Design and Evaluation of a Fast and Robust Worm Detection Algorithm

Tian Bu[†] Aiyou Chen[†] Scott Vander Wiel[†‡] Thomas Woo[†]
[†]Bell Labs, Lucent Technologies [‡]Los Alamos National Laboratory

***Abstract*— Fast spreading worms are a reality, as amply demonstrated by worms such as Slammer, which reached its peak propagation in a matter of minutes. With these kinds of fast spreading worms, the traditional approach of signature-based detection is no longer sufficient. Specifically, these worms can infect all vulnerable hosts well before a signature is available. To counter them, we must devise fast detection algorithms that can detect new worms without signatures as they first begin to appear. We present the design and evaluation of such an algorithm in this paper.**

The key to the algorithm is the identification of certain invariant characteristics of worm propagation. Specifically, we are able to demonstrate using real network traces how worm propagation can perturb the arrival process distribution of unsolicited packets. Our algorithm employs a novel two-step procedure that combines a first stage change point detection with a second stage growth rate inference to confirm the existence of a worm.

To evaluate the algorithm, we have applied it to multi-year network traces that cover many of the major worm outbreaks in recent years, including Slammer, Witty, Nimda and Blaster. In all cases, the new algorithm is able to detect the worm within a very short time, well before significant infection has taken place.

I. Introduction

The release and propagation of the Slammer [1] worm in 2003 was a watershed event in the study of Internet worm epidemiology. It not only demonstrated in an unprecedented way the scale and disruption that is possible in the real world with a relatively compact worm; it also showed the ineffectiveness of current techniques in detecting and countering these new super fast spreading worms. More specifically, in the early phase of Slammer propagation, it doubled in size every 8.5 seconds. It reached a maximum scan rate of of 55M addresses per second and was able to infect more than 90 percent of vulnerable hosts within 10 minutes [1]. Eventually, the spread of Slammer was slowed by its own self-interfering nature and a flawed pseudo-random number generator used in IP address generation. In the end, Slammer served as a wake-up call as it carried no malicious payload and its main damage was in resource (bandwidth and CPU) consumption.

Our defense against fast spreading worms needs to be improved significantly. Worms like Slammer must be countered with extremely fast detection and containment mechanisms. The traditional approach of human-mediated worm detection, dissection and signature development is no longer sufficient. By the time a worm signature can be manually obtained, typically in hours, all vulnerable hosts will be infected. Some studies estimate that the defense must be put in place for a new worm in minutes [2]. There have been some proposals on automating the detection process [3]–[7] and these are discussed in Section II.

In this paper, we propose and study a novel algorithm for detecting new worms, i.e., worms that appear for the first time and are thus without an identified signature. In this case, the detection must rely solely on observing some *invariant* properties of worm propagation. This paper does not solve the followup problem of reacting to a worm once it has been detected. Traditionally, the reaction process involves a human in the loop, and is therefore slow. With fast spreading worms, automated reaction is required. Typically, this involves some form of filtering, which can be based on network headers or deep packet inspection. The reaction can also involve automated generation of a worm signature and this will feed back into the detection stage. Reaction approaches can be very diverse, and are themselves topics of research in their own right. This paper does not study them further.

Basis of Worm Propagation

There are many types of worm, some spread by compromising well-known services and others spread using email as a carrier. In this paper, our interest is only in *scan-based* worms. These are worms that spread themselves by exploiting the presence of certain network service vulnerabilities at the new hosts. Fundamental to all scan-based worm propagation is some form of probing (transmission of packets) from an infected host to a new host, this is called a *scan*. In general without a signature of the worm's payload, it is not possible to determine whether a given service request is legitimate or the result of a worm that is scanning for new host to infect. However, at the edges or enterprise or service provider networks, where worm detection is often performed and where one may have knowledge of valid destination address ranges, one heuristic for detecting worms is to track the rate of probes to unallocated IP addresses because these are considered suspicious. We call such probes *unsolicited* scans in this paper.

Without a signature, a new worm must be identified by observing only its propagation characteristics, that is, the pattern of the scans generated as the worm spreads. To be accurate and robust, the key is to find a "signal" that is *invariant* across different breeds of worms and strong enough to be detected. In this paper, we use real network traces to demonstrate that the arrival process of unsolicited scans can offer such a signal. More specifically, the propagation of a worm can significantly alter the arrival rate of unsolicited

1-4244-0222-0/06/$20.00 (c)2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the Proceedings IEEE Infocom.

scans and exhibit an exponential growth pattern in the early phase, which is consistent with the epidemic model for worm propagation. Details of this characterization are presented in Section III. The core design of our algorithm is directly driven by this characterization and design details are presented in Section IV.

Requirements for Fast Worm Detection

General requirements for a fast worm detection algorithm are discussed below. An algorithm that meets these requirements is given in Section IV and evaluated in Section V.

Fast. There is always a race between propagation and detection. The faster a worm spreads, the faster it must be detection. Detection is useless after most of the vulnerable hosts have been infected. As discussed earlier, some recent worms have done their damage in a matter of minutes. A good detection algorithm should signal at the inception of the worm spread as it hits its exponential growth rate.

Accurate. A major complaint against existing intrusion detection systems is their poor accuracy: they have too many *false positives*, raising an alarm when an attack is not present, and *false negatives*, missing an ongoing attack. It should be obvious that a low false alarm rate is critical. This is even more so for fast propagating worms because the reaction process needs to be automated and then a false alarm could trigger defensive actions that could be exploited to cause damage. That would represent a new level of Denial-of-Service attack.

Robust. A detection algorithm is robust if it works well for various worms with different propagation characteristics under different network conditions. The detection algorithm should automatically adapt and produce desirable performance regardless of the worm's propagation rate or the size of the network. The latter is particularly important because a smaller network tends to produces fewer samples and thus may require a longer amount of time before an alarm can be triggered. This notion of *scale insensitivity* or *scale free* effectiveness is an important one.

Main Contributions

In summary, the main contributions of this paper are:

- Original observations and statistical characterizations on the arrival processes of unsolicited packet and unsolicited scanners
- A novel two-step worm detection algorithm with firm analytical foundation and whose design is motivated by the above observations
- A thorough evaluation of the effectiveness of the new algorithm against the design requirements (mentioned above) using real network traces that cover multiple years and many recent worm outbreaks. Our results demonstrate extremely fast—in seconds and minutes—worm detection for several well-known worms in the past few years including Nimda, Slammer, Blaster and Witty.

The balance of the paper is organized as follows. Section II surveys related work on worm detection. Section III presents observations and statistical characterization of unsolicited packets. A two-step worm detection algorithm and its analytical foundation are described in Section IV. Section V presents extensive results on the application of the new algorithm to real network traces. Section VI briefly describes a number of design and implementation issues and Section VIII gives our conclusions.

II. Related Work

Worm detection has been an active area of research, especially after several major outbreaks in recent years. Many approaches have been proposed. The following are most relevant to our work.

Singh et al. [8] and Gu et al. [4] both proposed to detect a worm by monitoring correlation in the content (headers or payload) of incoming and outgoing packets. However, the correlation may not be reliable. Some worms have started to randomize portions of their payloads [1]. To avoid being stopped by signature-based protection mechanisms, future worms will likely strive to have more and more random content in their payloads. For example, packet header correlation can take advantage of the fact that a unique destination port was used by an attacking worm. Unfortunately, Worm Witty may send to any destination port. A detection algorithm based on counting victims is proposed in [5]; it looks for increases in the rate of newly infected outside hosts. In [6], [7] a proposal is made to contain worms based on the observation that scanning worms cause high failed connection ratios. This technique requires keeping track of a large number of connection states. Our algorithm is not based on connection state and is thus more scalable and robust to spoofed address flooding. In addition, triggering on connection failures requires waiting for a preset timeout value to infer that a failure has occurred. A fast worm could grow very large before enough failed connections are identified.

Reference Zou et al [3] is the most closely related work. They are the first to propose the idea of detecting a worm by identifying an exponential growth trend at its early stage. However, their estimate of the exponential rate is based on a counting process for a given time interval. It is not obvious how to pick an interval that is appropriate for a variety of worms both fast and slow. In addition, the counting process represents a reduction of information from the arrival process and this information loss could make some worms too difficult to detect effectively with their approach. We show some examples using real traces in Section III.

Several authors analyze illegitimate scans due to worm propagation [1], [9], [10]. We use them to verify that our local view of network traffic is consistent with the views that other networks present.

Many tools have been created to monitor the Internet for illegitimate traffic. Symantec Corp. has an "early warning solution" [11] that collects IDS and firewall attack data from the security systems of thousands of partners to keep track of the latest attacks. The SANS Institute set up the "Internet Storm Center" [12] for gathering log data from intrusion detection sensors around the world. Moore et al. [13] present

the concept of distributed "network telescopes" that monitor dark (unused) IP addresses to observe security incidents in the Internet. Using a network of honey-pots to identify attacks and gather information is proposed in [14], [15]. The unsolicited packets gathered by these networks can be used as input to our worm detection algorithms.

Wang et al. [16] use a CUSUM for detection of SYN flood DoS attacks by tracking increases in incomplete SYN/FIN pairs. Detecting worms is more challenging than detecting DoS attacks because we look not just for a change, but for specific types of changes–those with the exponential increases associated with worms. For this purpose we utilize both CUSUM signals and a second stage exponential detector.

III. Characterizing Unsolicited Traffic

Bell Labs has collected traffic traces from the Internet gateway in Murray Hill, NJ and these have been used to shape the development of the worm detection algorithms. Performance evaluations are also based on these traces. This section describes the traces and several properties of the network traffic that are relevant to the detection algorithm.

A. Unsolicited traffic collection

The Bell Labs trace collector resides at the corporate firewall and records headers of all packets that reach it from either inside or outside of the corporate network. Bell Labs has two $/16$ subnets behind its firewall with less than 10% of the IP address space occupied and the remaining 90% unassigned. These unused addresses are used as a network telescope as defined in [13]. All packets sent to the unused portion of the address space are considered to be *unsolicited.* Packet header traces have been collected for about six years at this location with a small number of few missing days due to both technical and human factors.

Scan packets arriving from propagating worms are only a portion of the unsolicited traffic that hits the firewall. Our goal is to identify the outbreak of new worms by monitoring the arrival process of unsolicited packets or the arrival of new sources that send these packets. For worms that scan random IP addresses, successful detection requires monitoring a network of sufficient size [13]. The calculations in [13] indicate that the two $/16$ subnets at Bell Labs are large enough to detect even slow worms. Designing an effective detection algorithm, however, requires understanding the arrival process of unsolicited traffic. The remainder of this section presents some observations on inter-arrival times of unsolicited packets.

B. Inter-Arrival times of unsolicited packets and sources

Two different traffic streams are investigated in this section—the stream of all unsolicited packets and the stream of unsolicited packets from external sources that have not been observed in the previous t seconds. We call the latter stream a t-sample. Typically a large portion of the unsolicited packets are generated by a small fraction of observed sources in the absence of a fresh worm outbreak [17]. A t-sample, however, will not be dominated by a small number of sources and thus has the advantage that any erratic behavior of the top scanners will have little affect on the t-sample packet stream. As $t \to \infty$ it is obvious that the t-sample will record each unsolicited source exact once. Therefore, we refer a t-sample as the *scanner* arrival process.

Figure 1 plots one-second counts of scanning traffic in 12-hour periods containing the outbreaks of four worms—Nimda, Slammer, Blaster and Witty—with dates ranging from September 2001 to March 2004. Two panels are shown for each worm. The top panel, labeled "Scans", shows the number of unsolicited packets arriving from external sources to the firewall. A single source could contribute one or many packets to any of the counts. The bottom panel, labeled "Scanners", shows the number of unsolicited arrivals of a t-sample with $t = 5$. The black curves in each plot are 10 minute moving averages.

For each worm, the "Scan" counts are much more erratic than the corresponding "Scanner" counts. This makes sense because a single external machine can momentarily flood the firewall with unsolicited packets and drastically affect the Scan count but not the Scanner count. If external sources do not coordinate the *Scanner* counts will tend to behave like Poisson random variables because each source acts independently and has a small chance of hitting the firewall in a given one-second interval. Poisson variables have equal mean and variance. However, square-root scaling is used in Figure 1; the square root of a Poisson random variable has a standard deviation of about 1/2, independent of the mean. Thus, the width of the gray bands of Scanner counts is approximately 3 (six standard deviations) at any time of the night or day. This is not true for the *Scan* counts.

One difficulty with using counts at fixed intervals, such as the one-second counts of Figure 1, is that the appropriate count interval depends on both the baseline rate of new scanners and on the speed needed to detect new worms. For example, in the Blaster plot between 6 and 8 PM, many of the counts are equal to zero, one or two; if the baseline rate of new scanners were a factor of 10 smaller, then one-second count windows would be too wide. On the other hand, when Slammer arrives, the impact is sudden and the worm should be detected very quickly. Obviously using a count interval that is too long could slow down detection. For these reasons we develop a detection scheme based, not on counts, but on the inter-arrival times of the new scanner sources. Using inter-arrival times results in a scheme that is inherently self-scaling, in terms of both the baseline rate of new scanner arrivals and the speed of detection when a new worm arrives.

The above reasoning for scanner counts to behave like Poisson random variables also suggests that scanner arrival times come from a Poisson process. During normal time periods, the process should be relatively stationary, having a nearly constant mean. Slow trends or the outbreak of a worm would affect the mean, but we expect that a non-stationary Poisson process will be adequate to model these effects. In Section V we demonstrate, using real traces, that the early stage of worm propagation looks very much like a

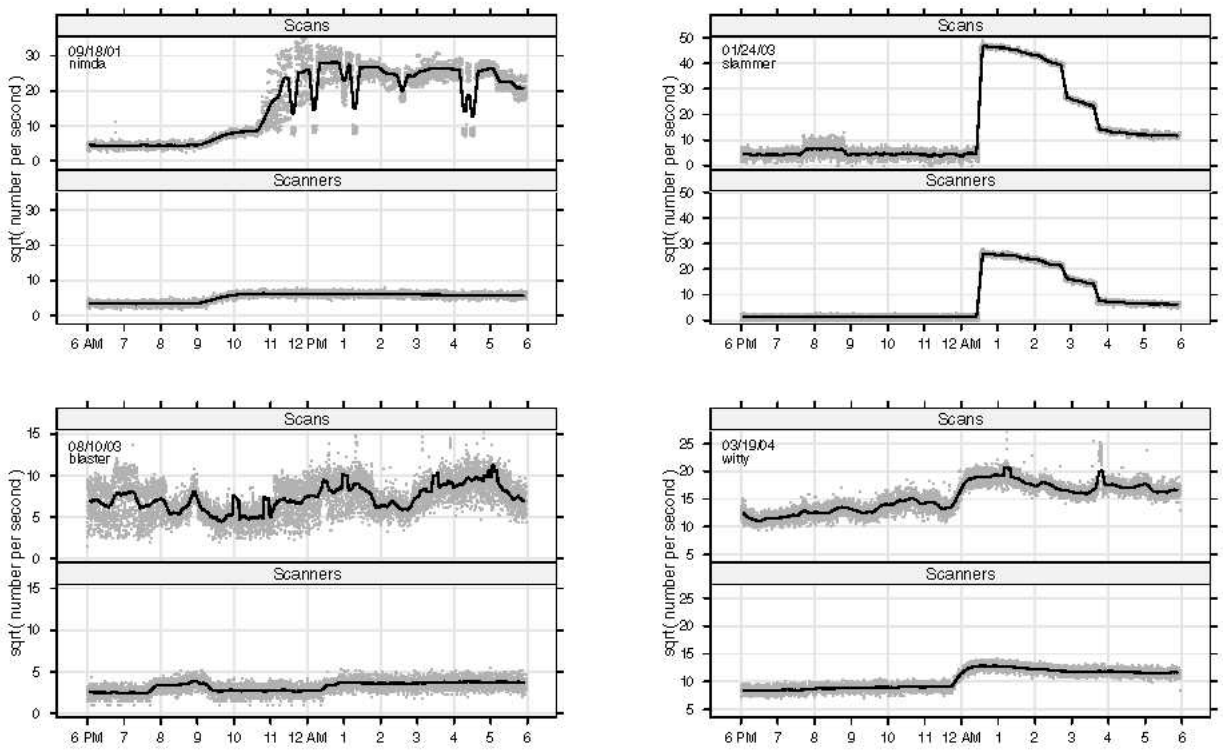

Fig. 1. One-second counts of scan traffic in 12 hour time-frames surrounding the arrivals of four different worms. *Scans* are unsolicited packets arriving at the firewall. *Scanners* are source IP addresses that generate unsolicited packets. The dark lines are 10 minute moving averages. Scanner counts are far cleaner than Scan counts.

non-stationary Poisson process with exponentially increasing rate.

If the stationary model is appropriate then scanner inter-arrival times should be exponentially distributed with constant mean over long periods of time, except during worm outbreaks or other events affecting a large number of external sources. Figure 2 shows probability plots of inter-arrival times in the quiet periods preceding the four worm outbreaks shown in Figure 1. The figure shows a log-log plot of sorted inter-arrival times against quantiles of the exponential distribution, labeled on a cumulative percentage scale. The 45 degree grid lines correspond to exponential distributions with different means. The scanner inter-arrival times clearly pass this check of the Poisson process model. Additional probability plots (not shown) over many different time periods, both before and after the worm outbreaks, are similar.

The worm detection algorithm proposed in [3] uses a Kalman filter to estimate the growth of infected hosts based on arrival counts in fixed time intervals. As discussed above, these counts will have Poisson distributions, and not the *constant variance* Gaussian distributions assumed by the Kalman filter. The new detection method developed in the following section is based on a Poisson process model for scanner inter-arrival times. It avoids using count data in fixed time intervals and thereby achieves the goal of being self-scaling—it can operate over a large range of baseline scanner arrival rates without having to be re-tuned.

To summarize, a t-sample records unsolicited packet arrivals from the same source at most once every t-seconds. Whereas the process of *all* unsolicited packet arrivals is erratic, a t-sample of scanners is nearly a stationary Poisson process outside of the brief periods following the release of a new worm. During a worm outbreak the scanner arrival rate is expected to increase exponentially. We have evaluated samples for various values of $t \geq 1$ and the results are similar to those presented for $t = 5$. In the rest of the paper we only consider t-samples and not the process of all unsolicited packets.

IV. Detection Algorithm: Design and Foundation

The previous section demonstrated, using packet header traces, that t-sample scanners have exponential inter-arrival times with locally constant mean in the absence of a worm outbreak. Upon the arrival of a new worm and during its early phase of propagation, we will model the scanner arrivals as a Poisson process with a non-stationary rate.

The worm detection algorithm developed in this section is strongly based on these observations. The algorithm follows a two-stage procedure. The first stage employs a change detection algorithm to detect an increase in the rate of unsolicited packet arrivals in a t-sample. When an increase is detected a second stage is launched in an attempt to verify that the arrival rate is, indeed, increasing *exponentially* as would be expected from a worm outbreak. The first stage uses a CUSUM procedure to detect an increase in the scanner arrival

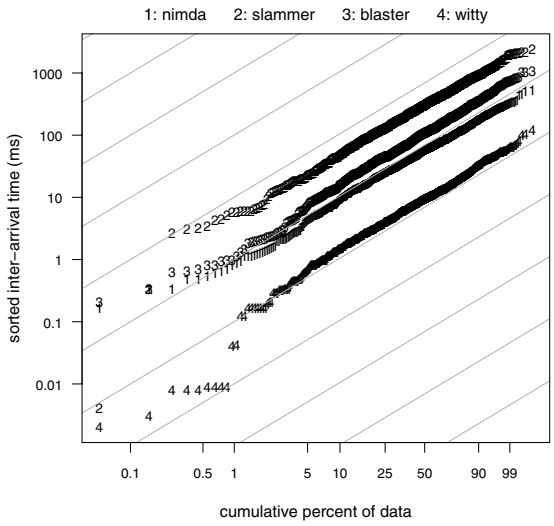

Fig. 2. Exponential probability plots of 1000 consecutive inter-arrival times of new scanners IP addresses. The exponential distribution is a good fit in each case. Diagonal grid lines correspond to exact exponential distributions.

rate. The second stage uses a Maximum Likelihood Estimation (MLE) procedure to fit a non-stationary Poisson process with exponentially increasing rate function. The second stage throws a worm alarm if the exponential rate estimate differs significantly from zero with a high level of confidence.

In Section IV-A, we explain how CUSUM is used for change detection. Section IV-B provides details on maximum likelihood inference of the worm propagation model. The two stages are combined in Section IV-C where the complete worm detection algorithm is presented.

A. Change Detection using CUSUM

Let T_n denote the arrival time of the n-th unsolicited packet in a t-sample, and let $X_n = T_n - T_{n-1}$ be the inter-arrival time where $T_0 = 0$. We assume that the inter-arrival times $\{X_n : 1 \leq n < n_w\}$ before the worm starts are i.i.d. exponential with mean μ, where T_{n_w} represents the time of the first worm scan. After a worm arrives, the inter-arrival times $\{X_n : n_w \leq n < \infty\}$ have a decreasing mean, which is less than μ. This shift in the distribution of inter-arrival times may be considered a change point in statistical terms and CUSUM schemes are optimal (in a sense made precise in [18]) for detecting changes from one distribution to another.

The CUSUM scheme can be applied as follows. Let $S_0 = 0$ and define

$$S_n \quad = \quad max(0, S_{n-1} + \mu - X_n - p\mu), \quad n = 1, 2, \cdots$$

where p depends on the expected drop in mean inter-arrival times due to a worm. Typically $p\mu$ is set to about half the size of drop in mean inter-arrival time that is crucial to detect quickly. A change of inter-arrival time is signaled whenever S_n exceeds a certain threshold h. The intuition behind CUSUM is that if the mean of X_n shifts from μ to something smaller than $\mu - p\mu$ at sample n_w then S_n will tend to accumulate positive increments after n_w and thus eventually cross the threshold h and signal a change. In practice, μ is not known; but an estimate, such as an Exponentially Weighted Moving Average (EWMA) can be used in its place. The details of the EWMA are provided in Section IV-C.

Choosing the threshold parameter, h, requires trading off between detection delay (i.e., sensitivity) and the false detection rate. Small values of h provide quick detection when changes are present but also give more false alarms. In our case, the CUSUM is not used to directly trigger alarms but only as a first stage toward detection. We prefer small h values to reduce the first stage detection delay and we rely on the second stage to separate out real worms from false alarms. The threshold h can be calculated from the expected time between false alarms, known as the Average Run Length (ARL) in quality control literature. Details for computing an appropriate h are provided in Appendix B.

Although worm arrivals always reduce mean inter-arrival times in t-samples, not all reductions are due to worms. Our experience with real traces suggests the arrival process of unsolicited packets in a t-sample is well-modeled as stationary over periods of many hours but shows slow drift over longer periods and is sometimes impacted by sudden non-worm events such as denial-of-service attacks. These effects can often cause S_n to exceed h incorrectly, contribute to false alarms and make the CUSUM less effective. It is possible to use a large p or h to filter out many changes in the background, but this approach may miss slow worms and delay the detection of fast worms. As a remedy, we design a second stage detection algorithm to filter out non-worm events; the second stage raises an alarm only when inter-arrival times show a definitive exponential pattern of decrease. Whereas CUSUM charts in quality control are usually intended to detect all process changes, the present application is more discriminating; alarms are desired only for process changes that can be positively identified as having the behavior of a worm.

B. Maximum Likelihood Inference of Worm Propagation Rate

A CUSUM signal triggers the second stage detection in which a worm propagation model is estimated. However, if a new worm outbreak is in progress, it is probable that some time has elapsed between the outbreak and the CUSUM signal. Let T_{n_0} denote the most recent time (prior to the current signal) when the CUSUM transitioned from a value of 0 to a positive value. If a worm exists, its arrival is most likely earlier than T_{n_0} (Case 1). It *is* possible for a worm to arrive between T_{n_0} and the CUSUM signaling time (Case 2), but this happens with small probability and the lag from worm to CUSUM signal is most likely small. The third case is that no worm exists (Case 3); this is the usual situation. We first focus on the statistical estimation of the worm propagation model based on Case 1; this also includes Case 3 and serves as a good approximation for Case 2, which we investigate further in Section VI-A.

Scanner arrivals in a t-sample before a worm outbreak are well-modeled as a Poisson process with rate $b(t)$ that changes slowly with time. Scanner that arise from a fresh worm outbreak can be modeled as a non-stationary Poisson process with rate

$$\lambda(t) = ae^{r(t-t_w)}I(t \geq t_w)$$

where t_w is the time when the first worm scan arrives; a is the expected number of worm scanner arrivals in the first second; r is the exponential propagation rate; and $I(x)$ is an indicator function having value 1 when x is true and 0 otherwise. We assume that the background scanners and the ones caused by a new worm are independent.[1] The superposition of background and worm scanners is thus modeled as a non-stationary Poisson process with rate

$$\lambda(t) \quad = \quad b(t) + ae^{r(t-t_w)}I(t \geq t_w).$$

Because the background traffic is approximately stationary, its rate $b(t)$ can be estimated easily using local averaging. Propagation characteristics are described by the parameters a and r that depend on the efficiency of the worm and the size of the network being monitored. We show later in this section that although a is not identifiable (i.e., cannot be estimated statistically) when t_w is unknown, the exponential rate r *is* identifiable. A worm alarm is triggered when the data indicates that r is *significantly* higher than a small tolerable rate r_0 with high statistical *confidence*.

For simplicity, assume that the worm starts at 0 (i.e., $t_w = 0$), unsolicited scanners arrive at times $T_1, T_2, \ldots$ according to a Poisson process with rate $\lambda(t) = b + ae^{rt}$, $t \geq 0$, and the corresponding CUSUM sequence $S_1, S_2, \ldots$ remains below the threshold h until some arrival T_{n_0} ($n_0 \geq 1$) when the CUSUM exceeds h and therefore signals. Define $\overline{T}_j = T_{n_0+j} - T_{n_0}$ for $j = 1, 2, \ldots, n$, where $\overline{T}_n$ is the current arrival relative to the signaling time T_{n_0}. Note that we can only observe $\overline{T}_1, \ldots, \overline{T}_n$ and not the complete stream of arrivals $T_1, \ldots T_{n_0}, T_{n_0+1}, \ldots, T_{n_0+n}$ because the worm outbreak time $t_w = 0$ is not generally known.

Thus any estimators of a and r must be based on $(\overline{T}_1, \ldots, \overline{T}_n)$ whose distribution depends on the unknowns n_0 and T_{n_0}. The following Theorem (proved in Appendix A) and its Corollary demonstrate that the r can be estimated from the $\overline{T}_j$, but a cannot.

Theorem 1. *Let $T_1, T_2, \ldots$ denote consecutive arrival times from a Poisson process with positive rate $\lambda(t) = b + ae^{rt}$ beginning at $t = 0$. Define $\overline{T}_j = T_{n_0+j} - T_{n_0}$ for $j = 1, 2, \ldots$ and for some $n_0 \geq 1$. Then given $T_{n_0} = t_0$, the relative times $\overline{T}_1, \overline{T}_2, \ldots$ are arrivals from a Poisson process with rate $\overline{\lambda}(t) = b + \overline{a}e^{rt}$, $t \geq 0$, where $\overline{a} = ae^{rt_0}$.* □

Corollary 1. *Under the conditions of Theorem 1 and assuming that $a > 0$, the parameters $\overline{a}, b$ and r are identified by the data $(\overline{T}_1, \ldots, \overline{T}_n)$ for $n \geq 3$ but the parameter a is not identified unless t_0 is known.* □

[1]This is reasonable in the early stages but propagation can eventually congest the network and cause normal traffic to back off in response.

The exception $a = 0$ corresponds to no worm and in this case the propagation rate r has no meaning. Fortunately, for the purpose of worm detection, r is the most interesting parameter and it can be estimated by maximum likelihood inference as discussed next.

Let $\overline{\Lambda}(t) = \int_0^t \overline{\lambda}(s)ds$. Then the normalized arrival times $\overline{\Lambda}(\overline{T}_1), \overline{\Lambda}(\overline{T}_2), \ldots$ follow a stationary Poisson process with rate 1 [19]. Let $l_n(r, \overline{a}) = \log p(\overline{T}_1, \ldots, \overline{T}_n | T_{n_0} = t_0)$ be the log-likelihood function for the $\overline{T}_j$'s conditional on T_{n_0}. By the density transformation formula [20],

$$\begin{aligned} l_n(r, \overline{a}) &= \sum_{j=1}^{n} \log \overline{\lambda}(\overline{T}_j) - \overline{\Lambda}(\overline{T}_n) \\ &= \sum_{j=1}^{n} \log(b + \overline{a}e^{r\overline{T}_j}) - \{b\overline{T}_n + \frac{\overline{a}}{r}(e^{r\overline{T}_n} - 1)\} \end{aligned}$$

The maximum likelihood estimates (MLE) are defined as

$$(\hat{r}, \hat{\overline{a}}) \quad = \quad \arg\max l_n(r, \overline{a}). \tag{1}$$

Let $\theta = (r, \overline{a})^T$ and $\hat{\theta} = (\hat{r}, \hat{\overline{a}})^T$. Denote $l_n(\theta) = l_n(r, \overline{a})$. Then the MLE $\hat{\theta}$ has good properties as summarized in Theorem 2 below.

Theorem 2. *Under the conditions of Theorem 1, if θ is bounded, then as $n \to \infty$,*

$$\hat{\theta} \quad \to \quad \theta,$$

in probability and

$$\sqrt{n}(\hat{\theta} - \theta) \quad \to \quad \mathcal{N}(0, \mathbf{I}(\theta)^{-1}),$$

in distribution where $\mathbf{I}(\theta)$ is the information matrix,

$$\mathbf{I}(\theta) \quad = \quad limit_{n\to\infty} - E[\frac{1}{n}\frac{\partial^2}{\partial\theta\partial\theta^T} l_n(\theta)],$$

and can be estimated consistently by

$$\hat{\mathbf{I}} \quad = \quad -\frac{1}{n}\frac{\partial^2}{\partial\theta\partial\theta^T} l_n(\hat{\theta}). \tag{2}$$

□

An explicit expression for $\hat{I}$ is straightforward to derive but omitted here. See Theorem VI.1.2 of [19] for a formal proof of the theorem.

The MLE $\hat{r}$ and its estimated asymptotic variance are used repeatedly in the second stage to test whether r is significantly positive. In particular, we test $r > r_0$ against $r \leq r_0$, where r_0 (say 0.0001) is the maximal rate that can be ignored. Let $se(\hat{r})$ be the asymptotic standard error of $\hat{r}$, that is,

$$se(\hat{r}) \quad = \quad \sqrt{[\hat{\mathbf{I}}^{-1}]_{11}/n}. \tag{3}$$

Since $Z_n \equiv (\hat{r} - r_0)/se(\hat{r})$ is asymptotically normal distributed with mean 0 and variance 1 [20] under the null hypothesis $r = r_0$, the second stage declares a worm outbreak when the $Z_n > q_c$ where q_c is a threshold such as the 99.99 percentile of the standard Normal distribution. For example $q_c = 3.8$ is the 99.99% quantile of the Normal distribution.

WormDetection(T)

1. $S_0 = 0$;
2 Initialize μ with the median of the first $c_0 = 100$ observed inter-arrival time divided by $\log(2)$
3. For each new arrival T_i, $X_i = T_i - T_{i-1}$
4. $S_i = \max(0, S_{i-1} + \mu - X_i - p\mu)$
5. $\mu = (1 - w) \cdot \mu + w \cdot X_i$
6. If $S_i > 0$
7. If $S_{i-1} = 0$
8. $j = 0; n_0 = i$; $S_{max} = S_i$
9. Else
10. $j = j + 1$; $S_{max} = max(S_{max}, S_i)$;
11. If $S_i < 0.8 \cdot S_{max}$, $S_i = 0$
12. If $S_i > h$
13. Estimate MLE of $\hat{r}$ and $se(\hat{r})$
14. from $T_{n_0} \ldots T_{n_0+j}$
15. If $(\hat{r} - r_0) > q_c \cdot se(\hat{r})$
16. Raise an alarm;
17. Skip estimate until CUSUM drop to zero;
18. EndFor

Fig. 3. *Worm detection algorithm*

C. The Complete Worm Detection Algorithm

Figure 3 shows the complete worm detection algorithm that operates a stream of scanner arrival times from a t-sample of unsolicited packets. After reviewing a number of implementation details the algorithm is explained line by line below.

In most CUSUM monitoring applications, the CUSUM statistic is reset to zero after a signal is triggered. In our scheme, however, a large CUSUM is required for the second stage to operate so the CUSUM is not reset immediately upon crossing the threshold h but the reset occurs only after a substantial downward trend is seen following the trigger. The algorithm identifies a downtrend a case in which the current CUSUM value is less than 80% of the maximum value recorded since the previous reset.

Although scanner arrivals, for the most part, resemble a locally stationary Poisson process, outliers do occasionally occur in our traces. These are cases where the inter-arrival time between scanners is abnormally large for one reason or another. These outliers never trigger a false alarm because the MLE does not yield a large r in the second stage. However, the outliers can easily lead to a CUSUM signal and thus needlessly trigger the MLE computations in the second stage. To reduce the impact of outliers, we implement the following random tail-draw technique proposed in [21]. Let μ_{n-1} be the most recent exponentially weighted moving average (EWMA) estimate of $E(X_n)$. If X_n lies outside of the 0.01% and 99.99% percentiles of the exponential(μ_{n-1}) distribution, then it is replaced with a random draw $\tilde{X}_n$ from the corresponding of the distribution for the purpose of calculating S_n.

Line by line, the algorithm proceeds as follows. Lines 1 and 2 initialize the CUSUM and an EWMA estimate of the mean inter-arrival time. Starting the EWMA based on the median of an initial sample provides robustness against outliers. Dividing the median by $log(2)$ produces an estimate of the mean. For each new unsolicited scanner packet, Line 4 computes the current CUSUM and Line 5 the current EWMA. No further action is required if the CUSUM is zero. The EWMA parameter w determines the depth of the memory and the relative weight between the current and previous data. Although there is no general rule for the optimal choice of w, in our experiments performance of the algorithm is similar for various values of w between 10^{-4} to 10^{-7}. Whenever the CUSUM becomes positive, Lines 7 and 8 initialize indices used to record the transition and track the local maximum: j is used to track the number of consecutive positive CUSUM's and S_{max} is the local maximum. If the CUSUM remains positive on subsequent steps then Line 10 updates j and S_{max} and Line 11 resets the CUSUM to zero if a downtrend is recognized with respect to the local maximum. Line 12 triggers estimation of the propagation rate in Lines 13 and 14 if the CUSUM has become large. Lines 15 through 17 test whether the data suggest a significantly large propagation rate with high confidence. If so, the alarm is raised until such time as the CUSUM is reset to zero again.

V. Evaluation using Real Traces

This section presents results of extensive evaluations of the algorithm using multi-year network traces that cover many of the well-known worm outbreaks in the last few years. For brevity, we selected four of the most well-known worms in this time period—Slammer, Witty, Nimda and Blaster—and provide snapshots of the operation of our algorithm during the periods surrounding the worm outbreaks. We would have liked to include the Code Red worm, but unfortunately traces for those days are missing.

In all cases the algorithm is able to detect the worm outbreak with a short delay relatively to the time it takes for the worms to infect all vulnerable hosts. These worms are quite diverse with respect to the method of exploitation, the vulnerable population size and the propagation rate. This diversity demonstrates the effectiveness of our algorithm across different breeds of worms with different characteristics.

A set of three plots for each worm shows performance of the algorithm. The first plot shows packet counts before, during and after the worm outbreak with reference lines to indicate outbreak and detection times. The second plot shows the CUSUM statistic growing to trigger the rate estimation stage of the algorithm. Finally, a QQ-plot is shown as a diagnostic to judge whether the model of exponential growth adequately describes the outbreak. In all experiments, we used $w = 10^{-5}$, $p = 1/32$, $q_c = 3.8$, and h selected to achieve the Average Running Length $ARL = 1000$ seconds (see Appendix B). We have experimented with other reasonable choices of these parameters and seen little variation in the results.

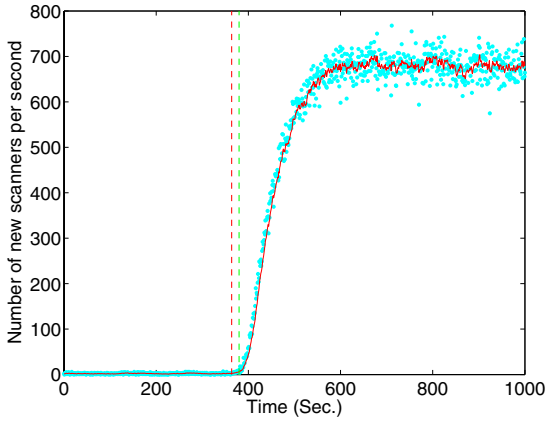

Fig. 4. Slammer is detected in 16 seconds

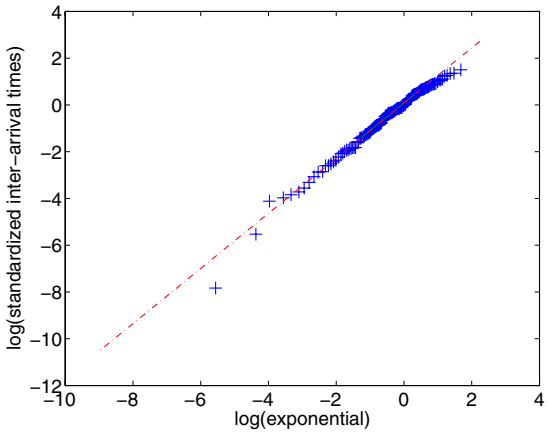

Fig. 6. Model fitting for Slammer: outlier less than 2%

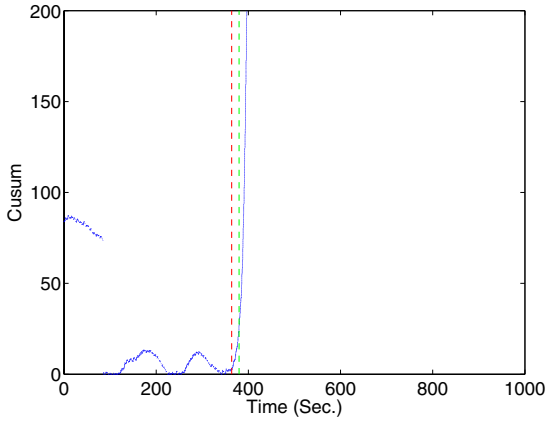

Fig. 5. CUSUM turns positive in 2 seconds after first Slammer scan

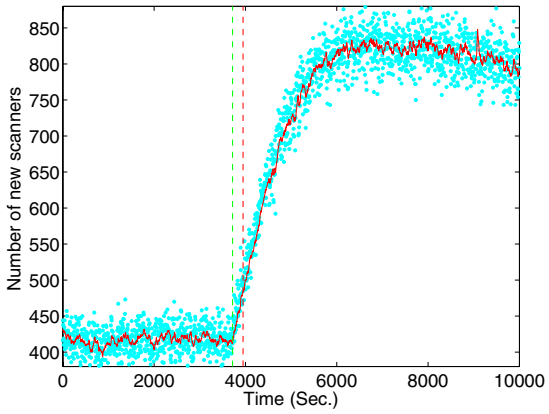

Fig. 7. Witty is detected in 230 seconds

A. *Slammer*

Figure 4 plots the number of scanners arriving at the Bell Labs firewall every second observed 1,000 seconds surrounding the outbreak of Slammer. The first (left) dashed vertical line marks the arrival the first Slammer scan at Bell Labs and the second (right) dashed vertical line marks when the worm detector signals the outbreak. The average number of unsolicited packets is about 2.5 per second before the first worm scan arrives at time 364 seconds. The alarm is raised at just 16 seconds after the initial Slammer scan and at the time the scanners rate has increased to about 6.5 per second. Scans from Slammer peak at about 600 seconds when almost all vulnerable hosts world-wide have become infected.

The algorithm was able to give a warning in as little as 6.7% of time it took for Slammer to infect all hosts. In the Bell Labs trace only 60 hosts had been affected before Slammer was detected whereas a total of 72,516 were infected in total.

Figure 5 plots the CUSUM statistic during the same period of time. The CUSUM turns positive within 2 seconds after the first Slammer scan arrival. Then worm-rate estimation operates for an additional 14 seconds before the data can confidently demonstrate a significantly large exponential rate of growth. Before the Slammer outbreak, the CUSUM statistic grew to large and significant values twice but no alarm was raised because the second stage MLE inference did not demonstrate a significant exponential propagation rate.

Figure 6 is a QQ-plot intended to help evaluate how well the estimated non-stationary Poisson model fits the data. As described in Appendix C, standardized inter-arrival times are computed using the estimated model of exponential growth The standardized inter-arrival times are expected to be exponential(1) random variables which is the reference distribution used on the horizontal axis. A perfect fit between the model and the data would be seen if all points plotted on the line $y = x$. The actual values are very close to the ideal other than a few points in the left tail that account for less than 2% of the data. This suggests the estimated Poisson model with exponential growth describes the scanner traffic fairly accurately.

B. *Witty*

Figure 7 shows counts of scanners arriving at Bell Labs in five second interval for 100,000 seconds surrounding the outbreak of Witty. Five-second intervals are used because Witty spreads much more slowly than Slammer where one second intervals were more appropriate. The first Witty scan

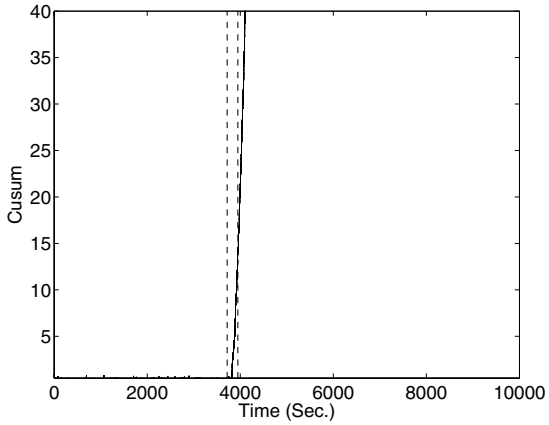

Fig. 8. CUSUM turns positive in 79 seconds after first Witty scan

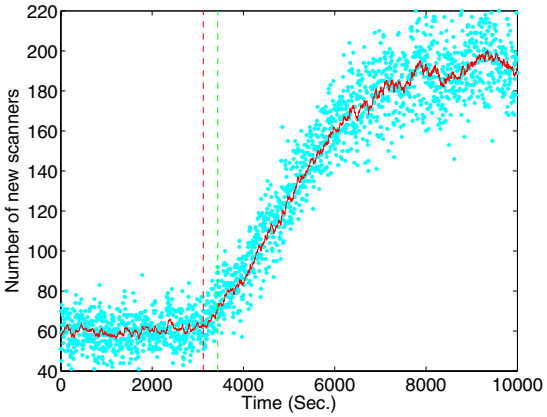

Fig. 10. Nimda is detected in 316 seconds

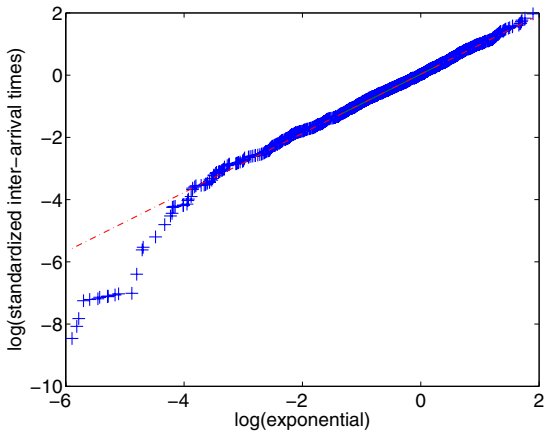

Fig. 9. Model fitting for Witty: outlier less than 3%

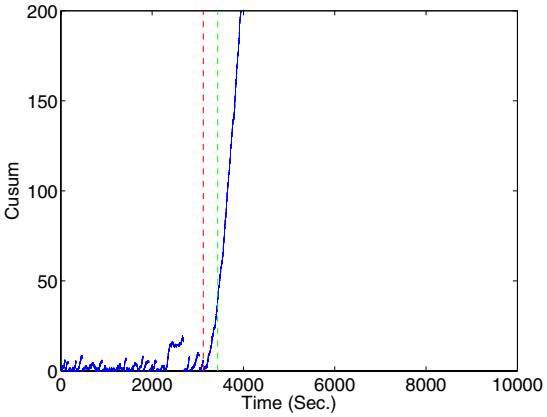

Fig. 11. CUSUM turns positive in 19 seconds after first Nimda scan

arrives at time 3720 seconds and is detected at 3950 seconds after a 230 second delay. The average scanner counts increase 410 to 480 during this period. The worm reaches its peak at about 6000 seconds when the number of unsolicited packets has grown to about 820 per five seconds. The detection algorithm signals a worm outbreak when the worm is only one tenth on its way to the peak. The detailed packet trace shows that only 320 hosts were infected by the time of the alarm whereas Witty eventually infected 11,171 hosts.

In Figure 8, 79 seconds elapse from the first Witty scan to the time the CUSUM statistic turns positive. Another 151 seconds elapse before the MLE $\hat{r}$ becomes significantly positive and the alarm is triggered. The QQ-plot in Figure 9 suggests a good fit between the real trace and our estimated model. Only about 3% of the data in the lower tail lie any marked distance from the line $y = x$.

C. Nimda

The first Nimda worm scan arrives at Bell Labs at time 3120 seconds in Figure 10. An alarm is raised after a 316 second delay. The peak is reached at 9410 seconds. Note that there is a dip about 1000 seconds before the peak and we conjecture that this may be an effect from some networks being disconnected from the Internet in response to the worm outbreak. The early detection algorithm signals an outbreak when the worm is 1/20 of its way to its peak. 19 seconds elapse between first Nimda scan and when the CUSUM turns positive and MLE estimation begins. Another 297 seconds elapse before the MLE confirms that the exponential rate is significantly positive and the alarm is raised. The QQ-plot in Figure 12 demonstrates that the model of exponential increase fits the scanner arrival data reasonably well. Only smallest three values, 0.15% of the points, are removed from the diagonal line.

D. Blaster

The Blaster worm starts at time 3040 seconds in Figure 13 and the detection is made at time 3381 second after a delay of 341 seconds, which is less than 1/10 of time to reach a peak at 6320 seconds. We observe from Figure 14 shows the CUSUM becoming positive at 3054 seconds, 14 seconds after the first scan, and the final alarm being raised after another 327 of rate estimation. In Figure 15, only about 1% of the points in the left tail are removed from the line $y = x$.

E. Summary

Table I summarizes the detection results for all four worms. Column 2 gives the date of outbreak. Column 3 has the

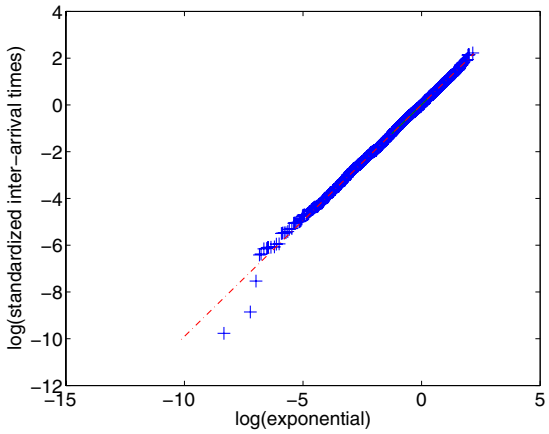

Fig. 12. Model fitting for Nimda: outlier less than 0.15%

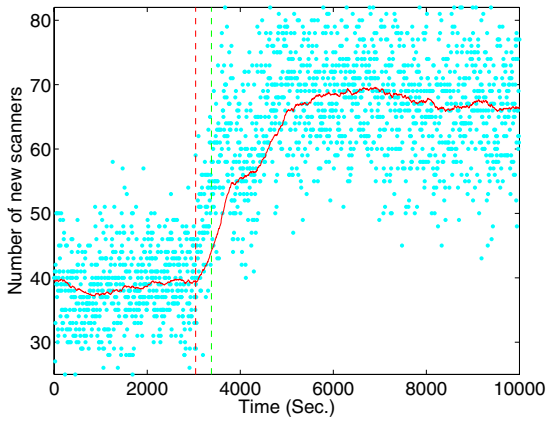

Fig. 13. Blaster is detected in 341 seconds

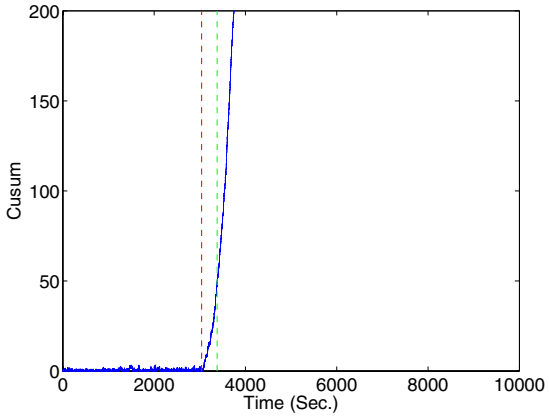

Fig. 14. CUSUM turns positive in 14 seconds after first Blaster scan

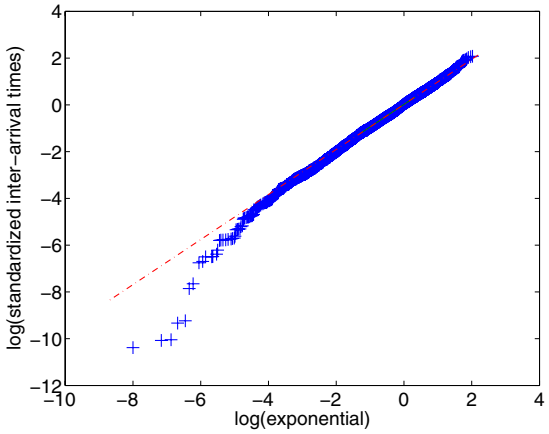

Fig. 15. Model fitting for Blaster: outliers less than 1%

estimated propagation rate at the time of the signal. Columns 4 and 5 record the time until detection and the total propagation time and the last column is the ratio of these times. Because worm growth is exponential, the ratio of infected hosts to vulnerable hosts is significantly lower that the fraction of total time until detection and this reinforces the value of fast detection.

Worm	Date	Growth Rate	Detection Time	Propag. Time	Early Warn. By Time
Slammer	1/25/03	0.1325	16	236	6.8%
Witty	2/19/04	0.0055	230	2280	10%
Nimda	9/18/01	0.0042	316	6290	5%
Blaster	8/11/03	0.0042	314	3280	9.6%

TABLE I
SUMMARY OF WORM DETECTION RESULTS

VI. DISCUSSION

A. Impact of MLE Start Time

Although the validity of our approach is only formally proved for Cases 1 and 3 in Section IV, we conjecture that Case 2 occurs with small probability and the impact on accuracy is limited even when it does occur. Because our collection of traces does not contain enough information to allow testing the conjecture, we verify it using simulation of a propagation process as follows. The first 200,000 arrivals follow a stationary Poisson process with rate $\lambda = b$ per second and the remaining 20,000 arrivals have a non-stationary rate of $\lambda(t) = b + ae^{rt}$ per second, where $b = 78$ counts per second, $a = 5$ and $r = 0.01$. The outbreak of the worm is at the change point from a stationary to a non-stationary process. The detection algorithm is run with $w = 0.0001$ and $h = 60/b^2$ and record is kept of the number of false alarms, the worm start time (t_w), the alarm time, and the latest time, t_0, that the CUSUM becomes positive before the worm alarm is raised. The simulation was repeated 500 times.

Case 1 occurs when $t_0 > t_w$ and Case 2 when $t_0 < t_w$. Figure 16 shows a histogram of $t_0 - t_w$. Case 2 occurs in only about 10% of simulations and when it does occur, the difference between t_0 and t_w is not large. The worm detection time is equal to the alarm time minus the worm start time. A histogram of these times for the 500 simulation runs is provided in Figure 17.

[2]This corresponds to a 12 minute ARL.

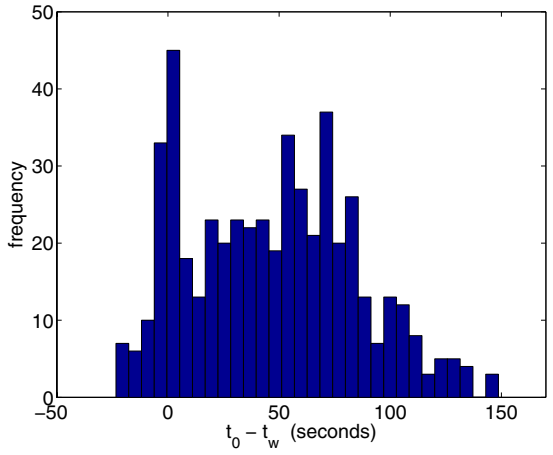

Fig. 16. Histogram of the delay between worm start time and CUSUM starting estimation time

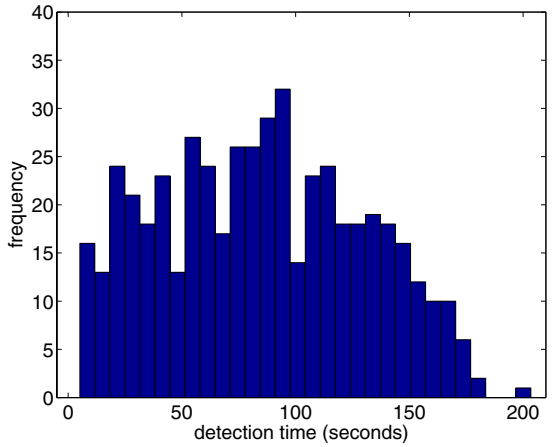

Fig. 17. Histogram of the detection time

B. False Alarms

In our experiments, CUSUM is triggered very often due to a rate change on the unsolicited packet arrival. However, most of these changes do not follow an exponential growth trend and thus the MLE inference does not produce a significant rate estimate, which in turn yields no alarm. We ran the algorithm on six non-consecutive months of traces to pinpoint the well known worms and saw only 19 unexpected alarms. Zooming into the traces that triggered these alarms, they are all cases with exponential-type increases in the rate of unsolicited scanners. We are able to confirm that 11 of these correspond to re-launches of some well-known worms (Re-launches of existing worms are also observed by [2]). For the remaining eight alarms, we conjecture that they are caused by worms that did not have widespread effects and are thus not publicized. Because the t-sampling technique filters the unsolicited packet arrivals in such a way that no small set of hosts is able to affect the arrival process, an exponential increases in the rate of uncoordinated scanning hosts is very unlikely to occur unless a new worm has been released. Unfortunately, our traces record only the packet headers so we do not have the information to perform additional analyses based on payload to find out what exactly triggered the exponential increase. We are in the process of obtaining traces from other parts of the network to delve further into the eight alarms of unknown origin.

VII. Robustness of unsolicited packet sampling

We have worked with t-samples of scanner arrivals rather than the full set of unsolicited packets because the scanner process is much better behaved due to elimination of the fluctuations introduced by a few top scanners. Implementation of the $t-$sample involves keeping track of the source addresses of unsolicited packets. However, source addresses may be spoofed. It has been showed in [17] that most of the scanners do not spoof their addresses. If a scanner spoofs its source address consistently for all scans with different hosts using different addresses, then there will be no effect on the statistics of the t-sample. This is because we sample based on distinct source addresses but not the exact value of address. The most challenging scenario is when all scanners spoof the same address, where we would get one count every t seconds. We can apply various spoof detection algorithm such as the one based on TTL field in packet header [22] to make the sampling more robust.

Another challenge to the source based sampling is NAT where many unsolicited source may share a single address. However, the unsolicited packets arrival rate will still grow exponentially but may at slightly lower rate. Since our detection algorithm has a very low threshold for worm propagation rate, the impact of NAT to our detection algorithm should be minimal.

VIII. Conclusion

Worm outbreaks are increasingly a major threat to the Internet. The release and propagation of Slammer provided ample evidence of the scale and disruption possible with a fast-spreading worm.

To counter these worms, we have devised a fast and robust worm detection algorithm that does not use a payload signature and relies solely on observing certain invariant characteristics of propagating worms. The algorithm has been applied to real network traces to demonstrate the effectiveness of the new approach.

We plan to further evaluate the algorithm using traces collected from a variety of Internet locations to test its sensitivity to the fraction of occupied addresses on the subnet being monitored and to the traffic volume on the link. We would also like to reduce the computational complexity of the algorithm. One possibility is to reduce the false signal rate of the CUSUM such that the more expensive MLE computations will be invoked less frequently. However, a better solution would be to replace the MLE algorithm with an online version that provides approximated MLEs. Two possible technologies for this approach are the particle filter and the extended Kalman filter that are under our further investigation.

APPENDIX

A. Proof of Theorem 1.

For $r > 0$ let $\Lambda(t) = \int_0^t \lambda(s)ds = bt + a(\exp(rt) - 1)/r$ and for $r = 0$ define , $\Lambda(t)$ by the limit as $r \to 0$. Then $\{T_j^* = \Lambda(T_j) : j = 1, 2, \cdots\}$ follows the standard Poisson process with stationary rate 1. By the probability density transformation formula, the joint distribution of $\{T_k, T_{k+1}, \cdots, T_{k+n}\}$ is :

$$\begin{aligned} & p(\{T_k, T_{k+1}, \cdots, T_{k+n}\}) \\ = & (\prod_{j=k}^{k+n} |\frac{\partial T_j^*}{\partial T_j}|) p(\{T_j^* : k \le j \le k+n\}) \\ = & (\prod_{j=k}^{k+n} \lambda(T_j)) p(T_k^*) \prod_{j=k}^{k+n-1} p(T_{j+1}^* | T_j^*) \\ = & (\prod_{j=k}^{k+n} \lambda(T_j)) g_k(\Lambda(T_k)) \exp(-(\Lambda(T_{k+n}) - \Lambda(T_k))), \end{aligned}$$

where g_k is the Gamma density function with degree k. Since $p(T_k) = \lambda(T_k) g_k(\Lambda(T_k))$, thus

$$\begin{aligned} & p(\{T_{k+1}, \cdots, T_{k+n}\} | T_k) \\ = & (\prod_{j=k+1}^{k+n} \lambda(T_j)) \exp(-(\Lambda(T_{k+n}) - \Lambda(T_k))). \end{aligned}$$

Let $\overline{T}_j = T_{k+j} - T_k$, $j = 1, \cdots, n$. Then

$$\begin{aligned} & p(\{\overline{T}_j : 1 \le j \le n\} | T_k = t_0) \\ = & (\prod_{j=1}^{n} \lambda(\overline{T}_j + t_0)) \exp(-(\Lambda(\overline{T}_n + t_0) - \Lambda(t_0))) \\ = & (\prod_{j=1}^{n} \overline{\lambda}(\overline{T}_j)) \exp(-(\overline{\Lambda}(\overline{T}_n))), \end{aligned}$$

which coincides with the non-stationary Poisson process with rate $\overline{\lambda}(t) = b + ae^{rt_0}e^{rt}$. Here $\overline{\Lambda}(t) = \int_0^t \overline{\lambda}(s)ds$.

B. Calculation of h.

Let τ be the first false alarm time with the background inter-arrival distribution F_0 with probability density f_0. That is, $\tau = \inf_n\{S_n \ge h,\ S_k < h \text{ for } k = 1, \cdots, n-1\}$, where S_n is defined in Section II.A. Then h is decided by the average run length (ARL) $E_0[\tau | S_0 = 0] = L_0$ with a pre-specified L_0, for example 1 million seconds (about half a month). By defining the ARL function $L(t) = E_0[\tau | S_0 = t]$, $0 \le t < h$, then h can solved by the following integral equation (let $\delta = (1-p)\mu$)

$$L(t) = 1 + L(0)(1 - F_0(t+\delta)) + \int_0^h L(s) f_0(t + \delta - s) ds,$$

with initial value $L(0) = L_0$. In this paper, f_0 and F_0 are the density and distribution functions of exponential(μ). The closed form solution to the above integral equation has been developed in [23].

C. QQ-plot for evaluating accuracy of MLE based model inference

Notice that if $\{\overline{T}_1, \cdots, \overline{T}_n\}$ are arrival times from a non-stationary Poisson process with cumulative rate $\overline{\Lambda}(t)$, then the transformed arrival process $\{\overline{\Lambda}(\overline{T}_1), \cdots, \overline{\Lambda}(\overline{T}_n)\}$ follows a stationary Poisson model with rate 1, whose inter-arrival times are i.i.d. exponential (1). We use this fact to test the goodness-of-fitting of our data. The steps are as follows: 1). calculate $\{\hat{\overline{\Lambda}}(\overline{T}_1), \cdots, \hat{\overline{\Lambda}}(\overline{T}_n)\}$, where $\hat{\overline{\Lambda}}(t) = bt + \frac{\hat{\overline{a}}}{\hat{r}}(\exp(\hat{r}t) - 1)$; 2). calculate transformed inter-arrival times $\overline{X}_j = \hat{\overline{\Lambda}}(\overline{T}_j) - \hat{\overline{\Lambda}}(\overline{T}_{j-1})$, $j = 1, \cdots, n$, $\overline{T}_0 = 0$; 3). compare the empirical quantiles of $\{\overline{X}_j : j = 1, \cdots, n\}$ with quantiles of exponential(1), so-called QQ-plot, in the logarithmic scale. The x-axis is the logarithmic quantiles of exponential (1) and the y-axis is the logarithmic quantiles of $\{\overline{X}_j : j = 1, \cdots, n\}$.

REFERENCES

[1] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "The spread of the sapphire/slammer worm," 2003.

[2] S. Staniford, V. Paxson, and N. Weaver, "How to own the internet in your spare time," in *USENIX Security Symposium*, 2002.

[3] C. C. Zou, W. Gong, D. Towsley, and L. Gao, "The monitoring and early detection of internet worms," *IEEE/ACM Transactions on Networking*.

[4] G. Gu, M. Sharif, X. Qin, D. Dagon, W. Lee, and G. Riley, "Worm detection, early warning and response based on local victim information," in *Annual Computer Security Applications Conference*, (Tucson, Arizona), dec 2004.

[5] J. Wu, S. Vangala, L. Gao, and K. Kwiat, "An effective architecture and algorithm for detecting worms with various scan techniques," in *Network and Distributed System Security Symposium*, Feb 2004.

[6] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan, "Fast portscan detection using sequential hypothesis testing," in *IEEE Symposium on Security and Privacy*, may 2004.

[7] N. Weaver, S. Staniford, and V. Paxson, "Very fast containment of scanning worms," in *USENIX Security Symposium*, aug 2004.

[8] S. Singh, C. Estan, G. Varghese, and S. Savage, "Automated worm fingerprinting," in *ACM/USENIX Symposium on Operating System Design and Implementation*, Dec 2004.

[9] C. Shannon and D. Moore, "The spread of the witty worm," 2004.

[10] D. Moore and C. Shannon, "Code-red worms: A global threat," 2002.

[11] "Symantec early warning solutions."

[12] "Internet storm center."

[13] D. Moore, C. Shannon, G. M. Voelker, and S. Savag, "Network telescopes," 2002.

[14] "Honeynet project: know your enemy."

[15] D. Dagon, X. Qin, G. Gu, W. Lee, J. Grizzard, J. Levine, and H. Owen, "Honeystat: Local worm detection using honeypots," in *Recent Advances In Intrusion Detection (RAID)*, 2004.

[16] H. Wang, D. Zhang, and K. G. Shin, "Change-point monitoring for detection of dos attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 4, 2004.

[17] V. Yegneswaran, P. Barford, and J. Ullrich, "Internet intrusions: Global characteristics and prevalence," in *ACM SIGMETRICS*, 2003.

[18] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes, Theory and Application*. Prentice Hall, 1993.

[19] P. Andersen, Q. Borgan, R. Gill, and N. Keiding, *Statistical Models Based on Counting Processes*. Springer-Verlag, 1992.

[20] P. Bickel and K. Doksum, *Mathematical Statistics, Basic Ideas and Selected Topics, Vol. I, second edition*. Prentice Hall, 2001.

[21] D. Lambert and C. Liu, "Dynamic thresholding: monitoring streams of network counts online," in *Journal of The American Statistical Association, to appear*, 2006.

[22] C. Jin, H. Wang, and K. Shin, "Hop-count filtering: An effective defense against spoofed dos traffic," in *ACM International Conference on Computer and Communications Security (CCS)*, Oct. 2003.

[23] F. Gan and K. Choi, "Computing average run lengths for exponential cusum schemes source," *Journal of Quality Technology*, vol. 26, no. 2, pp. 134–143.

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the Proceedings IEEE Infocom.