

The Limits of Global Scanning Worm Detectors in the Presence of Background Noise

David W. Richardson, Steven D. Gribble, and Edward D. Lazowska
Department of Computer Science & Engineering, University of Washington
{daverich,gribble,lazowska}@cs.washington.edu

ABSTRACT

Internet worms cause billions of dollars in damage each year. To combat them, researchers have been exploring global worm detection systems to spot a new random scanning worm outbreak quickly. These systems passively listen for worm probes on unused IP addresses, looking for anomalous increases in probe traffic to distinguish the emergence of a new worm from background Internet noise.

In this paper, we use analytic modeling, simulation, and measurement to understand how background noise impacts the detection ability of global scanning worm detectors. We investigate the relationship between the average background noise level, the number of IP addresses monitored, and the detection latency for two classes of global scanning worm detectors: scan packet-based and victims-based schemes. Our results show how worm detection latency degrades as a function of the background noise level. To compensate, global scanning worm detectors can increase the number of IP addresses that they monitor. However, given the growth trend of background noise levels, the number of IP addresses which must be monitored may quickly become unreasonable. Because of this, we conclude that global scanning worm detection schemes are unlikely to be competitive with local scanning and signature-based worm detection schemes.

Categories and Subject Descriptors

D.4.6 [Operating Systems]: Security and Protection—*Invasive software*
; I.6.4 [Simulation and Modeling]: Model Validation and Analysis

General Terms

Security Measurement Theory

Keywords

computer security, computer worms, scanning worms, worm detection, worm models

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WORM'05, November 11, 2005, Fairfax, Virginia, USA.
Copyright 2005 ACM 1-59593-229-1/05/0011 ...\$5.00.

1. INTRODUCTION

Researchers have been designing systems to detect new random scanning worms before they affect a large number of victims [1, 3, 7, 9, 12, 17, 18, 19, 24, 25]. These *global scanning worm detectors* exploit two properties of worms. First, because worms emit randomly directed probes to find victims, their probes can be detected by monitoring unused IP addresses. Second, because a worm outbreak grows quickly and in a mathematically predictable fashion [8, 25], the difference between background scan traffic from non-worm sources [16] and worm probes can be distinguished over time. By monitoring a block of unused IP addresses and looking for statistically significant trends in aggregate probe rates, a worm detector can spot the worm “signal” through the background noise and raise an alarm.

Internet *background noise* consists of packets sent to ports on IP addresses where no network service exists to handle them. These noise packets are abundant in today’s Internet and are generated by network scanners, DoS bots, lingering worms, or incorrectly configured systems [16]. While the issue of background noise has been acknowledged in previous work, there has not yet been a deep analysis of how noise impacts the effectiveness of scanning worm detectors. For example, one might naturally assume that a detector that monitors more unused IP addresses would be able to detect a worm epidemic sooner. However, the number of noise probes observed by a detector increases as more IP addresses are monitored, potentially confounding detection.

In this paper, we use a combination of analytic modeling, simulation, and measurement to understand the relationship between background noise and scanning worm detection fidelity. We examine two global scanning worm detection schemes: *scan packet-based* detectors that look for anomalous increases in the aggregate arrival rate of probes, and *victims-based* detectors that look for an anomalous increase in the rate at which new probe sources (i.e., victims) are observed. For each technique, we describe and validate an analytic model of background noise traffic, we simulate the propagation of a worm in the presence of varying noise levels, and we analyze and explain the success or failure of the worm detector to raise an alarm early in the worm’s life cycle. Our results demonstrate that:

- For both scan packet-based and victims-based detection schemes, larger background noise levels degrade the expected detection latency. Given that background noise levels are trending upwards over time, we expect that scanning worm detectors will grow less capable in the future, unless they monitor more IP addresses.

- For both detection schemes, increasing the number of monitored IP addresses improves the detection latency. Monitoring more IP addresses decreases the observed variation in the noise probe arrival rate, allowing a detector to use a tighter alarm threshold.
- Unfortunately, as the average noise rate grows larger, the number of additional IP addresses that must be monitored quickly becomes unreasonable.

These results strongly suggest that global scanning worm detection systems will have limited utility in the future, assuming the background noise signal continues to grow. Fortunately, alternative classes of detection systems are now being explored. Signature-based detectors examine probe payloads for content strings that occur repeatedly and which originate from a large number of sources [19, 12]; because signature-based schemes do not attempt to track the growth rate or size of a worm, they are much more robust against noise. Local scanning worm detectors attempt to monitor hosts within a network to detect when they transition from emitting benign traffic to emitting scanning, worm-like traffic [17, 18, 9]. Accordingly, we believe that worm detection systems of the future should be based on local scanning algorithms or signature-based schemes. These schemes are not prone to false positives from background noise, and they have the potential to detect and block any scanning pathogen, whether or not it is a self-propagating worm.

The rest of this paper is organized as follows. Section 2 describes related work on worm detection. Section 3 presents an analytical framework for scan packet-based worm detectors in the presence of noise. Section 4 presents an analytical framework for victims-based detectors in the presence of noise. Finally, we conclude the paper in Section 5.

2. RELATED WORK

The propagation and abatement of Internet worms have similar mathematical properties as biological epidemics. Several researchers have taken advantage of this similarity to construct models that describe the spread of Internet worms over time. Zou et al. applied classical stochastic epidemic models to model the spread of the Code Red worm [26]. Their model considered how Code Red’s probe rate slowed over time as networks grew clogged and countermeasures were deployed. In follow-on work, they described how the propagation of a worm could be retarded if a detection system quarantined activity on ports for which scan traffic was observed [27]. However, their analysis did not consider the issue of false positives caused by background noise traffic.

Chen et al. proposed a stochastic worm propagation model, compared its predictive power to classic epidemic models, and analyzed a simple scan packet-based detector [5]. While elements of their analysis are similar to ours, they did not consider how background noise affects worm detection.

By fusing analytic worm propagation models with a localized model of network topologies, Liljenstam et al. generated realistic synthetic worm traffic and examined its effect on network performance [13]. Additionally, they investigated the effectiveness of the DIB:S/TRAFEN worm detection system [4] assuming a simple Poisson background noise process. However, they did not investigate how or if the addition of noise in their analysis affects the worm detector’s results.

2.1 Analysis of Real Worms

Several researchers have reverse engineered worms or gathered measurement data during the propagation of a worm outbreak. An early example was the deconstruction of the original Morris worm [20], though no measurement results were described. Moore et al. performed a detailed measurement analysis of the Code Red worm incident, providing insight into its probe rate and propagation over time [15]. A similar study was performed of the Sapphire/Slammer worm incident, showing how it propagated to 90% of vulnerable hosts within 10 minutes [14]. Kienzle and Elder provide a recent survey of worms and their trends [11].

In many ways, real worms have naïve designs. Researchers have proposed strategies by which a worm could propagate more quickly, including taking advantage of knowledge of routing topologies [28], or constructing hit-list, permutation scanning, or flash worms [21].

2.2 Worm Detection Systems

In an early proposal for a global scanning worm detector, Bakos and Berk described a system which monitors ICMP destination unreachable packets, looking for the “exponential bloom” that signifies the rapid global growth of self-propagating code [1]. This study did not consider the effect of background noise, and it did not quantify detection latency or false positive rates. Zou et al. proposed a detection system that fits observations of probe traffic to an epidemic model, using Kalman filtering to estimate the rate at which an average worm victim successfully infects additional victims [25]. They observed that increasing the number of monitored IP addresses decreases the error in the estimated number of infected hosts. They acknowledged that background noise affects detection, but provided no analysis of the magnitude of its effect.

The closest work to ours is that of Wu et al. [24], which described scan packet-based and victims-based scanning worm detectors and quantified their effectiveness. While they correctly argued that victims-based detectors are less prone to false positives from background noise, they provided no analysis to explain why, nor did they provide data relating noise rates to detection probabilities. Similarly, while they argued that monitoring more IP addresses improves detection latency, they did not analyze why.

Barford et al. [2] is the only work that provides a similar analysis of the effects of noise on worm detectors. However, their analysis is at a very high-level, using a number of mathematical approximations to put rough bounds on detection probabilities and false positive rates. In contrast, our analysis and noise models are specifically tailored to understand the differences between scan packet-based and victims-based detectors. We also parameterize our noise models and worm growth models using data collected from real Internet measurements and actual worm epidemics.

Qin et al. used traces gathered on 25,600 unused IP addresses to evaluate the effectiveness of several worm detection schemes on real traffic [17]. Similar to us, they advocated that *local detection* may be more effective than global detection: once a local host is observed to exhibit worm-like behavior, one can immediately assert that a worm has been detected. They propose a HoneyPot-based scheme for detecting local worm-like behavior [7].

Schecter et al. considered the problem of detecting that a local host is maliciously scanning rather than trying to

forge benign connections [18]. Once classified as a scanner, a host can be quarantined or rate-limited. Gu et al. accomplished a similar goal with a different technique: if a host that received a packet on a certain port emits traffic on that port at an anomalous rate, it is likely infected by a worm [9]. Jung et al. [10] developed an effective online algorithm called Threshold Random Walk (TRW) based on sequential hypothesis testing for detecting scanners with very few (4-5) connection attempts. Weaver et al. [22] used a refined version of the TRW algorithm as the basis for scanning worm detectors. While their algorithm inherently relies on finding deviations from benign or noisy traffic patterns, their analysis does not explain how the properties of the noise affects their results.

Fundamental to the effectiveness of all scanning worm detectors is the selection of the set of unused IP addresses to monitor for malicious activity. Cooke et al. [6] investigated how and why different sets of unused IP addresses observe differing and localized Internet traffic patterns. Their work shows how certain sensor properties affect the sensor’s observations, suggesting that these outside influences must be taken into account when generalizing Internet traffic trends and anomalies to the entire address space. Our analysis does not take into account these global variations in traffic patterns. For the sake of clarity and simplicity, we assume that noise and worm traffic is distributed uniformly across IP address space.

Instead of looking for worm-like traffic patterns, signature-based systems instead attempt to generate content-based signatures of worm pathogens which can then be used to block their spread. Signature-based systems (such as Earlybird [19] and Autograph [12]) borrow local scanning techniques to identify suspicious traffic, but they then “sift” through the payload of this traffic to identify commonly re-occurring signatures. The signature-based system proposed by Whyte et al. [23] uses DNS anomalies to detect scanning worms. The idea is that while legitimate human users tend to use alphanumeric strings and DNS to contact hosts, scanning worms typically use numeric IP addresses to find new victims. This technique of looking for non-DNS-based connections can be a good behavioral signature for detecting worms.

By looking for signatures that occur often and that arrive from multiple sources, these signature-based schemes can identify and block both self-propagating and non-self-propagating pathogens. Moreover, the signatures generated can be used to filter out false positives arising from benign or previously known malicious traffic, making background noise a non-issue.

3. AN ANALYSIS OF SCAN PACKET-BASED WORM DETECTORS

In this section of the paper, we analyze the effectiveness of *scan packet-based* worm detectors in the presence of background noise. After providing an overview of how scan packet-based detectors work, we describe and validate an analytic model for background noise. Using this model, we derive the expected distribution of noise packets that arrive at a monitored IP address. Next, we use the well-known random constant spread (RCS) model to derive the expected distribution of worm scan packets that arrive at a monitored IP address. By fusing these two models together,

we analyze the behavior of a scan packet-based detector by calculating the probability that it will detect the worm over time. Because our analysis is model driven, we can easily vary parameters such as the mean level of noise in the Internet and the number of IP addresses the detector is monitoring, observing their effect on the probability of raising an alarm.

Our goal in this section is to answer three questions: (1) What is the tradeoff between detection fidelity and false positive rate? (2) How is detection fidelity affected by the Internet’s background noise level? (3) How is detection fidelity affected by the number of IP addresses monitored by the detector?

3.1 System Architecture

A scan-packet detector passively monitors a set of unused IP addresses, keeping track of the rate at which packets arrive. Abstractly, we consider each observed IP address to be a separate *monitor*, even if all are routed to a single host. For random scanning worms, it does not matter whether these unused IP addresses are distributed across the IP address space or clustered within a contiguous block. Our analysis assumes the set of monitors provide uniform, random sampling of the global noise and worm traffic.

Each monitor receives two types of packets: worm scan packets and noise packets. Worm scan packets are those generated by machines infected with a worm. Noise packets are packets generated by non-worm sources such as a DoS bot or an incorrectly addressed but valid TCP connection establishment request. Unfortunately, a monitor cannot distinguish between worm and noise packets *a priori*. Instead, the detector relies on observing an increase in the overall scan packet rate to determine that a worm is present and growing. Formally, a detection system with k monitors raises an alarm at time t whenever the total scan packet arrival rate $observed_{sp}(k, t)$ is greater than an *alarm threshold* θ , where the total packet arrival rate includes both worm and noise packets: $observed_{sp}(k, t) > \theta$.

The rationale behind this detection condition is that since worms grow rapidly, the rate of generated worm scan packets will also grow rapidly. Monitors can detect this growth by looking for an anomalous increase in the observed packet arrival rate above the alarm threshold. However, natural and random fluctuations in the observed noise rate are likely to occasionally exceed θ , causing the system to raise an alarm even when no worm attack is in progress. To minimize the probability of a false alarm, an alarm should only be raised when $observed_{sp}(k, t)$ is greater than the range of *likely* variations in the noise by a statistically significant amount.

3.2 Modeling Noise

While previous work has shown that different areas of the Internet experience differing local traffic patterns [6], for this paper we ignore any intrinsic global noise variations, and instead assume that each IP address observes a noise arrival process drawn from the same, global distribution. Using this assumption, we ran a simple experiment to understand the basic nature of Internet background noise in which we monitored a single unused IP address over the course of 25 days, and counted the number of TCP SYN packets that arrived during each hour of this time period. We observed a distribution of arrival rates, with a mean of 42 packets arriving each hour and a standard deviation of 57 (Figure 1a).

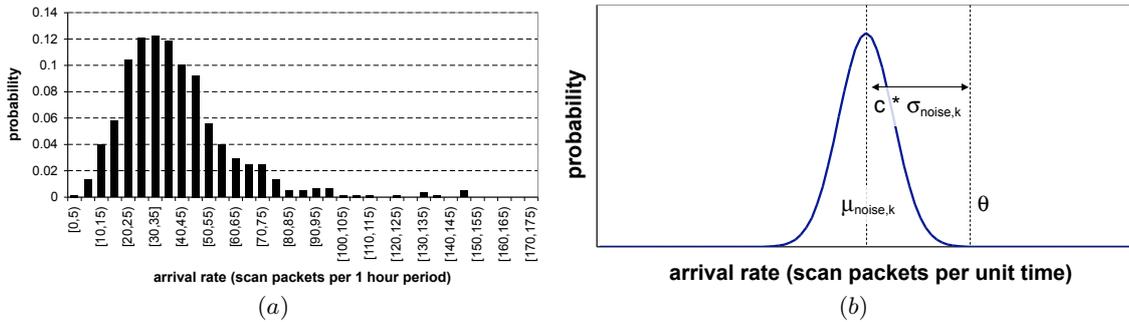


Figure 1: (a) Measured distribution of the number of noise packets that arrived per hour at a single, unused IP address. (b) A hypothetical distribution of the noise rate observed by a detection system that monitors k IP addresses. The alarm condition threshold θ is shown at $c * \sigma_{noise,k}$ above the mean noise rate $\mu_{noise,k}$.

We need to adopt a model for Internet noise that results in a similar variable arrival rate distribution at each monitor. As a starting point, we present a very simple model in which noise packets are emitted at a constant aggregate rate ϕ per time step, but each is sent to a randomly selected IP address within the Internet. This model represents each noise packet as a Bernoulli trial, where the probability that a single monitor will observe a noise packet is $1/2^{32}$.

For a system with k monitors, the probability that a noise packet will be detected is $p_k = \frac{k}{2^{32}}$. For a given time step, we can represent the number of the noise packets that are detected as a random variable N . This variable has a binomial distribution with a mean $\mu_{noise,k}$ and variance $(\sigma_{noise,k})^2$:

$$\mu_{noise,k} = \phi * p_k = \frac{\phi * k}{2^{32}}, \quad (1)$$

$$\begin{aligned} (\sigma_{noise,k})^2 &= \phi * p_k * (1 - p_k), \quad (2) \\ &= \mu_{noise,k} * \left(1 - \frac{k}{2^{32}}\right). \end{aligned}$$

Projecting back from the data in Figure 1a, under this model the aggregate rate θ at which noise packets are injected into the Internet is approximately 50.1 million packets per second! We can also arrive at a similar estimate based on data within a study of Internet background radiation [16]. Based on the data in Figure 6 within that paper, the estimated aggregate Internet noise rate is 46.6 million packets per second, which is remarkably close to the estimate from our data.

Note that the mean noise packet arrival rate grows linearly with both the number of monitored IP addresses k and the aggregate noise injection rate ϕ . The standard deviation grows with the aggregate noise rate, but decreases as more IP addresses are monitored. Overall, as the noise level on the Internet increases, the worm detector will see a *wider* noise distribution as well a higher mean noise arrival rate.

3.2.1 The Detection Condition

Given our noise model, we are now in a position to define the detection system alarm threshold θ . We want the system to raise an alarm when the observed scan packet arrival rate is higher than the mean noise arrival rate, but to a large enough degree that we can be statistically confident that it is not due to variations in the noise. To accomplish this, we define θ as some number of standard deviations c above

the mean noise rate: $\theta = \mu_{noise,k} + c * \sigma_{noise,k}$. The parameter c corresponds to the confidence level that an alarm is the result of increased worm traffic and not a random noise fluctuation; increasing c increases this confidence level. This detection condition is depicted graphically in Figure 1b.

An increase in the aggregate Internet noise rate ϕ raises the alarm condition θ for two reasons: the observed noise arrival rate grows, as does the width of the observed noise distribution. As the Internet background noise grows louder, the alarm threshold becomes harder to reach.

3.3 Modeling a Worm

To model worms, we adopt the well-known and widely used *random constant spread* (RCS) model, whose accuracy has been demonstrated for worm outbreaks such as Code Red v2 and Slammer [14, 15]. The RCS model assumes that each infected host randomly scans the IP address space to find new victim machines. The RCS model is a classic logistic curve that represents the way in which random-scanning worms initially spread at an exponential rate, but eventually slow down as the susceptible population becomes saturated with the pathogen.

Admittedly, the RCS model does not capture the specific dynamics of how saturation occurs, nor does it model other precise effects such as Slammer's diminishing scan rate over time [14]. The RCS model instead captures how all effective worms experience a period of exponential growth. Any practical worm detection system that hopes to stop the spread of an active worm must be able to detect the worm as early as possible in this exponential growth phase, otherwise detection is useless because the worm will have already infected far too many victims. Since beneficial worm detectors operate in this exponential growth phase, other secondary dynamics, especially at the end of a worm's life-cycle, have little impact on a detector's effectiveness. The RCS model removes these second-order effects to provide a clear and simple model that captures the most important aspects of a worm's growth behavior from the viewpoint of a worm detector.

Under the RCS model, the number of infected hosts $I(t)$ at time t after the worm is released is very closely approximated by the exponential function $I(t) = a^t$, for some constant a , up until the population saturates. As an example, an analysis of the Code Red v2 epidemic of July 2001 estimated

that $I(t) = 10.84^t$, where t is measured in hours [15]. If we assume that an infected host transmits s probes per unit time to random destinations, then at time t the overall worm probe rate $\psi(t)$ is modeled by $\psi(t) = s * I(t) = s * a^t$.

Much like our noise model, each worm scan packet can be considered to be a Bernoulli trial, where success means that the worm scan packet is observed by our detector. Given this, at a given time t , the number of worm packets observed by a detector monitoring k IP addresses is a binomial random variable $W(t)$, with a mean and standard deviation given by:

$$\mu_{worm,k}(t) = \frac{\psi(t) * k}{2^{32}}, \quad (3)$$

$$(\sigma_{worm,k}(t))^2 = \mu_{worm,k}(t) * \left(1 - \frac{k}{2^{32}}\right). \quad (4)$$

As time progresses, the mean arrival rate of worm probes at the detector grows exponentially. Much like with the noise processes, the arrival rate distribution widens as the arrival rate grows.

3.4 Modeling the Worm Detector

We now have all of the pieces in place that we need to understand how a scan packet-based worm detector will behave in the presence of noise. The total observed scan packet arrival rate at a worm detector at time t is described by the sum of the noise packet arrival rate N and the worm packet arrival rate $W(t)$. Thus, at time t , $observed_{sp}(k, t) = N + W(t)$. Because N and $W(t)$ are binomial variables whose means and standard deviations are known, so is $observed_{sp}(k, t)$.

We are now ready to tackle the problem of calculating the probability that the detector raises an alarm at a particular time t . Given our detection condition $\theta = \mu_{noise,k} + c * \sigma_{noise,k}$, the system will raise an alarm at time step t if $observed_{sp}(k, t) > \theta$. Given this, the probability that the worm detector raises an alarm at time step t is just the expression $Pr[observed_{sp}(k, t) > \theta]$. Since we have closed-form solutions for $observed_{sp}(k, t)$ and θ , we can calculate this probability for any given number of monitored IP addresses k , any given aggregate noise packet generation rate ϕ , and any given detection threshold c .

We now derive the key equation that we use for the rest of our analysis. This equation calculates the worm detector *fidelity*, which we define as the probability $Prob_{detect}(t)$ that the detector raises an alarm by time step t . Assuming that time step $t = 0$ is the point at which the first machine in the Internet becomes infected by the worm, $Prob_{detect}(t)$ gives us the probability that a system detects this worm by time t in its growth. Thus, the worm detector fidelity can be calculated as:

$$Prob_{detect}(t) = 1 - \prod_{i=0}^t (1 - Pr[observed_{sp}(k, i) > \theta]) \quad (5)$$

In the rest of this section, we will vary k , ϕ , and c , and use equation 5 to examine how the probability of detection is affected by each of these parameters.

3.5 Quantitative Results

Given our analytical model, we can now return to our three basic questions about scan packet-based worm detectors. To answer them, we use measurements of the Code Red

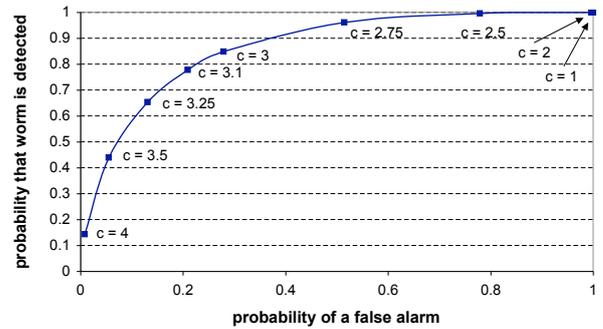


Figure 3: False alarm vs. worm detection probabilities. This figure illustrates the trade-off between the probability of raising a false alarm and the probability of detecting the worm by its critical time, for alarm conditions at various different numbers of standard deviations c above the mean noise level. The system considered had $k = 2^{18}$ monitors, and the noise ϕ was fixed at $\phi = 1.5$ billion packets per minute.

v2 worm [15] to parameterize our worm model. Then, we use our worm detector model to explore the effect of varying the background noise level ϕ and the number of monitored IP addresses k .

The Code Red v2 measurements show that the overall probe rate of the worm was approximately $\psi(t) = 660 * 10.84^{(t/60)}$, where t is measured in minutes since the worm began to propagate. If we define the time t_{crit} as the critical point by which our system should detect the worm, then the worm detector's fidelity is the probability $Prob_{detect}(t_{crit})$ that the worm is detected by t_{crit} . For our analysis, we chose $t_{crit} = 242$ minutes, the time by which 15,000 machines would be infected.

To pick a reasonable range of background noise levels ϕ and monitored IP addresses k to model, we used our data from Figure 1(a) that showed an actual aggregate Internet background noise rate of approximately 3 billion packets generated each minute. Our analysis modeled a range of background noise rates between $\phi = 0.5$ billion and $\phi = 20$ billion packets generated per minute to explore the effects of rising Internet noise rates. To understand the impact of different numbers of IP addresses monitored, we explored the values $k = \{1, 2^8, 2^{16}, 2^{18}, 2^{24}, \text{ and } 2^{32}\}$.

3.5.1 Detection Fidelity vs. False Positive Rate

When there is not a worm outbreak in progress, there is a non-zero probability that a scan-packet based worm detector will raise a false alarm due to random variations in the noise process above the alarm threshold θ . Accordingly, we define the false alarm probability as the likelihood that at least one false alarm is raised during a time window equal to t_{crit} . It should be noted that a worm detector false alarm is not catastrophic since false alarms conservatively flag benign traffic as malicious. As long as alarms are infrequent and manual verification can quickly rule them out, the cost of a false alarm would be limited.

By tuning the value of c , the number of standard deviations that the alarm threshold is set above the mean noise

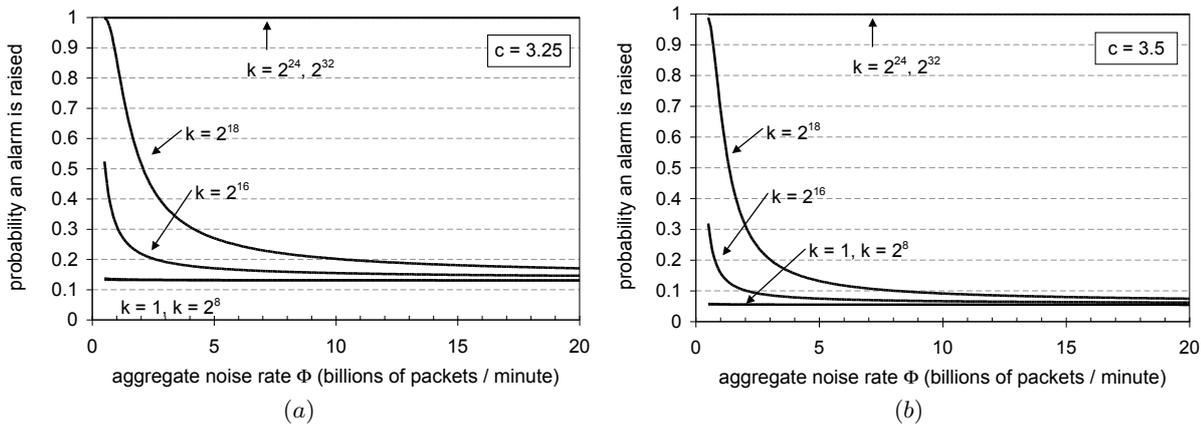


Figure 2: (a) The probability an alarm is raised by time t_{crit} for varying numbers of monitored IP addresses k between $k = 1$ and $k = 2^{32}$, where the number of standard deviations c used in calculating the detection threshold θ is $c = 3.25$. (b) The probability an alarm is raised by time t_{crit} for values of k between $k = 1$ and $k = 2^{32}$, where the number of standard deviations c is 3.5 .

arrival rate, we can affect both the false positive rate and the probability that a worm is detected. By increasing c , we require larger observed deviations from the mean noise rate to raise alarm, and therefore decrease the false alarm rate. But, if we raise c too much, we will also lower the worm detection rate. There is a fundamental tradeoff between detection fidelity and the false positive rate.

Figure 3 illustrates this tradeoff. We modeled an aggregate Internet noise rate of $\phi = 1.5$ billion packets generated per minute, and a detector that monitors $k = 2^{18}$ IP addresses. The graph demonstrates that increasing c decreases both the false alarm and detection probabilities. However, above $c > 3$ there is a region where the probability of detecting a worm is much larger than the probability of detecting a false alarm. This suggests that we should set $c > 3$ to avoid many false alarms while still maintaining adequate detection fidelity. For the rest of our analysis, we will use values $c = 3.25$ and $c = 3.5$.

3.5.2 Detection Fidelity vs. the Background Noise Level of the Internet

As the background noise level grows, the ability for a worm detector to raise an alarm degrades. A larger mean noise arrival rate will drown out the worm for a longer period of time, permitting the worm to affect more victims before it is noticed. As well, noise arrival rate distribution grows wider with a larger noise level, forcing the detector to adjust its worm detection threshold even higher to prevent false alarms.

We now quantify these effects. We use equation 5 to plot the probability $Prob_{detect}(t_{crit})$ that the detector raises an alarm by time t_{crit} , for varying values of the number of monitored IP addresses k , and for varying levels of Internet noise. As previously mentioned, we explore a range of values between $\phi = 0.5$ billion and $\phi = 20$ billion noise packets generated in aggregate on the Internet per minute; we estimated the current Internet noise level at $\phi = 3$ billion per minute. Figure 2b shows our results for $c = 3.25$ and $c = 3.5$.

These graphs demonstrate that as the noise level increases,

the probability that an alarm is raised rapidly decreases down to the baseline false alarm probability of the detector (13% for $c = 3.25$ and 5.5% for $c = 3.5$). As the baseline noise level and noise variations grow, the detector must become more tolerant to larger fluctuations in the number of observed scan packets, making it more difficult for the detector to spot a statistically significant increase in scan packet rate. The worm signal simply gets “lost in the noise”.

3.5.3 Detection Fidelity vs. the Number of Monitored IP Addresses

By increasing the number of monitored IP addresses k , the detector is able to reduce the width of the observed noise distribution. As a result, a smaller fluctuation will convince the detector that it has observed a worm signal, allowing it to spot the worm earlier. We once again consider the plots of Figures 2a and 2b. Focusing on the separate curves plotted for different values of k , increasing the number of monitored IP addresses increases the probability that an alarm is raised. However, even for moderately large values of k such as $k = 2^{16}$ (a class B network), the probability of detection remains low for realistic levels of background noise. For small values of k such as $k = 2^8$ (a class C network), the probability of detection is indistinguishable from the false alarm rate. Adding more monitors to a detector does increase detection fidelity, but for realistic noise levels, scan packet-based detectors are only effective for large values of k .

3.6 Summary

Scan packet-based detectors are highly sensitive to the amount of background noise in the Internet. As the amount of noise increases in the future, the effectiveness of scan packet-based detectors will degrade. Adding more monitored IP addresses increases the probability of detecting a worm. However, scan packet-based detectors must monitor a large number of IP addresses to be effective (i.e., $k > 2^{18}$).

4. AN ANALYSIS OF VICTIMS-BASED WORM DETECTORS

In the previous section of the paper, we analyzed the effectiveness of scan packet-based worm detectors in the presence of background noise. In this section, we perform a similar analysis for *victims-based* worm detectors. We first provide an overview of how victims-based detectors work, and we describe and validate an analytic model for Internet background noise from the perspective of a victims-based detector.

4.1 System Architecture

Victims-based worm detectors [24] have a similar architecture as scan packet-based detectors. The detector passively monitors a set of unused IP addresses to observe the arrival of unsolicited packets. Instead of simply monitoring the rate at which these packets arrive, a victims-based detector looks at their source IP addresses to determine which hosts transmitted them. A host that transmits an unsolicited packet is called a *victim*. A detector sees a victims arrival rate consisting of victims that are infected with a worm, and victims that are background noise sources.

A victims-based worm detector keeps track of which victims it has observed in the past, and calculates the observed rate of new victims. When the detector sees an increase that is statistically unlikely to be due to variations in the background noise, it raises an alarm. Formally, a victims-based worm detector with k monitors raises an alarm at time t if the total new victim arrival rate $observed_{vic}(k, t)$ exceeds the alarm threshold θ , where θ is greater than statistically likely variations in background noise: $observed_{vic}(k, t) > \theta$.

4.2 Modeling Noise

Because victims-based detectors keep track of the arrival rate of new victims, not the arrival rate of packets, we need to adopt a more sophisticated model for Internet noise than the one we employed for scan packet-based detectors. Our noise model must model the dynamics of the population of noise victims: how quickly new noise victims are born on the Internet, how long they remain noisy, and the per-victim scan rate.

We present a simple noise model that captures these dynamics. Our model assumes that new noise victims are born at a constant rate of λ new victims per time unit, and that a noise victim remains noisy for a fixed lifetime L before being “repaired.” While a noise victim is noisy, it emits noise packets at a rate of S_{rate} packets per time unit, with each packet being sent to a randomly selected IP address. Given this, the aggregate rate at which noise packets are generated, ϕ , is $\phi = L * \lambda * S_{rate}$.

With this model in place, we now need to derive the rate at which new noise victims are observed by a victims-based detector. To provide some intuition, in Figure 4 we show measured data obtained from monitoring a single, unused IP address over the course of 25 days. As with our scan packet-based noise model, we assume that this data is largely representative of values we would see at any IP address. Thus, the figure demonstrates that the rate at which new victims are observed fluctuates over time, and because of this fluctuation, the detector will observe a distribution of new victim arrival rates. In other words, the rate at which new victims are observed is governed by some random variable N with some distribution. In Figure 4, our measured distribution

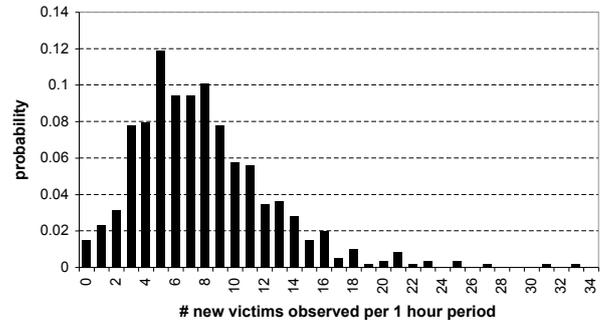


Figure 4: Measured Internet background noise victim arrival rate. This graph shows the measured distribution of the number of new noise victims that were observed per hour by a single, unused IP address.

for a single monitored IP address had a mean of 7.9 new victims observed per hour.

We now derive a closed-form expression for N . Though we have produced a formal derivation, we omit its details for brevity. To summarize it, our derivation shows that:

- The probability p_L that a noise victim is detected during its lifetime by a detector monitoring k IP addresses is $p_L = 1 - (1 - \frac{S_{rate} * k}{2^{32}})^L$.
- The mean number of new noise victims that the detector will observe per time unit (i.e., the mean new noise victim arrival rate $\mu_{noise,k}$) is $\lambda * p_L$.
- The detector will observe a Poisson distribution of new victim arrival rates, with mean $\mu_{noise,k}$. The variance of a Poisson distribution is equal to its mean, so $\sigma_{noise,k}^2 = \mu_{noise,k}$.

Putting this together, we have:

$$\mu_{noise,k} = (\sigma_{noise,k})^2 = \lambda * \left(1 - \left(1 - \frac{S_{rate} * k}{2^{32}}\right)^L\right) \quad (6)$$

4.2.1 Parameterizing the Noise Model from Internet Measurements

Our scan packet noise model had only one noise-related parameter: ϕ , the aggregate rate at which new noise packets were generated on the Internet. In contrast, our victims-based noise model has three parameters: the global birth rate of new noise victims λ , the lifetime of a victim L , and the per-victim scan rate S_{rate} . (As previously mentioned, $\phi = L * \lambda * S_{rate}$.) We must first find a way to estimate these three parameters before we can analyze the effectiveness of victims-based detectors under realistic noise conditions.

We could not find a good source of data from which to estimate L . However, we verified that the results in this section of the paper are not very sensitive to different values of L . Accordingly, we made a reasonable estimate of $L = 24$ hours: a machine will stay compromised and probe the Internet for approximately a day before being noticed and repaired.

Fortunately, given this L value, we can estimate λ and S_{rate} from our own measurements and from data published in the study of Internet background radiation [16]. Based on the data in Table 10 of that paper, a monitored Class

A network saw approximately 500,000 unique IP addresses generating noise over a 24 hour period. Since a Class A network contains 1/256th of all possible IP addresses, we expect that most noise sources will probe at least one address within this network during their lifetime, but to be conservative, we doubled this value to 1,000,000 noise sources observed over a 24 hour period. Given this, we estimate that there are $\lambda = 1,000,000/(24 * 60) = 694$ new noise victims born on the Internet each minute.

Substituting $\lambda = 694$ per minute, $\phi = 3$ billion per minute, and $L = 24$ hours into the relation $\phi = L * \lambda * S_{rate}$, we estimate that $S_{rate} = 3000$ scans per minute. We will use these computed estimates when we incorporate our noise model into the experiments in the quantitative analysis portion of this section

4.2.2 The Detection Condition

We can now define the alarm threshold θ for a victims-based detector. As before, we define θ to be some number of standard deviations c above the mean noise rate. Thus, the victims-based detector will raise an alarm when $observed_{vic}(k, t) > \theta$, where the alarm threshold $\theta = \mu_{noise,k} + c * \sigma_{noise,k}$.

4.3 Modeling a Worm

To model a worm, we use the same RCS model described previously in Section 3.3. We want to understand $W(t)$, the rate at which the detector will observe new worm victims as a function of time. Unfortunately, analytically deriving $W(t)$ is difficult, as the number of new victims observed at time t depends on which victims were previously observed. Instead, we simulated the growth of a Code Red v2-like worm to numerically generate $W(t)$ distributions at several different time values.

4.4 Modeling the Worm Detector

Using our noise and worm models, the total rate at which a detector with k monitors observes new victims at time t is described by the sum of two random variables: the distribution of noise victims N , and the distribution of worm victims over time $W(t)$. This gives us $observed_{vic}(k, t) = N + W(t)$. Because we numerically calculate $W(t)$, we cannot derive a closed-form expression for the distribution $observed_{vic}(k, t)$. However, given values for noise parameters λ , L , and S_{rate} , we can instead numerically calculate it. We can now calculate the probability that a victims-based worm detector raises an alarm by time t after the worm is released. Similar to the scan packet-based detector, this probability is:

$$Prob_{detect}(t) = 1 - \prod_{i=0}^{i=t} (1 - Pr[observed_{vic}(k, i) > \theta]) \quad (7)$$

In the rest of this section, we use our real-world estimates of noise parameters L and S_{rate} , and apply equation 7 to explore how different values of k , λ , and c affect the probability that a victims-based detector will successfully raise an alarm.

4.5 Quantitative Results

Given our new noise and worm models, we can now return to our three questions about victims-based worm detectors. To answer these questions, we again use measurements of the Code Red v2 worm [15] to parameterize our worm model. Then, we use our detector model to explore the effect of

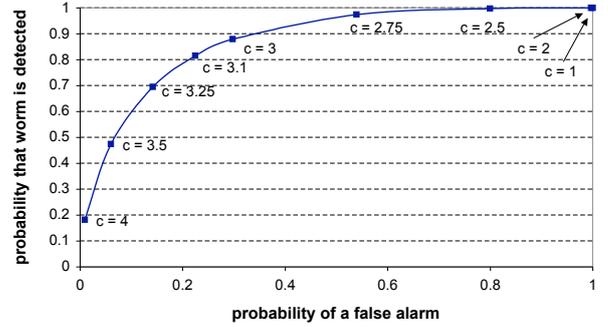


Figure 5: False alarm vs. worm detection probabilities. This figure illustrates the trade-off between the probability of raising a false alarm and the probability of detecting the worm by its critical time, for alarm conditions at various different numbers of standard deviations c above the average noise level. The system considered monitored $k = 2^{16}$ IP addresses, and the global new victim birth rate λ was fixed at $\lambda = 3500$ new victims per minute.

varying the global new victim arrival rate λ and the number of monitored IP addresses k . We continue to model the overall probe rate of Code Red v2 victims at time t as $\psi(t) = 660 * 10.84^{(t/60)}$, where t is measured in minutes. Again, we define the critical point by which the worm should be detected as $t_{crit} = 242$ minutes.

In the experiments to follow, we included our noise model by using the previously computed estimates of our noise model's parameters. Specifically, we chose to hold fixed the values of $L = 24$ hours and $S_{rate} = 3000$ scans per minute while varying the new victim arrival rate λ between 100 and 10,000 victims per minute to understand the impact of different noise levels on a victims-based detector.

4.5.1 Detection Fidelity vs. False Positive Rate

Much like with the scan packet-based detector, variations in the new victim arrival rate observed by the detector mean that a victims-based detector has a non-zero probability of raising a false alarm when a worm outbreak is not in progress. By tuning the value of c , we can once again trade off between the speed at which the worm will be detected and the false alarm probability.

Figure 5 illustrates this tradeoff. We modeled a global new victim birth rate of $\lambda = 3500$ victims per minute, and a detector that monitors $k = 2^{16}$ IP addresses. The graph demonstrates that increasing c decreases both the false alarm and detection probabilities. As before, values of $c = 3.25$ and $c = 3.5$ appear to give us a reasonable balance.

4.5.2 Detection Fidelity vs. the Background Noise Level of the Internet

The additional layer of complexity in our victims-based noise model required us to consider a different metric λ , the global birth rate of new victims, for assessing the background noise level in the Internet. However, given that we are holding the victim lifetime L and per-victim scan rate

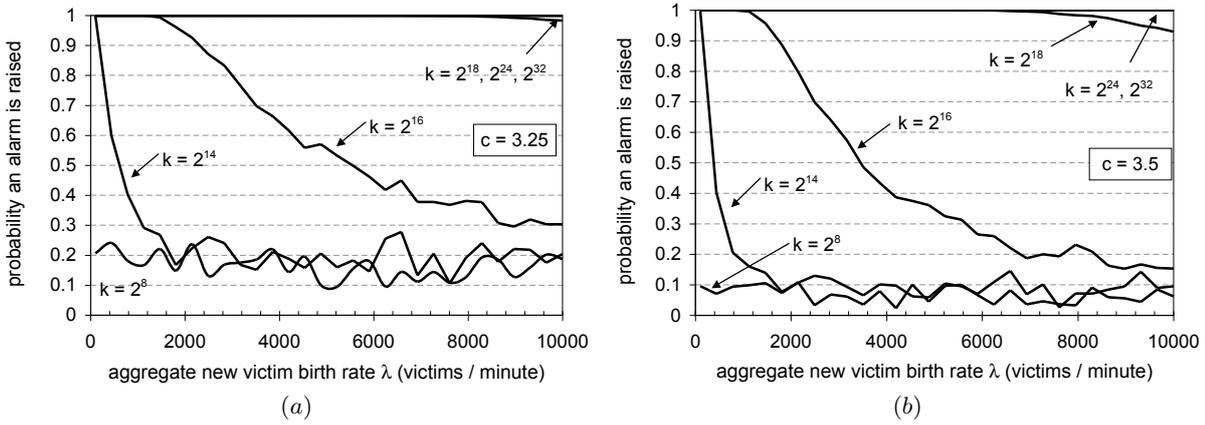


Figure 6: (a) The probability an alarm is raised by time t_{crit} for varying numbers of monitored IP addresses k between $k = 2^8$ and $k = 2^{32}$, where the number of standard deviations c used in calculating the detection threshold θ is $c = 3.25$. (b) The same graph, except the number of standard deviations c is set at 3.5. The fluctuations in these graphs are due to our numerical calculations of some of the distributions, as described in Sections 4.3 and 4.4.

S_{rate} constant, the aggregate background scan rate ϕ is directly proportional to λ . Under these assumptions, an increase in global noise is a direct consequence of an increase in the rate at which noise victims are born.

In this section, we vary the metric λ to quantify how an increased Internet noise level affects victims-based worm detectors. To do this, we use equation 7 to plot the probability $Prob_{detect}(t_{crit})$ that the detector raises an alarm by time t_{crit} , for varying values of the number of monitored IP addresses k , and for varying global new noise victim arrival rates λ . As previously mentioned, we explore a range of values between $\lambda = 100$ and $\lambda = 10,000$ new noise victims born per minute. Figure 6a and 6b show our results for $c = 3.25$ and $c = 3.5$ standard deviations in our detection threshold θ .

As the arrival rate of new noise victims on the Internet increases, the probability that an alarm is raised rapidly decreases down to the baseline false alarm probability of the detector (14% for $c = 3.25$ and 6% for $c = 3.5$). As the baseline noise level λ and noise variations grow, the detector must become more tolerant to larger fluctuations in the number of observed victims, making it more difficult for the detector to spot a statistically significant increase in the arrival rate of victims. As with scan packet-based detectors, increasing noise levels cause a victims-based worm detector to lose the worm signal in the noise.

Note that we did not plot any data for $k = 1$ in these graphs. For only a single monitored IP address, the expected rate at which new victims were observed was so small (a few victims per hour) that the detection threshold no longer is meaningful: the arrival of any new victim would cause an alarm to be raised.

4.5.3 Detection Fidelity vs. the Number of Monitored IP Addresses

To understand how the addition of more monitors impacts the detection fidelity, we once again consider Figures 6a and 6b. As with the scan packet-based detector, increasing the number of monitored IP addresses increases the probability that an alarm is raised by t_{crit} . However, comparing Fig-

ure 6 to the equivalent Figure 2 for the scan packet-based detector, we see that a victims-based detector is more effective for the same number of monitored IP addresses. Both require a large number of monitored IP addresses, but with 2^{16} IP addresses, a victims-based detector performs better than a scan packet-based detector that monitors 2^{18} IP addresses. Adding more monitors does increase detection fidelity. However, for growing, realistic noise levels, victims-based and scan packet-based detectors are only effective for large values of k .

4.6 Summary

Victims-based detectors are also highly sensitive to the amount of background noise in the Internet. As the amount of noise increases in the future, the effectiveness of victims-based detectors will also degrade substantially. Likewise, adding more monitored IP addresses increases the probability of detecting a worm. Though victims-based detectors can make better use of any given number of IP addresses, both the victims-based and scan packet-based detectors require large numbers of IP addresses in order to be effective.

5. CONCLUSIONS

Global scanning worm detectors monitor traffic arriving at unused IP addresses in an attempt to observe the scan traffic associated with a rapidly growing worm. However, a constant deluge of Internet “background noise” also arrives at these detectors. To spot a worm, the detectors must look for statistically significant increases in probe traffic that distinguish the worm signal from random variations in the noise.

In this paper, we examined how Internet background noise affects the ability of global scanning worm detectors to detect worms. To accomplish this, we proposed measurement-inspired statistical models of background noise, and combined them with the random constant spread (RCS) model of worm propagation to calculate the probability that a worm detector would be able to raise an alarm. We analyzed two global scanning worm detection schemes: scan

packet-based and victims-based detectors. We found that scan packet-based detectors are only effective if they monitor a very large number of IP addresses, and that as the noise level of the Internet grows, they quickly lose their ability to detect worms early. While victims-based detectors are somewhat less sensitive to the noise level, they too require a large number of unused IP addresses, and their detection fidelity, too, degrades as noise grows.

The global scanning worm detectors considered in our analysis would likely fall prey to more advanced and clever worms. Evasive techniques such as the use of hit-lists, source IP address spoofing, or slow scanning can easily fool such detectors and allow a worm to spread undetected. For these reasons, and for the limitations identified in our analysis, we conclude that global scanning worm detectors are not a viable long-term strategy for detecting worms early. Local detection schemes and signature-based detectors are much better equipped to deal with the increasing background Internet noise level and the rapidly escalating ability of worms to spread quickly and cleverly.

Acknowledgments

We gratefully acknowledge Krzysztof Gajos, who played an integral role in the genesis of this work, and Paul Beame and Brian Bershad for their valuable feedback. Finally, we also thank David L. Richardson at the University of Cincinnati for his insight. This work was supported in part by NSF CAREER award ANI-0132817 and gifts from Intel Corporation and Nortel Networks.

6. REFERENCES

- [1] G. Bakos and V. H. Berk. Early detection of Internet worm activity by metering ICMP destination unreachable messages. In *Proceedings of the SPIE Conference on Sensors, and Command, Control, Communications and Intelligent*, Orlando, FL, April 2002.
- [2] P. Barford, S. Jha, and V. Yegneswaran. Fusion and filtering in distributed intrusion detection systems. In *Proceedings of the 42nd Annual Allerton Conference on Communication, Control and Computing*, September 2004.
- [3] V. Berk, G. Bakos, and R. Morris. Designing a framework for active worm detection on global networks. In *Proceedings of the First IEEE International Workshop on Information Assurance (IWIA '03)*, Darmstadt, Germany, March 2003.
- [4] V. H. Berk, R. S. Gray, , and G. Bakos. Using sensor networks and data fusion for early detection of active worms. In *Proceedings of AeroSense 2003: SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*, Orlando, FL, April 2003.
- [5] Z. Chen, L. Gao, and K. Kwiat. Modeling the spread of active worms. In *Proceedings of INFOCOM*, San Francisco, CA, March-April 2003.
- [6] E. Cooke, M. Bailey, Z. M. Mao, D. Watson, F. Jahanian, and D. McPherson. Toward understanding distributed blackhole placement. In *Proceedings of the 2004 ACM Workshop on Rapid Malcode*, Fairfax, VA, October 2004.
- [7] D. Dagon, X. Qin, G. Gu, W. Lee, J. Grizzard, J. Levin, and H. Owen. Honeystat: Local worm detection using honeypots. In *Proceedings of RAID 2004*, Sophia Antipolis, France, Sept. 2004.
- [8] D. Daley and J. Gani. *Epidemic Modeling: An Introduction*. Cambridge University Press, 199.
- [9] 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 *20th Annual Computer Security Applications Conference (ACSAC 2004)*, Tucson, AZ, December 2004.
- [10] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, Oakland, CA, May 2004.
- [11] D. M. Kienzle and M. C. Elder. Recent worms: A survey and trends. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode*, Washington, DC, October 2003.
- [12] H.-A. Kim and B. Karp. Autograph: Toward automated, distributed worm signature detection. In *Proceedings of the 13th Usenix Security Symposium (Security 2004)*, San Diego, CA, August 2004.
- [13] M. Liljenstam, D. M. Nicol, V. H. Berk, and R. S. Gray. Simulating realistic network worm traffic for worm warning system design and testing. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode*, Washington, DC, October 2003.
- [14] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. *IEEE Security and Privacy*, 1(4):33–39, July 2003.
- [15] D. Moore, C. Shannon, and J. Brown. Code-red: a case study on the spread and victims of an Internet worm. In *Proceedings of the 2002 Internet Measurement Workshop*, Marseille, France, November 2002.
- [16] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of Internet background radiation. In *Proceedings of IMC 2004*, Sicily, Italy, October 2004.
- [17] X. Qin, D. Dagon, G. Gu, and W. Lee. Worm detection using local networks. Technical Report GIT-CC-04-04, College of Computing, Georgia Institute of Technology, February 2004.
- [18] S. E. Schechter, J. Jung, and A. W. Berger. Fast detecton of scanning worm infections. In *Proceedings of RAID 2004*, Sophia Antipolis, France, September 2004.
- [19] S. Singh, C. Estan, G. Varghese, and S. Savage. Automated worm fingerprinting. In *Proceedings of the OSDI '04*, San Francisco, CA, December 2004.
- [20] E. H. Spafford. The Internet worm program: An analysis. *ACM SIGCOMM Computer Communication Review*, 19(1):17–57, January 1989.
- [21] S. Staniford, V. Paxson, and N. Weaver. How to Own the Internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium (Security '02)*, San Francisco, CA, August 2002.
- [22] N. Weaver, S. Staniford, and V. Paxson. Very fast containment of scanning worms. In *Proceedings of the 13th Usenix Security Symposium (Security 2004)*, San

- Diego, CA, August 2004.
- [23] D. Whyte, E. Kranakis, and P. van Oorschot. Dns-based detection of scanning worms in an enterprise network. In *Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS 2005)*, San Diego, CA, February 2005.
- [24] J. Wu, S. Vangala, L. Gao, and K. Kwiat. An effective architecture and algorithm for detecting worms with various scan techniques. In *Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS '04)*, San Diego, CA, February 2004.
- [25] C. Zou, L. Gao, W. Gong, and D. Townsley. Monitoring and early warning for Internet worms. In *Proceedings of the 10th ACM Conference on Computer and Communications Security*, Washington, DC, Oct. 2003.
- [26] C. Zou, W. Gong, and D. Towsley. Code red worm propagation modeling and analysis. In *Proceedings of the 9th ACM Conference on Computer and Communication Security*, Washington, DC, November 2002.
- [27] C. Zou, D. Towsley, and W. Gong. Worm propagation modeling and analysis under dynamic quarantine defense. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode*, Washington, DC, October 2003.
- [28] C. Zou, D. Towsley, W. Gong, and S. Cai. Routing worm: A fast, selective attack worm based on IP address information. Technical Report TR-03-CSE-06, University of Massachusetts, November 2003.