Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2006-074 November 10, 2006

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

Jaeyeon Jung, Rodolfo A. Milito, and Vern Paxson

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

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

Jaeyeon Jung jyjung@csail.mit.edu
Rodolfo A. Milito rodolfo@consentry.com
Vern Paxson 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 with TRW [7]. The sequential hypothesis testing methodology enables engineering the detectors to meet false positives and false negatives targets, 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 a pure RBS and at the other end as pure TRW, and extends RBS's power in detecting worms that scan randomly selected IP addresses. Using three traces from two 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, internal Web crawlers, and a network monitoring tool 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 host 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 [4, 10, 14].

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 [4, 14] 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 [13, 17]: 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 [5, 12]. A related technique is the use of *meta-servers*, such as worms that query search engines for likely victims [6]. 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 [4, 10, 18, 19].

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 [7]. Wald's sequential hypothesis testing [15] 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. Such data allow us to assess RBS's efficacy in detecting worms that remain inside an enterprise, rather than just those that manifest in a site's external Internet traffic (a limitation of previous studies). 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 [10] 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 find that when n is small, RBS requires more empirical data to arrive at a detection decision; for example, it requires on average 10.4 first-contact connections when $n = 5$. However, when n is larger, RBS provides accurate and fast detection. On the other hand, we show that a fixed-threshold rate-based scheme will inevitably require more difficult tradeoffs between false positives and false negatives.

§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 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 [20] and implemented a virus throttle that confines a host to sending packets to no more than one new host a second [14]. 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 [4], number of failed first-contact connections [10, 18], and the absence of preceding DNS lookups [19]. 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 [7], 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.* [18] 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.* [10] 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 [8, 11, 16]. 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 slow-propagating (stealthy) worms that are indistinguishable from benign hosts by their traffic behaviors.

3 Data Analysis

We hypothesize that we can bound a benign user'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 those 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`. While ideally we would have such traces from a number of different sites in order to assess the range of behavior normally seen, such traces have to date been unavailable. Indeed, we believe the internal traces to which we have access are unique or nearly so for the research community at present. Thus, we view them as highly valuable, if fundamentally limited, though we need to continually keep in mind the caution that we should not readily generalize from them. (A much larger dataset, `Lab II`, later became available from this same site. We use it in §6 to assess RBS + TRW.)

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 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.

Outgoing connections	
to internal hosts	32,967
to external hosts	16,082
total	49,049
Local hosts	≥ 652

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

For each 10-minute trace, we observe that the number of active internal hosts that initiated any outbound traffic during the observation period varies. The last row in Table 1 shows that the largest number of active internal hosts in a 10-minute trace is 652. (Because each trace was anonymized separately, we are unable to tell how many distinct internal hosts appear across all of the traces.)

We plot the cumulative distribution of per-host fan-out in Figure 1. We see that over 99.5% of hosts contacted fewer than 60 different hosts in 10 minutes, which results in less than 0.1/sec fan-out rate on average. However, the top 0.5% most active hosts greatly stretch out the tail of the distribution. In what follows, we examine those hosts with a high fan-out rate to understand what distinguishes their behavior from that of worm propagation. Then, we find a set of "purely" benign hosts, which we use to develop a model that captures their fan-out rate statistics.

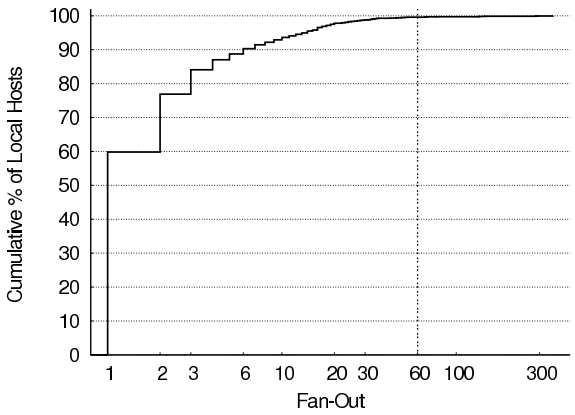


Figure 1: Fan-out distribution of an internal host's outbound network traffic for a 10 minute observation period

3.1 Separating Benign Hosts

Our starting point is the assumption that a host is benign if its fan-out rate is less than 0.1/sec averaged over a 10-minute monitoring period. (Note that Twycross and Williamson [14] use a 1/sec fan-out rate 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. Table 2 shows the average fan-out per each type of scanners detected from the `Lab` dataset. Note that we do not include the mail servers here, as they are not scanners per se, but rather applications that happen in this environment to exhibit high fan-out.

Type	Count	Average fan-out
Internal scanner	5	196.4
Internal crawler	2	65.5

Table 2: Scanners detected from the `Lab` dataset

We exclude the scanners from our subsequent analysis, because including them would greatly skew the fan-out statistics of benign hosts. Given their low number, it is reasonable to presume that sites could maintain white-lists of such hosts to filter out detections of their activities by our algorithm.

3.2 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.

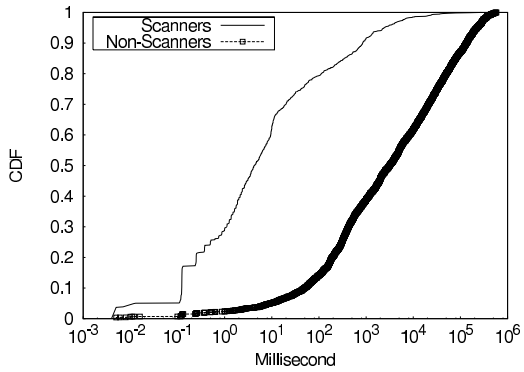


Figure 2: Distribution of first-contact interarrival time, per host

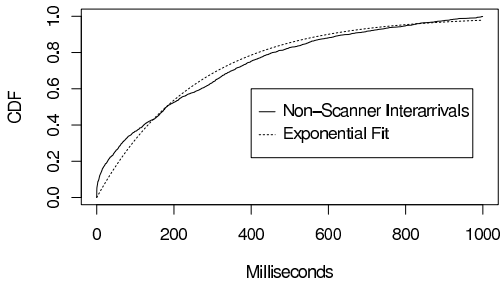


Figure 3: First-contact interarrivals initiated by benign hosts roughly follow an exponential distribution with mean $\mu = 261$ msec.

Figure 2 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), listing two groups, `scanners` and `non-scanners`. We see that scanners have a much shorter average interarrival time (1.1 sec) compared to the non-scanners (39.2 sec). Yet, the average is deceptive because of the uneven distribution of time intervals. Although the average non-scanner 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 3. Here the fit to `non-scanners` uses $\mu = 261$ msec. In comparison, `scanners` on average wait no more than 69 msec between first-contact connections. We note that a scanner could craft its probing scheduling such that its fine-grained scanning behavior matches 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 the previous section we examined network traces and found that benign hosts often initiate more than one first-contact connection request per second, but in such cases we can approximate the interval between these connections with an exponential distribution. 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. 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 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, α, and a target detection rate, β [15]:

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

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

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 $\frac{1-\beta}{1-\alpha}$ [15]. 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 $\frac{1}{1-\alpha} \approx 1$).

There are four parameters to set in order to run RBS. First, α and β give the false positive rate and the detection rate we want to achieve with the detection algorithm. In addition, we need *priors*, λ_1 and λ_0, as the reference fan-out rates. We base their selection on our simple models of the network behavior of scanners versus non-scanning hosts.

Since a host's instantaneous first-contact rate can vary a great deal over time, RBS needs to estimate whether the current behavior provides evidence strong enough to choose one hypothesis against the other. 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 elapsed time, T_{H_1}, below which we arrive at an H_1 (detection) decision:

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

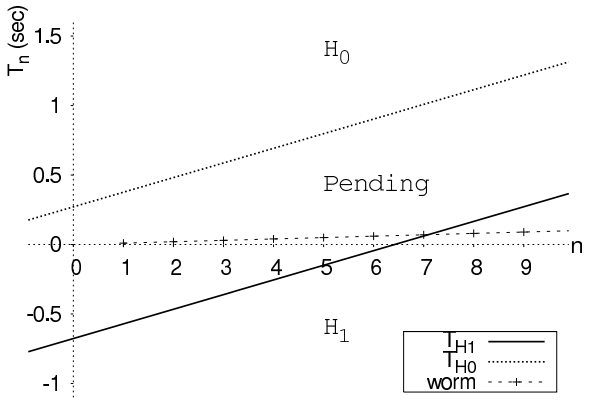
Likewise, we can obtain a threshold elapsed time T_{H_0}, above which we conclude H_0 (benign host):

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

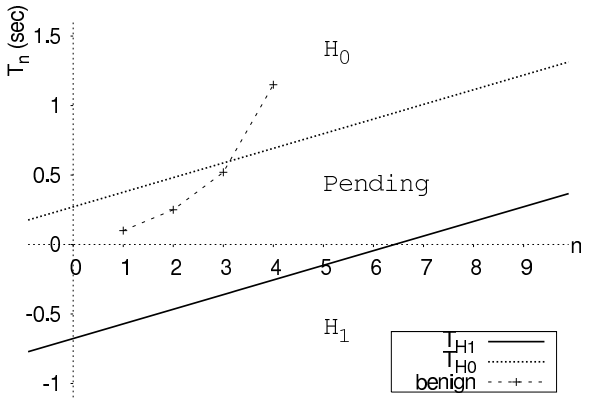
Figure 4 shows how those threshold elapsed times, T_{H_1} and T_{H_0}, partition the area into three decision regions—H_1, H_0, and `Pending`. Figure 4(a) illustrates T_n of a host issuing first-contact connections at 100/second. At the 8^{th} event, T_8 falls below T_{H_1}, which drives the likelihood ratio to reach the H_1 decision. 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. 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 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 [10].

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



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



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

Figure 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

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 $\ln(x) < x - 1$.

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) (6) show that RBS's decision is made based 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 evaluate the performance of RBS in terms of false positives, false negatives, and the detection speed using a trace-driven simulation of the `Lab` dataset. The dataset contains 650 active hosts, including 14 hosts that are legitimate but exhibit network behavior resembling that of fast targeting worms. We then discuss the robustness of RBS to the bursty nature of benign traffic that a naïve fixed threshold based algorithm is unable to capture.

Each line in the `Lab` dataset represents a connection seen by the Bro NIDS [9], sorted by the timestamp of the first packet belonging to the connection. Connection information includes a source address, s, a destination address, d, and a connection start time. For each connection, our trace-driven simulator checks if s has previously accessed d. If not, it updates the likelihood of s being a scanner as described in §4. Figure 5 illustrates the sequence of processes that the simulator takes every time the likelihood ratio is updated.

In addition to the hosts identified in Table 2, by operating RBS we found 9 more hosts whose network activity involved some sort of scanning: 2 more instances of the internal scanner (not found using the previous simple fan-out threshold of 60 destinations over 10 minutes, because their scanning was limited to 7 and 34 destinations,

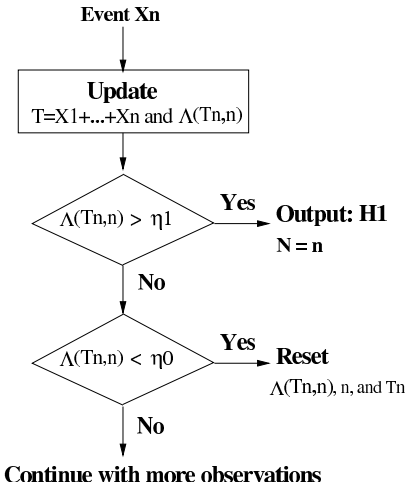


Figure 5: Flow diagram of the real-time detection algorithm

$\lambda_1 =$	$2\lambda_0$	$3\lambda_0$	$4\lambda_0$	$5\lambda_0$	$10\lambda_0$	$15\lambda_0$	$20\lambda_0$	$25\lambda_0$
`scanners` (7)	4	5	6	6	7	7	7	7
`Web crawlers` (2)	2	2	2	2	2	2	2	2
`WhatsUp` (1)	0	1	1	1	1	1	1	1
`iprint` (6)	0	0	1	2	3	3	6	6
Total detection (16)	6	8	10	11	13	13	16	16
False positives	0	0	0	0	0	0	0	0
$\overline{N}\|H_1$	30.2	18.1	13.8	10.4	6.6	5.7	5.2	5.0
Theoretical bound (Hz)	> 5.5	> 7.0	> 8.3	> 9.5	> 15.0	> 19.8	> 24.3	> 28.5

Table 3: Trace-driven simulation results of RBS varying λ_1 when $\lambda_0 = 3.83$ Hz, $\alpha = 10^{-5}$, and $\beta = 0.99$: $\overline{N}|H_1$ represents the average number of first-contact connections initiated by the flagged hosts until detection. The final line gives the theoretical bound on the slowest worm the scanner can detect (Equation (7)).

respectively); a network monitoring system, "WhatsUp" [3], which contacted 14 hosts in 0.4 sec; and 6 instances of an `iprint` printer management client [2] that occasionally accesses all the printers to check availability. These all exhibit greater than 5/sec fan-out rates averaged over a second because of their bursty first-contact connections.

For high accuracy, we set $\beta = 0.99$ (99% target detection rate) and $\alpha = 0.00001$ (0.001% target false alarm rate). Note that α is set very low because the detection algorithm executes at every first-contact connection initiated by a local host, which adds up to a very large number of tests. We choose λ_0 such that $1/\lambda_0$ equals 261 (msec), the mean time interval to visit new destinations of benign hosts as shown in §3.2. However, there is no obvious pick for λ_1 since a worm can choose an arbitrary rate to propagate. If λ_1/λ_0 is close to 1, RBS takes longer to make a decision. It can also miss short bursts; but on the other hand, it can detect slower scanners than for higher λ_1/λ_0 ratios, per Equation (7).

Table 3 shows the simulation results of RBS for the `Lab` dataset as we vary λ_1 as a multiple of $\lambda_0 = 3.83$ Hz. With increasing λ_1, we see that RBS's detection rate increases without incurring false positives. Hosts that RBS often fails to detect include an internal scanner that probed only 7 hosts in a 10 minute trace, and 6 `iprint` hosts that accessed only 10 or fewer other printers, sometimes in two separate 5-connection bursts, which thins out the source's average fan-out rate, making them difficult to detect.

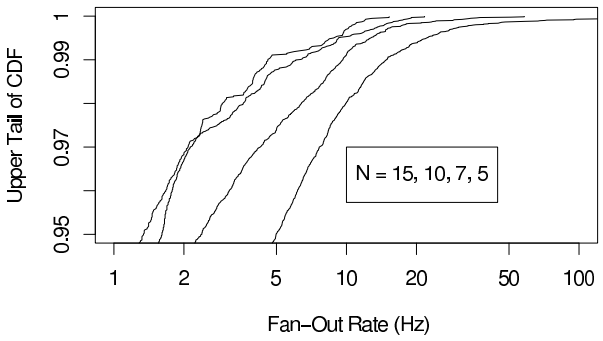


Figure 6: CDF of fan-out rates of non-scanner hosts using a window size of 15, 10, 7 and 5 (from left to right).

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.

5.1 Limitations of Simple Rate-Base Thresholds

An issue to consider is whether we really need RBS's more sophisticated approach, or if a simpler scheme using a fixed-rate threshold suffices. We model such a simpler scheme as one that, upon each connection arrival, compares the fan-out rate, n/T, with a threshold η, alerting if the rate exceeds the threshold. Here, n is the number of first-contact connections from a host and T the elapsed time over which they occurred.

In this section we evaluate such schemes and find that they suffer from either significant false alarms, due to legitimate bursty connections, or significant false negatives. RBS is more robust to such bursts as it demands consistency over a larger number of observations before reaching a decision.

We compute a host's instantaneous fan-out rate as follows. For an outgoing connection initiated by the host at time t_c, we look back in time for the $n-1$ most recent first-contact connections. If the time of the first of these is t_p, then we calculate the fan-out rate as the ratio of $n/T = n/(t_c - t_p)$.

After removing the scanners listed in Table 3, Figure 6 shows the upper tail of the distribution of the fan-out rate of the remaining hosts, as a function of the aggregation window size n. Recall that any detection of these connections constitutes a false positive. So, for example, for windows of size $n = 7$, the 99th percentile occurs right around 10 Hz. Thus, using a window of size 7, to detect scanners that scan as slowly as 10 Hz, we must accept a false positive rate of 1% *per window*. With a window size of 7, this would mean over our dataset the detector would generate 118 false positives. While higher values of n reduce the false positive rate, they also will increase false negatives, such as the bursty scanners discussed in the previous section.

Comparing these curves with the slowest scanners detectable by RBS, per Table 3, we see that RBS gains significant power in both detecting slower or briefer scanners and in avoiding false positives. The essential advantage of RBS over the simpler schemes is that RBS effectively can *adapt* its window n and threshold η, rather than having to use single, fixed values for these.

6 Combined Approach: RBS + TRW

RBS uses *fan-out rate* to differentiate benign traffic from scanners, 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 [7, 18, 10]. TRW [7] 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 [7], 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:

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

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

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

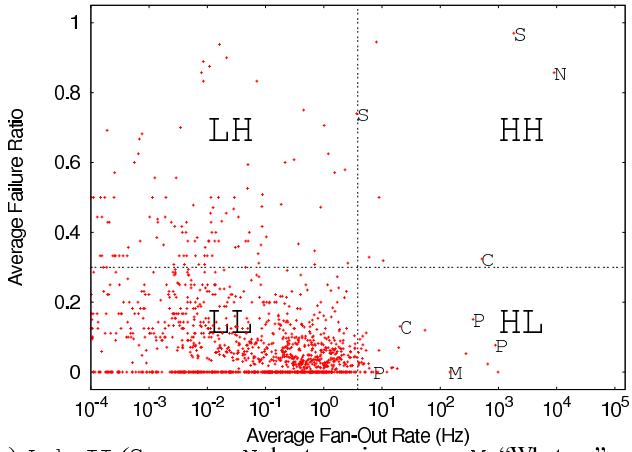
It is interesting to note that $\Lambda(S_n, T_n)$ is just the product of Λ_{TRW} and Λ_{RBS}. Moreover, $\Lambda(S_n, T_n)$ reduces to Λ_{TRW} when there is no difference in fan-out rates between benign and scanning hosts ($\lambda_1 = \lambda_0$). Likewise, $\Lambda(S_n, T_n)$ reduces to Λ_{RBS} when 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.

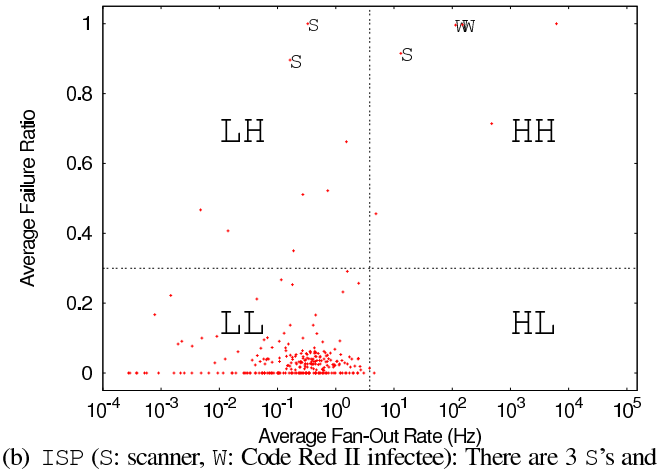
- **Class LH** (low fan-out rate, high failure ratio): Slow-scanning worms or scanners that probe blindly (randomly or sequentially) will likely generate many failures, triggering TRW with a high probability.
- **Class HH** (high fan-out rate, high failure ratio): Fast-scanning worms (e.g., Code Red, 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 [17] 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 [17], 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`, `M`: "Whatsup" monitor, `C`: internal Web crawler, `P`: print management host): There are 2 `S`'s, 1 `N`, 1 `M`, 2 `C`'s and 3`P`'s in the plot



(b) `ISP` (`S`: scanner, `W`: Code Red II infectee): There are 3 `S`'s and 2 `W`'s in the plot. Two `W`'s are almost overlapped with each other

Figure 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 3.83 Hz fan-out rate (λ_0) and 0.3 failure ratio (1-θ_0) as baselines in order to categorize hosts in our trace, where the setting for λ_0 comes from §3 and that for θ_0 from [10]. 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 3.83 Hz fan-out rate and 0.3 failure ratio thresholds.

			Lab II	ISP
Outgoing Connections			796,049	1,402,178
Duration			137 hours	10.5 hours
H O S T S	**HH**	H_1	3	3
		H_0	4	3
	LH	H_1	1	2
		H_0	147	6
	HL	H_1	5	0
		H_0	32	1
	LL	H_1	0	0
		H_0	1195	255
	$\leq$ 5 first-contact connections		2,621	119
	Total	H_1	9	5
		H_0	3,999	384
		Total	4,008	389

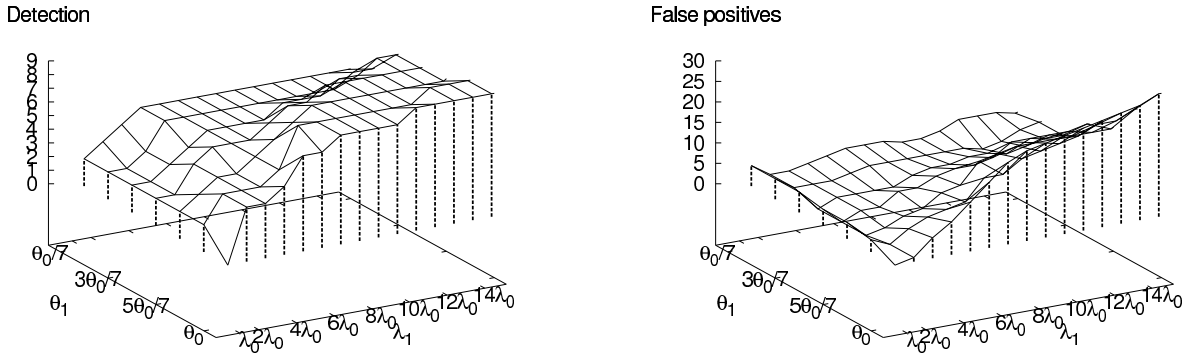
Table 4: Evaluation datasets

Table 4 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 135 one-hour long traces from Dec. 2004 and Jan. 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.).

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**[1] reveals that there are 9 (`Lab II`) and 5 (`ISP`) hosts whose behavior qualifies them as proxies for the topological worms that we aim to detect (H_1) because of their high-fan-out behaviors: For `Lab II`, the 3 **HH** hosts are one internal vulnerability scanner, one host that did a fast `nmap` [1] scan of 7 other hosts, and one internal Web crawlers that occasionally contacted tens of internal Web servers to update search engine databases; 1 **LH** host is another internal vulnerability scanner, whose average fan-out rate was 3.68 (slightly lower than the threshold); 5 **HL** hosts are one internal Web crawler, one "WhatsUp" monitor, and 3 printer management hosts. 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 4 **HH** hosts in the `Lab II` dataset that we classify as benign (H_0) turn out to all be benign NetBIOS clients that often made connection requests to absent hosts. The 3 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 both datasets 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).

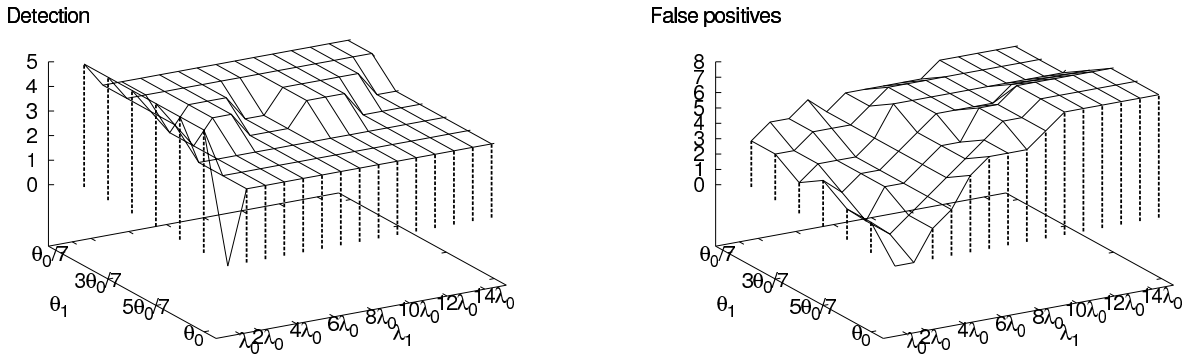
[1] We looked into each host in those three classes for the `ISP` dataset, and the 66 of such hosts for the `Lab II` dataset that generated more than 20 first-contact connections in a one-hour monitoring period.



(a) Detection (out of 9 targets)

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

Figure 8: Simulation results of RBS + TRW for the `Lab II` dataset, varying λ_1 and θ_1



(a) Detection (out of 5 targets)

(b) False alarms (out of 389 hosts)

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

Table 4 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 187 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 14 target hosts while significantly reducing false positives.

We evaluate RBS + TRW by varying λ_1 from λ_0 to $15\lambda_0$, and θ_1 from $\frac{1}{7}\theta_0$ to θ_0. We fix $\lambda_0 = 3.83$ Hz, $\theta_0 = 0.7$, $\beta = 0.99$, and $\alpha = 10^{-5}$. Figures 8 and 9 show the number of detection 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_1 axis), and when $\theta_1 = \theta_0$, to RBS (dashed vertical lines along the λ_0 axis).

Table 5 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, when $\lambda_1 = 11\lambda_0$ and $\theta_1 = \theta_0$, RBS has 0 false negatives for `Lab II`, but misses 2 **LH** scanners in `ISP`. In comparison,

	λ_1	θ_1	Lab II False -	Lab II False +	Lab II $\overline{N}\|H_1$	ISP False -	ISP False +	ISP $\overline{N}\|H_1$
RBS	$11\lambda_0$	$=\theta_0$	0	25	5.6	2	8	6.4
TRW	$=\lambda_0$	$\frac{1}{7}\theta_0$	7	5	18.5	0	3	10.0
RBS + TRW	$10\lambda_0$	$\frac{4}{7}\theta_0$	1	9	6.9	1	6	5.0

Table 5: Evaluation of RBS + TRW vs. RBS and TRW. `Lab II` has 9 scanners and `ISP` has 5 scanners. $\overline{N}|H_1$ represents the average number of first-contact connections originated from the detected hosts upon detection.

when $\lambda_1 = \lambda_0$ and $\theta_1 = \frac{1}{7}\theta_0$, TRW has 0 false negatives for `ISP`, but misses 7 scanners in `Lab II`, including the **HL** hosts, 1 Web crawler and the **LH** nmap scanner.

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 $\lambda_1 = 10\lambda_0$ and $\theta_1 = \frac{4}{7}\theta_0$ misses only two low-profile target hosts (one iprint host from `Lab II` and a slow HTTP scanner from `ISP`) while generating no more than 15 false positives, per Table 5. Had we run RBS and TRW in parallel, we could have eliminated all the false negatives, but at the cost of 41 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 ($\lambda_1 \in [10\lambda_0, 15\lambda_0]$ and $\theta_1 \in [\frac{4}{7}\theta_0, \frac{6}{7}\theta_0]$), 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 (§6), 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.* [18] 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 [21].

Post-detection response: The results in Table 5 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 can 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 [10], 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 [15] 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; for example, it requires on average 10.4 first-contact connections when $n = 5$, but the theoretical bound shows that it can detect any scanners that sustain more than 9.5 first-contact connections per second. In addition, as n grows larger RBS provides accurate and quick detection.

We then presented RBS + TRW, a hybrid of RBS and TRW [7] 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 worm or scanning worm). 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 and a network monitoring tool

that we use as proxies for topological worms. In doing so, it generates less than 1 false alarm per hour.

References

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

[2] Novell:iprint overview. `http://www.novell.com/products/netware/printing/quicklook.html`.

[3] Whatsup gold — the standard for network monitoring systems. `http://www.areawidetech.com/whatsup_gold.htm`.

[4] 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).

[5] 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).

[6] F-SECURE. F-Secure Virus Descriptions : Santy. `http://www.f-secure.com/v-descs/santy_a.shtml`.

[7] 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).

[8] 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).

[9] PAXSON, V. Bro: a system for detecting network intruders in real-time. *Computer Networks (Amsterdam, Netherlands: 1999) 31*, 23–24 (1999), 2435–2463.

[10] 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).

[11] 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).

[12] 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.

[13] STANIFORD, S., PAXSON, V., AND WEAVER, N. How to own 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.

[14] TWYCROSS, J., AND WILLIAMSON, M. M. Implementing and Testing a Virus Throttle. In *Proceedings of the 12th USENIX Security Symposium* (Aug. 4–8, 2003).

[15] WALD, A. *Sequential Analysis*. J. Wiley & Sons, New York, 1947.

[16] 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).

[17] 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.

[18] 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).

[19] 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).

[20] 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).

[21] 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).