Chapter 1

DEGRADING INTERNET SERVICES BY INTERMITTENT FALSE FEEDBACKS AND THE COUNTERMEASURES

Yajuan Tang, Xiapu Luo and Rocky Chang

Abstract Feedback control is an important element in the engineering of many stable Internet services. However, the feedback channels are vulnerable to various Internet attacks. In this paper, we show analytically that the recently proposed low-rate denial-of-service (DoS) attacks can effectively degrade these Internet services by generating intermittent false feedback signals. To evaluate the attacks' effectiveness, we employ both control-theoretic approach for a general feedback control system and a detailed analysis for a specific system. On the countermeasures, we propose a nonparametric algorithm to detect an attack based on the changes in the traffic distribution.

Keywords: feedback control, low-rate DoS attack, attack detection and defense

1. Introduction

Feedback control is a fundamental building block for many dependable computing systems, network protocols, and Internet services which are required to handle dynamic service demands. As a classic Internet example, the TCP congestion control dynamics with an active queue management (AQM) scheme at a router can be modeled as a feedback control system. Web servers increasingly rely on feedback controllers to provide stable and scalable performance (e.g., [1, 3, 4]). Moreover, feedback control is a central element in the emerging autonomic computing and communications systems (e.g., [6, 7]).

However, there is much less attention paid to the insecurity of the feedback control mechanisms. In this paper, we consider the threat of denial-of-service (DoS) attacks; in particular, we concentrate on a class of low-rate DoS attacks that send out intermittent pulses of malicious requests to a victim. Examples of such attacks include *reduction of quality* (RoQ) attack [9, 10] and *pulsing denial of service* (PDoS) attacks [11, 12]. These low-rate attacks are much more flexible than the Shrew attacks [13] that require a fixed period between attack pulses. In the rest of this paper, we do not further distinguish between the RoQ and PDoS attacks; instead, we refer them to as low-rate DoS (LRDoS) attacks.

The LRDoS attacks are particularly effective in attacking feedback control systems. When the system encounters an attack pulse, it will be temporarily overloaded. There are two consequences to this overloading: (1) new requests will be refused during the attack, because the resources are depleted by the malicious requests, and (2) it will take some time for the system to recover to the normal state using the feedback controller. That is, during the recovery period, many new requests will also be turned down. Therefore, a sequence of properly spaced attack pulses will induce intermittent false feedback signals which could force the server to persistently operate in a low-throughput region.

Contributions and roadmap of this paper There are two parts to this paper. In the first part, we seek to understand the potential effects of the LRDoS attacks on feedback-based Internet services. We start by analyzing these services using a control-theoretic approach (in §1.3). After that, we analyze performance degradation of Web server under different attack scenarios (in §1.4). In the second part, we propose a new nonparametric algorithm to detect the LRDoS attacks based on changes in the traffic distribution (in §1.5). Furthermore, we will present simulation results to evaluate the attack and the detection algorithm (§1.6).

2. Related works

Related attacks Guirguis et al. propose the RoQ attack that exploits the transients of adaptation [9, 10]. They have specifically considered the effect on a Web server equipped with a feedback-based admission controller. Chan et al. [14] propose an attack that exploits the relative update scheme in computer systems to prevent normal users from joining the service. The procedure of generating the arrival time of the next update is essentially a feedback loop. Luo et al. [11, 15] have analyzed the effects of the PDoS attack on TCP throughput with different AQM schemes and have proposed a two-stage detection algorithm.

Detection of LRDoS attacks Sun et al. [16] present a dynamic time warping (DTWP)-based detection scheme which however incurs a high computation complexity. Chen and Hwang [17] present a spectral template-matching approach to identify a Shrew attack. However, the template is generated from simulation, and the test is based on parametric distributions which may not be representative in a real environment. Furthermore, both DTWP and spectrum-based methods may not be able to handle aperiodic LRDoS attacks. Luo and Chang [11] propose a two-stage detection scheme. They also propose Vanguard [18] to cover the situation where the attack rate is less than or equal to bandwidth bottleneck which cannot be handled in [11]. However, both schemes require bi-directional data.

3. General vulnerabilities under LRDoS attacks

In this paper, we model the feedback-based Internet services as a typical feedback control loop shown in Figure 1. The two major components are *process* and *controller*. The process could represent any Internet services, such as Web, email, routing, and media streaming [2]. The output of the process $\rho(t)$ could be any measurable *process output* (e.g., system utilization, queue length) that is fed back to the controller. $\alpha(t)$ is the *control signal* generated by the controller in order to regulate the process output according to a desired value ρ^*. Therefore, the controller is driven by a *control error*: $e(t) = \rho^* - \rho(t)$, and its output is $\alpha(t)$.

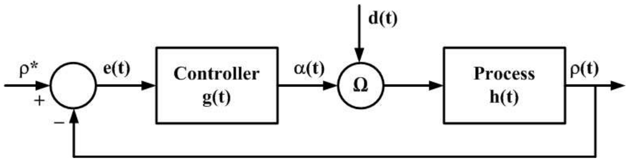


Figure 1. The feedback-based Internet services are modeled as a typical feedback control loop.

On the other hand, we model the normal requests for the services and the malicious requests from an LRDoS attack combined as a *disturbance input* $d(t)$. Moreover, we use $\lambda_n(t)$ to denote the arrival rate of the 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. That is, the attack pulses are periodic with a period of τ. The input to the process is therefore driven by both $\alpha(t)$ and $d(t)$ through an operator Ω. We consider both additive Ω [19] and multiplicative Ω [10] in this paper.

In this section, we first use a control-theoretic approach to analyze how an LRDoS attack can effectively degrade the performance of such 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 Ω.

Table 1 summarizes the major symbols used in this paper. The upper group of rows lists the parameters associated with the process, some of which will be used for the Web server in §1.4. The middle group lists the parameters associated with the controller, and K will be used for the proportional controller in §1.4. The bottom group lists the parameters for the disturbance inputs.

Table 1. The main notations used in this paper.

Notations	Description
$\alpha(\cdot)$	admission rate
$\rho(\cdot)$ (see Note 1)	process output
ρ^* (see Note 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
$\alpha(\cdot)$ (see Note 2)	control signal
$d(\cdot)$	disturbance input
K	controller's parameter
$\lambda_n(\cdot)$	arrival rate of normal requests
$\lambda(\cdot)$	the total arrival rate
λ_a	attack intensity
τ	attack period
N	total number of attack pulses

Note 1: We will use $\rho(\cdot)$ and ρ^* to refer to system utilization in §1.4.

Note 2: We will use $\alpha(\cdot)$ to refer to admission rate in §1.4.

In the lack of space, we will derive the results only for the additive Ω; the results for the multiplicative Ω can be derived similarly. Recall that $d(t) = \lambda_n(t) + \sum_{k=1}^{N} \lambda_a \delta(t - k\tau) = d_n(t) + d_a(t)$; 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 controller's and the process's transfer function, 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.

Under an attack-free environment, the feedback control loop enables the process's output $\rho(t)$ to converge to ρ^*; consequently, the whole system could attain the best performance according to its design. However, Theorem 1 shows that an LRDoS attack will impede this convergence by introducing oscillations to the output $\rho(t)$. This is an undesirable phenomenon for Internet services, because these oscillations will 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 the 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 Eq. (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;* $\rho(t)$*'s trajectory is therefore a stable, periodic function.*

COROLLARY 2 *Under an LRDoS attack,* $e(t)$ *oscillates in the time domain and its magnitude is proportional to the pulse intensity:* $e(t) \sim e_n(t) - \lambda_a \sum_{k=1}^{N} f(t - k\tau)$, *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 Eq. (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* $t \rightarrow \infty$. *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 last section, we analyze in this section a specific feedback-based Internet service: Web server.

We will next describe the service model and then analyze the service degradation caused by a single attack pulse. After that, we consider a sequence of attack pulses and analyze the service degradation for four cases of attack periods.

The service model under consideration follows Figure 1 with the following components. First, the system uses utilization ($\rho(t)$) as the system output; the utilization is a piecewise linear function of $n(t)$ (see Eq. (5)). The controller is a PI controller with parameter K that 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 will be dropped. As a result, this service model's state vector consists of $\alpha(t)$, $\rho(t)$, and $n(t)$; their evolutions 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 used in [10] except that we use a constant μ and a continuous-time model, both of them are essential for analytical tractability. Moreover, in the rest of the paper, we assume a constant rate for the normal requests; that is, $\lambda_n(t) = \lambda_n$.

4.1 An analysis for 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 different stages before reaching the same state before the attack: saturation, recovery I, and recovery II. The durations for these three stages are denoted as 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 the saturation stage in which we assume that the aggregated arrival rate will result in a 100% utilization (i.e., $\rho(t) = 1$). The utilization stays at 100% throughout this period, because $\alpha(t)\lambda(t) > \mu$. Since $\rho(t) = 1$, according to Eq. (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 $\dot{n}(t) = \lambda_n\alpha(t) - \mu$.

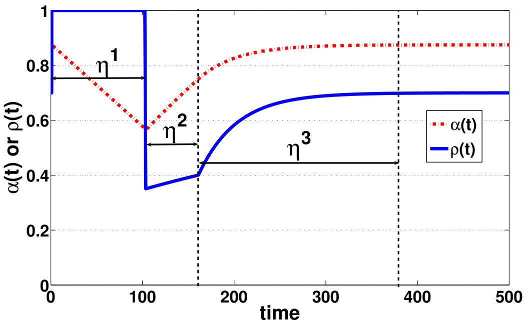


Figure 2. The effect of one attack pulse at $t = 0$ on $\alpha(t)$ and $\rho(t)$ of the system. The system parameters are: with $A = 0.00267$, $B = 0.2$, $C = 0.024$, $D = -1.4$, $\ell = 75$, $K = 0.01$, $\mu = 90$, $\rho^* = 0.7$ (we adopt this set of parameters from [10]).

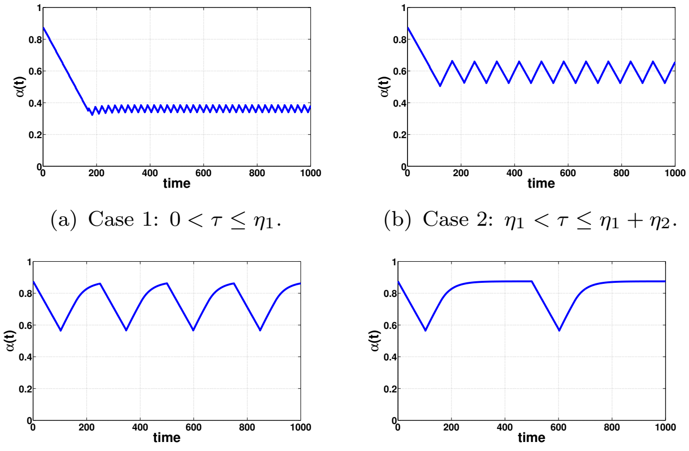
This stage will end when all the pending requests have been processed; that is, the rate of admitted requests is equal to the service rate. Therefore, we can obtain η_1 by solving $\lambda_a \alpha_0 + \int_0^{\eta_1} \lambda_n \alpha(t) dt = \eta_1 \mu$ with the initial conditions $[\alpha_0, \rho_0, n_0]$, 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 $\rho_0 = \rho(0^+) = 1$:

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

Recovery stage I: At the beginning of this 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 will 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 will end when $n(t) = \ell$, we can obtain η_2 by solving $\rho(\eta_2) = A\ell + B$ with the initial conditions $(\alpha(\eta_1^+), \rho(\eta_1^+), \lambda_n \alpha(\eta_1))$:

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

Recovery stage II: The only differences between recovery stages I and II are the parameters and initial conditions. The initial conditions here 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 last stage will end when the utilization reaches the desired value. Therefore, we can obtain η_3 by solving $\rho(\eta_3) =$



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

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

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

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

Figure 3. Effects of the attack period on the admission rate: Four possible cases.

ρ^* 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 An analysis for multiple attack pulses

When there are multiple attack pulses, the attack period inflicts different degrees of damage on the victim. We will consider four different choices of τ in this section ([10] considered only the last case). First of all, Figure 3 illustrates how an attack launched at $t = 0$ degrades the admission rates for the four cases. All four cases show that there is a relatively long saturation period at the beginning of the attack (this period is more noticeable for a small τ). After that, their admission rates all converge to oscillating patterns, similar to what we have discussed in section 1.3. Moreover, the oscillating periods increase with τ; the peaks of the admission rates also increase with τ.

Case 1. $0 < \tau \le \eta_1$**:** Same as the single-pulse case, the admission rate will drop linearly (i.e., $\dot{\alpha} = K(\rho^* - 1)$). During this declining period, there are more attack pulses arriving at the victim. However, they do not cause further damage, because the admission rate is already very low. Similar to the single-pulse case again, the system eventually serves all the pending requests, and the recovery stage starts. However, another attack pulse arrives before the system restores the admission rate to the normal level when there is no attack. 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. $\eta_1 < \tau \leq \eta_1 + \eta_2$**:** Same as case 1, the admission rate drops linearly at 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 kth 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., $\frac{\partial \eta_1}{\partial \alpha_0} > 0$). Subsequently, the time to recover before the next pulse arrival, which is given by $k\tau - \eta_1$, is longer. Unfortunately, the admission rate still could not 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, at the next attack pulse's arrival, 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.

Case 3. $\eta_1 + \eta_2 < \tau \leq \eta_1 + \eta_2 + \eta_3$**:** The evolution of the system state in this case is similar to that in case 2 except that there is an additional recovery part governed by parameters C and D. We therefore omit the details here.

Case 4. $\tau > \eta_1 + \eta_2 + \eta_3$**:** In this case, the system state can always return to the steady state before the next pulse arrival. Since this case has been analyzed in [10], we do not further consider it here.

We have in fact proved the convergence of the system state for the four cases and have derived the maximal and minimal values of $\alpha(t)$ after convergence, denoted by α_{max} and α_{min}, respectively. However, in the lack of space, we just present α_{max} and α_{min} without a 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; as recalled from Figure 3, the magnitude increases with τ.

5. Detecting the LRDoS attacks

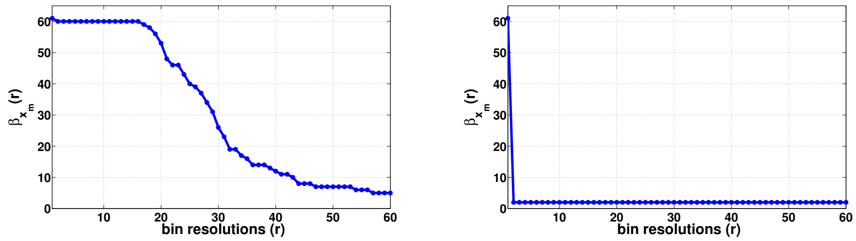
In this section, we propose a new anomaly-based detection scheme to discover LRDoS attacks. Our approach is based on the fact that the high-intensity request bursts sent out in an attack will disturb the distribution of the normal requests' arrival rate. Our detection scheme therefore consists of two components: (1) model the distribution using a histogram, and (2) use a nonparametric outlier detection algorithm to determine whether there is a change in the histogram (which is assumed due to an LRDoS attack). There is a similar outlier detection approach proposed in [20], in which samples are clustered under various resolutions. Our method is different from it in that we use histogram instead of clustering techniques.

Suppose that x_i, $i \in \mathbb{Z}$, are samples of the request rate. For every window of W samples, say $\{x_{m-W+1}, \ldots, x_m\}$, our detection algorithm will determine whether the latest sample x_m has disturbed the distribution. To do so, we use histograms 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 then plots the number of samples falling into each bin, or the *bin size*. Therefore, we could uncover the attack by detecting changes in the histogram for $\{x_{m-W+1}, \ldots, x_m\}$ by comparing it with the histogram for $\{x_{m-W+1}, \ldots, x_{n-1}\}$.

However, the main drawback of the histogram approach is to determine a proper bin resolution. We resolve this issue by applying the approach 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 both normal requests and malicious requests. As shown, the $\beta_{x_m}(r)$ values for the normal request decrease more gradually from 60 (when $r = 1$) to 1 (when $r = 60$). On the other hand, the $\beta_{x_m}(r)$ values for the malicious requests see a drastic drop from $r = 1$ to $r = 2$, because x_m, a sample from the attack pulse, is an outlier as compared with the normal request samples. Therefore, we could detect the attack by measuring the change in the $\beta_{x_m}(r)$ values as r increases; for this purpose, we define a cumulated 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 Eq. (11) can detect the 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 the normal



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

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

(or attack) traffic. If the attack intensity is high enough, the sample x_m from the attack traffic will mostly be the first one 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, consider that x_m is from the normal traffic. Furthermore, if the normal traffic intensity is uniformly distributed, then $\beta_{x_m}(r) = \frac{1}{r}$. Therefore $R_n(x_m) = \sum_{r=1}^{W-1} \frac{r+1}{r} < \sum_{r=1}^{W-1} \frac{r+r}{r} = 2(W-1) = R_a(x_m)$.

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

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

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 the normal traffic, $\bar{x}_n(r)$ should not be too far away from $\bar{x}$. If x_m is from the attack traffic, $\bar{x}_n(r)$ is much more closer to x_m because of the intensity of the attack traffic. Therefore, we revise the $R(x_m)$ values for a normal traffic sample and an attack traffic sample as:

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

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

The last step is to choose a threshold θ such that the detection outcome is positive if $R(x_m) \geq \theta$. We determine θ as follows. Denote σ_x as the standard deviation of the first $W - 1$ samples in the detection window: $\sigma_x = \sqrt{\frac{1}{W-1}\sum_{i=m-W+1}^{n-1}(x_i - \bar{x})^2}$. Note that if x_m is from the attack traffic, $\bar{x}_m - \bar{x} > \sigma_x$; if x_m is from the normal traffic, $\bar{x}_m - \bar{x} \approx \sigma_x$. Moreover, we have $\sum_{r=2}^{W-1}\frac{1}{r} < W - 1$. Taking these into consideration, we set $\theta = (2W-2)\sigma_x/\bar{x}$. Moreover, to cater for the burstiness in the normal traffic, we introduce a weight w_d to the threshold value: $\theta = (2W-2)w_d\sigma_x/\bar{x}$. In §1.6, we will evaluate the performance of the detection algorithm for different values of w_d.

6. Simulation results

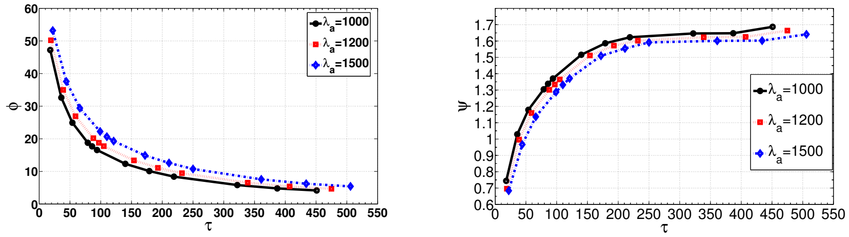
In this section, we present MATLAB simulation results for the evaluation of the attack capability and detection performance. We have again used the following parameters in our simulation experiments: $A = 0.00267$, $B = 0.2$, $C = 0.024$, $D = -1.4$, $\ell = 75$, $K = 0.01$, $\mu = 90$, and $\rho^* = 0.7$.

Attack capability In §1.4, we have analyzed the effects of τ on the system output and the admission rate of the victim system. Here we use simulations to further quantify the effects; we also study the effects for different values of λ_a. We first define two metrics to characterize the capability of an LRDoS attack: (1) the percent of normal requests dropped due to an attack, denoted by ϕ, and (2) the number of normal requests dropped due to an attack per λ_a, denoted by ψ. Therefore,

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

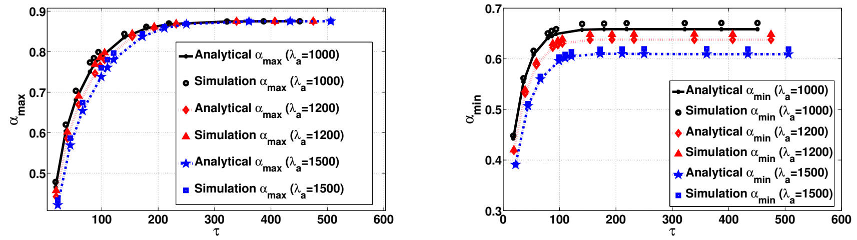
where T is the observation period (all N attack pulses arrive during T), and α^c is the admission rate when the system is in the steady state without attacks. Therefore, ϕ measures the absolute service degradation, whereas ψ measures the attack effectiveness in terms of the amount of service degradation caused by one attack pulse.

In Figure 5(a), we report the values ϕ and ψ for a set of 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 §1.4. By comparing the corresponding results in Figure 5(a) and Figure 5(b), we have observed that ϕ increases with λ_a, but ψ decreases with λ_a. Given the same λ_a, ϕ decreases with τ; however, ψ increases with τ. Moreover, as $\tau \to \infty$, $\phi \to 0$, because this is similar to the case of one attack pulse 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.



(a) The attack capability in terms of ϕ. (b) The attack capability in terms of ψ.

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

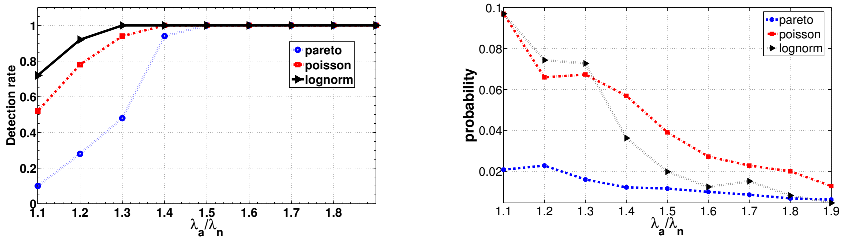


(a) α_{max} for different values of τ and λ_a. (b) α_{min} for different values of τ and λ_a.

Figure 6. The analytical and simulation results of α_{max} and α_{min}.

In Figure 6, we show the values of α_{max} and α_{min} obtained from simulations; we also include the analytical results from Eq. (9) and Eq. (10) for the purpose of comparison. First of all, the simulation results match very closely with the analytical results. Moreover, both α_{max} and α_{min} are increasing functions of τ which can be validated by taking the derivatives of Eq. (9) and Eq. (10). Moreover, α_{max} and α_{min} will eventually reach plateaus which indicate that the corresponding range of τ belongs to case 4. Moreover, the α_{max} values for the three λ_a cases converge to the same value; however, the α_{min} values do not—a higher λ_a gives a lower α_{min}. As a result, a higher attack intensity increases the magnitude of the oscillations.

Detection performance To evaluate the performance of the detection algorithm proposed in §1.5, we generate the background normal traffic using three different distributions: log-normal, Pareto, and Poisson. For the log-normal (or Pareto) distribution, we set the location parameter to 4.6027 (or 91.6667) and the scale parameter to 0.0707 (or 12). For the Poisson distribution, we set the rate to 100. By doing so, the mean arrival rates for these three distributions are all equal to 100 requests per



(a) Detection rate.

(b) False alarm rate.

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

second. Moreover, the samples for the request arrival rates are computed every second.

Figure 7(a) and Figure 7(b) show the detection rates and false alarm rates obtained from the simulation, respectively. The detection rate increases with $\frac{\lambda_a}{\lambda_n}$ for all three distributions; all attacks can be detected (i.e., the detection rate is 1) when $\frac{\lambda_a}{\lambda_n}$ reaches 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. On the other hand, the false alarm rate also improves with $\frac{\lambda_a}{\lambda_n}$. The false alarm rates for both log-normal and Pareto distributions exhibit a similar trend, whereas that for the Poisson distribution stays at a very low level.

7. Conclusions and future works

In this paper, we have analyzed the effects of the recently proposed low-rate DoS attacks on the feedback-based Internet services. By sending intermittent attack pulses, these attacks induce the victim system to generate false feedback signals, which could cause the system to decrease the request accepting rate. On the countermeasure side, we have designed a nonparametric algorithm to detect changes in the traffic distributions. We have conducted extensive simulation experiments to validate our findings.

There are a few extensions to this work. We are investigating how to optimize an LRDoS attack. One such problem is to optimize a given number of attack pulses by computing an optimal schedule of sending the pulses. We are also experimenting with a number of TCP variants, especially those available in the Linux system (e.g., Veno, Hybla, Westwood+) and evaluating the impact of the LRDoS attacks on them.

Acknowledgments

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) and a grant from the Cisco University Research Program Fund at Community Foundation Silicon Valley. The authors would also like to thank the reviewers for their useful comments.

References

[1] A. Robertsson, B. Wittenmark, M. Kihl, and M. Andersson, Design and evaluation of load control in Web-server systems, *Proceedings of American Control Conference*, 2004.

[2] J. Hellerstein, Y. Diao, S. Parekh, and D. Tilbury, *Feedback Control of Computing Systems*, Wiley-IEEE Press, 2004.

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

[4] 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 IEEE Real-Time and Embedded Technology and Applications Symposium*, 2003.

[5] Y. Diao, J. Hellerstein, S. Parekh, R. Griffith, G. Kaiser, and D. Phung, Self-managing systems: A control theory foundation, *Proceedings of IEEE Conf. Engineering of Computer-Based Systems*, 2005.

[6] Y. Diao, S. Parekh, R. Griffith, G. Kaiser, D. Phung, and J. Hellerstein, A control theory foundation for self-managing systems, *IEEE Journal on Selected Areas of Communications*, vol. 23, no. 12, 2005.

[7] R. Lotlika, R. Vatsavai, M. Mohania, and S. Chakravarthy, Policy schedule advisor for performance management, *Proceedings of IEEE Conference on Autonomic Computing*, 2005.

[8] R. Zhong, C. Lu, T. Abdelzaher, and J. Stankovic, Controlware: A middleware architecture for feedback control of software performance, *Proceedings of IEEE International Conference on Distributed Computing Systems*, 2002.

[9] M. Guirguis, A. Bestavros, I. Matta, and Y. Zhang, Exploiting the transients of adaptation for RoQ attacks on Internet resources, *Proceeedings of IEEE International Conference on Network Protocols*, 2004.

[10] M. Guirguis, A. Bestavros, I. Matta, and Y. Zhang, Reduction of quality RoQ attacks on Internet end-systems, *Proceedings of IEEE Annual INFOCOM Conference*, 2005.

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

[12] X. Luo, R. Chang, and E. Chan, Performance analysis of TCP/AQM under denial-of-service attacks, in *Proceedings of IEEE International Symposium on Modeling, Analysis, and Simulation*, 2005.

[13] A. Kuzmanovic and E. Knightly, Low-rate TCP-targeted denial of service attacks (the shrew vs. the mice and elephants), *Proceedings of ACM Annual SIGCOMM Conference*, 2003.

[14] M. Chan, E. Chang, L. Lu and S. Ng, Effect of malicious synchronization, *Proceedings of Applied Cryptography and Network Security*, 2006.

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

[16] H. Sun, J. Lui, and D. Yau, Defending against low-rate TCP attack: Dynamic detection and protection, *Proceedings of IEEE International Conference on Network Protocols*, 2004.

[17] Y. Chen and K. Hwang, Collaborative detection and filtering of shrew DDoS attacks using spectral analysis, *Journal of Parallel and Distributed Computing*, 2006.

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

[19] 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, no. 1/2, 2002.

[20] H. Fan, O. Zaïane, A. Foss, and J. Wu, A nonparametric outlier detection for effectively discovering top-N outliers from engineering data, *Proc. Pacific-Asia Conf. Knowledge Discovery and Data Mining*, 2006.