Chapter 18

PROTECTING INTERNET SERVICES FROM LOW-RATE DOS ATTACKS

Yajuan Tang, Xiapu Luo and Rocky Chang

Abstract Feedback control is an important element in the engineering of stable Internet services. However, feedback channels are vulnerable to various Internet attacks. This paper shows analytically that the recently proposed low-rate denial-of-service (DoS) attacks can degrade Internet services by generating intermittent false feedback signals. The effectiveness of the attacks is evaluated using a control-theoretic approach for a general feedback control system and detailed analysis for a specific system. A nonparametric algorithm based on changes in traffic distribution is proposed for detecting attacks.

Keywords: Feedback control, low-rate DoS attacks, detection, countermeasures

1. Introduction

Feedback control is a fundamental building block for many dependable computing systems, network protocols and Internet services that are required to handle dynamic service demands. A classic example is modeling TCP congestion control dynamics with an active queue management (AQM) scheme at a router as a feedback control system. Web servers increasingly rely on feedback controllers to provide stable and scalable performance (see, e.g., [11, 17, 19]). Moreover, feedback control is a central element in emerging autonomic computing and communications systems (see, e.g., [4, 10]).

However, relatively little attention has been paid to the lack of security of feedback control mechanisms. This paper focuses on denial-of-service (DoS) attacks; in particular, low-rate DoS (LRDoS) attacks that send out intermittent pulses of malicious requests to victims. Examples of these attacks include reduction of quality (RoQ) attacks [6, 7] and pulsing denial-of-service (PDoS) attacks [14, 16]. These low-rate attacks are much more flexible than shrew attacks [9], which require a fixed time period between attack pulses. In the rest of this paper, we do not distinguish between RoQ and PDoS attacks; instead, we refer them to as low-rate DoS (LRDoS) attacks.

LRDoS attacks are particularly effective on feedback control systems. When a system encounters an attack pulse, it is temporarily overloaded. There are two consequences to this overloading: (i) new requests are refused during the attack, because resources are depleted by the malicious requests, and (ii) it takes some time for the system to recover to its normal state using a feedback controller. Many new requests are also turned down during this recovery period. Therefore, a sequence of properly-spaced attack pulses induces intermittent false feedback signals, which could force the server to persistently operate in a low-throughput region.

This paper has two parts. The first part examines the potential effects of LRDoS attacks on feedback-based Internet services. A control-theoretic approach is used to model these services and analyze the performance degradation of a web server under different attack scenarios. The second part of the paper presents a new nonparametric algorithm to detect LRDoS attacks based on changes in traffic distribution. Simulation results are used to analyze the attacks and evaluate the detection algorithm.

2. Related Work

Several researchers have studied LRDoS attacks. Guirguis, *et al.* [6, 7] originally proposed the RoQ attack, which exploits the transients of adaptation. They specifically considered the effects on a web server equipped with a feedback-based admission controller. Chan, *et al.* [1] proposed a related attack, which exploits the relative update scheme in computer systems to prevent normal users from joining the service. The procedure for generating the arrival time of the next update is essentially a feedback loop. Luo, *et al.* [14, 15] analyzed the effects of PDoS attacks on TCP throughput with different AQM schemes and proposed a two-stage detection algorithm.

Sun, *et al.* [18] presented an LRDoS attack detection scheme based on dynamic time warping (DTWP), but the scheme incurs high computation complexity. Chen and Hwang [2] devised a spectral template matching approach to identify shrew attacks. However, the template is generated from simulation and the test is based on parametric distributions, which may not be representative of real-world environments. Furthermore, the DTWP- and spectrum-based methods may not be able to handle aperiodic LRDoS attacks. Luo and Chang [14] developed a two-stage detection scheme. They also proposed the Vanguard scheme [13] to cover situations where the attack rate is less than or equal to a bandwidth bottleneck that cannot be handled [14]. However, both these schemes require bi-directional data to be effective.

3. Vulnerabilities to LRDoS Attacks

We model feedback-based Internet services as a typical feedback control loop shown in Figure 1. The two major components are the “process” and

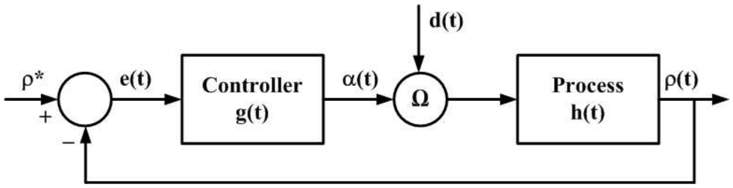


Figure 1. Feedback control loop.

the "controller." The process could represent any Internet service (e.g., email, web services, routing or streaming media) [8]. The output of the process $\rho(t)$ is any measurable "process output" (e.g., system utilization or queue length) that is fed back to the controller. $\alpha(t)$ is the "control signal" generated by the controller to move the process output to the desired value ρ^*. The controller is driven by a "control error" $e(t) = \rho^* - \rho(t)$.

On the other hand, the combination of normal requests for services and malicious requests from an LRDoS attack are modeled as a "disturbance input" $d(t)$. Moreover, we use $\lambda_n(t)$ to denote the arrival rate of normal requests. And we model an LRDoS attack as a sequence of Dirac signals: $\sum_{k=1}^{N} \lambda_a \delta(t - k\tau)$, where λ_a is the attack intensity of each pulse, τ is the time elapsed between two adjacent attack pulses, and N is the total number of pulses in the attack. Thus, the attack pulses are periodic with period τ. The input to the process is driven by both $\alpha(t)$ and $d(t)$ through an operator Ω. We consider additive [12] and multiplicative [7] operators Ω in this paper.

In this section, we first use a control-theoretic approach to analyze how an LRDoS attack can degrade the performance of Internet services. Note that the results obtained here apply to any controller and process. In the next section, we will analyze the effect on a web server consisting of a proportional controller, a constant service rate model and a multiplicative operator Ω.

Table 1 summarizes the symbols used in this paper. The upper rows list the parameters associated with the process, some of which are used for the web server in Section 4. The middle rows list the parameters associated with the controller, and K is used for the proportional controller in Section 4. The bottom rows list the parameters for the disturbance inputs.

Due to the lack of space, we will derive the results only for the additive operator Ω; the results for the multiplicative Ω can be derived in a similar manner. Let $d(t) = \lambda_n(t) + \sum_{k=1}^{N} \lambda_a \delta(t - k\tau) = d_n(t) + d_a(t)$; its Laplace transform is given by $D(s) = \mathcal{L}(\lambda_n(t)) + \lambda_a \sum_{k=1}^{N} e^{-k\tau s} = D_n(s) + D_a(s)$. Moreover, let $G(s)$ and $H(s)$ be the Laplace transforms of the transfer functions of the controller and the process, respectively. Therefore, the system output for the additive Ω in the s-plane is:

$$Y(s) = \frac{R(s)G(s) + D_n(s)}{1 + G(s)H(s)} H(s) + \frac{D_a(s)}{1 + G(s)H(s)} H(s), \tag{1}$$

where $Y(s)$ and $R(s)$ are the Laplace transforms of $\rho(t)$ and ρ^*, respectively.

Table 1. Notation.

Notation	Description
$\alpha(\cdot)$	Admission rate
$\rho(\cdot)$[1]	Process output
ρ^*[1]	Desired process output
$n(\cdot)$	Number of backlogged requests
μ	Servicing rate
A, B, C, D, ℓ	Constants for determining $\rho(\cdot)$
$e(\cdot)$	Control error
$\alpha(\cdot)$[2]	Control signal
$d(\cdot)$	Disturbance input
K	Controller parameter
$\lambda_n(\cdot)$	Arrival rate of normal requests
$\lambda(\cdot)$	Total arrival rate
λ_a	Attack intensity
τ	Attack period
N	Total number of attack pulses

[1] We use $\rho(\cdot)$ and ρ^* to refer to system utilization in Section 4.

[2] We use $\alpha(\cdot)$ to refer to admission rate in Section 4.

In an attack-free environment, the feedback control loop enables the process output $\rho(t)$ to converge to ρ^*. Consequently, the entire system could attain the best performance according to its design. However, Theorem 1 shows that an LRDoS attack impedes this convergence by introducing oscillations to the output $\rho(t)$. This is an undesirable phenomenon for Internet services, because the oscillations result in performance degradation and unstable services. Moreover, Corollary 2 shows that $e(t)$, which affects the control signal $\alpha(t)$, will also fluctuate periodically, and its amplitude is modulated by the attack intensity. Therefore, $e(t)$ cannot converge to zero as long as attack pulses are present. In other words, the attacker could inflict different scales of damage by tuning the attack intensity.

Theorem 1 *Under an LRDoS attack, the system output comprises a response caused by the normal requests and an additional oscillating component due to the attack:* $\rho(t) \sim \rho_n(t) + \lambda_a \sum_{k=1}^{N} f(t - k\tau)$.

Proof 1 *We prove the theorem for an additive* Ω. *By taking an inverse Laplace transform of Equation 1,* $\rho(t)$ *comprises an attack-free component and an attack-induced component:* $\rho(t) = \rho_n(t) + \rho_a(t)$. *Moreover,*

$$\rho_a(t) = \mathcal{L}^{-1}\left(\frac{\lambda_a \sum_{k=1}^{N} e^{-k\tau s}}{1 + G(s)H(s)} H(s)\right) = \lambda_a \sum_{k=1}^{N} f(t - k\tau). \tag{2}$$

It is not difficult to see that $f(t) = \mathcal{L}^{-1}(\frac{H(s)}{1+G(s)H(s)})$ is the system output excited by $\delta(t)$ when $\rho^* = 0$, because $\mathcal{L}(\delta(t)) = 1$. Moreover, being a stable system, $\rho_n(t)$ converges to a steady state. Therefore, the trajectory of $\rho(t)$ is a stable, periodic function.

Corollary 2 *Under an LRDoS attack, $e(t)$ oscillates in the time domain and its magnitude is proportional to the intensity of the pulse $e(t) \sim e_n(t) - \lambda_a \sum_{k=1}^{N} f(t - k\tau)$, where $e_n(t)$ is the error caused by the normal requests and $f(\cdot)$ is a function introduced by the attack.*

Proof 2 *We prove the corollary for an additive Ω. Since $E(s) = R(s) - Y(s)$, we have $E(s) = E_n(s) - E_a(s)$. From Equation 2, $e_a(t) = \lambda_a \sum_{k=1}^{N} f(t - k\tau)$. Moreover, being a stable system, $e_n(t) = \mathcal{L}^{-1}(E_n(s))$ vanishes as $t \to \infty$. Therefore, the attack introduces an oscillation to the error signal with an amplitude proportional to the attack intensity.*

4. LRDoS Attack on a Web Server

Having established the general results in the previous section, we examine a specific feedback-based Internet service – a web server. First, we describe the service model and analyze the service degradation caused by a single attack pulse. Next, we consider a sequence of attack pulses and analyze the service degradation for various attack periods.

The service model under consideration follows Figure 1 with the following components. First, the utilization ($\rho(t)$) is employed as the system output; the utilization is a piecewise linear function of $n(t)$ (see Equation 5). The controller is a PI controller with parameter K, which takes in $\rho^* - \rho(t)$ and generates an admission rate ($\alpha(t)$) as the control signal. Therefore, the rate of the admitted requests is given by $\lambda(t)\alpha(t)$; the unadmitted requests are dropped. As a result, the state vector of the service model comprises $\alpha(t)$, $\rho(t)$ and $n(t)$; their evolution and relationships are summarized below:

$$\dot{\alpha}(t) = K(\rho^* - \rho(t)),\ \alpha(t) \in [0, 1] \tag{3}$$

$$\dot{n}(t) = \lambda(t)\alpha(t) - \mu,\ n(t) \in [0, +\infty) \tag{4}$$

$$\rho(t) = \begin{cases} An(t) + B & \text{if } n(t) < \ell \\ Cn(t) + D & \text{if } n(t) \geq \ell \end{cases},\ \rho(t) \in [0, 1] \tag{5}$$

Note that this service model is the same as the one in [7], except that it uses a constant μ and a continuous-time model, both of which are essential for analytical tractability. In the rest of the paper, we assume a constant rate for normal requests, i.e., $\lambda_n(t) = \lambda_n$.

4.1 Analysis of a Single Attack Pulse

Suppose that an attack pulse arrives at $t = 0$ when the system is in the steady state and its state vector is $[\alpha_0, \rho_0, n_0]$. The system will evolve through three

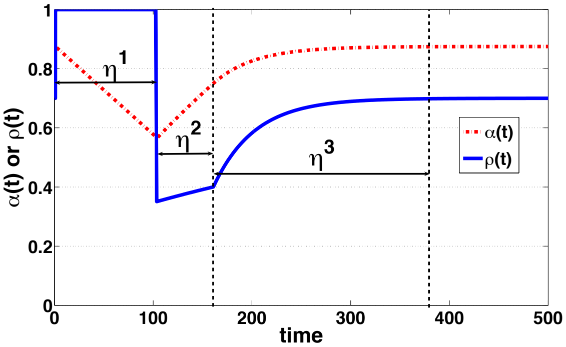


Figure 2. Effect of one attack pulse at $t = 0$ on $\alpha(t)$ and $\rho(t)$ ($A = 0.00267$, $B = 0.2$, $C = 0.024$, $D = -1.4$, $\ell = 75$, $K = 0.01$, $\mu = 90$, $\rho^* = 0.7$ (from [7])).

different stages before reaching the same state after the attack: saturation, recovery I and recovery II. The durations of these three stages are denoted by η_1, η_2 and η_3, respectively. The two recovery stages are due to the two piecewise linear relationships between $\rho(t)$ and $n(t)$.

Saturation Stage: As soon as the first attack pulse arrives, the system enters into a saturation stage in which we assume that the aggregated arrival rate results in a 100% utilization (i.e., $\rho(t) = 1$). The utilization stays at 100% throughout this period because $n(t) > \mu$. Since $\rho(t) = 1$, according to Equation 3, the admission rate decreases linearly as shown in Figure 2. Therefore, the state evolution during this stage is characterized by $\rho(t) = 1$, $\dot{\alpha}(t) = K(\rho^* - 1)$ and $\dot{n}(t) = \lambda_n \alpha(t) - \mu$.

This stage ends when all the pending requests have been processed, i.e., the total number of processed requests is equal to the total number of admitted requests. Therefore, we can obtain η_1 by solving $\lambda_a \alpha_0 + \int_0^{\eta_1} \lambda_n \alpha(t) dt = \eta_1 \mu$ with the initial conditions $[\alpha_0, \rho_0, n_0]$, where $\alpha_0 = \alpha(0^-) = \alpha(0^+)$, $n_0^+ = (\lambda_n + \lambda_a)\alpha_0 + n_0^-$ (this is due to the arrival of the attack pulse at $t = 0$), and $\rho_0 = \rho(0^+) = 1$:

$$\eta_1 = \frac{(\lambda_n \alpha_0 - \mu) + \sqrt{(\lambda_n \alpha_0 - \mu)^2 - 2\lambda_n K(\rho^* - 1)((\lambda_n + \lambda_a)\alpha_0 + n_0)}}{\lambda_n K(1 - \rho^*)}. \quad (6)$$

Recovery Stage I: At the beginning of this recovery stage, we have $n(\eta_1^-) = n(\eta_1^+) = \lambda_n \alpha(\eta_1^+)$, $\alpha(\eta_1^-) = \alpha(\eta_1^+) = \alpha_0 + K(\rho^* - 1)\eta_1$ and $\rho(\eta_1^+) = A\lambda_n \alpha(\eta_1^+) + B$. Since the utilization is now below the desired level, both the admission rate and utilization increase in this stage. Their evolutions are given by $\rho(t) = A\lambda_n \alpha(t) + B$, $\dot{\alpha}(t) = K(\rho^* - \rho(t))$ and $\dot{n}(t) = \lambda_n \alpha(t) - \mu$. Since this stage ends when $n(t) = \ell$, we can obtain η_2 by solving $\rho(\eta_2) = A\ell + B$ with the initial conditions $(\alpha(\eta_1^+), \rho(\eta_1^+), \lambda_n \alpha(\eta_1))$:

$$\eta_2 = \frac{1}{A\lambda_n K} \ln \frac{A\lambda_n \alpha(\eta_1) + B - \rho^*}{A\ell + B - \rho^*}. \quad (7)$$

Recovery Stage II: The two recovery stages differ only in their parameters and initial conditions. The initial conditions for the second recovery stage are: $\alpha(\eta_2^-) = \alpha(\eta_2^+) = \frac{\ell}{\lambda}$, $\rho(\eta_2^-) = \rho(\eta_2^+) = A\ell + B$, and $n(\eta_2^-) = \dot{n}(\eta_2^+) = \lambda_n \alpha(\eta_2^+) - \mu$. This stage ends when the utilization reaches the desired value. Therefore, we can obtain η_3 by solving $\rho(\eta_3) = \rho^*$ with the initial conditions $(\alpha(\eta_2^+), \rho(\eta_2^+), \lambda_n \alpha(\eta_2))$:

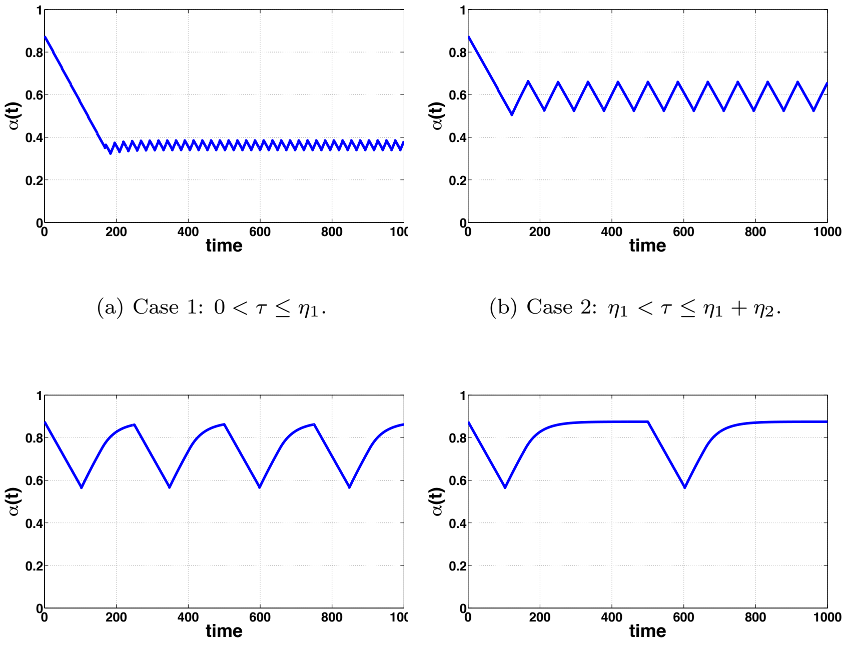
$$\eta_3 = \frac{1}{C\lambda_n K} \ln \frac{C\lambda_n \alpha(\eta_2) + D - \rho^*}{b\rho^* - \rho^*}, \textit{ where } b \approx 1 \textit{ and } \alpha(\eta_2) = \frac{\ell}{\lambda}. \quad (8)$$

4.2 Analysis of Multiple Attack Pulses

When there are multiple attack pulses, different degrees of damage are produced for different attack periods. This section considers four different choices of τ (only one choice was examined in [7]). Figure 3 illustrates how an attack launched at $t = 0$ degrades the admission rates for the four cases. In all four cases there is a relatively long saturation period at the beginning of the attack (this period is more noticeable for smaller values of τ). After that, the admission rates converge to oscillating patterns, similar to what we have discussed in Section 3. Moreover, the oscillating periods and the peaks of the admission rates increase with τ.

Case 1 ($0 < \tau \leq \eta_1$): As in the single pulse case, the admission rate drops linearly (i.e., $\dot{\alpha} = K(\rho^* - 1)$). During this declining period more attack pulses arrive at the victim. However, they do not cause further damage, because the admission rate is already very low. Again, as in the single pulse case, the system eventually serves all the pending requests and the recovery stage starts. However, another attack pulse arrives before the system can restore the admission rate to the normal pre-attack level. As a result, the admission rate drops linearly again. But this time, the system recovers much faster because the admission rate is already at a very low value when the attack pulse arrives. Consequently, the admission rate converges to an oscillation pattern with a small peak value.

Case 2 ($\eta_1 < \tau \leq \eta_1 + \eta_2$): As in Case 1, the admission rate drops linearly in the beginning and the additional attack pulses arriving during this period do not cause further damage. When the recovery stage first starts, the next attack pulse, say the k^{th} pulse, arrives when the admission rate has not yet been restored (i.e., $\alpha((k-1)\tau) < \alpha_0$). This $\alpha((k-1)\tau)$ induces a shorter η_1 for the next attack period because η_1 is an increasing function with respect to the initial admission rate (i.e., $\frac{\partial \eta_1}{\partial \alpha_0} > 0$). Consequently, the time to recover before the next pulse arrival, which is given by $k\tau - \eta_1$, is longer. Unfortunately, the admission rate still cannot climb back to α_0. To see why, suppose that at $t = k\tau^-$ the admission rate is large enough that $\alpha(k\tau^-) = \alpha((k-1)\tau^-)$. Therefore, when the next attack pulse arrives, the same number of attack requests is accepted, which forces the system to oscillate again. As a result, the admission rate converges to an oscillating pattern with a peak value less than α_0.



(a) Case 1: $0 < \tau \leq \eta_1$.

(b) Case 2: $\eta_1 < \tau \leq \eta_1 + \eta_2$.

(c) Case 3: $\eta_1 + \eta_2 < \tau \leq \eta_1 + \eta_2 + \eta_3$.

(d) Case 4: $\eta_1 + \eta_2 + \eta_3 < \tau$.

Figure 3. Effects of the attack period on the admission rate (four cases).

Case 3 ($\eta_1 + \eta_2 < \tau \leq \eta_1 + \eta_2 + \eta_3$): The evolution of the system state is similar to that in Case 2, except that there is an additional recovery part governed by parameters C and D. The details are, therefore, omitted.

Case 4 ($\tau > \eta_1 + \eta_2 + \eta_3$): In this case, the system state always returns to the steady state before the next pulse arrives. Since this case has already been analyzed in [7], we do not discuss it again.

This proves that the system state converges in all four cases. We have derived the maximal and minimal values of $\alpha(t)$ after convergence, denoted by α_{max} and α_{min}, respectively. However, due to a lack of space, we present the expressions for α_{max} and α_{min} without proof:

$$\alpha_{max} = \frac{e^{-A\lambda_n K\tau}}{A\lambda_n}(Ay + A\lambda_n\alpha_{\max} + B - \rho^*)e^{\frac{A}{\rho^*-1}y} + \frac{\rho^* - B}{A\lambda_n}. \tag{9}$$

$$\alpha_{min} = K(\rho^* - 1)\eta_1 + \alpha_{max}, \tag{10}$$

where $y = -(\lambda_n\alpha_{max} - \mu) - \sqrt{(\lambda_n\alpha_{max} - \mu)^2 - 2\lambda_n K(\rho^* - 1)(\lambda_n + \lambda_a)\alpha_{max}}$. Note that $\alpha_{max} - \alpha_{min}$ measures the magnitude of the oscillations, and as shown in Figure 3, the magnitude increases with τ.

5. Detecting LRDoS Attacks

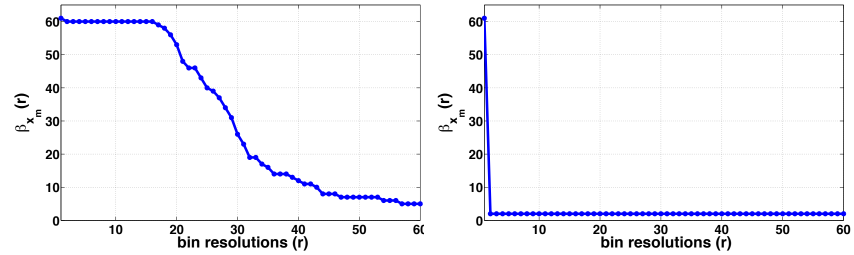
This section describes a new anomaly-based detection scheme for LRDoS attacks. The approach is based on the fact that high-intensity request bursts in an attack disturb the distribution of the arrival rates of normal requests. The detection scheme, therefore, has two components: (i) modeling the distribution using a histogram, and (ii) using a nonparametric outlier detection algorithm to determine whether there is a change in the histogram (assumed to be due to an LRDoS attack). A similar outlier detection approach has been proposed in [5]. However, it differs from our histogram approach in that it uses a clustering technique to group samples under various resolutions.

Suppose that x_i, $i \in \mathbb{Z}$, are request rate samples. For every window of W samples, say $\{x_{m-W+1}, \ldots, x_m\}$, our detection algorithm determines whether the latest sample x_m has disturbed the distribution. This is accomplished by using a histogram to model the distribution. A histogram consists of a set of equally-spaced intervals of sample values each of which is called a "bin;" the total number of bins is called the "bin resolution." Given a set of samples, the histogram plots the number of samples falling into each bin ("bin size"). An attack is detected when changes are seen in the histogram for $\{x_{m-W+1}, \ldots, x_m\}$ compared with the histogram for $\{x_{m-W+1}, \ldots, x_{m-1}\}$.

Determining the proper bin resolution is the main drawback of the histogram approach. The issue is resolved by applying the detection strategy to a range of bin resolutions. We first let $\beta_{x_m}(r)$ be the size of the bin that contains the sample x_m when the bin resolution is r. Figure 4 plots the values of $\beta_{x_m}(r)$ for $r \in [1, 60]$ obtained from simulation experiments for normal requests and malicious requests. Note that the $\beta_{x_m}(r)$ values for normal requests decrease more gradually from 60 (when $r = 1$) to 1 (when $r = 60$). On the other hand, the $\beta_{x_m}(r)$ values for malicious requests drop drastically from $r = 1$ to $r = 2$, because x_m, a sample from the attack pulse, is an outlier compared with the normal request samples. Therefore, the attack can be detected by measuring the changes in $\beta_{x_m}(r)$ values as r increases. For this purpose, we define a cumulative ratio for x_m:

$$R(x_m) = \sum_{r=1}^{W-1} \frac{\beta_{x_m}(r)}{\beta_{x_m}(r+1)}. \tag{11}$$

To see how the statistic in Equation 11 is used to detect an attack, let $R_n(x_m)$ (or $R_a(x_m)$) be the value of $R(x_m)$ when x_m is a sample from normal (or attack) traffic. If the attack intensity is high enough, the sample x_m from the attack traffic will most likely be the first sample that is separated from other samples when r is increased beyond one. In the most extreme case, $\beta_{x_m}(1) = W$ and $\beta_{x_m}(r) = 1$, $r > 1$; therefore, $R_a(x_m) = 2W - 2$. On the other hand, suppose that x_m is from normal traffic. Then, if the normal traffic intensity is uniformly distributed, $\beta_{x_m}(r) = \frac{1}{r}$. Therefore, $R_n(x_m) = \sum_{r=1}^{W-1} \frac{r+1}{r} < \sum_{r=1}^{W-1} \frac{r+r}{r} = 2(W-1) = R_a(x_m)$.



(a) x_m is a sample from normal requests. (b) x_m is a sample from malicious requests.

Figure 4. Change in x_m's bin size when the bin resolution is increased from 1 to 60.

However, to deal with the fact that the normal traffic distribution is usually heavy tailed, we introduce appropriate weights to the ratios in Equation 11 by assigning a higher weight to a $\beta_{x_m}(r)$ that has a higher traffic intensity:

$$R(x_m) = \sum_{r=1}^{W-1} \left(\frac{\beta_{x_m}(r)}{\beta_{x_m}(r+1)} \left| \frac{\bar{x}_n(r+1) - \bar{x}}{\bar{x}} \right| \right), \tag{12}$$

where $\bar{x}_n(r)$ is the mean of the samples in x_m's bin and $\bar{x}$ is the mean of all W samples. Moreover, if x_m is from normal traffic, $\bar{x}_n(r)$ should not be too far away from $\bar{x}$. If x_m is from attack traffic, $\bar{x}_n(r)$ is much closer to x_m because of the intensity of the attack traffic.

Therefore, the $R(x_m)$ values for a normal traffic sample and an attack traffic sample are revised as follows:

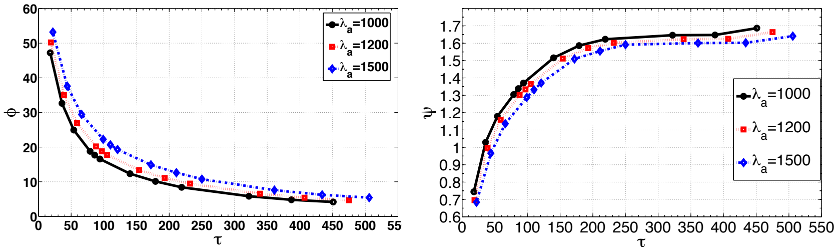
$$R_a(x_m) = (2W-2) \left| \tfrac{\bar{x}_n - \bar{x}}{\bar{x}} \right| \approx (2W-2) \left| \tfrac{\lambda_a - \bar{x}}{\bar{x}} \right|. \tag{13}$$

$$R_n(x_m) = \textstyle\sum_{r=1}^{W-1} \left(\tfrac{r+1}{r} \left| \tfrac{\bar{x}_n(r) - \bar{x}}{\bar{x}} \right| \right) < (2W-2) \left| \tfrac{\bar{x}_n - \bar{x}}{\bar{x}} \right| = R_a(x_m). \tag{14}$$

The final step is to choose a threshold θ such that the detection outcome is positive if $R(x_m) \geq \theta$. The threshold θ is determined as follows. Denote σ_x as the standard deviation of the first $W-1$ samples in the detection window: $\sigma_x = \sqrt{\frac{1}{W-1} \sum_{i=m-W+1}^{m-1} (x_i - \bar{x})^2}$. Note that if x_m is from attack traffic, then $\bar{x}_m - \bar{x} > \sigma_x$; if x_m is from normal traffic, $\bar{x}_m - \bar{x} \approx \sigma_x$. Also, we have $\sum_{r=2}^{W-1} \frac{1}{r} < W-1$. Taking these into consideration, we set $\theta = (2W-2)\sigma_x/\bar{x}$. To handle the burstiness in normal traffic, we introduce a weight w_d to the threshold value: $\theta = (2W-2)w_d\sigma_x/\bar{x}$.

6. Simulation Results

This section evaluates the performance of the detection algorithm for different values of w_d. MATLAB simulation results are used to assess attack capabilities as well as detection performance. The simulation experiments use



(a) Attack capability in terms of ϕ. (b) Attack capability in terms of ψ.

Figure 5. Capability of LRDoS attacks for different values of τ and λ_a.

the same parameters as before: $A = 0.00267$, $B = 0.2$, $C = 0.024$, $D = -1.4$, $\ell = 75$, $K = 0.01$, $\mu = 90$, and $\rho^* = 0.7$.

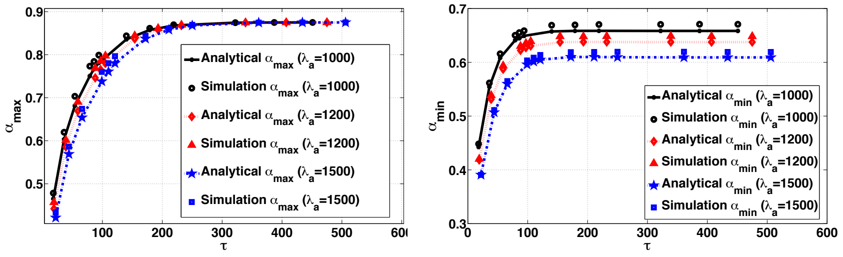
Attack Capability: In Section 4, we analyzed the effects of τ on the system output and admission rate of a victim system. Here we use simulations to further quantify the effects. We also study the effects for different values of λ_a. First, we define two metrics that characterize the capability of LRDoS attacks: (i) the percent of normal requests dropped due to an attack (denoted by ϕ), and (ii) the number of normal requests dropped due to an attack per λ_a (denoted by ψ). Therefore,

$$\phi = \frac{\int_0^T (\alpha^c - \alpha(t))\lambda_n dt}{\int_0^T \alpha^c \lambda_n dt} \times 100 \quad and \quad \psi = \frac{\int_0^T (\alpha^c - \alpha(t))\lambda_n dt}{N\lambda_a},$$

where T is the observation period (all N attack pulses arrive during T), and α^c is the admission rate when the system is in the steady state and not under attack. Therefore, ϕ measures the absolute service degradation. On the other hand, ψ measures the attack effectiveness in terms of the amount of service degradation caused by one attack request.

Figure 5 presents the ϕ and ψ values for attacks with $\lambda_a = 1000, 1200$ and 1500 requests per second, and $\tau \in [20, 500]$ seconds. For the three λ_a values, it is easy to verify that the range of τ covers the four cases discussed in Section 4. Figures 5(a) and 5(b) show that ϕ increases with λ_a while ψ decreases with λ_a. Furthermore, for a given value of λ_a, ϕ decreases with τ while ψ increases with τ. Moreover, as $\tau \to \infty$, $\phi \to 0$, because this is similar to the case of one attack pulse being encountered over a very long observation period. On the other hand, ψ converges to a value below 2, which is the maximal service degradation in terms of ψ.

Figure 6 presents the values of α_{max} and α_{min} obtained from simulations; it also includes the analytical results from Equations 9 and 10 for comparison. Note that the simulation results closely match the analytical results. Also, both α_{max} and α_{min} are increasing functions of τ, which can be validated by



(a) α_{max} for different values of τ and λ_a. (b) α_{min} for different values of τ and λ_a.

Figure 6. Analytical and simulation results for α_{max} and α_{min}.

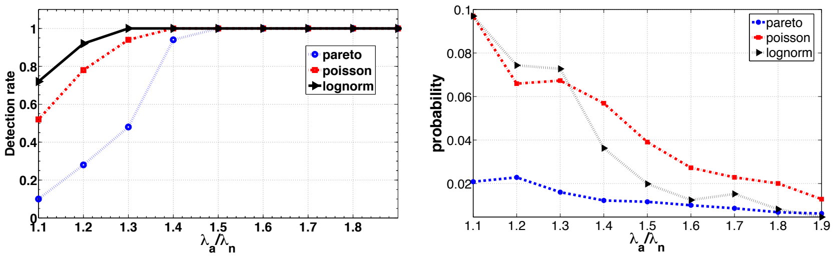
computing the derivatives of Equations 9 and 10. Furthermore, α_{max} and α_{min} eventually reach plateaus, which indicates that the range of τ corresponds to Case 4 in Section 4.2. Finally, the α_{max} values for the three λ_a values converge to the same value, but the α_{min} values do not—a higher λ_a gives a lower α_{min}. In other words, a higher attack intensity increases the magnitude of the oscillations.

Detection Performance: To evaluate the performance of our detection algorithm, normal traffic was generated using log-normal, Pareto and Poisson distributions. For the log-normal and Pareto distributions, the location parameter was set to 4.6027 and 91.6667, respectively, and the scale parameter to 0.0707 and 12, respectively. For the Poisson distribution, the rate was set to 100. These parameter values yielded mean arrival rates of 100 requests per second for all three distributions. Also, samples for the request arrival rates were computed every second.

Figures 7(a) and 7(b) show the simulation results for the detection rates and false alarm rates, respectively. The detection rate increases with $\frac{\lambda_a}{\lambda_n}$ for all three distributions. Also, all the attacks can be detected (i.e., detection rate is 1) when $\frac{\lambda_a}{\lambda_n}$ is around 1.5. Moreover, the detection algorithm achieves the highest detection rate for the log-normal distribution before the detection rate reaches 1.0. Note that the variance of the log-normal distribution is approximately 50; this shows that the detection algorithm works well even under bursty normal traffic. The false alarm rate also improves (decreases) with $\frac{\lambda_a}{\lambda_n}$. The false alarm rates for the log-normal and Pareto distributions exhibit similar decreasing trends, whereas the rate for the Poisson distribution is at a low (albeit decreasing) level.

7. Conclusions

Low-rate DoS (LRDoS) attacks can significantly degrade feedback-based Internet services. By sending intermittent attack pulses, these attacks induce victim systems to generate false feedback signals, which cause them to decrease



(a) Detection rate.

(b) False alarm rate.

Figure 7. Performance of the new detection algorithm for values of $\frac{\lambda_a}{\lambda_n}$.

their request acceptance rates. The nonparametric algorithm presented in this paper analyzes changes in traffic distribution to detect attacks. Extensive simulation results demonstrate that the algorithm is very effective at detecting LRDoS attacks.

Our current research is investigating strategies for optimizing LRDoS attacks and improving detection capabilities. We are also experimenting with several TCP variants, especially those targeting Linux systems (e.g., Veno, Hybla, Westwood+).

Acknowledgements

This work was partially supported under Project No. PolyU 5080/02E from the Research Grant Council of the Hong Kong Special Administrative Region, China, and by a grant from the Cisco University Research Program Fund.

References

[1] M. Chan, E. Chang, L. Lu and S. Ngiam, Effect of malicious synchronization, in *Applied Cryptography and Network Security (LNCS 3989)*, J. Zhou, M. Yung and F. Bao (Eds.), Springer, Berlin-Heidelberg, Germany, pp. 114–129, 2006.

[2] Y. Chen and K. Hwang, Collaborative detection and filtering of shrew DDoS attacks using spectral analysis, *Journal of Parallel and Distributed Computing*, vol. 66(9), pp. 1137–1151, 2006.

[3] Y. Diao, J. Hellerstein, S. Parekh, R. Griffith, G. Kaiser and D. Phung, Self-managing systems: A control theory foundation, *Proceedings of the Twelfth IEEE International Conference and Workshops on the Engineering of Computer-Based Systems*, pp. 441-448, 2005.

[4] Y. Diao, S. Parekh, R. Griffith, G. Kaiser, D. Phung and J. Hellerstein, A control theory foundation for self-managing computing systems, *IEEE Journal on Selected Areas of Communications*, vol. 23(12), pp. 2213–2222, 2005.

[5] H. Fan, O. Zaïane, A. Foss and J. Wu, A nonparametric outlier detection for effectively discovering top n outliers from engineering data, in *Advances in Knowledge Discovery and Data Mining (LNCS 3918)*, W. Ng, M. Kitsuregawa, J. Li and K. Chang (Eds.), Springer, Berlin-Heidelberg, Germany, pp. 557–566, 2006.

[6] M. Guirguis, A. Bestavros and I. Matta, Exploiting the transients of adaptation for RoQ attacks on Internet resources, *Proceedings of the Twelfth IEEE International Conference on Network Protocols*, pp. 184–195, 2004.

[7] M. Guirguis, A. Bestavros, I. Matta and Y. Zhang, Reduction of quality (RoQ) attacks on Internet end-systems, *Proceedings of the Twenty-Fourth Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 1362–1372, 2005.

[8] J. Hellerstein, Y. Diao, S. Parekh and D. Tilbury, *Feedback Control of Computing Systems*, John Wiley, New York, 2004.

[9] A. Kuzmanovic and E. Knightly, Low-rate TCP-targeted denial-of-service attacks: The shrew vs. the mice and elephants, *Proceedings of the Conference on Applications, Technologies, Architectures and Protocols for Computer Communications*, pp. 75–86, 2003.

[10] R. Lotlika, R. Vatsavai, M. Mohania and S. Chakravarthy, Policy schedule advisor for performance management, *Proceedings of the Second International Conference on Autonomic Computing*, pp. 183–192, 2005.

[11] Y. Lu, T. Abdelzaher, C. Lu, L. Sha and X. Liu, Feedback control with queueing-theoretic prediction for relative delay guarantees in web servers, *Proceedings of the Ninth IEEE Real-Time and Embedded Technology and Applications Symposium*, pp. 208–217, 2003.

[12] C. Lu, J. Stankovic, G. Tao and S. Son, Feedback control real-time scheduling: Framework, modeling and algorithms, *Journal of Real-Time Systems*, vol. 23(1-2), pp. 85–126, 2002.

[13] X. Luo, E. Chan and R. Chang, Vanguard: A new detection scheme for a class of TCP-targeted denial-of-service attacks, *Proceedings of the Tenth IEEE/IFIP Network Operations and Management Symposium*, pp. 507–518, 2006.

[14] X. Luo and R. Chang, On a new class of pulsing denial-of-service attacks and the defense, *Proceedings of the Twelfth Annual Network and Distributed System Security Symposium*, 2005.

[15] X. Luo and R. Chang, Optimizing the pulsing denial-of-service attacks, *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 582–591, 2005.

[16] X. Luo, R. Chang and E. Chan, Performance analysis of TCP/AQM under denial-of-service attacks, *Proceedings of the Thirteenth IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pp. 97–104, 2005.

[17] A. Robertsson, B. Wittenmark, M. Kihl and M. Andersson, Design and evaluation of load control in web-server systems, *Proceedings of the American Control Conference*, vol. 3(30), pp. 1980–1985, 2004.

[18] H. Sun, J. Lui and D. Yau, Defending against low-rate TCP attacks: Dynamic detection and protection, *Proceedings of the Twelfth IEEE International Conference on Network Protocols*, pp. 196–205, 2004.

[19] M. Welsh and D. Culler, Adaptive overload control for busy Internet servers, *Proceedings of the Fourth USENIX Symposium on Internet Technologies and Systems*, p. 4, 2003.

[20] R. Zhang, C. Lu, T. Abdelzaher and J. Stankovic, Controlware: A middleware architecture for feedback control of software performance, *Proceedings of the Twenty-Second International Conference on Distributed Computing Systems*, p. 301, 2002.