Real-Time Detection of Stealthy DDoS Attacks Using Time-Series Decomposition

Haiqin Liu and Min Sik Kim
School of Electrical Engineering and Computer Science
Washington State University
Pullman, Washington 99164-2752, U.S.A.
Email: {hliu,msk}@eecs.wsu.edu

Abstract—Recently, many new types of distributed denial of service (DDoS) attacks have emerged, posing a great challenge to intrusion detection systems. In this paper, we introduce a new type of DDoS attacks called stealthy DDoS attacks, which can be launched by sophisticated attackers. Such attacks are different from traditional DDoS attacks in that they cannot be detected by previous detection methods effectively. In response to this type of DDoS attacks, we propose a detection approach based on time-series decomposition, which divides the original time series into trend and random components. It then applies a double autocorrelation technique and an improved cumulative sum technique to the trend and random components, respectively, to detect anomalies in both components. By separately examining each component and synthetically evaluating the overall results, the proposed approach can greatly reduce not only false positives and negatives but also detection latency. In addition, to make our method more generally applicable, we apply an adaptive sliding-window to our real-time algorithm. We evaluate the performance of the proposed approach using real Internet traces, demonstrating its effectiveness.

I. INTRODUCTION

Nowadays, distributed denial of service (DDoS) attacks pose one of the most serious security threats to the Internet [1]–[3]. DDoS attacks can result in a great damage to network services. The DDoS attackers usually utilize a large number of puppet machines to launch attacks against one or more targets, which can exhaust the resources of the victim side. That makes the victim host lose the capability to serve the legitimate customers. Since DDoS attacks can greatly degrade the performance of the network and are difficult to detect, they have become one of the most serious security challenges to the current intrusion detection systems (IDS) [1], [4], [5]. Concerning the current state of the network, every corner of the world is likely to be the target of DDoS attacks. However, as long as they are detected early, the loss can be reduced to the minimum. Therefore, DDoS attack detection still attracts much concern from researchers.

Over the past decade, many efforts have been devoted to the detection of DDoS attacks. A typical approach to detecting DDoS attacks in a network is to detect whether the amount of the total flow or other similar metrics exceeds a certain threshold, which is determined based on the traffic history. However, the problem of identifying attack traffic is generally difficult because currently the pattern of the normal network flow is so dynamic and changing that those fixed thresholds can result in a high false positive rate. There are methods that utilize adaptive thresholds which can change according to the network conditions [6]. The main drawback of such methods is that those thresholds are still hard to be determined to get high detection accuracy, and that the effectiveness greatly depends on the previous training by using historical data. Moreover, attackers can still manipulate their traffic and packets to defeat detection.

We introduce a new type of DDoS attacks called stealthy DDoS attacks, which can be launched by sophisticated attackers. For example, a smart attacker can inject the attack traffic in a very slow speed in order to increase those thresholds and thus achieve the final attack goal. Such DDoS attacks are called shrew DDoS attacks or low-rate DDoS attacks [7], and are a subclass of stealthy DDoS attacks. Cheng et al. firstly utilized the power spectral density (PSD) of network flow to detect general TCP SYN flood [8]. Chen and Hwang also used the power spectral density (PSD) of network flow to detect shrew attacks [9]. The attack types they focused on are stealthy, periodic, pulsing, low-rate, and embedded in TCP or UDP flows. Luo and Chang studied the characteristics of shrew attacks with a wavelet-based approach [10]. Lu and his group [11] proposed a network-wide detection scheme for DDoS attacks by exploiting spatial and temporal correlation of attack traffic. Their study mainly focused on the spoofed address attack, which is unsuitable for detecting recent attacks launched from true source IP addresses [12]. In addition, a number of PCA-based approaches [13], [14] were proposed to detect network-wide DDoS attacks by decomposing the original traffic into the normal and abnormal subspaces and then finding out statistical outliers in the abnormal subspace. Since the PCA-based approaches usually require pretreatment on the feature matrix like the zero-mean transformation, which is necessary for the detection accuracy of them, we argue that the PCA methods is hard, if not impossible, to be adopted in real-time detection. Unfortunately, none of the defense schemes above can identify and filter out general stealthy DDoS attacks effectively and accurately. The main reason is that zombies can increase the number of attack packets in a very low pace, which will be able to defeat the traditional baseline-based detection schemes by stealthily promoting those baselines. In order to detect them, new approach needs to be proposed.

In this paper, we adopt flow connection entropy (FCE)
series to reduce the dimensionality of the high-dimensional traffic sequence, and then use the time-series decomposition method to divide the FCE series into a trend component and a steady random component. We then analyze those components to detect the anomaly of both long-term and short-term trends in the traffic. By separately analyzing each component and synthetically evaluating the result, we can get obtain the overall behavior of the traffic for the real-time anomaly detection.

The remainder of this paper is organized as follows. Section II introduces stealthy DDoS attacks. Section III describes the time series properties of the network traffic and the rationale and methodology of traffic decomposition. A real-time stealthy DDoS detecting algorithm is proposed in Section IV. In Section V, experimental results are presented and the conclusion follows in Section VI.

II. STEALTHY DDoS ATTACKS

We introduce the fundamentals of stealthy DDoS attacks and compare their properties with traditional DDoS attacks. Based on the comparison, we present in detail why this kind of DDoS attacks cannot be easily detected by the conventional methods.

![Fig. 1. A typical low-rate DDoS attack pattern](image1)

Length of burst $L$

Burst rate $R$

Period of attack $T$

The terminology “stealthy DDoS attacks” proposed in this paper mainly refers to shrew DDoS attacks and slowly-increasing-intensity DDoS attacks (SIDA). The shrew DDoS attacks were firstly introduced in [7], which was followed by a series of related research [10], [15], [16]. Generally, a shrew DDoS attack refers to a periodic, pulsing, and low-rate attack traffic embedded in TCP or UDP flows. A typical low-rate DDoS attack is illustrated as a periodic waveform shown in Fig. 1, where $T$ is the time period of an attack, $L$ is the length of a burst period, and $R$ is the burst rate. We consider the shrew DDoS attacks as a subclass of general stealthy DDoS attacks. In this paper, our research focuses on the latter one, which can be launched by a patient, intelligent attacker in no great rush. Fig. 2 shows a generic example of a slowly-increasing-intensity DDoS attack, where the parameter $I$ means the initial intensity of the attack traffic. $T$ is the length of the period of burst and $\Delta I$ is the increment of the attack intensity each time. The value of $\Delta I$ can be manipulated by a smart attacker and be controlled within a very small range to hide the attack behavior. The overview of the SIDA curve is slowly-increasing scalariform. That means the sophisticated hacker can stealthily enhance the threshold for judging legitimate traffic and thus defeat those previous detection methods that detect based on thresholds. This type of DDoS attacks is even more dangerous than those traditional attacks because it is harder to detect, especially when embedded in a large amount of traffic, and thus can greatly delay the anomaly detection.

To the best of our knowledge, none of the previous research focus on the detection of SIDA. In order to detect the new attack type, a new approach needs to be employed.

III. INTERNET TRAFFIC ANALYSIS

The Internet traffic is complex network flows with strong outburst and instability. In this paper, we sample the flow connection entropy (FCE) series from the Internet traffic. By calculating the distribution of the FCE series, we can obtain a coarse-grained estimation of the traffic. We define FCE, which reflects the change of the flow distribution caused by DDoS attacks, as follows.

**Definition** A flow $f_i$ is a 3-tuple $\{sip_i, dip_i, dport_i\}$, where $sip_i$ represents source IP address, $dip_i$ destination IP address, and $dport_i$ destination port number.

This definition of flow can reflect main characteristics of DDoS attack traffic, because a typical DDoS attack pattern is that each zombie machine from a BotNet launches one connection to target at a certain service port of one or more victim machines.

**Definition** The flow connection entropy (FCE) of a set of flows is defined as

$$FCE = -\sum_{i=1}^{n} p(f_i) \log_2 p(f_i)$$

(1)

where $p(f_i)$ is the probability of receiving a packet belonging to flow $f_i$.

During certain time period $\Delta t$, we consider the frequency of the packets belonging to $f_i$ as the estimation of $p(f_i)$.

From the definitions, we can see that a DDoS attack will result in an abnormal increase in the FCE of the network traffic.
A. Statistical Property of FCE Series

We sample network traffic with sampling period $\Delta t$ and calculate FCE of every interval. Therefore, the packet arrivals are modeled by FCE time series: $Z(N, \Delta t) = \{FCE_i, i = 1, 2, ..., N\}$, where $N$ is the length of the time series. The $k$th order auto-correlation coefficient of the series $Z$ is then defined as:

$$\rho_k = \frac{\sum_{i=1}^{N-k} (FCE_i - \bar{FCE})(FCE_{i+k} - \bar{FCE})}{\sum_{i=1}^{N} (FCE_i - \bar{FCE})^2}$$  \hspace{1cm} (2)

where $\bar{FCE}$ is the mean of FCE series.

We then obtain the double auto-correlation (DA) coefficient $\rho_k'$ by re-calculating auto-correlation coefficient of the original auto-correlation coefficient series. Fig. 3(a) shows the DA coefficient of normal traffic. From that result, we can see that $\rho_k'$ is near 0 when the order $k$ is larger than 1. If FCE is stationary, $\rho_k'$ should decay rapidly as $k$ increases [17]. Otherwise, $\rho_k'$ fluctuates as $k$ increases. Since $\rho_k'$ in Fig. 3(a) stabilizes as $k$ increases, we can conclude that the FCE series for the normal traffic are stationary. That is to say, for the normal network traffic FCE series at present time is independent of the previous network status. However, the DA coefficient of normal traffic with injected SIDA traffic, shown in Fig. 3(b), is much larger than 0 even when the order $k$ is greater than 1. Besides, its value decreases in a very slow pace. Clearly, the injected attack traffic makes the FCE series of network traffic non-stationary. Thus, the traditional auto-regressive (AR) model does not perform well.

B. Decomposition Model of Short-Term Traffic

Today’s Internet traffic depends on many factors, such as user behavior, network architecture, and unexpected accidents, which make the network traffic contain both stationary and non-stationary components. According to a study conducted by Guang et al. [18], generally the long time-scale traffic exhibit the characteristics of trend, period, mutation, and randomness. Specifically, they divided the large-timescale time series such

as $FCE_i$ into trend component $A_i$, period component $P_i$, mutation component $B_i$, and random component $R_i$. Therefore, for long time-scale traffic, it can be modeled as:

$$FCE_i = A_i + P_i + B_i + R_i$$  \hspace{1cm} (3)

where $A_i$ and $P_i$ belong to long-term changes which represents the smooth process of network traffic behavior while $B_i$ and $R_i$ belong to short-term changes reflecting uncertainty in network traffic.

In order to come up with a real-time detection algorithm which can be executed in an incremental way, we introduce a sliding window into our algorithm. Generally, the size of the sliding window should be kept small (10 minutes in our implementation) to adapt to real-time detection requirements. With larger window size, more memory resources will be consumed.

For a short period of time, we claim that the mutation component and the period component are negligible. In other words, the network traffic within the window can be considered as the consequence of the interaction between the trend component and random component only. This assumption greatly simplifies the real-time detection algorithm and makes our approach more practical. Under this assumption, the FCE series is modeled as

$$\begin{align*}
FCE_i &= FCE_{i-1} + R_i \\
R_i &= \eta_i + \varepsilon_i
\end{align*}$$  \hspace{1cm} (4)

where $FCE_{i-1}$ denotes the trend component, $R_i$ is the random component containing $\eta_i$ and $\varepsilon_i$, which represents the steady random component and the measurement error (white noise) component, respectively. The value of $\varepsilon_i$ can reflect the absolute error between the prediction sequence and the measured sequence.

Fig. 4 demonstrates the overview of the detection process based on the time series decomposition method. Given a series of FCE data, we first obtain the trend and random components by decomposition of the series. Then, anomaly detection methods are applied to each component separately. Results are collected at the decision module to make a comprehensive decision. Because each anomaly detection method has its own strength under certain conditions, the proposed synthetic approach has a potential to be more effective than previous approaches in detecting SIDA and general DDoS attacks.

We use a modified exponentially weighted moving average (EWMA) to produce estimate the trend component. The current estimation of FCE is obtained by combining the current FCE with the FCE from the previous period corrected for trend
as follows:

\[
\begin{align*}
FCE_i &= \alpha_i FCE_i + (1 - \alpha_i) FCE_{i-1} \\
\alpha_i &= \alpha_{\text{max}} (1 - e^{-C \beta_i}) \\
\beta_i &= \frac{|FCE_i - FCE_{i-1}|}{FCE_{i-1}}
\end{align*}
\]

where \(\beta_i\) reflects the extent of the fluctuation. Generally, if \(FCE\) series are subject to large changes, then the proportion of the history should be small so as to quickly attenuate the effect of old observations. In contrast, when \(FCE\) series smoothly change as time goes, the proportion of the history should be large to minimize random variations. The value of the parameter \(C\) is determined empirically through experiments.

After we subtract the long-term trend component from the original series, the remainder of the series can be considered as the random part of the traffic. Hence, the estimated steady random series is obtained from

\[ R_i = FCE_i - FCE_i. \]

C. Anomaly Detection of Each Component

By decomposing the FCE series, the trend component will reflect the majority of slowly-increasing signal while the steady random signal is contained in the random component. After we decompose the original series into two separate components, appropriate approaches will be applied to each component separately.

Conventional approaches can perform well in detecting anomaly with certain properties. For instance, the cumulative sum (CUSUM) technique, which is based on the Sequential Change Point Detection [19], can determine whether the observed time series is statistically homogeneous, and find the time when the change occurs. However, its effectiveness will be weakened when the network signal contains a large proportion of increasing- or declining-intensity signals, because those signals may result in statistical bias, causing false positives. In our approach, we apply the non-parametric CUSUM method [20] to the random component, which has all the advantages of sequential and non-parametric tests with light computational overhead.

For the trend component, we calculate the double autocorrelation coefficient series and examine the first few elements. The calculation of the double autocorrelation coefficient series at phase \(i\) is as follows:

\[ y_i = (y_{i-1} + x_i)^+ \]

\[ y_0 = 0 \]

A. Self-Adaptive CUSUM Technique for Random Component

In order to accurately and timely detect the mutation point of random series \(R_t\), we adopt the CUSUM technique in our approach. The basic idea of CUSUM is to accumulate those small offsets during the process to amplify the varying statistical feature and thus improve the detection sensitivity. CUSUM can detect a small deviation of the mean effectively. It is generally defined as:

\[
\left\{ \begin{array}{l}
y_i = (y_{i-1} + x_i)^+ \\
y_0 = 0
\end{array} \right.
\]

where \(x_i\) is the observed original value and \(\Delta^+\) is \(\Delta\) if \(\Delta > 0\) and 0 otherwise. When \(x_i\) becomes positive from a negative value, \(y_i\) becomes larger, and its exceeding a threshold \(TH_{\text{CUSUM}}\) indicates the change point of the original series. In our implementation, we use \(R_t = R_i - R_H\) as the original series, where \(R_H\) is the upper bound of \(R_i\) series during normal process.

Although the original CUSUM algorithm can quickly and efficiently detect attacks, after an attack period ends, the alarm will remain active. To stop the alarm, we consider the attack is over if \(y_i\) does not grow for \(C_{\text{AttackEnd}}\Delta_t\). When the alarm stops, we reset \(y_i\) to 0.

B. Double Autocorrelation Technique for Trend Component

As shown in Fig. 3(b), the double autocorrelation coefficient can be used as an indicator of the existence of SIDA in network traffic; the injected SIDA traffic increases the double autocorrelation coefficient. It results from the high internal dependency of SIDA traffic. The FCE series obtained from the same traffic also exhibit this property. Based on these observations, we can detect the anomaly in the trend component using the following condition:

\[ \rho_k^*(i) > TH_{\text{DA}}, \quad 2 \leq k \leq K_{\text{max}} \]

where \(\rho_k^*(i)\) is the double autocorrelation coefficient series at phase \(i\). If all the first \(K_{\text{max}}\) elements of the double autocorrelation coefficient series (except the very first element which always equals to 1) exceed the threshold \(TH_{\text{DA}}\), then we conclude that there is SIDA traffic embedded in the traffic.

Considering the fact the SIDA traffic is injected to the normal traffic in a very slow pace, another condition should be introduced in order to detect the SIDA traffic at an early stage. During \(\theta\Delta t\) period, if the sum of the first \(K_{\text{max}}\) element (except the very first one) keeps growing \(|C_{\text{DA}}\theta|\) times, then we consider a SIDA attack is initialized by attackers as follows:

\[
\sum_{j=i-\theta+1}^{i} \left\{ \sum_{k=2}^{K_{\text{max}}} \rho_k^*(j) > \sum_{k=2}^{K_{\text{max}}} \rho_k^*(j-1) \right\} \geq |C_{\text{DA}}\theta| \]

where \(C_{\text{DA}}\) is an empirically-determined constant. The above condition can be checked by the following way. We firstly compare the sum of the first \(K_{\text{max}}\) element of the double autocorrelation coefficient series at phase \(i - \theta + 1\) with that at phase \(i - \theta + 2\). If the former is greater than latter, then 1 will
be accumulated into the left side of the formula, otherwise 0. If the cumulated value during the previous $\theta$ phase is greater than $|C_{DA}\theta|$, then an anomaly is indicated.

C. Adaptive Sliding Window

We use an adaptive size sliding windows into our algorithm, which means the size of the introduced window can be automatically adjusted according to the current network traffic condition. An adaptive sliding window is necessary for improving the performance of the proposed algorithm. Intuitively, when the double autocorrelation coefficient becomes larger than those in previous phases, the window size should be enlarged so as to capture a more obvious increasing or declining trend. Therefore, the following ratio $R_{RA}$ can represent that trend.

$$R_{RA}^{i} = \frac{K_{max}}{K_{max}} \sum_{k=2}^{K_{max}} \rho_k^2(i) \sum_{k=2}^{K_{max}} \rho_k^2(i - 1)$$

(10)

For the random component, the CUSUM technique does not require a specific length of the time series; however, the window size can be smaller in order to reduce the detecting latency.

The initial size of the sliding window is denoted by $L_0$. Every time the window slides forward, the size of the window changes. Based on the discussion above, the size of the adaptive sliding window is adjusted as follows:

$$L_i = \begin{cases} L_{i-1} + C_1 \left[ R_{DA}^{i-1} \right] - C_2 & \text{if } L_i \geq L_0 \\ L_0 & \text{otherwise} \end{cases}$$

(11)

where $C_1$ and $C_2$ are both positive integers, which can be optimally determined by experiments.

D. Overall Algorithm Description

Based on the above discussions, the proposed algorithm can be described as follows. After initializing those values of parameters, we sample the initial FCE series with window length $L_0$. Then, we decompose the FCE series inside the window by the improved EWMA technique proposed in the previous section. The following flow contains two branches. That is to separately apply the DA and CUSUM techniques to trend component and random component, respectively. Note that the value of $y_i$ needs to be reset to 0 when the attack terminates. If one of these two branches detects an anomaly, then an appropriate attack type will be reported. After computing the next $L_i - L_{i-1} + 1$ FCE series, the sliding window advances to the next, and then the algorithm repeats. As we can see, all of the techniques adopted in this algorithm can be implemented using simple processes and small computing resource.

V. EVALUATION

We use actual network traffic downloaded from the MIT Lincoln Laboratory [21] to evaluate our approach. The traffic we tested is synthetically generated by merging the attack traffic and the normal traffic.

We developed our own tool to generate the SIDA traffic, and the increasing rate is adjustable. The duration of one SIDA was normally distributed with mean 15 minutes and variance 1. The attack traffic was periodically generated and the duration of the period was exponentially distributed, with mean 60 minutes. General outburst-like SYN flood attack traffic was also injected into the tested trace, and the duration of each outburst followed the normal distribution with mean 20 seconds and variance 5 seconds. The interval between each outburst was exponentially distributed with mean 1000 seconds. From Fig. 5, we can see that the merged traffic (top) seems to be normal, especially when we compare the traffic volume with the ground truth shown in Fig. 6.

We considered the detection accuracy and the detection latency as the main performance metrics. Here, the detection accuracy contains two aspects: false positive ratio (FPR) and false negative ratio (FNR). Unless otherwise noted, the default setting for the parameters we adopted in our experiment were $\Delta t = 5s$ for sampling FCE series, $\alpha_{max} = 0.4$, $C = 10$ for time series decomposition, $C_{\text{AttackEnd}} = 3$, $TH_{\text{CUSUM}} = 10$ for CUSUM technique detection, $K_{max} = 6$, $TH_{DA} = 0.4$, $C_{DA} = 0.8$, $\theta = 3$ for DA technique, and $L_0 = 50$, $C_1 = 3$, $C_2 = 5$ for the adaptive sliding window.

Fig. 5 shows that after we calculated the FCE series, the
SIDA attack can be greatly amplified from the background traffic. Hence, several periodical SIDA attacks are observed from the FCE series. Such slowly-increasing trends are more obvious when we apply the time series decomposition approach into the original FCE series. Several strong long-period tendencies are observed from the trend component, and it provides the basis for the DA technique to detect its high internal correlation. Concerning the random component, those outbursts, which indicates general DDoS attack stream, can be quickly detected by the improvedCUSUM technique.

Fig. 6 shows the detection result of experiments, which was demonstrated in the same time period as Fig. 5 for comparison. We use alarm series to represent our detecting result, of which the value is 1 when an attack is detected and 0 otherwise. From Fig. 6, nearly all the SIDA can be detected in a very early stage. The average of the detection latency for those SIDAs in our experiment is around 32 seconds, which is very small when compared with the 900 seconds whole period of SIDA. Compare with the ground truth series, the overall FPR in our experiment is around 4.3% and the overall FNR is around 9.8%. Higher accuracy is achievable by finding a better combination of those parameter values, which is still our ongoing work.

VI. CONCLUSION

In this paper, we introduce a new type of DDoS attacks called stealthy DDoS attacks, which can be launched by a sophisticated attacker. Such attacks are different from traditional DDoS attacks and cannot be easily detected by traditional detection methods. For this type of DDoS attacks, we propose a detection approach based on the decomposition of time series, which divides the time series into trend and steady random components. We then analyze different components to detect the anomaly in both long-term and short-term changes of the traffic. By analyzing each component separately and evaluating results synthetically, the approach can greatly reduce both false negatives and false positives. Furthermore, to make our method more generally applicable, we apply the adaptive sliding window to our approach. The experimental results using real Internet traces show the effectiveness of this approach.

REFERENCES