Vanguard: A New Detection Scheme for a Class of TCP-targeted Denial-of-Service Attacks

Xiapu Luo, Edmond W. W. Chan and Rocky K. C. Chang
Department of Computing
The Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong, SAR, China
Email: {csxluo|cswwchan|csrchang}@comp.polyu.edu.hk

Abstract— **A few low-rate, TCP-targeted Denial-of-Service (DoS) attacks have been recently proposed, including the Shrew attack, Reduction of Quality (RoQ) attack, and Pulsing DoS (PDoS) attack. All of them use periodic attack pulses to throttle TCP flows. These attacks could potentially become major threats to the Internet's stabiliity and therefore they have motivated the development of a number of detection mechanisms for such attacks. However, those detection mechanisms are designed for specific attacks. Moreover, they assume that the period of the attack pulses is a nonzero constant. Unfortunately, these assumptions can be easily thwarted by more sophisticated attack strategies. In this paper, we propose a new detection system called Vanguard to identify a wide range of the aforementioned low-rate, DoS attacks, including the traditional flooding-based attacks as a special case. Vanguard can also detect attacks with randomized attack periods. We have validated Vanguard's efficacy based on extensive test-bed experiments. We have also compared Vanguard with other recently proposed detection systems.**

I. Introduction

Denial-of-Service (DoS) attacks have emerged as a major threat to the Internet's stability today [1], [2], [3]. Existing mechanisms for detecting DoS attacks can be classified into two approaches [4]: pattern-based detection and anomaly-based detection. The pattern-based detection makes use of stored patterns or signatures of known attacks to facilitate detection. An obvious drawback is its inability to detect unknown attacks. Snort [5] is a classical example of pattern-based detection system. To overcome this limitation, some approaches have been proposed recently to automatically extract hidden traffic patterns [6], [7], [8]. However, the attacker could still evade these schemes by employing polymorphic attack patterns and spoof packets. On the other hand, the anomaly-based detection compares the current status of systems or networks with their normal status, and raises alarms whenever the deviation exceeds a given threshold. Although this approach is capable of identifying unknown attacks, it induces a relatively high false alarm rate. Various anomaly-based detection mechanisms have been proposed and they are based on traffic rate [9], [10], TTL value [11], packet header information [12], statistical analysis [13], [14], data mining [15], [16] and signal processing [17], [18].

Traditionally, the DoS attack floods a victim with a large amount of attack packets. We refer this kind of attack to as flooding-based DoS (FDDoS) attacks. However, many recently proposed DoS attacks, such as [19], [20], [21], [22], use an average attack rate smaller than the bottleneck bandwidth, and they target at TCP flows. These low-rate DoS attacks was started from the seminal work of *Shrew attack* [19]. By sending out attack pulses with a constant period that matches with the TCP's minimum retransmission timeout value, i.e. 1 second recommended in RFC2988 [23], the Shrew attack can force victim TCP flows to frequently enter the timeout (`TO`) state. The *Pulsing DoS attack* (PDoS) [21] further generalizes the Shrew attack by exploiting the vulnerability of the TCP's congestion control mechanism. The *RoQ attack* [20] sends out periodic attack pulses to RED routers in order to induce an unusually high dropping probability.

On the defense side, a dynamic time warping (DTWP)-based [24] and a spectrum-based [25] mechanisms were proposed to detect a Shrew attack. The former uses the DTWP algorithm to compare the feature of the sampled incoming traffic with pre-defined attack traffic patterns, while the latter looks into the frequency-domain characteristics of the incoming traffic to discover anomalies due to the attack. On the other hand, for the PDoS attack detection, a discrete wavelet transform (DWTM)-based detection mechanism [21] was proposed which is based on two types of traffic anomalies—severe fluctuations in the incoming TCP traffic and a decline in the trend of the outgoing TCP ACK traffic.

However, there are 2 major shortcomings common to the proposed detection mechanisms. First, they were designed for a specific kind of low-rate DoS attack, and they may not be effective for detecting other kinds of low-rate attacks or even the traditional FDDoS attack.

Second, they all assume that the attack period is a constant [19], [21], [20], but attackers can always launch their attacks with randomized attack periods, or even with a null idle time between two attack pulses, i.e., low-rate FDDoS attacks. Since the attacks can have various attack patterns, i.e. randomized attack periods, we name them as *Polymorphic DoS (PMDoS) attacks*. Based on extensive experiment results, we have discovered that the PMDoS attack can evade the aforementioned detection mechanisms. In particular, the PMDoS attack with randomized attack periods are very effective at thwarting the DTWP-based and spectrum-based mechanisms, but it can still be detected by the DWTM-based mechanism. Moreover, the PMDoS attack with a null idle period can render both the DTWP-based and DWTM-based mechanisms ineffective.

The main contribution of this paper is to propose a single detection scheme for the PMDoS attacks. To this end, we first propose a general model for the PMDoS attacks, including Shrew attack, PDoS attack, RoQ attack and FDDoS attack. Next, we propose a new detection scheme, called Vanguard, which is based on detecting anomalies in 3 types of network traffic statistics: The outgoing TCP ACK traffic, the ratio of the incoming TCP data traffic and the outgoing TCP ACK traffic, and the volume distribution of the incoming TCP data traffic. We have also implemented Vanguard as a Snort plug-in [5]. The results obtained from extensive test-bed experiments show that Vanguard is more effective than other approaches in detecting the PMDoS attack.

The rest of the paper is organized as follows. Section II presents the model and analytical results for the PMDoS attack. Section III discusses the mechanism of Vanguard and other proposed detection schemes. In section IV, we first investigate the impact of the PMDoS attack on TCP flows, and then present some experimental results to evaluate Vanguard, and to compare Vanguard with the other detection approaches. We finally conclude this paper with future work in section V.

II. The polymorphic DoS attack

A. Modelling the PMDoS attack

We model a PMDoS attack as an *Alternating Renewal Process*, which controls the sending rate of attack packets that can be in one of two states: R_{attack} for a period of T_{on}, and 0 *bps* (bits per second) for a period of T_{off}. Therefore, based on the results from *Renewal Theory* [26], we obtain the average attack rate $R_{average}$ of a PMDoS attack in Lemma 1.

Lemma 1: The average attack rate of a PMDoS attack is given by

$$R_{average} = \frac{E[T_{on}]R_{attack}}{E[T_{on}] + E[T_{off}]}, \tag{1}$$

where T_{on} and T_{off} are i.i.d. random variables. Besides, according to the definition of the low-rate attack, the PMDoS attack should satisfy the following constraint:

$$R_{average} \leq R_{bottle}, \tag{2}$$

where R_{bottle} is the bandwidth of the bottleneck.

Based on Eq. (1), we know that the PMDoS attack is equivalent to a *PDoS attack* when T_{on} and T_{off} are degenerate random variables (constant values). Moreover, if T_{off} is close to 1 second and T_{on} is approximate to the round-trip time (RTT) of victim TCP flows, then the PMDoS attack is equivalent to a *Shrew attack*. Furthermore, when T_{off} goes to 0, the PMDoS attack becomes a flooding-based DoS attack with a constant sending rate, for which Eq. (2) becomes

$$R_{attack} \leq R_{bottle}. \tag{3}$$

Definition 1: The normalized throughput degradation cased by a PMDoS attack, denoted by Γ, is defined as

$$\Gamma = 1 - \frac{\Psi_{attack}}{\Psi_{normal}}, \tag{4}$$

where Ψ_{attack} and Ψ_{normal} denote the amount of data (bytes) successfully sent by the victim TCP flows in the presence of and in the absence of a PMDoS attack within the same period, respectively.

Definition 2: The normalized cost of a PMDoS attack, denoted by γ, is defined as

$$\gamma = \frac{R_{average}}{R_{bottle}}. \tag{5}$$

B. Analyzing the PMDoS attack

By exploiting the TCP congestion control mechanism, the PMDoS attack attempts to throttle the victim TCP flows by restraining their congestion window sizes (`cwnd`). Based on the relationship between R_{attack} and R_{bottle}, we divide the PMDoS attack into two categories and analyze their impact on `cwnd`, respectively.

1) When $R_{attack} \leq R_{bottle}$, the PMDoS attack will behave like a constant-bit-rate (CBR) source. We denote this class of PMDoS attacks by $\mathbb{A}^-$,
2) When $R_{attack} > R_{bottle}$, the PMDoS attack will dispatch intermittent attack pulses because $T_{off} > 0$. We denote this class of PMDoS attacks by $\mathbb{A}^+$,

Proposition 1: If the average value of `cwnd` in the absence of PMDoS attack is $\mathbf{E}[W]$, then under $\mathbb{A}^-$ PMDoS attack it becomes

$$\mathbf{E}[W^-] = (1 - \frac{R_{attack}}{R_{bottle}})\mathbf{E}[W]. \tag{6}$$

Proof: Since the victim TCP senders will increase its packet transmission rate whenever there is available bandwidth, their flows will make full use of the available bandwidth ($R_{bottle} - R_{attack}$) after the $\mathbb{A}^-$ PMDoS attack occupies R_{attack} portion of the bottleneck's bandwidth R_{bottle} [27], i.e. TCP's packet transmission rate $\mathbf{E}[R_{tcp}] = R_{bottle} - R_{attack}$. Since R_{tcp} can be modeled as $\mathbf{E}[R_{tcp}] = \frac{3\mathbf{E}[W]}{4RTT} MSS$ [28], where MSS is the maximum segment size and RTT is assumed to be a constant value as [28], [29], $\mathbf{E}[W]$ will decrease the same percent $(1 - \frac{R_{attack}}{R_{bottle}})$ as R_{tcp} does. ■

For the $\mathbb{A}^+$ PMDoS attack, since $R_{attack} > R_{bottle}$, the bottleneck would be blocked for a period of $\frac{R_{attack} T_{on}}{R_{bottle}}$. We assume that during such period all the victim TCP flows would suffer from packet loss and then enter the fast retransmit/fast recovery (FR) state, during which each TCP flow will employ additive increase and multiplicative decrease (AIMD) algorithm to control its cwnd. In this paper, we consider a general AIMD algorithm specified by $AIMD(a, b),\ a > 0,\ 1 > b > 0$ as follows [30]: the sender decreases its cwnd from W to $b \cdot W$ whenever it enters the FR state, and increases its cwnd from W to $W + a$ per RTT until receiving another congestion signal. Many TCP protocols, such as Tahoe, Reno, and New Reno, use $AIMD(1, 0.5)$. Moreover, many TCP implementations do not send an ACK for every received packet. Instead, they send a delayed ACK after receiving d consecutive full-sized packets, where d is typically equal to 2 [31]. In this case, the sender's cwnd is only increased by $\frac{a}{d}$ per RTT.

Therefore, we can model the value of cwnd in the presence of $\mathbb{A}^+$ PMDoS attacks as follows:

$$W^+(n+1) = bW^+(n) + \frac{a}{d} \frac{T_{on}(n) + T_{off}(n)}{RTT} \quad (7)$$

Eq. (7) belongs to the general class of classical stochastic difference equation shown in Eq. (8) [32], [33].

$$Y(n+1) = A(n)Y(n) + B(n), \quad n \geq 0. \quad (8)$$

Since the attacker can send an arbitrary sequence of attack pulses, we consider a general attack pattern, whose $T_{period}(n) = T_{on}(n) + T_{off}(n)$ is a stationary and ergodic stochastic process.

Based on Theorem 1 in [32], we can obtain the solution to Eq. (7) in Proposition 2.

Proposition 2: If $T_{period}(n)$ is a stationary and ergodic stochastic process, then there is a unique stationary solution for Eq. (7) as follows:

$$W^{+*}(n) = \frac{a}{dRTT} \sum_{j=0}^{\infty} b^j (T_{period}(n - j - 1)), \quad (9)$$

Proof: Let $A(n) = b$ and $B(n) = \frac{a}{d} \frac{T_{period}(n)}{RTT}$. It can easily be proved that $A(n)$ and $B(n)$ fulfill the following requirements in [32], [33]:

$$-\infty \leq \mathbf{E}[log(|A(0)|)] < 0 \quad (10)$$

$$\mathbf{E}[log(|B(0)|)]^+ < \infty,\ x^+ = max(0, x), \forall x \in \mathbb{R} \quad (11)$$

■

We use the similar technique in [34] to characterize the cwnd under the $\mathbb{A}^+$ PMDoS attack. Since $W^+(n)$ will converge to $W^{+*}(n)$ absolutely almost surely according to Theorem 1 in [32], we can obtain the converged value by calculating the expectation of $W^{+*}(n)$.

Corollary 1: The expectation of $W^{+*}(n)$ is given by

$$\mathbf{E}[W^{+*}(n)] = \frac{a\mathbf{E}[T_{period}(n)]}{dRTT(1-b)}. \quad (12)$$

If both T_{on} and T_{off} are constant values and let $T_{AIMD} = T_{on} + T_{off}$, then we get the same converged value (i.e. $W_C = \frac{aT_{AIMD}}{dRTT(1-b)}$) in [21].

Proof: By taking expectation on both sides of Eq. (9). ■

III. Vanguard: A new anomaly-based detection scheme for the PMDoS attack

In this section, we will describe and analyze three kinds of traffic anomalies caused by the PMDoS attack, and then introduce a new anomaly-based detection scheme, Vanguard, to identify the attack. Although attackers can use UDP packets, arbitrary IP packets or ICMP packets to launch the attack, they would prefer TCP packets as attack packets. The reason is that the router would dispatch different types of packets into separate service queues, so that different types of flows would not affect one another. Hence, we only consider attack packets of TCP type in this paper. Moreover, we do not consider the case that the attacker use non-compliant TCP flows to occupy the bottleneck because some existing approaches can be used to identify and punish such malicious flows [35].

A. Traffic anomalies caused by PMDoS attacks

Since the PMDoS attacks target at TCP flows, they will inevitably induce anomalies in the TCP traffic. In the following, we discuss three types of traffic anomalies that have been confirmed through analysis and extensive experiments, and are employed as the attack indicators.

1) Indicator I - Decline in the outgoing TCP ACK traffic: Since the PMDoS attack will cause packet loss in victim TCP data traffic, the number of the corresponding TCP ACK packets will decrease also. Such a simple indicator of the PMDoS attack has been used in [21]. However, it may cause false alarms if the decline in the incoming TCP data traffic does not correspond to the attack. Moreover, it may miss the attack if an advanced attacker manipulates the normal TCP flows to trigger more ACK packets, e.g. by exploiting reordered TCP data packets, to compensate for the decline of the ACK packets due to the attack.

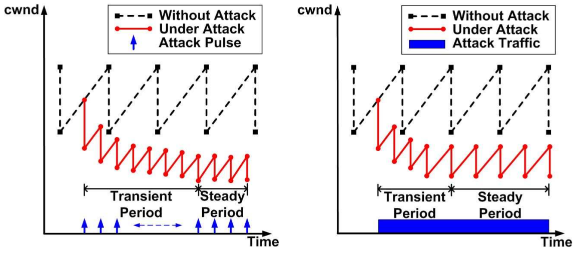
2) Indicator II - The increase in the ratio of the incoming TCP traffic to the outgoing TCP ACK traffic: In order to eliminate the potential false alarms caused by *Indicator I*, we can monitor the ratio I_{Ratio} of the incoming TCP data traffic to the outgoing TCP ACK traffic. When there is no packet loss, $I_{Ratio} = d$, where d, which is typically equal to 2 [31], denotes the number of consecutive incoming legitimate TCP data packets before an outgoing ACK packet is sent. When there is packet loss or packet reordering, $I_{Ratio} = 1$ since the receiver will send an immediate duplicate ACK whenever an out-of-order packet arrives [31]. With this indicator, decline in the outgoing ACK traffic due to decline in the normal TCP data traffic will not trigger false alarm. On the other hand, the attack has to trigger more ACK packets to conceal its attack packets. A similar technique has been used in D-WARD [36] to detect DDoS attack. However, while D-WARD is installed in attackers' source networks, Vanguard is located at the victim's network. Moreover, D-WARD cannot handle the case when the attacker makes use of reordered TCP packets to trigger more ACK packets.

3) Indicator III - The change in the distribution of the incoming TCP traffic: Based on *Indicators I* and *II*, we can discover most of the PMDoS attacks, except for those sophisticated variants that trigger more outgoing TCP ACK traffic to conceal its attack packets. However, we can observe a third kind of traffic anomaly—the change in the distribution of the incoming TCP traffic. As a result of the PMDoS attack, victim TCP flow distribution would be different from the original one. For example, in the presence of an $\mathbb{A}^+$ PMDoS attack, the cwnd of the senders would be forced to converge to a low value due to the periodic packet loss as shown in Fig. 1(a). On the other hand, $\mathbb{A}^-$ PMDoS attack behaves like a CBR flow, the cwnd would also be constrained. However in this case, the fluctuation of cwnd is modulated by the limited bandwidth instead of the attack. In both cases, the incoming TCP traffic distribution would be changed.

B. Vanguard: a new detection scheme

In this subsection, we propose a new detection scheme, Vanguard, which detects ongoing PMDoS attacks by monitoring the three aforementioned indicators. The detection procedure consists of two main steps:

1) Locate change points in statistics extracted from the three indicators.
2) Raise an alarm if the decision statement is true.



(a) cwnd under A^+ attack (b) cwnd under A^- attack

Fig. 1. The cwnd under PMDoS attack.

In order to detect anomaly in the outgoing ACK traffic indicated in its trend, we apply the *moving average* algorithm in time series [37]. We use the following notations:

1) $D_{ACK}(i), i = 1, 2, \ldots$ is the ith sample of the outgoing ACK traffic;
2) T_{sample} is the length of each sampling interval in seconds;
3) W_D is the length of detection window in seconds;
4) $N_{sample} = W_D/T_{sample}$ be the number of samples in a detection window.

$I_{ACK}(n)$ is the statistic for the outgoing ACK traffic as follows:

$$I_{ACK}(n) = \frac{1}{W_D} \sum_{i=(n-1)N_{sample}+1}^{nN_{sample}} D_{ACK}(i). \quad (13)$$

We define the ratio of the incoming TCP traffic to the outgoing ACK traffic as

$$I_{Ratio}(n) = \frac{I_{Data}(n)}{I_{ACK}(n)}, \quad (14)$$

where $I_{Data}(n) = \frac{1}{W_D} \sum_{i=(n-1)N_{sample}+1}^{nN_{sample}} D_{Data}(i)$ and $D_{data}(i), i = 1, 2, \ldots$ is the ith sample of the incoming TCP data traffic. According to the analysis in section III-A, I_{ACK} will decrease in the presence of non-sophisticated PMDoS attack due to the decline of the victim incoming TCP traffic, while I_{Data} may or may not reduce depending on whether the attack packets can compensate for the dropped TCP data packets. I_{Ratio} would be larger than d if the attack packets cannot trigger more outgoing ACK packets.

We employ the *color histogram indexing* method [38] to capture the change in the distribution of the incoming TCP traffic. In the field of image retrieval, this method is a robust approach to computing similarity of two images [39]. Our basic idea is to measure the similarity index (SI) of the distribution of the incoming TCP traffic and that of the normal TCP traffic. Since the PMDoS attack will change the incoming TCP traffic distribution, there will be an abrupt change in the series of SI. The algorithm consists of three steps: First, we compute a traffic histogram for each detection window of samples. Given minimum (D_{data}^{min}) and maximum (D_{data}^{max}) values of the incoming TCP traffic samples, we divide the range $[D_{data}^{min}, D_{data}^{max}]$ into B disjoint subregions of equal size, named as *histogram bins*. The traffic histogram $h(n)$ of the nth detection window is then obtained by counting the number of samples $h_{n,i}$ that falls in the histogram bin $i,\ 1 \leq i \leq B$, i.e. $h(n) = [h_{n,1}, \ldots, h_{n,B}]$. Second, a cumulative histogram $H(n) = [H_{n,1}, \ldots, H_{n,B}]$ is obtained by $H_{n,i} = \sum_{j \leq i} h_{n,j}$. Third, with the cumulative histogram of the normal TCP traffic $\widehat{H} = [\widehat{H}_1, \ldots, \widehat{H}_B]$ and $H(n) = [H_{n,1}, \ldots, H_{n,B}]$, define the SI of nth detection window as

$$I_{Dis}(n) = \sqrt{\sum_{j=1}^{B} (H_{n,j} - \widehat{H}_j)^2}. \qquad (15)$$

We obtain I_{ACK}, I_{Ratio} and I_{Dis} at the end of each detection window, and the system raises an alarm if the following decision statement is true:

$$I_{Ratio} \uparrow \vee \{I_{ACK} \downarrow \wedge I_{Dis} \uparrow\}, \qquad (16)$$

where $\uparrow$ and $\downarrow$ represent abrupt increase and decrease, respectively. Accordingly, Vanguard first locates any abrupt change in each indicator, and raises the alarm if the indicator I_{Ratio}, or both indicators I_{ACK} and I_{Dis} are found to be abnormal.

The CUSUM, a non-parametric change point detection algorithm [40], is used to capture abrupt changes in the sequences $\{I_{ACK}(n)\}$, $\{I_{Ratio}(n)\}$, and $\{I_{Dis}(n)\}$. This algorithm assumes that the mean of the variables being monitored will change from negative to positive. Since $I_{ACK}(n)$, $I_{Ratio}(n)$ and $I_{Dis}(n)$ are always non-negative, we first transform them into three random sequences, $P_{ACK}(n)$, $P_{Ratio}(n)$ and $P_{Dis}(n)$, which have negative mean values under the normal period, as follows:

$$\begin{aligned} P_{ACK}(n) &= \beta_{ACK} - I_{ACK}(n), & (17)\\ P_{Ratio}(n) &= I_{Ratio}(n) - \beta_{Ratio}, & (18)\\ P_{Dis}(n) &= I_{Dis}(n) - \beta_{Dis}, & (19) \end{aligned}$$

where β_{ACK}, β_{Ratio} and β_{Dis} are constants for determining the mean values of $\{I_{ACK}(n)\}$, $\{I_{Ratio}(n)\}$ and $\{I_{Dis}(n)\}$, respectively. Normally, we can set β_{ACK} to $\overline{I_{ACK}(n)} - P_{tolerance}[\triangle(I_{ACK}(n))]$, where $\triangle(I_{ACK}(n))$ is the standard deviation of $I_{ACK}(n)$, and $P_{tolerance}$ defines the sensitivity to the decline in the outgoing ACK traffic by controlling the allowable decrease during the transformation of $\{I_{ACK}(n)\}$. We set β_{Ratio} and β_{Dis} to the upper bound of $\{I_{Ratio}(n)\}$ and $\{I_{Dis}(n)\}$, respectively.

With $y_{P_{ACK}}(n-1)$, the CUSUM values of $P_{ACK}(n-1)$, and $P_{ACK}(n)$, Vanguard computes the CUSUM value $y_{P_{ACK}}(n)$ as:

$$y_{P_{ACK}}(n) = \max\{0, y_{P_{ACK}}(n-1) + P_{ACK}(n)\}. \quad (20)$$

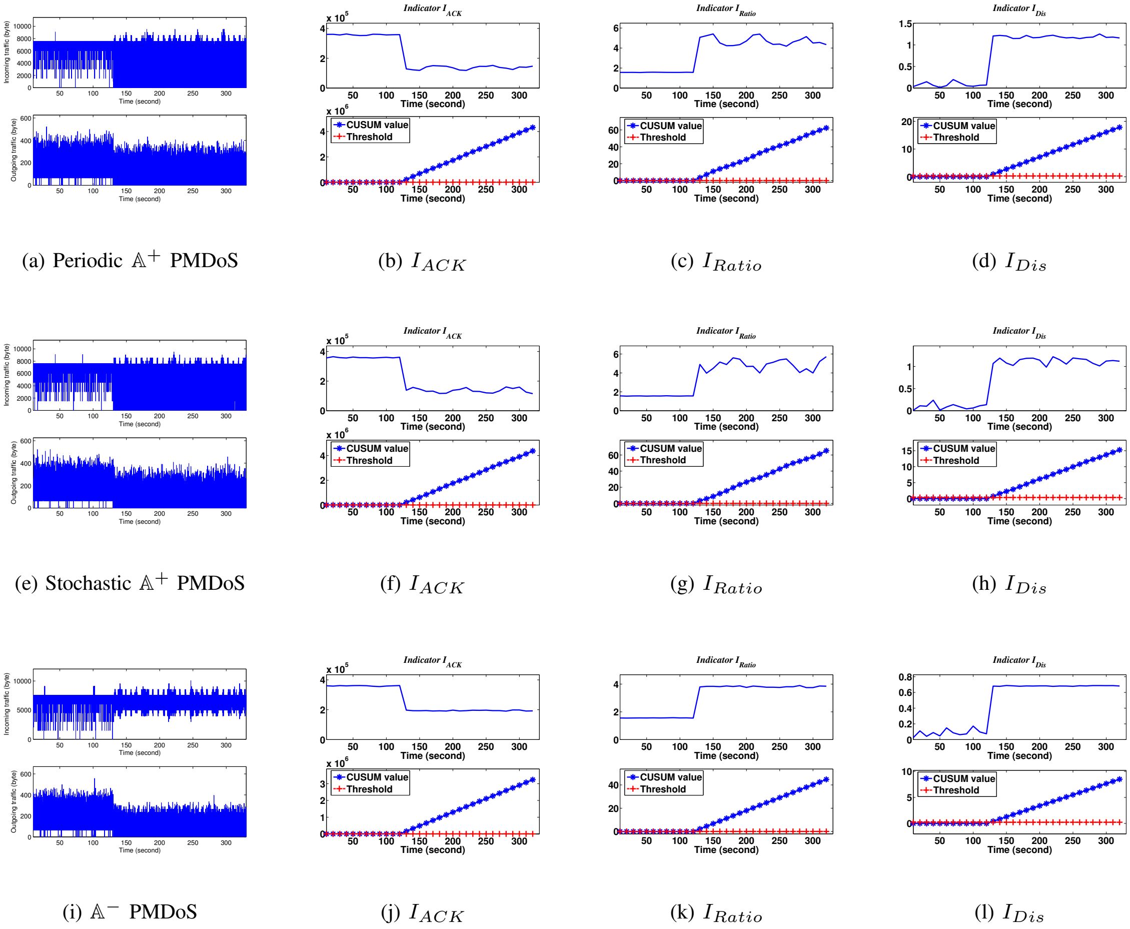
Thus, the presence of the abnormal decline in the outgoing ACK traffic is confirmed if $y_{P_{ACK}}(n) > C_{ACK}^{CUSUM}$, where C_{ACK}^{CUSUM} is the corresponding CUSUM threshold. Similarly, by computing the CUSUM values $y_{P_{Ratio}}(n)$ and $y_{P_{Dis}}(n)$ and by comparing with the corresponding CUSUM thresholds C_{Ratio}^{CUSUM} and C_{Dis}^{CUSUM} same as the above, we can confirm the presence of the increase in $I_{Ratio}(n)$ and the change in the distribution of incoming TCP traffic.

Fig. 2 demonstrates the detection process of Vanguard on a periodic $\mathbb{A}^+$ PMDoS attack, a stochastic $\mathbb{A}^+$ PMDoS attack, and a $\mathbb{A}^-$ PMDoS attack. The data is obtained from test-bed experiments with the presence of cross traffic and a bottleneck capacity of 10Mbps, which will be discussed in more detail in Sec. IV-A. Both $\mathbb{A}^+$ PMDoS attacks operate with $R_{attack} = 25Mbps$ and $R_{average} = 6Mbps$, while the $\mathbb{A}^-$ PMDoS attack operates with $R_{attack} = R_{average} = 6Mbps$. All the attacks have the same attack cost $\gamma = 0.6$. An attacker starts the attack at the 131th second from the beginning of the experiment. For each row of subfigures, the first subfigure shows the raw incoming TCP traffic in the upper panel and the raw outgoing ACK traffic in the lower panel. The second, third and forth subfigures plot the indicators $I_{ACK}(n)$, $I_{Ratio}(n)$ and $I_{Dis}(n)$, respectively. For each set of those subfigures, the upper panel shows the raw data of the indicators, and the lower panel illustrates the CUSUM detection results of these indicators. We observe that the PMDoS attack results in different kinds of incoming TCP traffic and outgoing ACK traffic patterns. However, the abnormal change in the traffic can be instantly revealed from the three indicators, and thus effectively captured by Vanguard.

C. Other detection schemes

Since the defense system may need to process a huge volume of incoming packets, having a low computational complexity is a very important consideration in designing a practical defense system. In this subsection, we compare the computational complexity of Vanguard with other proposed detection schemes. We consider the DWTM-based detection scheme [21], spectrum-based scheme [18], [25], and DTWP-based algorithm [24].

Since all the detection schemes under comparison make decision after collecting and manipulating N data samples in a detection window, their lowest complexity is $\Theta(N)$. For Vanguard, the values of indicators I and II



(a) Periodic $\mathbb{A}^+$ PMDoS (b) I_{ACK} (c) I_{Ratio} (d) I_{Dis}

(e) Stochastic $\mathbb{A}^+$ PMDoS (f) I_{ACK} (g) I_{Ratio} (h) I_{Dis}

(i) $\mathbb{A}^-$ PMDoS (j) I_{ACK} (k) I_{Ratio} (l) I_{Dis}

Fig. 2. Demonstration of the detection process of Vanguard

can be updated upon receiving each data sample. By using bins with the same size, the data sample can quickly locate the bin it belongs to. After that, the burden of computing indicator III is determined by B, which is the number of bins and is usually less than the number of samples (N) in each detection window. The CUSUM algorithm's complexity is $\Theta(1)$ [40]. Therefore, the complexity of Vanguard is $\Theta(N)$.

A discrete wavelet transform (DWTM)-based detection scheme is proposed in [21] to detect the presence of PDoS attacks. Originally, it detects two kinds of traffic anomalies near the TCP receiver: Periodic fluctuations in the incoming TCP data traffic and a decline in the trend of the outgoing ACK traffic. The scheme uses discrete wavelet transform (DWTM) to extract traffic anomalies caused by the PDoS attack, based on which the PDoS attack can be detected through non-parametric change-point algorithms. Therefore, its computational complexity depends on that of the DWTM, which is $\Theta(N)$ [41]. Since the two-stage detection scheme relies on the abnormal fluctuations in the incoming traffic, it will not be able to handle the $\mathbb{A}^-$ PMDoS attack, which constrains the normal fluctuation of TCP flow to a smaller bandwidth, i.e. the bottleneck bandwidth subtracted by the CBR traffic rate.

The spectrum-based detection has been used to differentiate between single-source DoS attacks and multi-source DoS attacks [18], which is employed to detect Shrew attack by observing the change in the power spectral density (PSD) of the incoming TCP traffic [25]. Hence, its computational complexity is mainly determined by that of computing the PSD, which is $\Theta(NlogN)$ [42]. However, the spectral analysis cannot handle the PMDoS attacks which could exhibit various frequencies under different settings of attacks. Just like the hop-frequency techniques, the frequency of the PMDoS attack (except for the Shrew attack) can be changed easily.

A dynamic time warping (DTWP)-based algorithm is proposed in [24] to identify the Shrew attack by matching the pattern of the incoming TCP data traffic with that of Shrew attack traffic. It first employs auto-correlation to extract the signatures of the incoming traffic periodically and then compares the extracted signatures of the incoming traffic with the signatures of Shrew attack traffic through a slightly modified DTWP algorithm. Since the computational complexity of the auto-correlation processing is $\Theta(N^2)$ and that of DTWP is $\Theta(NM)$, (M is the length of selected signatures of Shrew attack), the DTWP-based algorithm's computational complexity is $\Theta(N^2)$. However, the DTWP method may fail if the attack pulses are not separated by a constant interval. Moreover, the DTWP method will not be able to detect the $\mathbb{A}^-$ attacks as there will not be significant square-wave patterns in the incoming traffic.

Table I summarizes the computational complexities of the detection schemes. Together with the DWTM-based scheme, Vanguard achieves the lowest computational complexity. Moreover, as will be shown in section IV, Vanguard can achieve the highest detection rate for the class of PMDoS attacks.

TABLE I

COMPLEXITY OF DIFFERENT DETECTION SCHEMES

Scheme	Computational complexity
Vanguard	$\Theta(N)$
DWTM-based scheme	$\Theta(N)$
Spectrum-based scheme	$\Theta(N \log N)$
DTWP-based scheme	$\Theta(N^2)$

D. A Snort implementation of Vanguard

Fig. 3 depicts the architecture of our implementation of Vanguard. We have implemented this scheme as a preprocessor plug-in in Snort [5] to facilitate real-time PMDoS attack detection. The overall detection mechanism can be summarized as follows:

```
for each intercepted packet $P$ {
  if(Collected $N_{sample}$ continuous samples) {
    performs network traffic analysis
    to evaluate $I_{ACK}(n)$, $I_{Ratio}(n)$ and $I_{Dis}(n)$;
    if(during training period)
      evaluate $\beta_{ACK}$, $\beta_{Ratio}$, $\beta_{Dis}$, $C_{ACK}^{CUSUM}$,
      $C_{Ratio}^{CUSUM}$ and $C_{Dis}^{CUSUM}$;
    else
      perform CUSUM change-point detection;
  }
  if($P$ is a TCP packet originated from or
  transmitted to $H$)
    update corresponding packet counter of
    the current sample;
}
```

In our architecture, after the Vanguard preprocessor is registered in the Snort's preprocessor list through the function `AddFuncToPreprocList()`, it starts intercepting the incoming TCP data traffic and outgoing ACK traffic for a specific host H. When a packet originated from or transmitted to H arrives, the preprocessor records its size and updates corresponding packet counter of the current sampling interval. Whenever N_{sample} continuous samples (a detection window) have been collected, the preprocessor performs the network traffic analysis to evaluate the indicators I_{ACK}, I_{Ratio} and I_{Dis} according to the Eqns. (13)-(15), respectively.

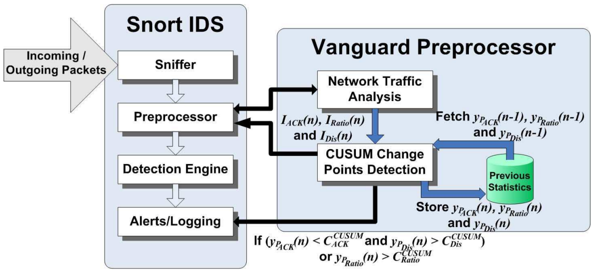

Snort IDS
Incoming / Outgoing Packets
Sniffer
Preprocessor
Detection Engine
Alerts/Logging
Vanguard Preprocessor
Network Traffic Analysis
$I_{ACK}(n)$, $I_{Ratio}(n)$ and $I_{Dis}(n)$
CUSUM Change Points Detection
Fetch $y_{P_{ACK}}(n-1)$, $y_{P_{Ratio}}(n-1)$ and $y_{P_{Dis}}(n-1)$
Previous Statistics
Store $y_{P_{ACK}}(n)$, $y_{P_{Ratio}}(n)$ and $y_{P_{Dis}}(n)$
If ($y_{P_{ACK}}(n) < C_{ACK}^{CUSUM}$ and $y_{P_{Dis}}(n) > C_{Dis}^{CUSUM}$) or $y_{P_{Ratio}}(n) > C_{Ratio}^{CUSUM}$

Fig. 3. Architecture of the implementation of Vanguard.

Before the Vanguard preprocessor actually begins the PMDoS attack detection process, it is necessary for the preprocessor to first determine the constant values for the traffic—β_{ACK}, β_{Ratio}, β_{Dis}, C_{ACK}^{CUSUM}, C_{Ratio}^{CUSUM} and C_{Dis}^{CUSUM}. Accordingly, the preprocessor allows users to specify the number of continuous detection windows N_{W_D} for the training period. After the training period, it evaluates the β_{ACK}, β_{Ratio} and β_{Dis}; and determines the C_{ACK}^{CUSUM}, C_{Ratio}^{CUSUM} and C_{Dis}^{CUSUM} by evaluating the means of the sequences $\{|P_{ACK}(n)|\}_{n=1}^{N_{W_D}}$, $\{|P_{Ratio}(n)|\}_{n=1}^{N_{W_D}}$ and $\{|P_{Dis}(n)|\}_{n=1}^{N_{W_D}}$, respectively.

After the training period, the preprocessor performs the CUSUM change-point detection process to identify the presence of PMDoS attacks. Whenever the statistics arrive, it first verifies the condition $y_{P_{Ratio}}(n) > C_{Ratio}^{CUSUM}$ to determine if any change point occurs in $\{I^{Ratio}(n)\}$. If it holds, the preprocessor will further verify that if either of the conditions $y_{P_{ACK}}(n) > C_{ACK}^{CUSUM}$ and $y_{P_{Dis}}(n) > C_{Dis}^{CUSUM}$ holds. If it is also true, the preprocessor will immediately call the function `SnortEventqAdd()` to pass a PMDoS attack alert to the Snort's Alert/Logging module.

IV. PERFORMANCE EVALUATION

In this section, we explore the effectiveness of PMDoS attacks as well as the performance of Vanguard. Next, we compare our detection scheme with the other existing detection schemes described in section III-C. We have carried out our evaluation using the topology in Fig. 4, which consists of n routers shared by a number of legitimate TCP users and an attacker. All the links, except the bottleneck link, between the routers R_n (the bottleneck router) and R_{n+1} have a one-way propagation delay of t_l ms and a capacity of r_l Mbps. The link are shared by n_t long-lived legitimate TCP flows, an attack flow and cross traffic generated by n_c long-lived TCP flows. The bottleneck link has a one-way propagation delay of t_b ms and a capacity of r_b Mbps, and carries the traffic excluding the cross traffic.

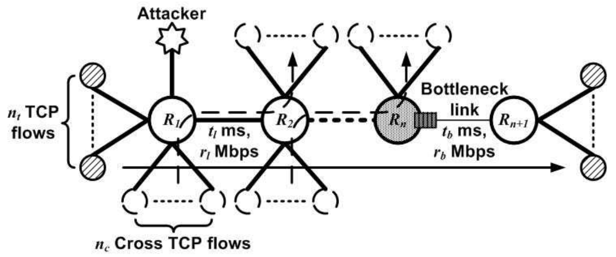


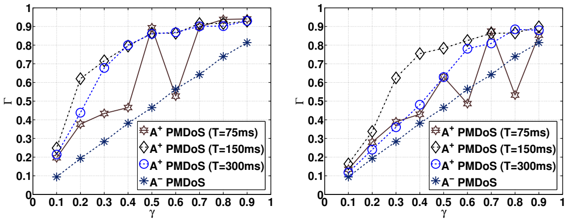
Fig. 4. The network topology for the performance evaluation.

A. The impact of PMDoS attacks

To demonstrate the impact of different variants of PMDoS attacks (periodic $\mathbb{A}^+$, stochastic $\mathbb{A}^+$, and $\mathbb{A}^-$ PMDoS attacks) on the n_t long-lived legitimate TCP flow aggregates, we have conducted extensive test-bed experiments and `ns-2` simulations. Due to the space limitation, we only present the test-bed experiment results in this paper.

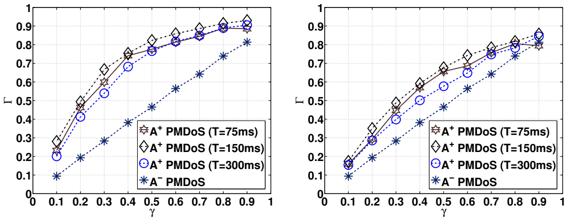
We use the following parameter settings: $n = 2$, $n_t = 15$ (New Reno), $n_c = 10$ (New Reno), $t_l = 15$ms, $t_b = 30$ms, $r_l = 100$Mbps, and $r_b = 10$Mbps. Each of the legitimate TCP flows experiences a fixed RTT of 150ms and employs a minimum `RTO` of 1000ms. The queue at the bottleneck router R_1 is a droptail queue of size $Q = RTT \times r_b$. The period of each experiment is 730s. After 250s, the attacker launches a PMDoS attack until the end of the experiment period. Both the legitimate flows and the cross traffic are generated using Iperf [43]. To simplify the notations, we use T and R to represent T_{on} and R_{attack} in the following figures, respectively.

Fig. 5 depicts the experiment results of Γ resulted from the periodic $\mathbb{A}^+$ PMDoS attacks with $R_{attack} = \{25, 50\}$Mbps and $T_{on} = \{75, 150, 300\}$ms versus the attack cost γ. Fig. 6 depicts the experiment results for the stochastic $\mathbb{A}^+$ PMDoS attacks. For the purpose of comparison, each of the sub-figures also includes the experiment results for the $\mathbb{A}^-$ PMDoS attacks.



(a) $R_{attack} = 25Mbps$ (b) $R_{attack} = 50Mbps$

Fig. 5. The normalized throughput degradation under the periodic $\mathbb{A}^+$ and $\mathbb{A}^-$ PMDoS attacks versus the attack cost.



(a) $R_{attack} = 25Mbps$ (b) $R_{attack} = 50Mbps$

Fig. 6. The normalized throughput degradation under the stochastic $\mathbb{A}^+$ and $\mathbb{A}^-$ PMDoS attacks versus the attack cost.

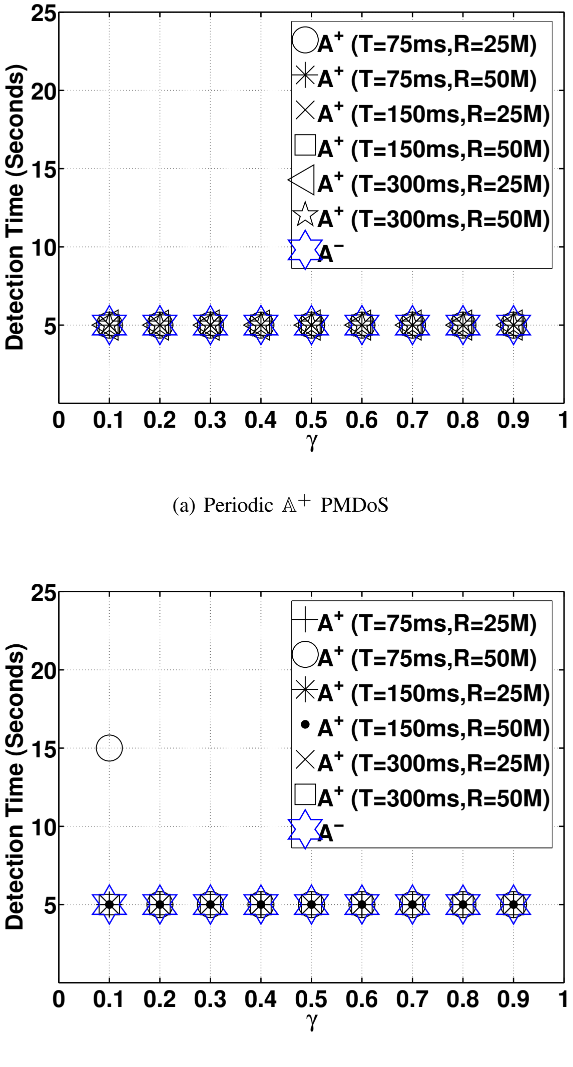
These figures give insight into the performance difference among the 3 variants of PMDoS attacks. First, note that the throughput of the TCP flow aggregates is significantly reduced by both the $\mathbb{A}^+$ and $\mathbb{A}^-$ PMDoS attacks. However, the impact of the periodic and stochastic $\mathbb{A}^+$ PMDoS attacks are generally far more significant than the $\mathbb{A}^-$ PMDoS attack: As shown in Figs. 5(a) and 6(a), while both the periodic and stochastic $\mathbb{A}^+$ PMDoS attacks with $\gamma = 0.5$ have already induced throughput degradation of flow aggregates by more than 80%, the $\mathbb{A}^-$ PMDoS attack with the same cost can only reduce the aggregates' throughput by not more than 50%. In order for the $\mathbb{A}^-$ PMDoS attack to achieve the similar level of degradation, the attacker has to increase its γ to 0.9, which also increases the chance of exposure to the detection mechanisms.

Besides, we notice that the choice of the attack pulse width T_{on} would affect the performance of the periodic $\mathbb{A}^+$ PMDoS attacks more significantly than the stochastic ones. Figs. 5(a) and 5(b) show that when the attack pulse width of the periodic $\mathbb{A}^+$ PMDoS attack ($T_{on} = 75$ms) is smaller than the RTT of the flow aggregates, i.e., 150ms, the normalized aggregate TCP throughput degradation may experience an unexpected decline even when the attack cost increases, e.g., $\gamma = 0.6$ in Fig. 5(a) and $\gamma = \{0.6, 0.8\}$ in Fig. 5(b). For those attack scenarios, since T_{on} does not cover the RTT of the flow aggregates, a portion of the flow aggregates could possibly be affected by the attack traffic periodically. Those survived TCP flows can therefore utilize more available bandwidth, resulting in a smaller aggregate TCP throughput degradation. On the contrary, the randomized attack period of the stochastic $\mathbb{A}^+$ PMDoS attack enables its attack pulses to throttle different legitimate TCP flows. As a result, we observe from Figs. 6(a) and 6(b) that when the attack operates with $E[T_{on}] = 75$ms, the throughput degradation generally increases with the attack cost.

B. The performance of Vanguard

In this section, we present the experiment results of Vanguard through a number of test-bed experiments. Our evaluation is based on the detection delays required to identify different PMDoS attack variants versus attack costs. We use the same procedures and settings for the test-bed experiments discussed in the previous section, except that we have installed a Snort IDS with the Vanguard preprocessor at R_{n+1} to sniff incoming TCP data traffic and outgoing ACK traffic. For the preprocessor configuration, $T_{Sample} = 0.005$s and $W_D = 5$s to achieve a small detection delay, and $N_{W_D} = 40$ to obtain a training period of 200s. Moreover, we set $B = 25$ and $P_{tolerance} = 2$.

Fig. 7 plots the detection time of PMDoS attacks against attack costs. Fig. 7(a) represents the results for the periodic $\mathbb{A}^+$ PMDoS attacks with $T_{on} = \{75, 150, 300\}$ms and $R_{attack} = \{25, 50\}$Mbps, while Fig. 7(b) represents the results for the stochastic $\mathbb{A}^+$ PMDoS attacks. Each subfigure also includes the detection times for the $\mathbb{A}^-$ PMDoS attacks. Note that Vanguard can identify all PMDoS attacks with various attack costs within 3 detection windows (15s). Specifically, it identifies all the periodic $\mathbb{A}^+$ and the $\mathbb{A}^-$ PMDoS attacks immediately after a detection window. However, it requires slightly more time to identify the stochastic $\mathbb{A}^+$ PMDoS attacks with $\gamma = 0.1$, $T_{on} = 75$ms and $R_{attack} = 50$Mbps. It is because small attack costs correspond to a weak attack power, and the randomized attack period of the stochastic $\mathbb{A}^+$ PMDoS attack induces a longer attack periods, thus requiring more time to produce significant impact on the legitimate traffic. As a result, Vanguard also needs a longer period to identify the attacks.



(a) Periodic $\mathbb{A}^+$ PMDoS

(b) Stochastic $\mathbb{A}^+$ PMDoS

Fig. 7. Detection time for the $\mathbb{A}^+$ PMDoS attacks and the $\mathbb{A}^-$ PMDoS attacks under Vanguard.

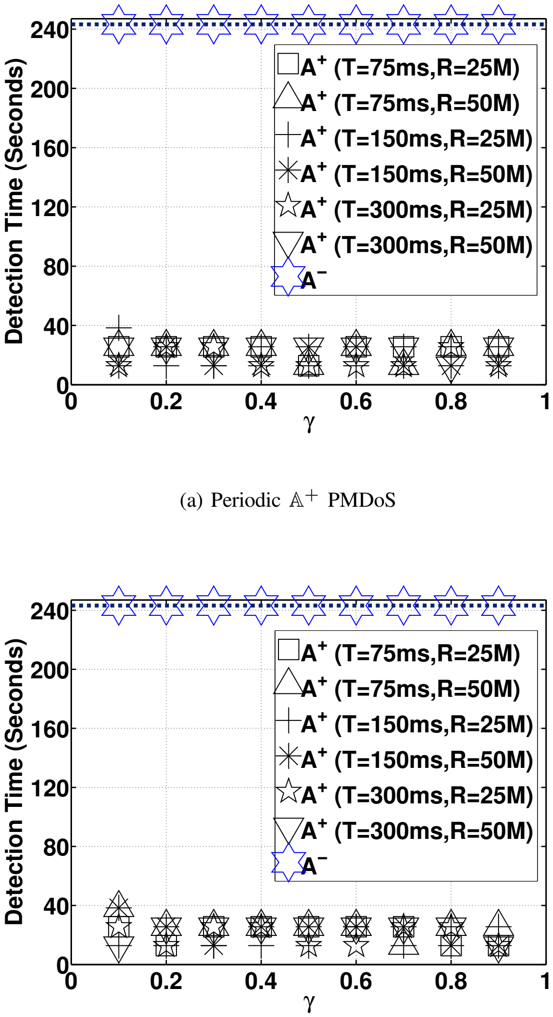
C. Comparison with other proposed detection schemes

In this sub-section, we further evaluate the performance of the DWTM-based, the DTWP-based, and the spectrum-based detection schemes using test-bed experiments, in order to compare their performance with Vanguard.

1) DWTM-based detection scheme: Fig. 8 shows the detection times required for the DWTM-based detection scheme to discover the $\mathbb{A}^+$ and $\mathbb{A}^-$ PMDoS attacks with various attack costs. We perform our experiments using the same network parameter settings. Each experiment lasts for 370s and a PMDoS attack begins at 130s. For the configuration of the DWTM-based detection system, we set each window of continuous samples last for 12.8 seconds to achieve a small detection delay, and $N_{W_D} = 6$ to obtain a training period of 76.8 seconds. Moreover, we follow similar configurations used in [21]. We set $P_{tolerance} = 1$; and the detection scheme employs the Haar wavelet [44] to capture the fluctuation in the incoming TCP data traffic, and the Daubechies wavelet with 4 vanishing moments ($DB(4)$) to extract the trend in the outgoing TCP ACK traffic. In each subfigure, any marker coinciding with the dashed line represents that the corresponding PMDoS attack is undetected by the detection scheme.

From the figures, it is clear that the DWTM-based detection scheme shows a poorer performance than Vanguard, with the average detection rate of 89.18%.

Specifically, while this mechanism can discover all the ongoing periodic and stochastic $\mathbb{A}^+$ PMDoS attacks within 3 detection windows (38.4s), it is incapable of discovering any $\mathbb{A}^-$ PMDoS attacks. Since the $\mathbb{A}^-$ PMDoS attack traffic constantly occupies a fixed portion of the bottleneck link capacity, the incoming TCP data traffic adapts to the remaining bandwidth without significant fluctuations. As a result, the attack traffic can elude the detection from the incoming traffic.



(a) Periodic $\mathbb{A}^+$ PMDoS

(b) Stochastic $\mathbb{A}^+$ PMDoS

Fig. 8. Detection time for the $\mathbb{A}^+$ PMDoS attacks and the $\mathbb{A}^-$ PMDoS attacks under the DWTM-based scheme.

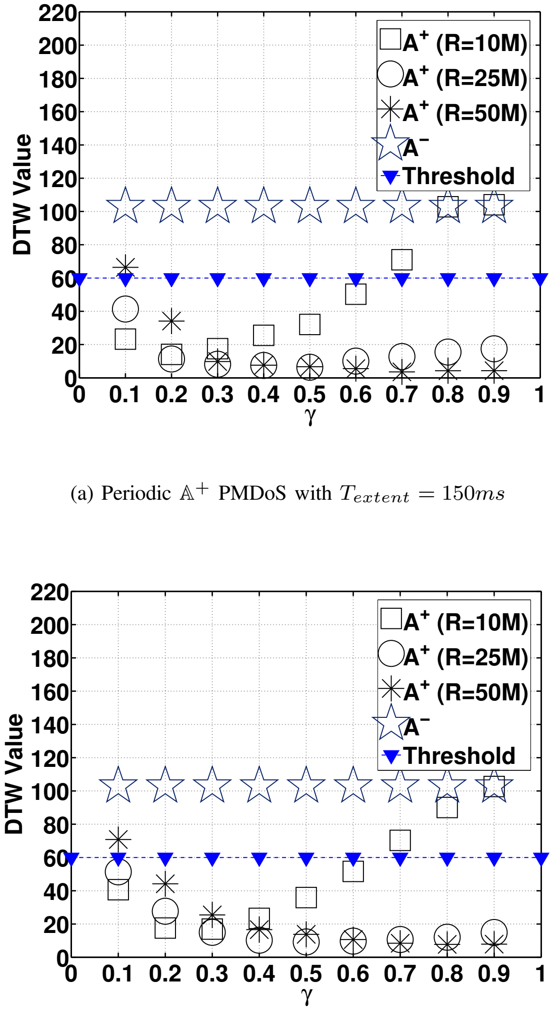
2) DTWP-based detection scheme: In Fig. 9, we report the experiment results of the DTWP-based detection scheme under the $\mathbb{A}^+$ and $\mathbb{A}^-$ PMDoS attacks with different attack costs. Each subfigure reports results for $R_{attack} = \{10, 25, 50\}$Mbps. The dashed line with downward-pointing triangles (-▼-) is the DTWP threshold (DTWP=60) recommended by [24] differentiating the normal traffic from the attack traffic. If the DTWP value is less than the threshold, the algorithm will confirm the presence of a PMDoS attack. We only present the experiment results for $T_{on} = 150$ms since they are similar to those with $T_{on} = \{75, 300\}$ms. From those figures, we can observe that the DTWP-based scheme can identify many periodic and stochastic $\mathbb{A}^+$ PMDoS attacks, but it cannot detect any $\mathbb{A}^-$ PMDoS attacks. It is due to the fact that this detection algorithm is designed specifically for the Shrew attack by matching the pattern of the incoming TCP data traffic with that of Shrew attack traffic. Thus, this scheme is not able to detect the $\mathbb{A}^-$ PMDoS attacks operated with CBR. Moreover, its average detection rate is only 76.9%, which is relatively less than that obtained by Vanguard and DWTM-based detection schemes.

3) Spectrum-based detection scheme: Fig. 10 illustrates the experiment results of spectrum-based detection scheme under the $\mathbb{A}^+$ and $\mathbb{A}^-$ PMDoS attacks. In each subfigure, the area between the solid line with downward-pointing triangles (-▼-) and the dashed line contains range of frequencies for single-source DoS attacks. On the other hand, the area between the solid line and the dashed line with upward-pointing triangles (-▲-) contains range of frequencies for multi-source DoS attacks. The experiment results show that the values of $(F(60\%))$ of the $\mathbb{A}^+$ PMDoS attacks do not concentrate on a small range. Instead, they spread from low frequencies to high frequencies. Since all the experiments are actually conducted in a single-source manner, this implies that the spectrum-based detection scheme cannot distinguish those single-source $\mathbb{A}^+$ PMDoS attacks from the multi-source ones.

V. Conclusions and Future Work

In this paper we have proposed Vanguard to detect a class of low-rate DoS attacks, which we refer them to as polymorphic DoS (PMDoS) attacks. Unlike the traditional flooding-based DoS (FDDoS) attack, the PMDoS attack can effectively degrade the performance of victim TCP flows with a much lower average attack rate. Moreover, the PMDoS attack may induce various traffic patterns according to different parameter settings, and consequently different impact on victim TCP flows.

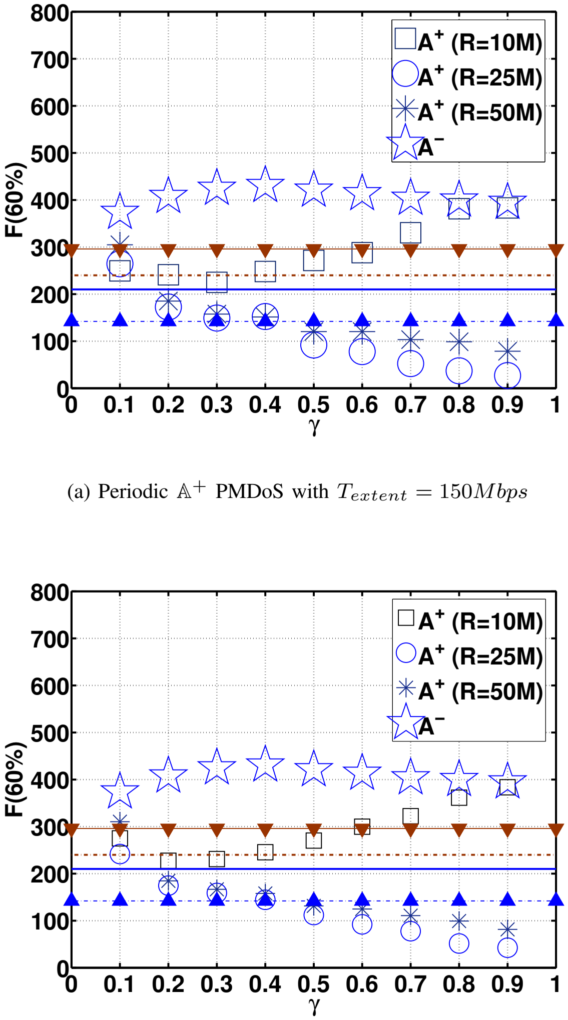
The detection engine in Vanguard is based on three indicators—a drastic decline in the outgoing ACK traffic, an unusually high variability in the ratio of the incoming TCP traffic and the outgoing TCP ACK traffic, and a significant change in the incoming TCP traffic distribution. We have implemented it as a Snort plug-in and experimented with it on a test-bed. The experiment



(a) Periodic $\mathbb{A}^+$ PMDoS with $T_{extent} = 150ms$

(b) Stochastic $\mathbb{A}^+$ PMDoS with $T_{extent} = 150ms$

Fig. 9. Detection time for the $\mathbb{A}^+$ PMDoS attacks and the $\mathbb{A}^-$ PMDoS attacks under the DTWP-based scheme.



(a) Periodic $\mathbb{A}^+$ PMDoS with $T_{extent} = 150Mbps$

(b) Stochastic $\mathbb{A}^+$ PMDoS with $T_{extent} = 150Mbps$

Fig. 10. Detection time for the $\mathbb{A}^+$ PMDoS attacks and the $\mathbb{A}^-$ PMDoS attacks under the spectrum-based scheme.

results show that Vanguard is very effective at detecting the PMDoS attack. Moreover, we compare Vanguard with other proposed detection systems. Vanguard incurs the lowest computational complexity, while achieving the highest detection rate. In the future work, we will investigate how to defend against the PMDoS attack through rate-limiting and other mechanisms.

Acknowledgment

The work described in this paper was partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 5080/02E), a grant from the Areas of Excellence Scheme established under the University Grants Committee of the Hong Kong Special Administrative Region, China (Project No. AoE/E-01/99), a grant from the Cisco University Research Program Fund at Community Foundation Silicon Valley, and a PolyU Research Grant (Project No. G-T848). We also thank Dr. Masayoshi Ejiri and the anonymous reviewers for their useful comments.

References

[1] D. Moore, G. Voelker, and S. Savage. Inferring internet denial of service activity. In *Proc. USENIX Security Symp.*, 2001.

[2] X. Yang and D. Wetherall and T. Anderson. A DoS-limiting network architecture. In *Proc. ACM SIGCOMM*, 2005.

[3] D. Dittrich. `http://staff.washington.edu/dittrich/misc/ddos/`.

[4] J. Mirkovic and P. Reiher. A taxonomy of DDoS attacks and defense mechanisms. *ACM SIGCOMM Computer Communications Review*, 34(2):39–54, Apr. 2004.

[5] The open source network intrusion detection system: Snort. www.snort.org.

[6] C. Estan, S. Savage, and G. Varghese. Automatically inferring patterns of resource consumption in network traffic. In *Proc. ACM SIGCOMM*, 2003.

[7] F. Hao, M. Kodialam and T. Lakshman. Real-time detection of hidden traffic patterns. In *Proc. IEEE ICNP*, 2004.

[8] P. Chhabra, A. John, and H. Saran. Pisa: Automatic extraction of traffic signatures. In *Proc. IFIP Networking*, 2005.

[9] R. 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. IEEE Workshop Information Assurance and Security*, June 2001.

[10] J. Baras, A. Cardenas, and V. Ramezani. On-line detection of distributed attacks from space-time network flow patterns. In *Proc. 23rd Army Science Conf.*, 2002.

[11] C. Jin, H. Wang, and K. Shin. Hop-count filtering: an effective defense against spoofed DDoS traffic. In *Proc. ACM Conf. Computer and Communications Security (CCS)*, 2003.

[12] Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, and J. Ucles. Hide: A hierarchical network intrusion detection system using statistical preprocessing and neural network classification. In *Proc. of IEEE Man Systems and Cybernetics Information Assurance Workshop*. IEEE, 2001.

[13] H. Wang, D. Zhang, and K. Shin. Detecting SYN flooding attacks. In *Proc. IEEE INFOCOM*, 2002.

[14] K. Wan and R. Chang. Engineering of a global defense infrastructure for DDoS attacks. In *Proc. of IEEE Intl. Conf. Networks*, pages 419–427, Aug. 2002.

[15] S. Mukkamala and A. Sung. Detecting denial of service attacks using support vector machines. In *Proc. of IEEE Intl. Conf. Fuzzy Systems*, volume 2, pages 1231 –1236, 2003.

[16] C. Manikopoulos and S. Papavassiliou. Network intrusion and fault detection: a statistical anomaly approach. *IEEE Communications Magazine*, 40(10):76 –82, Oct. 2002.

[17] P. Barford, J. Kline, D. Plonka, and A. Ron. A signal analysis of network traffic anomalies. In *Proc. ACM Internet Measurement Workshop*, 2002.

[18] A. Hussain and J. Heidemann and C. Papadopoulos. A framework for classifying denial of service attacks. In *Proc. ACM SIGCOMM*, 2003.

[19] A. Kuzmanovic and E. Knightly. Low-rate TCP-targeted denial of service attacks (the shrew vs. the mice and elephants). In *Proc. ACM SIGCOMM*, Aug. 2003.

[20] M. Guirguis, A. Bestavros, and I. Matta. Exploiting the transients of adaptation for RoQ attacks on Internet resources. In *Proc. IEEE ICNP*, 2004.

[21] X. Luo and R. Chang. On a new class of pulsing denial-of-service attacks and the defense. In *Proc. Network and Distributed System Security Symp. (NDSS)*, Feb. 2005.

[22] Q. Li, E. Chang, and M. Chan. On the effectiveness of DDoS attacks on statistical filtering. In *Proc. IEEE INFOCOM*, 2005.

[23] V. Paxson and M. Allman. Computing TCP's retransmission timer. RFC 2988, Nov. 2000.

[24] H. Sun, J. Lui, and D. Yau. Defending against low-rate TCP attack: Dynamic detection and protection. In *Proc. IEEE ICNP*, 2004.

[25] Y. Chen and K. Hwang and Y. Kwok. Collaborative defense against periodic Shrew DDoS attacks in frequency domain. Technical Report TR 2005-11, USC Internet and Grid Computing Lab, 2005.

[26] D. R. Cox. *Renewal Theory*. Chapman & Hall, 1962.

[27] X. Luo, R. Chang, and E. Chan. Performance analysis of TCP/AQM under denial-of-service attacks. In *Proc. of IEEE Intl. Symp. Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, 2005.

[28] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. The macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3), Jul. 1997.

[29] M. Hassan and R. Jain. *High performance TCP/IP Networking: Concepte, Issues, and solutions*. Pearson Education, Inc., 2004.

[30] Y. Yang and S. Lam. General AIMD congestion control. In *Proc. IEEE ICNP*, 2000.

[31] M. Allman, V. Paxson, and W. Stevens. TCP congestion control. RFC 2581, Apr. 1999.

[32] A. Brandt. The stochastic equation $Y_{n+1} = A_n Y_n + B_n$ with stationary coefficients. *Advances in Applied Probability*, 18(1), 1986.

[33] U. Horst. The stochastic equation $y_{t+1} = a_t y_t + b_t$ with nonstationary coefficients. *Journal of Applied Probability*, 38(1), 2001.

[34] E. Altman, K. Avrachenkov, and C. Barakat. A stochastic model of TCP/IP with stationary random losses. *IEEE/ACM Trans. Networking*, 13(2), 2005.

[35] K. Chandrayana and S. Kalyanaraman. Uncooperative congestion control. In *Proc. ACM SIGMETRICS*, 2004.

[36] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. In *Proc. IEEE ICNP*, 2002.

[37] C. Chatfield. *The Analysis of Time Series: An Introduction*. Chapam & Hall/CRC, sixth edition, 2003.

[38] M. Stricker and M. Orengo. Similarity of color images. In *Proc. SPIE Conf. Storage and Retrieval for Image and Video Databases III*, 1995.

[39] A. Smeulders, M Worring, S. Santini, A. Gupta and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2000.

[40] B. Brodsky and B. Darkhovsky. *Non-Parametric Statistical Diagnosis Problems and Methods*. Kluwer Academic Publishers, The Netherlands, 2000.

[41] M. Weeks and M. Bayoumi. Discrete wavelet transform: Architectures, design and performance issues. *Journal of VLSI Signal Processing*, 35, 2003.

[42] A. Oppenheim, A. Willsky, and S. Nawab. *Signals and Systems*. Prentice Hall, second edition edition, 1996.

[43] NLANR/DAST: Iperf 1.7.0 - The TCP/UDP bandwidth measurement tool. Available from `http://dast.nlanr.net/Projects/Iperf/`, 2003.

[44] I. Daubechies. *Ten lectures on wavelets*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.