of 10 intervals (10 minutes). The intensity of attacks is 75% of mean actual receiving mails rate.

Figure 4a shows the results of CUSUM method, and Figure 4b shows the results of leaky integrate-and-fire method. In order to evaluate the detection performance, we select several parameters from Figure 4. The results of selected parameter pairs are included in the rectangle in Figure 4. For each algorithm, we select 6 parameter pairs to finish the performance evaluation. The details

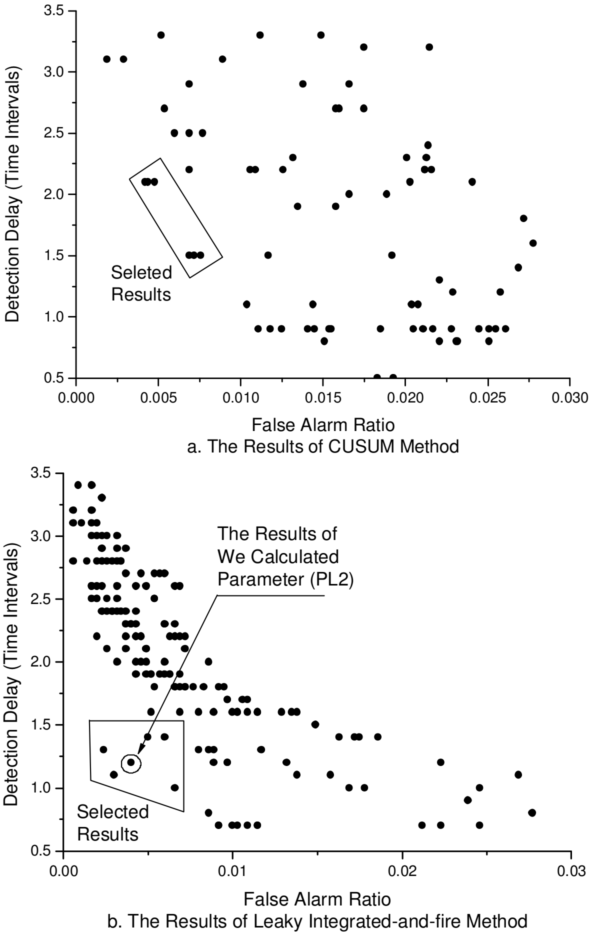


Figure 4: Detection results of different parameter pairs

Table 1: Selected Parameters for CUSUM method

ID	a	*Threshold*	FAR	DD
PC1	1.0	3.2	0.0052	2.2
PC2	1.0	3.3	0.0053	2.2
PC3	1.0	3.4	0.0055	2.2
PC4	1.1	2.1	0.0080	1.5
PC5	1.1	2.2	0.0082	1.5
PC6	1.2	2.3	0.0085	1.5

Table 2: Selected Parameters for CUSUM method

ID	a	*Threshold*	FAR	DD
PL1	4	2.5	0.0033	1.1
PL2	5	2.4	0.0037	1.2
PL3	5	2.5	0.0028	1.3
PL4	6	2.4	0.0052	1.4
PL5	6	2.5	0.0065	1.4
PL6	7	2.5	0.0070	1.0

of selected parameters are shown in Table 1 for CUSUM method and Table 2 for our method.

We can find the results we get with our select parameters have smaller false alarm ratio and lower detection delay in all reserved parameter pairs. Considering the tradeoff between false alarm ratio and detection delay, the parameter pairs we select for performance evaluation are suitable. Here we should note that the PL2 is the parameters we get by the algorithm described in Section 3.4. The column named ID is used for indicating the legends in Figure 7 and Figure 8 shown in the following section.

4.2 Evaluation of Cumulating Factor

Figure 5 shows how the cumulating factor K affects the false alarm ratio and detection delay, where the threshold H is calculated by the algorithm described in Section 3.4 and the detection probability is 100% in all tests. The Figure 5 is obtained by taking the average of 10 runs.

The cumulating factor K decides the length of history that the detection method uses and the behavior of computing. From Section 3.4 we know that the bigger K, the longer history are referred in making decision. At the same time, the bigger K, the lower weight of current delivering status is considered in detection results. It means that the current delivering status influent results less. The longer history may induce long detection time because current delivery status is not sensitive to the final detection result. The smaller K, the shorter history is considered and the bigger weight of current networking status has, the faster we can detect the anomaly, at the same time, the final detection results are more sensitive to the current delivering status. The smaller K will make more false alarms. In our test set, when $K = 5$, the detection results has a good tradeoff between detection delay and false alarm ratio.

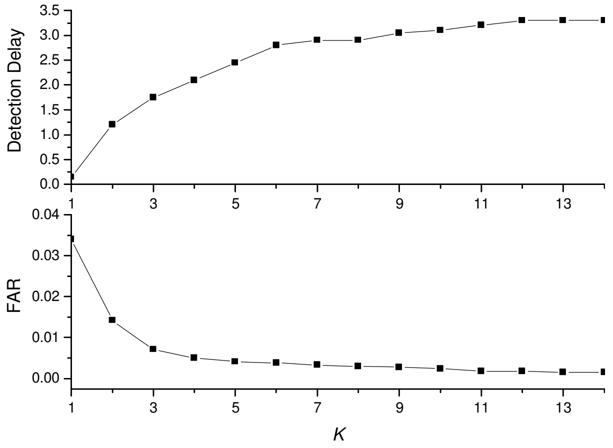


Figure 5: Effect of accumulating factor K

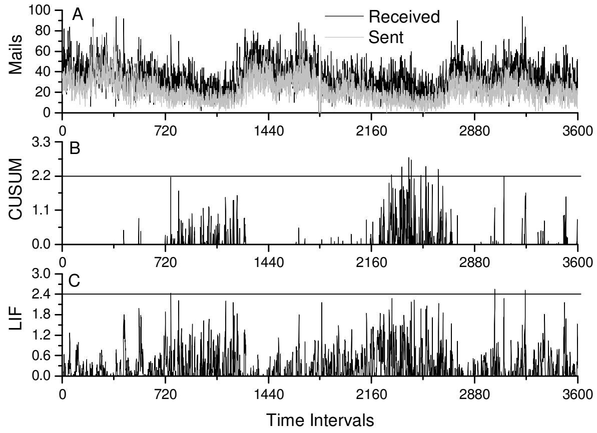


Figure 6: Performance of normal condition

From the Figure 5 We can see clearly that this evaluation proves the results of Section 3.4.

4.3 Evaluation of Normal SMTP Behavior

We use the mail trace shown in Figure 3 to represent the normal SMTP behavior at our mail server, and we have applied the leaky integrate-and-fire method on this trace without injecting flooding mails. The test statistics, $\{L(n)\}$, for this mail trace is plotted in Figure 6C. We also have applied the CUSUM method on this set without injecting any attack mails. And the result $\{g_n\}$ is plotted in Fig6B. The Figure 6A shows the same mail trace with Figure 3 for reference. The parameters used in this evaluation are PC1 and PL2 described in Table.1 and Table.2.

For this test set, Most of $L(n)'s$ and g'_ns are much smaller than the thresholds. Here the threshold for leaky integrate-and-fire method is 2.4 and for CUSUM method is 2.2. The $L(n)'s$ and g_n's bigger than corresponding thresholds are false alarm in this scenario. We can see clearly from Figure 6B that CUSUM method generates 8 false alarms in 2.5 days with false alarm ratio 0.22% and from Figure 6C, we can find that our leaky integrate-and-fire method only yields 3 false alarms in 2.5 days with false alarm ratio 0.08%.

Figure 6A also shows that under the normal condition, mail traffic exhibits clear diurnal patterns, although the number of mail sending and receiving may be burst on a small time scale, and slowly varying on the large time scale, but the difference between numbers of mail sending and mail receiving is small, as compared to the total number of mails exchanges.

4.4 Evaluation of Detecting Attacks with Constant Intensity

This experiment considers attacks with constant intensities, i.e. the attacks reach their max amplitudes in one time interval. We generate a serial of different intensity attacks to evaluate our detection performance and compare our method with CUSUM algorithm described in [13]. The attack serial is from low intensity to high intensity. In low constant intensity attack, the added attacks' amplitude is 15 mails, and it is about 44% of mean normal SMTP receiving speed. The high intensity attacks' is about 85% of mean normal SMTP receiving speed. The average results with 10 runs are shown in Figure 7. The horizontal axis in Figure 7 is attack mails injected per interval, and the legends in the figure are indicated by Tables 1 and 2.

As shown in the above graphs, our method has good performance in both low intensity attacks and high intensity attacks.

Figure 7a shows the detection probability evaluation results. We can see clearly that when injected attack mails bigger than 23 mails per interval, the detection probabilities of two detecting method with all parameter settings can archive 100%. We can find that all results of our method are better than the results of CUSUM method. Our method with parameters PL5, PL2 and PL6 archives 100% detection probability when the attack intensity is 17 mails per interval and PL1, PL3 and PL4 get the same detection probability when the attack intensity is 19 mails per interval. However, the CUSUM method gets 100% probability only when the attack intensity is 21 mails per interval.

Figure 7b shows the false alarm ratio results, and it is obviously that the results of our method are better than CUSUM method described in [13]. The range of the false alarm ratios of our method with PL1-PL6 is between 0.002 and 0.005, while the false alarm ratio range of CUSUM is between 0.005 and 0.008.

We can see clearly from Figure 7c that larger attack intensity leads to faster and easier detection of attacks. When the attack intensity is 15 mails per interval, our

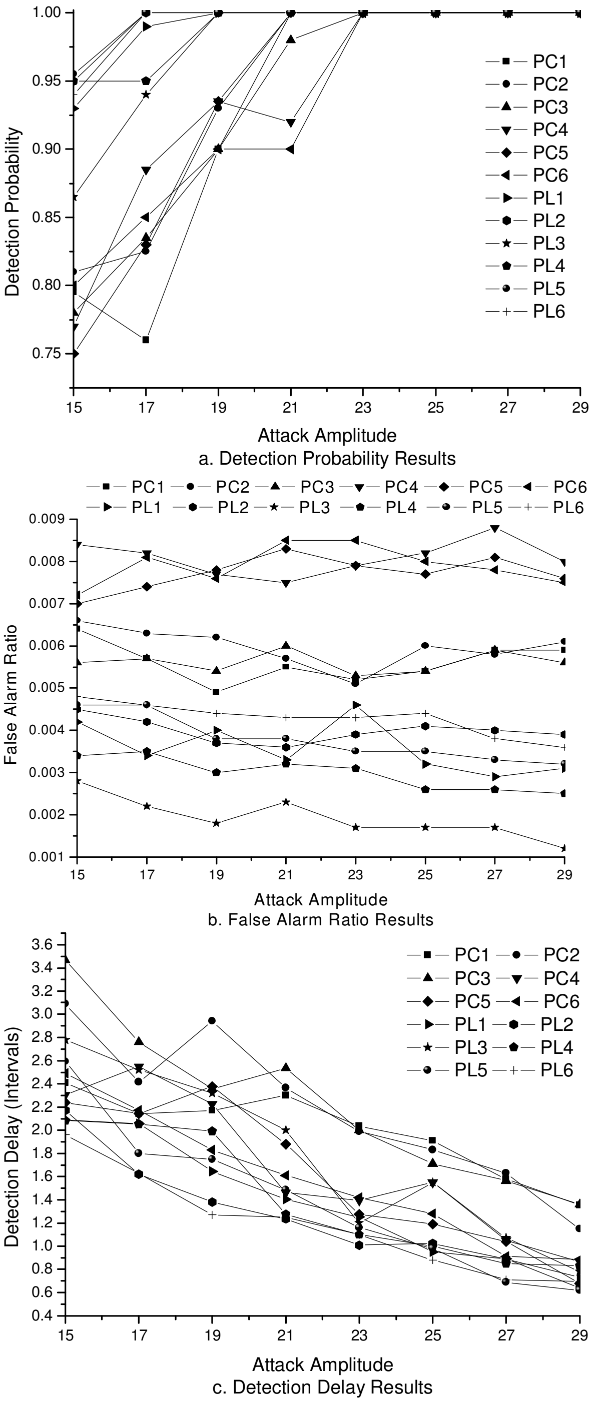


Figure 7: Average detecting results of different intensity attack

method needs 1.9 - 2.5 intervals to raise an alarm and the CUSUM method needs 2.2 - 3.5 intervals. And when the attack intensity is 29 mails per interval, the detection times of our method change to 0.6 - 1.0 intervals and CUSUM method changes to 0.8 - 1.4. Here we can see our weighted sum method has better detecting performance than the way that CUSUM does.

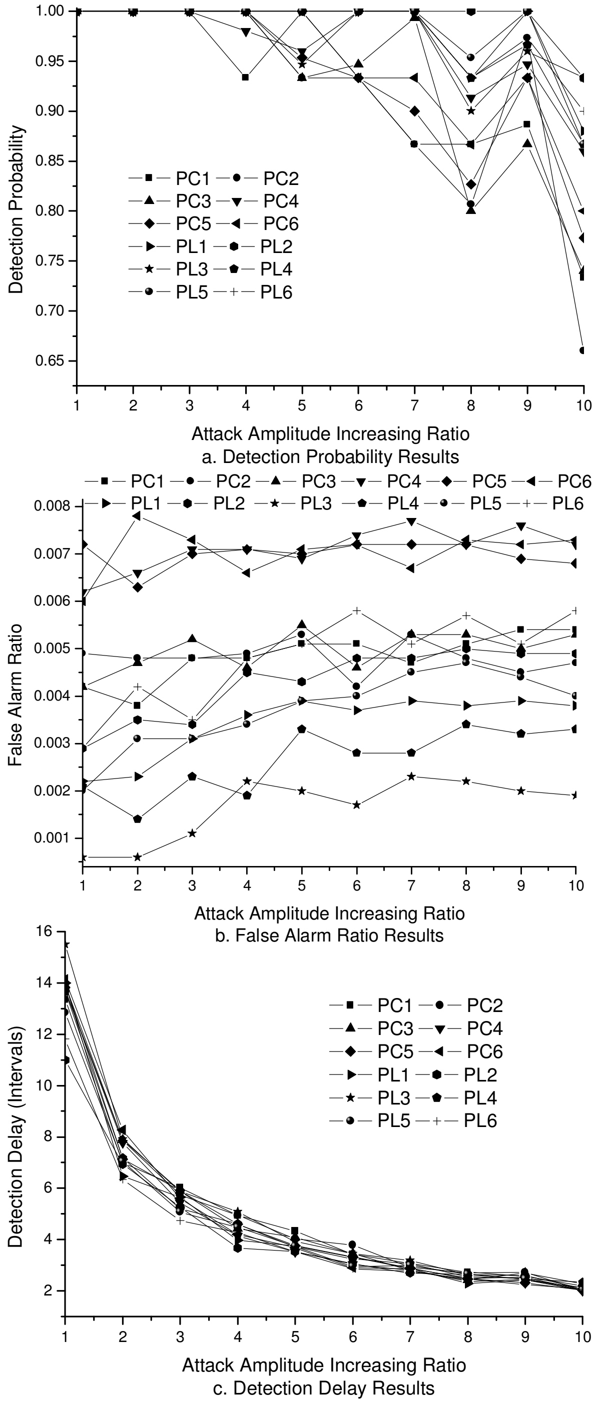
Here we should note the detecting results of the parameters PL2 that is mentioned in Section 3.4. In low intensity attacks, PL2 yields a detection probability of 96.2% and false alarm ratio 0.45%. In high intensity attacks, our method gets 100% detection probability and 0.40% false alarm ratio. The average false alarm ratio of our method in all attacks is about 0.4%. The detection delay of PL2 is 2.2 intervals, and in high intensity attacks with 29 attacks mails are injected per interval, it only need 0.8 intervals to raise alarms. Although the results of PL2 are not the best in all the parameters get, but they have better tradeoff among the evaluations. For example, the results of PL3 have the smallest false alarm ratio in the evaluations, but the results of detection probability and detection delay are the worst.

4.5 Evaluation of Detecting Attacks with Increasing Intensity

Our last experiment considers attacks with increasing intensities. In this experiment, the attacks will increase their intensities continuously until reach max amplitudes. We generate a serial of attacks with different increasing rates to compare our method with CUSUM algorithm. The increasing rates of attack's intensity are from 1 mail per interval to 10 mails per interval, and they have 40 attacks mails per interval when the attacks reach their max amplitudes, the max amplitude is about 120% of mean normal SMTP receiving speed. The average results with 10 runs are shown in Figure 8. The horizontal axis in Figure 8 is attack mails increased per interval, and the legends in the figure are indicated by Tables 1 and 2

Figure 8 shows the similar results with above section. We can see clearly that the detection probability of our method is higher than CUSUM algorithm in high intensity increasing ratio. The higher intensity ratio means duration attack duration is shorter. It is easy to see from Figure 8a that most parameter settings of our method get 100% detection probability when the increasing ratio is smaller than 7 mails per interval except PL3, and the result with PL2 even gets 100% detection probability when intensity increasing ratio is 9 mails per interval (In this scenario, the duration of attacks only 4 intervals as our max attack amplitude is set as 40 mails per interval). The results of the false alarm ratio shown in Figure 8b are similar with Figure 7b, and the results show the stable performance of our method in different type attacks.

Figure 8c shows the results of detection delay. The detection delays in this experiment vary with the intensity increasing ratio. When the ratio is small, the abnormal behavior induces a sample of observations with a



a. Detection Probability Results

b. False Alarm Ratio Results

c. Detection Delay Results

Figure 8: Average detection results of increasing intensity attack

mean very close to the existing regime. Therefore, the test statistic requires a sample large enough to cross the predefined threshold, resulting in a relatively long delay to detection. On the other hand, when the increasing ratio is high, the anomaly induces a mean significantly apart from the existing regimes, and detection takes a short time. From Figure 8c, it is easy to find out the detection delay of our method is shorter than CUSUM's, especially when the increasing ratio is low.

In this experiment, the parameters PL2 calculated by Section 3.4 are also proved to have a good detecting performance.

4.6 Discussion

The difference in the performance of our detecting method and CUSUM method is that our method uses weighted sum method to cumulate the behavior deviation, but CUSUM method treats all deviation fairly. Hence our method is more sensitive to current network status than CUSUM method, and our method has better detection probability and smaller detection delay than CUSUM algorithm, especially in low intensity attacks. Detection of low intensity attacks is important because early detection of anomaly with increasing intensity attacks would enable defensive action to be taken earlier. On the other hand, our method uses a weighted sum factor to drop the older history profile, but the CUSUM method has no such mechanism. Our method focuses on the up-to-date change of SMTP traffic status. Therefore our method has fewer false alarms than CUSUM method.

In summary, we note that all tested cases show that the detection is accurate. The most important thing is that the parameters PL2 got by our algorithm is proved to have good detecting performance in all scenarios. Therefore, it is safe to affirm that our method has a good detection capability across all possible anomalies that induce change in statistical characteristics of monitored variables without adjusting the parameters of detecting algorithm. In principle, our approach can be easily deployed across different mail servers.

5 Conclusions

In this paper, we propose an effective and robust mechanism for detecting SMTP traffic anomaly. Our detection method cumulates the deviation of current delivering status based on the leaky integrate-and-fire model, which is a weighted sum method. The characteristics of our method are:

1) The proposed method requires neither the set of anomalies to be detected, nor the thresholds to be supplied by the user. This is an important characteristic for real use as it is difficult to ask the system administer to find the appropriate parameters for the detection algorithm in different scenario.

2) Our method needs not store history profile and has low computation overhead, which make the detection method itself immune to attacks.

Our results show that leaky integrate-and-fire method is quite effective at detecting attacks. Compared with other method, our detection method has shorter detection delay and higher detection accuracy, especially in low intensity attacks. Detection of low intensity attacks is important because early detection of anomaly with increasing intensity attacks would enable defensive action to be taken earlier.

References

[1] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," in *ACM SIGCOMM*, pp. 71-82, 2002.

[2] R. Caceres, P. Danzig, S. Jamin, and D. Mitzel, "Characteristics of widearea TCP/IP conversations. Computer Communication Review," in *ACM SIGCOMM*, pp. 101-112, 1991.

[3] CERT/CC Co-ordination Center Advisories, *Carnegie Mellon*, 1988-1998. (http://www.cert.org/advisories/ index. html)

[4] B. Harris, R. Hunt, "TCP/IP security threats and attack methods," *Computer Communications*, vol. 22, no. 10, pp. 885-897, 1999.

[5] J. Hellerstein, F. Zhang, and P. Shahabuddin, "A statistical approach to predictive detection," *Computer Networks*, vol. 35, no. 1, pp. 77-95, 2001.

[6] B. McWilliams, *Wired News: Time-Travel Spammer Strikes Back*, Lycos, Inc. (http://www.wired.com/news/technology/0,1282,61026,00.htm

[7] D. Moore, G. Voelker, and S. Savage, "Inferring internet denial of service activity," in *Proceedings of USENIX Security Symposium 2001*, pp. 9-22, 2001.

[8] R. D. Patterson, M. H. Allerhand, and C. Giguere, "Time-domain Modelling of Peripheral Auditory Processing, A Modular Architecture and a Software Platform," *Journal of the Acoustical Society of America*, vol. 98, pp. 1890-1894, 1995.

[9] J. Postel, *Simple Mail Transfer Protocol*, RFC 821, 1982.

[10] V. Siris, and F. Papagalou, "Application of anomaly detec tion algorithms for detecting SYN flooding attacks," in *Proceedings of IEEE Global Telecommunications Conference*, pp. 2050-2054, 2004.

[11] L. S. Smith, "Onset-based sound segmentation," *Advances in Neural Information Processing Systems*, pp. 729-735, MIT Press, 1996.

[12] M. Thottan, *Fault Detection in IP Networks*, Ph.D. disser-tation, Rensselaer Polytech. Inst., Troy, NY, 2000.

[13] H. Wang, D. Zhang, and K. G. Shin, "Detecting syn flooding attacks," in *IEEE INFOCOM*, pp. 1530-15399, 2002.

[14] N. Yamai, K. Okayama, T. Miyashita, S. Maruyama, and M. Nakamura, "A protection method against massive error mails caused by sender spoofed spam mails," in *Proceeding of the 2005 Symposium on Application and the Internet*, pp. 384-390, IEEE Computer Society, 2005.

Hao Luo was born in 1979. He is a PhD and Assistant Professor of Research Center of Information Intelligent and Information Security Institute of Computing Technology, Chinese Academy of Sciences His research interest is network security.

Binxing Fang was born in 1960. He is a professor and doctor supervisor of Department of Computer Science and Engineering, Harbin Institute of Technology. His current research interests include network security and parallel computing.

Xiaochun Yun was born in 1971. He is a professor and doctor supervisor of Department of Computer Science and Engineering, Harbin Institute of Technology. His current research interests include network security and parallel computing.

Zhigang Wu was born in 1972. He is a post-doctor in the Research Center of Information Intelligent and Information Security Institute of Computing Technology, Chinese Academy of Sciences His research interest is network security.