On the Adaptive Real-Time Detection of Fast-Propagating Network Worms

Jaeyeon Jung[1], Rodolfo A. Milito[2], and Vern Paxson[3]

[1] Mazu Networks
jyjung@mazunetworks.com
[2] Consentry Networks
rodolfo@consentry.com
[3] International Computer Science Institute
and Lawrence Berkeley National Laboratory
vern@icir.org

Abstract. We present two light-weight worm detection algorithms that offer significant advantages over fixed-threshold methods. The first algorithm, RBS (*rate-based sequential hypothesis testing*), aims at the large class of worms that attempts to quickly propagate, thus exhibiting abnormal levels of the rate at which hosts initiate connections to new destinations. The foundation of RBS derives from the theory of sequential hypothesis testing, the use of which for detecting randomly scanning hosts was first introduced by our previous work developing TRW [6]. The sequential hypothesis testing methodology enables us to engineer detectors to meet specific targets for false-positive and false-negative rates, rather than triggering when fixed thresholds are crossed. In this sense, the detectors that we introduce are truly adaptive.

We then introduce RBS+TRW, an algorithm that combines fan-out rate (RBS) and probability of failure (TRW) of connections to new destinations. RBS+TRW provides a unified framework that at one end acts as pure RBS and at the other end as pure TRW. Selecting an operating point that includes both mechanisms extends RBS's power in detecting worms that scan randomly selected IP addresses. Using four traces from three qualitatively different sites, we evaluate RBS and RBS+TRW in terms of false positives, false negatives, and detection speed, finding that RBS+TRW provides good detection of high-profile worms as well as internal Web crawlers that we use as proxies for targeting worms. In doing so, RBS+TRW generates fewer than 1 false alarm per hour for wide range of parameter choices.

1 Introduction

If a network worm penetrates a site's perimeter, it can quickly spread to other vulnerable hosts inside the site. The infection propagates by the compromised hosts repeatedly attempting to contact and infect new potential victims. The traffic pattern of fast worm propagation—a single host quickly contacting many different hosts—is a prominent feature across a number of types of worms, and detecting such patterns constitutes the basis for several worm detection approaches [2, 8, 13].

The problem of accurately detecting such worm scanning becomes particularly acute for enterprise networks comprised of a variety of types of hosts running numerous, different applications. This diversity makes it difficult to tune existing worm detection methods [2, 13] that presume preselected thresholds for connection rates and window sizes over which to compute whether a host's activity is "too quick." First, finding a single threshold rate that accommodates all (or almost all) benign hosts requires excessive tuning because of diverse application behaviors (e.g., a Web browser generating multiple concurrent connections to fetch embedded objects vs. an SSH client connecting to a server). Second, the window size chosen to compute the average rate affects the detection speed and accuracy; if too small, the detection algorithm is less resilient to small legitimate connection bursts, but if too big, the detection algorithm reacts slowly to fast propagating worms, for which brisk response is vital.

In this paper, we first develop an algorithm for detecting fast-propagating worms that use high-quality *targeting* information. We base our approach on analyzing the rate at which hosts initiate connections to new destinations. One such class of worms are those that spread in a *topological* fashion [11, 16]: they gather information on the locally infected host regarding other likely victims. For example, the Morris worm examined *.rhosts* files to see what other machines were known to the local machine [4, 10]. A related technique is the use of *meta-servers*, such as worms that query search engines for likely victims [5]. These targeting worms can spread extremely quickly, *even using relatively low-rate scanning*, because the vulnerability density of the addresses they probe is so much higher than if they use random scanning. Furthermore, these worms can evade many existing worm defense systems that rely on the artifacts of random scanning such as number of failed connections and the absence of preceding DNS lookups [2, 8, 17, 18].

Our detection algorithm, *rate-based sequential hypothesis testing* (RBS), operates on a per-host and per-connection basis and does not require access to packet contents. It is built on a probabilistic model that captures benign network characteristics, which allows us to discriminate between benign traffic and worm traffic. RBS also provides an analytic framework that enables a site to tailor its operation to its network traffic pattern and security policies.

We then present RBS+TRW, a unified framework for detecting fast-propagating worms independent of their scanning strategy. RBS+TRW is a blend of RBS and our previous *threshold random walk* (TRW) algorithm, which rapidly discriminates between random scanners and legitimate traffic based on their differing rates of connection failures [6]. Wald's sequential hypothesis testing [14] forms the basis for RBS+TRW's adaptive detection.

We begin with an overview of related work in §2. §3 then presents an analysis of network traces we obtained from two *internal* routers of a medium-size enterprise. The traced traffic includes more than 650 internal hosts, about 10% of the total at the site. We examine the distribution of the time between consecutive *first-contact connection requests*, defined by [8] as a packet addressed to a host with which the sender has not previously communicated. Our analysis finds that for benign network traffic, these interarrival times are bursty, but within the bursts can be approximately modeled using exponential distributions with a few hundred millisecond average intervals.

In §4, we develop the RBS algorithm, based on the same sequential hypothesis testing framework as TRW. RBS quickly identifies hosts that initiate first-contact connection requests at a rate n times higher than that of a typical benign host. RBS updates its decision process upon each data arrival, triggering an alarm after having observed enough empirical data to make a distinction between the candidate models of (somewhat slower) benign and (somewhat faster) malicious host activity.

In §5, we evaluate RBS using trace-driven simulations. We show that computing a simple trimmed mean suffices to automatically discover an effective set of parameters for running RBS. Moreover, we show that RBS triggers few false positives when n is small (0 false positives when $n \leq 5$) when assessed against a trace that includes a variety of applications.

§6 presents RBS+TRW, which automatically adapts between the rate at which a host initiates first-contact connection requests and observations of the success of these attempts, combining two different types of worm detection. Using datasets that contain active worms caught in action, we show that RBS+TRW provides fast detection of scanners and two hosts infected by Code Red II worms, while generating less than 1 false alarm per hour.

2 Related Work

Williamson first proposed limiting the rate of outgoing packets to new destinations [19] and implemented a virus throttle that confines a host to sending packets to no more than one new host a second [13]. While this virus throttling slows traffic that could result from worm propagation below a certain rate, it remains open how to set the rate such that it permits benign traffic without impairing detection capability. For example, Web servers that employ content distribution services cause legitimate Web browsing to generate many concurrent connections to different destinations, which a limit of one new destination per second would significantly hinder. If the characteristics of benign traffic cannot be consistently recognized, a rate-based defense system will be either ignored or disabled by its users.

Numerous efforts have since aimed to improve the simple virus throttle by taking into account other metrics such as increasing numbers of ICMP host-unreachable packets or TCP RST packets [2], number of failed first-contact connections [8, 17], and the absence of preceding DNS lookups [18]. However, these supplementary metrics will be not much of use if worms target only hosts that are reachable and have valid names (e.g., topological worms).

This work is inspired by our previous paper [6], which first used sequential hypothesis testing for scan detection. Our previous paper develops the threshold random walk (TRW) portscan detection algorithm based on the observation that a remote port scanner has a higher probability of attempting to contact a local host that does not exist or does not have the requested service running.

Weaver *et al.* [17] present an approximation to TRW suitable for implementation in high-performance network hardware for worm containment. For the same problem of detecting scanning worms, Schechter *et al.* [8] combine credit-based rate-limiting and reverse sequential hypothesis testing optimized to detect infection instances. In comparison, our RBS+TRW provides a unified framework built on sequential hypothesis testing with two metrics, a rate and a probability of success of a first-contact connection, that cover a broad range of worms, mostly independent of their scanning strategy or propagation speed.

There have been recent developments of worm detection using *content sifting* (finding common substrings in packets that are being sent in a many-to-many pattern) and automatic signature generation [7, 9, 15]. These approaches are orthogonal to our approach based on traffic behavior in that the former require payload inspection, for which computationally intensive operations are often needed. Moreover, although our approach requires a few parameter settings, it requires no training nor signature updates. However, content-based approaches are capable of detecting slowly-propagating (stealthy) worms that are indistinguishable from benign hosts by their connection-level traffic behaviors.

3 Data Analysis

We hypothesize that we can bound a benign host's network activity by a reasonably low fan-out per unit time, where we define fan-out as the number of first-contact connection requests a given host initiates. This fan-out per unit time, or *fan-out rate*, is an important traffic measure that we hope will allow us to separate benign hosts from relatively slowly scanning worms. In this section, we analyze traces of a site's internal network traffic, finding that a benign host's fan-out rate rarely exceeds a few first-contact connections per second, and time intervals between these connections can be approximately modeled as exponentially distributed.

We analyze a set of 22 anonymized network traces, each comprised of 10 minutes' of traffic recorded at `Lab` on Oct. 4, 2004. These were traced using `tcpdump` at two *internal* routers within `Lab`, enabling them to collect bidirectional traffic originated by internal hosts to both *external* hosts outside `Lab` and to other *internal* hosts inside `Lab`. Although we present the results from one particular site in this section, we studied 4 additional traces collected from three different sites. We used the additional traces to double-check empirical findings and later to evaluate our detection algorithm.

Table 1 summarizes the `Lab` dataset after some initial filtering to remove periodic NTP traffic and "triggered" connections in which a connection incoming to a host causes the host to initiate a secondary connection outbound. Such triggered connections should not be considered as first-contact connections when assessing whether a host is probing. The table shows that the traffic between internal `Lab` hosts consists of about 70% of the total outbound traffic recorded in the datasets. Had we traced the traffic at the site's border, we would have seen much less of the total network activity, and lower first-contact connections accordingly.

For each 10-minute trace, we observe a varying number of internal hosts initiating outbound traffic during the observation period. The last row in Table 1 shows that the largest number of active internal hosts in a 10-minute trace is 652.[4]

[4] Because each trace was anonymized separately, we are unable to tell how many distinct internal hosts appear across all of the traces.

Table 1. `Lab` dataset summary: This analysis does not include `NTP` traffic or triggered outgoing connections such as `Ident`, `Finger`, and `FTP` data-transfer

Outgoing connections	49,049 (100%)
to internal hosts	32,967 (67.21%)
to external hosts	16,082 (32.79%)
Internal hosts	≥ 652

From the traces we observe that over 99.5% of the hosts contacted fewer than 60 different hosts in 10 minutes, corresponding to an average fan-out rate below 0.1/sec. We categorize these hosts as benign. (Note that Twycross and Williamson [13] use fan-out rate of 1/sec as a maximum allowed speed for throttling virus spreads.)

Only 9 hosts exceed this threshold in this trace. Of these, 4 were aliases (introduced by the traces having separate anonymization namespaces) for an internal scanner used by the site for its own vulnerability assessment. Of the remainder, 3 hosts are main mail servers that forward large volumes of email, and the other 2 hosts are internal web crawlers that build search engine databases of the content served by internal Web servers. By manual inspection, we also later found another appearance of the internal scanner that we missed using our 0.1/sec fan-out rate threshold, as in that instance the scanner contacted only 51 different IP addresses during the 10-minute period. We exclude the scanners and the crawlers[5] from our subsequent analysis. In what follows, we develop a model that captures fan-out rate statistics of this set of "purely" benign hosts.

3.1 Time Interval to Visit New Destinations

A host engaged in scanning or worm propagation will generally probe a significant number of hosts in a short time period, yielding an elevated first-contact connection rate. In this section, we analyze our dataset to determine the distribution of first-contact interarrivals as initiated by benign hosts. We then explore the discriminating power of this metric for a worm whose first-contact connections arrive a factor of n more quickly.

Figure 1 shows the distribution of the amount of time between first-contact connections for individual hosts. Here we have separated out the scanners (identified as discussed above). While the average interarrival time is 39.2 sec, we often see benign, non-scanner hosts initiating multiple first-contact connections separated by very little ($<$ 1 sec) time. In fact, these short time intervals account for about 40% of the total intervals generated by benign hosts, which makes it impractical to use 1/sec fan-out rate to identify possible worm propagation activity.

However, when focusing on sub-second interarrivals, we find that a benign host's short-time-scale activity fits fairly well to an exponential distribution, as illustrated in Figure 2. Here the fit to the empirical data uses $\mu = 261$ msec. We note that a scanner could craft its probing scheduling such that its fine-grained scanning behavior matches

[5] Note that we do not include the mail servers in the set of scanners, as they are not scanners per se, but rather applications that happen in this environment to exhibit high fan-out.

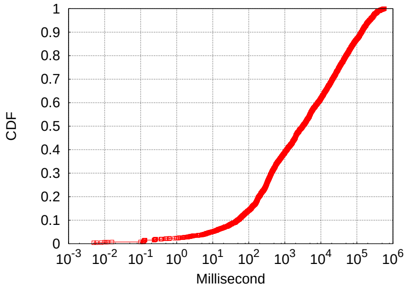


Fig. 1. Cumulative distribution of first-contact connections' interarrival time, per host

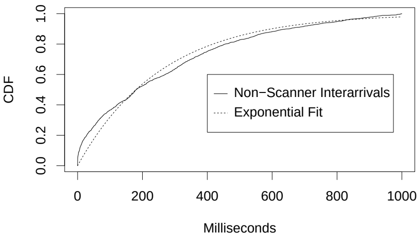


Fig. 2. First-contact interarrivals initiated by benign hosts roughly follow an exponential distribution with mean $\mu = 261$ msec.

that of benign users, or at least runs slower than what we model as benign activity. However, this will significantly slow down the scanning speed, so compelling attackers to make this modification constitutes an advance in the ongoing "arms race" between attackers and defenders.

We also note that we could extract significantly more precise interarrival models—including differing mean interarrival rates—if we partitioned the traffic based on its application protocol. While investigating this refinement remains a topic for future work, in our present effort we want to explore the efficacy of as *simple* a model as possible. If our algorithm can prove effective without having to characterize different protocols separately, we will benefit a great deal from having fewer parameters that need to be tuned operationally.

In the next section, based on these characteristics of benign activity, we develop our detection algorithm, RBS, for quickly identifying scanners or worm infectees with a high accuracy.

4 RBS: Rate-Based Sequential Hypothesis Testing

In this section, we develop a rate-based sequential hypothesis testing algorithm, RBS, which aims to quickly identify hosts issuing first-contact connections at rates higher than what we model as benign activity.

Let H_1 be the hypothesis that a given host is engaged in worm propagation, and let H_0 be the null hypothesis that the host exhibits benign network activity. A host generates an *event* when it initiates a connection to a destination with which the host has not previously communicated, i.e., when the host initiates a first-contact connection. As discussed in the previous section, we assume that the interarrival times of such events follow an exponential distribution with mean $1/\lambda_0$ (benign host) or $1/\lambda_1$ (scanner). When a host generates the i^{th} event at time t_i, we can compute an interarrival time,

$X_i = t_i - t_{i-1}$ for $i \geq 1$ and t_0 the initial starting point, and update the likelihood ratio of the host being engaged in scanning (or benign).

Define X_1, X_2, $\ldots$, X_n as a sequence of such interarrival times. Since we model each X_i as IID non-negative exponential random variables, their sum, T_n, is the n-Erlang distribution:

$$f_n(T_n|H_1) = \frac{\lambda_1(\lambda_1 T_n)^{n-1}}{(n-1)!} \exp^{-\lambda_1 T_n} \tag{1}$$

Based on Equation (1), we can develop a sequential hypothesis test in which we define the likelihood ratio as:

$$\Lambda(n, T_n) = \frac{f_n(T_n|H_1)}{f_n(T_n|H_0)} = \left(\frac{\lambda_1}{\lambda_0}\right)^n \exp^{-(\lambda_1 - \lambda_0)T_n} \tag{2}$$

and the detection rules as:

$$\text{Output} = \begin{cases} H_1 & \text{if } \Lambda(n, T_n) \geq \eta_1 \\ H_0 & \text{if } \Lambda(n, T_n) \leq \eta_0 \\ \text{Pending} & \text{if } \eta_0 < \Lambda(n, T_n) < \eta_1 \end{cases}$$

where we can set η_1 and η_0 in terms of a target false positive rate (the proportion of benign hosts that are erroneously reported as scanners), α and a target detection rate (the proportion of scanners that are correctly reported as scanners), β [14]:

$$\eta_1 \leftarrow \frac{\beta}{\alpha} \tag{3}$$

$$\eta_0 \leftarrow \frac{1-\beta}{1-\alpha} \tag{4}$$

Wald shows that setting thresholds as above guarantees that the resulting false positive rate is bounded by $\frac{\alpha}{\beta}$ and the false negative rate is by $\frac{1-\beta}{1-\alpha}$ [14]. Given that β is usually set to a value higher than 0.99 and α to a value lower than 0.001, the margin of error becomes negligible (i.e., $\frac{1}{\beta} \approx 1$ and $\frac{1}{1-\alpha} \approx 1$).

An essential advantage of RBS over a simpler scheme using a fixed-rate threshold is that RBS is more robust to legitimate bursty connections. Figure 3 illustrates how an average arrival rate can fluctuate a great deal depending on the window size over which we compute the average. However, RBS effectively can *adapt* its window size until it finds consistency over a sufficient number of observations to reach a decision.

For instance, if a host has initiated n first-contact connections and the elapsed time for the n^{th} connection is T_n, RBS chooses H_1 (scanner) only if the likelihood ratio $\Lambda(n, T_n)$ exceeds η_1. Using Equations (2) and (3), we can obtain a threshold on the

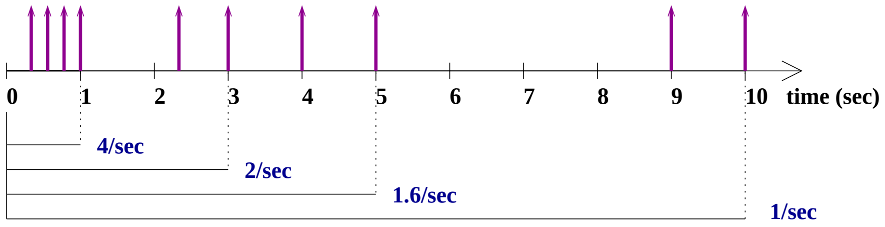


Fig. 3. 10 first-contact connection arrivals in 10 seconds: The figure illustrates that the average arrival rate can vary depending on the window size.

elapsed time, T_{H_1}, below which we arrive at an H_1 (scanner) decision:

$$\begin{aligned}\frac{\beta}{\alpha} &\leq \Lambda(n, T_n)\\ \frac{\beta}{\alpha} &\leq \left(\frac{\lambda_1}{\lambda_0}\right)^n \exp^{-(\lambda_1-\lambda_0)T_n}\\ \ln\frac{\beta}{\alpha} &\leq n\ln\frac{\lambda_1}{\lambda_0} - (\lambda_1-\lambda_0)T_n\\ T_n &\leq n\frac{\ln\frac{\lambda_1}{\lambda_0}}{\lambda_1-\lambda_0} - \frac{\ln\frac{\beta}{\alpha}}{\lambda_1-\lambda_0} = T_{H_1} \qquad (5)\end{aligned}$$

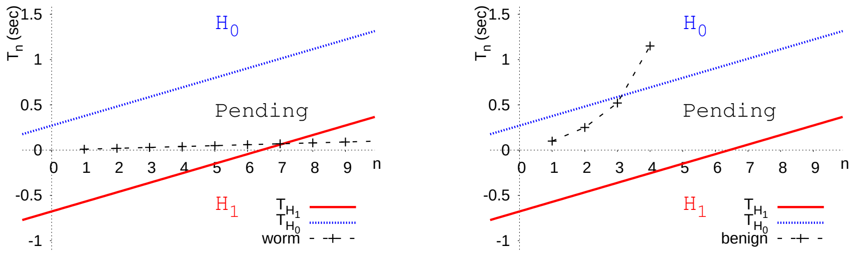
Likewise, we can obtain a threshold elapsed time T_{H_0}, above which we conclude H_0 (benign host):

$$T_{H_0} = n\frac{\ln\frac{\lambda_1}{\lambda_0}}{\lambda_1-\lambda_0} - \frac{\ln\frac{1-\beta}{1-\alpha}}{\lambda_1-\lambda_0} \qquad (6)$$

Figure 4 shows how those threshold elapsed times, T_{H_1} and T_{H_0}, partition the area into three decision regions—H_1, H_0, and `Pending`. Figure 4(a) illustrates T_n of a host issuing first-contact connections at 100/second. At the 8^{th} event, T_8 falls below T_{H_1}, which drives the likelihood ratio to reach the H_1 decision. Note that with the set of parameters used in Figure 4, RBS defers making a decision until it sees at least 7 events; this occurs because the elapsed time, T_n, is always greater than T_{H_1} up to $n = 6$. (T_i is a non-negative, non-decreasing random variable and T_{H_1} becomes positive when $n > 6.1$, given λ_0 =3/sec, λ_1 =20/sec, $\alpha = 10^{-5}$, and $\beta = 0.99$.) This initial holding period makes RBS robust against small traffic bursts. We can shorten this initial holding period, however, if we use a smaller β or larger α.

In general, Equation (5) provides important insights into the priors and the performance of RBS. T_{H_1} is a function of n, taking a form of $g(n) = a(n-c)$, where $a = (\ln\frac{\lambda_1}{\lambda_0})/(\lambda_1 - \lambda_0)$ and $c = (\ln\frac{\beta}{\alpha})/(\ln\frac{\lambda_1}{\lambda_0})$:

1. α and β affect only c, the minimum number of events required for detection (i.e., the minimum window size). For fixed values of λ_1 and λ_0, lower values of α or higher values of β (i.e., greater accuracy in our decisions) let more initial connections escape before RBS declares H_1. One can shorten this initial holding period by



(a) Fast spreading worm with 100 first-contact connections/second will be detected by RBS at the 8th connection attempt

(b) Benign host with 4 first-contact connections/second will bypass RBS at the 4th connection attempt

Fig. 4. T_{H_1} and T_{H_0} when $\lambda_0 = 3$/sec, $\lambda_1 = 20$/sec, $\alpha = 10^{-5}$, and $\beta = 0.99$. The X axis represents the n^{th} event and Y axis represents the elapsed time for the n^{th} event

increasing α or decreasing β. But we can only do so to a limited degree, as c needs to be greater than the size of bursty arrivals that we often observe from Web or P2P applications, in order to avoid excessive false alarms. Another different way to prevent damage from those initially allowed connection attempts is to hold them at a switch until proven innocent [8].

2. λ_0 and λ_1 determine a, the slope of T_{H_1} over n. The inverse of the slope gives the minimum connection rate that RBS can detect. Any host generating first-contact connections at a higher rate than λ_1 intercepts $g(x)$ with probability 1. There is a built-in robustness in this, because the slope is strictly larger than $\frac{1}{\lambda_1}$ (what we model as a scanner), which follows from the inequality $\ln(x) < x - 1, 0 < x < 1$.
3. Although we use λ_1 to model a scanner's first-contact connection rate, RBS can detect any scanner with a rate λ' provided that:

$$\lambda' > \frac{1}{a} = \frac{\lambda_1 - \lambda_0}{\ln \lambda_1 - \ln \lambda_0} \tag{7}$$

because a host with a rate higher than λ' will eventually cross the line of T_{H_1} and thus trigger an alarm.

Finally, Equations (5) and (6) show that RBS bases its decision on two parameters—the number of attempts, n, and the elapsed time, $T(n)$—and not the actual realization of the arrival process.

5 Evaluation

We evaluated the performance of RBS in terms of false positives using a trace-driven simulation of the `Enterprise` dataset. RBS is in essence an algorithm that provides a tight bound of benign hosts' fan-out rate, enabling us to detect worms and scanners that employ higher-than-normal fan-out rates.

The `Enterprise` packet trace was captured at internal routers of a small enterprise network in November 2006. The trace contains 184 active hosts that initiated 238,407 TCP connections during the 1-hour collection period. To establish a ground truth, we extensively analyzed the trace using well-known application signatures and the Ethereal program [3] and found that about 76 applications were running at the time, including P2P clients such as BitTorrent and KaZaA, and VoIP programs such as Skype. Moreover, we found no infected machines nor scanners in the trace, making it suitable for testing RBS's accuracy in terms of false positives.

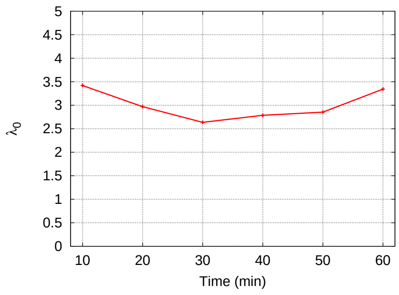
We need to set four parameters (α, β, λ_0, and λ_1) in order to run RBS. For high accuracy, we set $\beta = 0.99$ (99% target detection rate) and $\alpha = 10^{-6}$ (0.0001% target false alarm rate). Note that we set α very low because the detection algorithm executes for every first-contact connection initiated by a local host, which adds up to a very large number of tests.

The typical fan-out rate of benign hosts (λ_0) can change according to time (e.g., weekdays vs. weekend) and site (e.g., a small company where most network traffic is related to database transactions vs. a big ISP). To accommodate such changes, rather than asking an administrator to provide a magic number, we automatically infer the parameter λ_0 as follows:

- **Observation**: We observe interarrival times of first-contact connections generated by each host (i) and keep a list of mean interarrival times per host (μ_1, μ_2, $\mu_3, \ldots$) for a 10-minute period.
- **Inference**: At the end of an observation run, we compute a 10% trimmed mean [12] of the μ_i's: we first sort the data and remove the top and bottom 10% of the data before evaluating the arithmetic mean. As such, the inferred mean will not be affected by newly infected machines as long as the population of the infected machines stays below 10%. We set $1/\lambda_0$ equal to the inferred mean. Figure 5 shows the inferred values of λ_0 for the `Enterprise` dataset.

However, there is no obvious pick for λ_1, since a worm can choose an arbitrary propagation rate. If λ_1/λ_0 is close to 1, RBS takes longer to make a decision; but on the other hand, it can detect slower scanners than for higher λ_1/λ_0 ratios, per Equation (7).

Figure 6 shows the simulation results of RBS for the `Enterprise` dataset as we vary λ_1 as a multiple of λ_0. As described above, both λ_0 and λ_1 get updated every 10 minutes. RBS generates no false positives when λ_1/λ_0 is less than 6. However, RBS erroneously triggers for 2 hosts (a BitTorrent client and a chatty Web browser) when the ratio is higher than 7. The main reason for these false positives is short bursts. As discussed in §4, when λ_1/λ_0 is high, RBS becomes sensitive to short bursts, making it prone to generating false positives. Given that bursty connections are somewhat prevalent among many applications, this result leads us to recommend a small λ_1/λ_0 ratio.



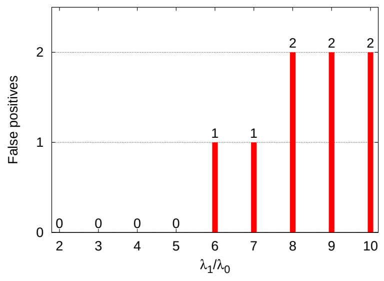


Fig. 5. 10% trimmed mean of first-contact connection arrival rate updated every 10 minutes

Fig. 6. Trace-driven simulation results of RBS varying λ_1 when $\alpha = 10^{-6}$, and $\beta = 0.99$

A caveat of using a small ratio is that RBS may miss carefully crafted scan traffic if the scanner repeatedly generates short bursts followed by a long idle time.

Thus, while this assessment is against a fairly modest amount of data, we find the results promising. We conduct a more extensive evaluation in §6.

6 Hybrid Approach: RBS+TRW

RBS uses *fan-out rate* to differentiate benign traffic from scanners (or targeting worms), which we model as Poisson processes with rates λ_0 (benign) and λ_1 (scanner), with $\lambda_0 < \lambda_1$. Another discriminatory metric proved to work well in detecting scanners is the *failure ratio* of first-contact connections [6, 17, 8]. TRW [6] works by modeling Bernoulli processes with **success** probabilities, θ_0 (benign) and θ_1 (scanner), with $1 - \theta_0 < 1 - \theta_1$. In this section, we develop a combined worm detection algorithm that exploits *both* a fan-out rate model and a failure ratio model. We evaluate the hybrid using trace-driven simulation, finding that this combined algorithm, RBS+TRW, improves both overall accuracy and speed of detection.

Suppose that a given host has initiated connections to n different destinations, and that the elapsed time until the n^{th} connection is T_n. Among those n destinations, S_n accepted the connection request (success) and $F_n = n - S_n$ rejected or did not respond (failure). Applying the models from RBS and TRW [6], we obtain a conditional probability distribution function for scanners:

$$\begin{aligned} f[(S_n, T_n)|H_1] &= P[S_n|T_n, H_1] \times f[T_n|H_1] \\ &= \binom{n}{S_n} \theta_1^{S_n} (1 - \theta_1)^{F_n} \\ &\quad \times \frac{\lambda_1 (\lambda_1 T_n)^{n-1}}{(n-1)!} \exp^{-\lambda_1 T_n} \end{aligned}$$

where $P[S_n|T_n, H_1]$ is the probability of getting S_n success events when each event will succeed with an equal probability of θ_1, and $f[T_n|H_1]$ is an n-Erlang distribution in which each interarrival time is exponentially distributed with mean $1/\lambda_1$.

Analogous to $f[(S_n, T_n)|H_1]$, for benign hosts we can derive:

$$\begin{aligned} f[(S_n, T)|H_0] = \binom{n}{S_n} \theta_0^{S_n} (1-\theta_0)^{F_n} \\ \times \frac{\lambda_0 (\lambda_0 T_n)^{n-1}}{(n-1)!} \exp^{-\lambda_0 T_n} . \end{aligned}$$

We then define the likelihood ratio, $\Lambda(S_n, T_n)$, as

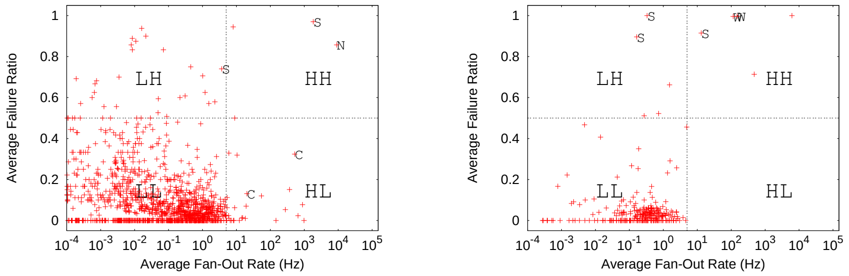
$$\begin{aligned} \Lambda(S_n, T_n) &= \frac{f[(S_n, T_n)|H_1]}{f[(S_n, T_n)|H_0]} \\ &= \left(\frac{\theta_1}{\theta_0}\right)^{S_n} \left(\frac{1-\theta_1}{1-\theta_0}\right)^{F_n} \\ &\quad \times \left(\frac{\lambda_1}{\lambda_0}\right)^{n} \exp^{-(\lambda_1 - \lambda_0) T_n} . \end{aligned}$$

It is interesting to note that $\Lambda(S_n, T_n)$ is just the product of Λ_{TRW} and Λ_{RBS}. Moreover, $\Lambda(S_n, T_n)$ reduces to Λ_{TRW} when there is no difference in fan-out rates between benign and scanning hosts (λ_1 = λ_0). Likewise, $\Lambda(S_n, T_n)$ reduces to Λ_{RBS} when there is no difference in failure ratios ($\theta_1 = \theta_0$).

We evaluate this combined approach, RBS+TRW, using two new sets of traces, each of which contains different types of scanners that happen to wind up contrasting the strengths of RBS and TRW. We first categorize hosts into four classes based on their fan-out rates and failure ratios. In what follows, we discuss types of scanners falling into each region and detection algorithms capable of detecting such hosts.

– **Class LH** (low fan-out rate, high failure ratio): Slow-scanning worms or scanners that probe blindly (randomly or sequentially) will likely generate many failures, triggering TRW with a high probability.
– **Class HH** (high fan-out rate, high failure ratio): Fast-scanning worms (e.g., Code Red, Slammer) that exhibit both a high fan-out rate and a high failure ratio will very likely to drive both TRW and RBS to quickly reach their detection thresholds.
– **Class HL** (high fan-out rate, low failure ratio): Flash, metaserver, and topological worms [16] belong to this class. These worms build or acquire a list of target hosts and then propagate over only those potential victims, so their connection attempts tend to succeed. While these targeting worms can bypass TRW, their high fan-out rate should trigger RBS.
– **Class LL** (low fan-out rate, low failure ratio): Most benign hosts fall into this class, in which their network behavior is characterized by a low fan-out rate and a low failure ratio. Typically, a legitimate host's fan-out rate rarely exceeds a few first-contact connections per second. In addition, benign users do not initiate traffic to hosts unless there is reason to believe the host will accept the connection request, and thus will exhibit a high success probability. Neither TRW nor RBS will trigger

hosts in this class, which in turn, allows particularly stealthy worms, or passive "contagion" worms that rely on a user's behavior for propagation [16], to evade detection. Worms of this type represent a formidable challenge that remains for future work to attempt to address.



(a) `Lab-II` (`S`: scanner, `N`: host running `nmap`, `C`: internal Web crawler)

(b) `ISP` (`S`: scanner, `W`: Code Red II infectee)

Fig. 7. Classification of hosts present in the evaluation datasets: Each point represents a local host that generated more than 5 first-contact connections

We use an average 5 Hz fan-out rate (λ_0) and 0.5 failure ratio (1-θ_0) as baselines in order to categorize hosts in our trace. Ideally, we should investigate all the hosts in the traces to obtain a ground truth, but because of the sheer amount of traffic volume (more than 2 million connections), we resort to this screening process to sift out the many hosts with quite limited activity.

We compute a fan-out rate with a sliding window of size 5 in order to capture bursty arrivals that often result from concurrent Web connections addressed to different Web sites for embedded objects. Figure 7 classifies hosts in the datasets based on the 5 Hz fan-out rate and 0.5 failure ratio thresholds.

Table 2 shows the details of the datasets we use for evaluation. The `Lab-II` dataset was collected at the same enterprise network as `Lab`. It is composed of 137 one-hour long traces from December 2004 and Janunary 2005, recorded at internal routers connecting a variety of subnets to the rest of the enterprise and the Internet. The `ISP` dataset was recorded using `tcpdump` at the border of a small ISP in April 2003. It contains traffic from 389 active hosts during the 10-hour monitoring period (The high number of connections is due to worm infections during the time of measurement.).

Table 2. Evaluation datasets: scanning hosts include vulnerability scanners, worm infectees, and hosts that we use proxies for targeting worms because of their anomalous high-fan-out rate.

			Lab-II	ISP
Outgoing Connections			796,049	1,402,178
Duration			137 hours	10.5 hours
	HH	scanning	2	3
H		benign	1	2
	LH	scanning	1	2
O		benign	34	3
	HL	scanning	2	0
S		benign	26	0
	LL	scanning	0	0
T		benign	1321	260
	$\leq$ 5 first-contact connections		2,621	119
S	Total	scanning	5	5
		benign	4,003	384
		Total	4,008	389

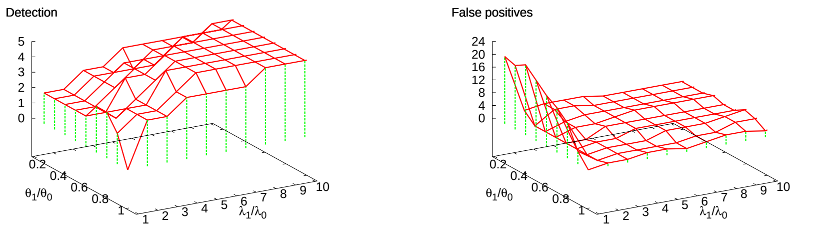
The table shows the division of the internal hosts into the four categories discussed above. Manual inspection of the hosts in **HH**, **HL**, and **LH**[6] reveals that there are 5 hosts each in both of Lab-II and ISP whose behavior qualifies them as scanners and worms that we aim to detect (H_1) because of their high-fan-out or high-failure behaviors: For Lab-II, the 2 **HH** hosts are one internal vulnerability scanner and one host that did a fast nmap [1] scan of 7 other hosts; 1 **LH** host is another internal vulnerability scanner; 2 **HL** hosts are internal Web crawlers that occasionally contacted tens of internal Web servers to update search engine databases. For ISP, the **HH** hosts are two Code Red II infectees plus an HTTP scanner, and the **LH** hosts are 2 slower HTTP scanners.

The one **HH** host in the Lab-II dataset that we classify as benign (H_0) turns out to be a NetBIOS client that often (benignly) made connection requests to absent hosts. The 2 benign **HH** hosts in the ISP dataset are all clients running P2P applications that attempt to contact a large number of transient peers that often do not respond. Most benign **LH** hosts are either low-profile NetBIOS clients (Lab-II) or P2P clients (ISP), and most benign **HL** hosts from Lab-II are caused by Web clients accessing Web sites with many images stored elsewhere (e.g., a popular news site using Akamai's content distribution service, and a weather site having sponsor sites' images embedded).

Table 2 also shows that while those two thresholds are useful for nailing down a set of suspicious hosts (all in either **HH**, **LH**, or **HL**), a simple detection method based on fixed thresholds would cause 66 false positives because of benign hosts scattered in the **LH** and **HL** regions, as shown in Figure 7. However, using dynamic thresholds based on the previously observed behavior, RBS+TRW accurately identifies those 10 target hosts while significantly reducing false positives.

[6] We looked into each host in those three classes for the ISP dataset, and the 66 of such hosts for the Lab-II dataset that generated more than 20 first-contact connections in a one-hour monitoring period.

We evaluate RBS+TRW by varying λ_1 from λ_0 to $10\lambda_0$, and θ_1 from $0.2\theta_0$ to θ_0. As discussed in §5, we infer λ_0 and θ_0 using 10% trimmed means.[7] We set $\beta = 0.99$, and $\alpha = 10^{-6}$. Figures 8 and 9 show the number of detections and false positives for each pair of λ_1 and θ_1. In particular, for $\lambda_1 = \lambda_0$, the combined algorithm reduces to TRW (dashed vertical lines along the θ axis), and when $\theta_1 = \theta_0$, to RBS (dashed vertical lines along the λ axis).



(a) Detection (out of 5 targets)

(b) False alarms (out of 4,008 hosts)

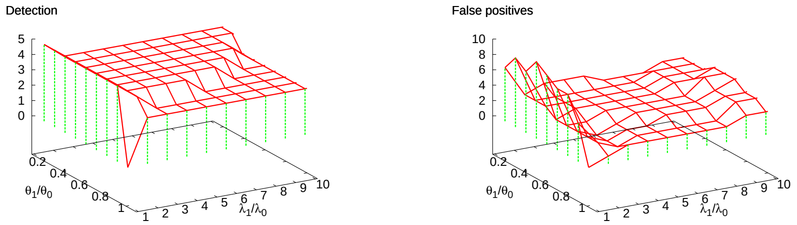
Fig. 8. Simulation results of RBS+TRW for the `Lab-II` dataset, varying λ_1 and θ_1

Table 3. Evaluation of RBS+TRW vs. RBS and TRW. Both `Lab-II` and `ISP` each have 5 scanners. $\overline{N}|H_1$ represents the average number of first-contact connections originated by the detected hosts upon detection.

			`Lab-II`			`ISP`		
	λ_1	θ_1	False -	False +	$\overline{N}\|H_1$	False -	False +	$\overline{N}\|H_1$
RBS	$10\lambda_0$	$= \theta_0$	0	2	5.6	2	3	6.4
TRW	$= \lambda_0$	$0.2\theta_0$	3	21	18.5	0	7	10.0
RBS+TRW	$5\lambda_0$	$0.6\theta_0$	0	3	6.9	1	3	5.0

Table 3 compares the performance of the combined algorithm against that of RBS and TRW alone. First, we find the priors that make RBS (TRW) the most effective (0 false negatives) in identifying scanners in the `Lab-II` (`ISP`) dataset. The nature of our test datasets keeps either algorithm from working better across both datasets. In fact,

[7] We placed an upper bound (0.9) on θ_0, since a small value of θ_0 (e.g., 0.9999) causes TRW to trigger for a few spurious failures.



(a) Detection (out of 5 targets)

(b) False alarms (out of 389 hosts)

Fig. 9. Simulation results of RBS+TRW for the `ISP` dataset, varying λ_1 and θ_1

when $\lambda_1 = 10\lambda_0$ and $\theta_1 = \theta_0$, RBS has 0 false negatives for `Lab-II`, but misses 2 **LH** scanners in `ISP`. In comparison, when $\lambda_1 = \lambda_0$ and $\theta_1 = 0.2\theta_0$, TRW has 0 false negatives for `ISP`, but misses 3 scanners in `Lab-II`, including the two Web crawlers.

We could address the problem of false negatives for either algorithm by running TRW and RBS in parallel, raising an alarm if either algorithm decides so. However, this approach comes at a cost of an increased number of false alarms, which usually result from **LH** hosts (e.g., Windows NetBIOS connections, often made to absent hosts) or **HL** hosts (e.g., a busy mail server or a Web proxy).

In general, improving the accuracy of a detection algorithm requires iterative adjustments of decision rules: first improving the detection rate by loosening the decision rule, and then decreasing the false positive rate by tightening the decision rule without losing too many correct detections. For this iteration, our combined algorithm, RBS+TRW provides two knobs, λ_1 and θ_1, that we can adjust to tune the detector to a site's traffic characteristics.

The trace-driven simulation shows that RBS+TRW with $\lambda_1 = 5\lambda_0$ and $\theta_1 = 0.6\theta_0$ misses only one low-profile target host (a slow HTTP scanner from `ISP`) while generating no more than 6 false positives, per Table 3. Had we run RBS and TRW in parallel, we could have eliminated all the false negatives, but at the cost of 33 false alarms altogether.

Overall, RBS+TRW provides the good detection of high-profile worms and scanners (no more than 2 misses across both datasets) while generating less than 1 false alarm per hour for a wide range of parameters ($\lambda_1 \in [4\lambda_0, 8\lambda_0]$ and $\theta_1 \in [0.4\theta_0, 0.7\theta_0]$), and reaching its detection decisions quickly (less than 7 first-contact connections on average).

7 Discussion

This section discusses several technical issues that may arise when employing RBS+TRW in practice. While addressing these issues is beyond the scope of this paper, we outline ideas and directions based on which we will pursue them in future work.

Operational issues: A worm detection device running RBS+TRW needs to maintain per local host information. For each host, a detector must track first-contact connections originated by the host, their failure/success status, and the elapsed time. The state thus increases proportional to the number of local hosts in the network (N) and the sum of all their currently pending first-contact connections. Given that RBS+TRW requires ≤ 10 first-contact connections on average to reach a decision ($\S6$), we can estimate amount of state as scaling on the order of $10N$. Note that every time RBS+TRW crosses either threshold, it resets its states for the corresponding host.

When constrained by computation and storage resources, one can employ cache data structures suggested by Weaver *et al.* [17] that track first-contact connections with a high precision. However, we note that running RBS+TRW on aggregate traffic across hosts (as opposed to the per-host operation for which it is designed) can significantly affect the detection performance due to the uneven traffic distribution generated by each end-host [20].

Post-detection response: The results in Table 3 correspond to RBS+TRW generating 0.07 false alarms per hour at the `Lab-II` site and 0.57 per hour at the `ISP` site. This low rate, coupled with RBS+TRW's fast detection speed, make it potentially suitable for automated containment, crucial to defending against fast-spreading worms. Alternatively, a network operator could employ connection rate-limiting for hosts detected by RBS+TRW, automatically restricting such hosts to a low fan-out rate.

Extensions: One can complement RBS+TRW with a classification engine and run the algorithm with specific parameters per application. For instance, many peer-to-peer applications probe other neighboring hosts in order to find the best peer from which to download a file. For a peer-to-peer client having a large number of transient peers, this probing activity can generate many failed connections, leading to an alarm. In such a case, grouping peer-to-peer traffic and running a separate instance of RBS+TRW with the parameters particularly tuned for this application should significantly improve the algorithm's performance.

Limitations: As indicated in Figure 7, RBS+TRW is unable to detect targeting worms using high-quality hit lists comprised of at least 70% active hosts and spreading no faster than several first-contact connections per second. Detecting such worms might be possible by working on larger time scales. For example, a scanner that generates first-contact connections at a rate of 1 Hz will end up accessing 3,600 different hosts in an hour, far outnumbering the sustained activity of a typical benign host. Thus, a natural avenue for future work is assessing the operation of RBS on longer timescales.

Finally, attackers can game our detection algorithm by tricking end users into generating first-contact connections either at a high rate (RBS), or that will likely end up failing (TRW). For instance, similar to an attack in [8], an attacker could put content on a web site with numerous embedded links to non-existent destinations.

8 Conclusion

We have presented a worm detection algorithm, RBS (*rate-based sequential hypothesis testing*), that rapidly identifies high-fan-out behavior by hosts based on the rate at which the hosts initiate connections to new destinations. RBS uses the sequential hypothesis testing [14] framework. While built using a model that the time between connection attempts to new destinations is exponentially distributed (which we show is a reasonable approximation for bursts of activity), RBS decisions reflect the aggregate measurement of the total elapsed time over a number of attempts, not the characteristics of individual arrivals. We define RBS in terms of a single discriminating metric—the rate of connection attempts—which differs substantially between benign hosts and an important class of worms. While the choice of such a metric evokes the measurement of an average rate over a window of certain size (and the comparison of the measured rate to a fixed threshold), RBS is more elaborate. The algorithm draws from sequential hypothesis testing the ability to adapt its decision-making in response to the available measurements in order to meet specified error requirements. We can view this as an adaptation of both the window size (i.e., how many attempts to make a decision) and the threshold (i.e., what is the minimum measured rate over that window that leads to a trigger). This adaptation gives RBS a robustness unseen in fixed window/threshold schemes.

We evaluated RBS using trace-driven simulations. We find that when the factor of speed difference, n, between a scanner and a benign host is small, RBS requires more empirical data to arrive at a detection decision but stays robust against short bursts. When n is less than 6, RBS generates no false positives for a 1-hour trace that includes P2P clients and VoIP programs known to connect to a set of peers.

We then presented RBS+TRW, a hybrid of RBS and TRW [6] which combines *fan-out rate* and *probability of success* of each first-contact connection. RBS+TRW provides a unified framework for detecting fast-propagating worms independent of their scanning strategy (i.e., topological or scanning worms). Using two traces from two qualitatively different sites, containing 389 active hosts and 4,008 active hosts, we show that RBS+TRW provides fast detection of hosts infected by Code Red II, as well as the internal Web crawlers that we use as proxies for topological worms. In doing so, it generates less than 1 false alarm per hour.

9 Acknowledgements

Our thanks to anonymous reviewers for their helpful comments. This work was supported by NSF Awards STI-0334088, NSF-0433702 and CNS-0627320, for which we are grateful. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors or originators and do not necessarily reflect the views of the National Science Foundation.

References

1. Nmap — free security scanner for network exploration & security audits. `http://www.insecure.org/nmap/`.

2. CHEN, S., AND TANG, Y. Slowing Down Internet Worms. In *Proceedings of the 24th International Conference on Distributed Computing Systems (ICDCS'04)* (Tokyo, Japan, Mar. 2004).
3. EHTEREAL.COM. Ethereal. `http://www.ethereal.com/`.
4. EICHIN, M. W., AND ROCHLIS, J. A. With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988. In *Proceedings of the IEEE Symposium on Research in Security and Privacy* (1989).
5. F-SECURE. F-Secure Virus Descriptions : Santy. `http://www.f-secure.com/v-descs/santy_a.shtml`.
6. JUNG, J., PAXSON, V., BERGER, A. W., AND BALAKRISHNAN, H. Fast Portscan Detection Using Sequential Hypothesis Testing. In *Proceedings of the IEEE Symposium on Security and Privacy* (May 9–12, 2004).
7. KIM, H.-A., AND KARP, B. Autograph: Toward Automated Distributed Worm Signature Detection. In *Proceedings of the 13th USENIX Security Symposium* (Aug. 9–13, 2004).
8. SCHECHTER, S. E., JUNG, J., AND BERGER, A. W. Fast Detection of Scanning Worm Infections. In *Proceedings of the Seventh International Symposium on Recent Advances in Intrusion Detection (RAID 2004)* (Sept. 15–17, 2004).
9. SINGH, S., ESTAN, C., VARGHESE, G., AND SAVAGE, S. Automated Worm Fingerprinting. In *Proceedings of the 13th Operating Systems Design and Implementation OSDI* (Dec. 2004).
10. SPAFFORD, E. H. A Failure to Learn from the Past. In *Proceedings of the 19th Annual Computer Security Applications Conference* (Dec. 8–12, 2003), pp. 217–233.
11. STANIFORD, S., PAXSON, V., AND WEAVER, N. How to 0wn the Internet in Your Spare Time. In *Proceedings of the 11th USENIX Security Symposium* (Berkeley, CA, USA, Aug. 5–9 2002), USENIX Association, pp. 149–170.
12. TURKEY, J. W. A survey of sampling from contaminated distributions. In *Contributions to Probability and Statistics* (1960), Stanford University Press.
13. TWYCROSS, J., AND WILLIAMSON, M. M. Implementing and Testing a Virus Throttle. In *Proceedings of the 12th USENIX Security Symposium* (Aug. 4–8, 2003).
14. WALD, A. *Sequential Analysis*. J. Wiley & Sons, New York, 1947.
15. WANG, K., CRETU, G., AND STOLFO, S. J. Anomalous payload-based worm detection and signature generation. In *Proceedings of the Eighth International Symposium on Recent Advances in Intrusion Detection (RAID 2005)* (Sept. 2005).
16. WEAVER, N., PAXSON, V., STANIFORD, S., AND CUNNINGHAM, R. A Taxonomy of Computer Worms. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode* (Oct. 27, 2003), ACM Press, pp. 11–18.
17. WEAVER, N., STANIFORD, S., AND PAXSON, V. Very Fast Containment of Scanning Worms. In *Proceedings of the 13th USENIX Security Symposium* (Aug. 9–13, 2004).
18. WHYTE, D., KRANAKIS, E., AND VAN OORSCHOT, P. DNS-based Detection of Scanning Worms in an Enterprise Network. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'05)* (Feb. 2005).
19. WILLIAMSON, M. M. Throttling Viruses: Restricting propagation to defeat malicious mobile code. In *Proceedings of The 18th Annual Computer Security Applications Conference (ACSAC 2002)* (Dec. 9–13, 2002).
20. WONG, C., BIELSKI, S., STUDER, A., AND WANG, C. Empirical analysis of rate limiting mechanisms. In *Proceedings of the Eighth International Symposium on Recent Advances in Intrusion Detection (RAID 2005)* (Sept. 2005).