The Monitoring and Early Detection of Internet Worms

Cliff C. Zou, *Member, IEEE*, Weibo Gong, *Fellow, IEEE*, Don Towsley, *Fellow, IEEE*, and Lixin Gao, *Member, IEEE*

***Abstract*—After many Internet-scale worm incidents in recent years, it is clear that a simple self-propagating worm can quickly spread across the Internet and cause severe damage to our society. Facing this great security threat, we need to build an early detection system that can detect the presence of a worm in the Internet as quickly as possible in order to give people accurate early warning information and possible reaction time for counteractions. This paper first presents an Internet worm monitoring system. Then, based on the idea of "detecting the trend, not the burst" of monitored illegitimate traffic, we present a "trend detection" methodology to detect a worm at its early propagation stage by using Kalman filter estimation, which is robust to background noise in the monitored data. In addition, for uniform-scan worms such as Code Red, we can effectively predict the overall vulnerable population size, and estimate accurately how many computers are really infected in the global Internet based on the biased monitored data. For monitoring a nonuniform scan worm, especially a sequential-scan worm such as Blaster, we show that it is crucial for the address space covered by the worm monitoring system to be as distributed as possible.**

***Index Terms*—Computer network security, early detection, Internet worm, network monitoring.**

I. Introduction

SINCE the Morris worm in 1988 [33], the security threat posed by worms has steadily increased, especially in the last several years. Code Red appeared on July 19, 2001 [27], which began the new wave of Internet-scale worm attacks. After that, Code Red II, Nimda, Slammer, Blaster, Sasser, and Witty have repeatedly attacked the Internet [9] and caused great damage to our society.

Currently, some organizations and security companies, such as the CERT, CAIDA, and SANS Institute [7], [8], [32], are monitoring the Internet and paying close attention to any abnormal traffic. When they observe abnormal network activities, their security experts immediately analyze these incidents. Given the fast-spreading nature of Internet worms and their severe damage to our society, it is necessary to set up a nation-scale worm-monitoring and early-warning system. (The U.S. Department of Homeland Security launched a "Cybersecurity Monitoring Project" in October 2003 [40]).

A straightforward way to detect an unknown (zero-day) worm is to use various anomaly detection systems. There are many well-studied methods or systems in the anomaly "intrusion detection" research area, for example, the "IDES" [13], "NIDES" [5] and "eBayes" [39] from SRI International; the anomaly intrusion detection method [15] based on "sequences of system calls"; the automatic model-construction intrusion detection system based on data-mining of audit data [24], etc.

Anomaly intrusion-detection systems usually concentrate on detecting attacks initiated by hackers. In the case of Internet worm detection, we find that we can take advantage of the difference between a worm's propagation and a hacker's intrusion attack. A worm code exhibits simple attack behaviors; all computers infected by a worm send out infection traffic that has similar statistical characteristics. Moreover, a worm's propagation in the Internet usually follows some dynamic models because of its large-scale distributed infection. On the other hand, a hacker's intrusion attack, which is more complicated, usually targets one or a set of specific computers and does not follow any well-defined dynamic model in most cases.

Based on this observation, we present a new detection methodology, "*trend detection*," by using the principle "detecting monitored traffic *trend*, not *burst*" [45]. Our "trend detection" system attempts to detect the dynamic *trend* of monitored traffic based on the fact that, at the early stage, a worm propagates exponentially with a *constant*, *positive* exponential rate. The "trend" we try to detect is the exponential growth trend of monitored traffic.

Based on worm propagation dynamic models, we detect the presence of a worm in its early propagation stage by using the *Kalman filter* estimation algorithm, which is robust to background noise existing in the monitored data. The Kalman filter is activated when the monitoring system encounters a surge of illegitimate scan activities. If the infection rate estimated by the Kalman filter, which is also the exponential growth rate of a worm's propagation at its early stage, *stabilizes* and *oscillates* slightly around a *constant positive* value, we claim that the illegitimate scan activities are mainly caused by a worm, even if the estimated worm infection rate is still not well converged. If the monitored traffic is caused by nonworm noise, the traffic will not have the exponential growth trend, and the estimated value of the infection rate would converge to zero or oscillate around zero. In other words, the Kalman filter is used to detect the presence of a worm by detecting the *trend*, not the *burst*, of the observed illegitimate traffic. In this way, the noisy illegiti-

Manuscript received February 13, 2004; revised August 17, 2004; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor V. Paxson. This work was supported in part by the Army Research Office under Contract DAAD19-01-1-0610, the Defense Advanced Research Projects Agency under Contract DOD F30602-00-0554, the National Science Foundation under Grants EIA-0080119, ANI9980552, and ANI-0208116, and the Air Force Research Laboratory.

C. C. Zou is with the School of Computer Science, University of Central Florida, Orlando, FL 32816-2362 USA (e-mail: czou@cs.ucf.edu).

W. Gong and L. Gao are with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, MA 01003 USA (e-mail: gong@ecs.umass.edu; lgao@ecs.umass.edu).

D. Towsley is with the Department of Computer Science, University of Massachusetts, Amherst, MA 01003-9264 USA (e-mail: towsley@cs.umass.edu).

Digital Object Identifier 10.1109/TNET.2005.857113

mate traffic in the Internet we observe everyday will not cause too many false alarms in our detection system.

In addition, we present a formula to predict a worm's vulnerable population size when the worm is still at its early propagation stage. We also present a formula to correct the bias in the number of infected hosts observed by a monitoring system. This bias has been mentioned in [10] and [29], but neither of them has presented methods to correct it. In this way, we can know how many computers in the global Internet are really infected based on local monitored data. Furthermore, we point out that in designing a worm monitoring system, the address space covered by a monitoring system should be as distributed as possible in order to monitor and detect nonuniform scan worms, especially a sequential scan worm such as Blaster.

The rest of this paper is organized as follows. Section II surveys related work. Section III introduces the worm-propagation models used in this paper. Section IV describes briefly the monitoring system. Data collection and the bias correction formula for monitored biased data are described in Section V. Section VI presents the Kalman filters for early worm detection, and the formula to predict the vulnerable population size. We conduct extensive simulation experiments and show the major results in Section VII. In Section VIII, we discuss limitations and possible future work. Section IX concludes this paper.

II. Related Work

In recent years, people have paid attention to the necessity of monitoring the Internet for malicious activities. Symantec Corporation has an "enterprise early warning solution" [1], which collects IDS and firewall attack data from the security systems of thousands of partners to keep track of the latest attack incidents. The SANS Institute set up the Internet Storm Center [17], which could gather the log data from participants' intrusion detection sensors distributed around the world.

In the academic research area, Moore *et al.* [29] presented the concept of "network telescope" to use a small fraction of unused IP space for observing security incidents in the global Internet. Pang *et al.* [30] called the abnormal traffic to unused IP space "background radiation," and presented detailed measurement analysis and characterization of such monitored traffic. From another perspective, Berk *et al.* [6] proposed a monitoring system by collecting ICMP "Destination Unreachable" messages generated by routers for packets to unused IP addresses. In "honeypot" research, Honeynet [16] is a network of honeypots to gather comprehensive information of attacks; "Honeyd" presented by Provos [31] is a virtual honeypot framework to simulate many virtual computer systems at the network level.

The monitoring system we present in this paper can be incorporated into the current monitoring systems such as the SANS Internet Storm Center. Our contribution in this context is to point out the infrastructure specifically for worm monitoring, and what data should be collected for early detection of worms. We also emphasize the functionality of egress monitors, which has been overlooked in previous research. Worm monitors can be set up as ingress and egress filters on routers, which cover more IP space and gather more comprehensive information than the log data collected from intrusion detection sensors or firewalls for current monitoring systems.

In the area of worm modeling, Kephart, White, and Chess of IBM performed a series of studies from 1991 to 1993 on viral infection based on epidemiology models [21], [20], [22]. Staniford *et al.* [37] used the classical epidemic model to model the spread of Code Red right after the Code Red incident on July 19, 2001; they also proposed several more vicious worms in the same paper. Zou *et al.* [46] presented a "two-factor" worm model that considered both the effect of human countermeasures and the effect of the congestion caused by extensive worm scan traffic. Chen *et al.* [10] presented a discrete-time version worm model that considered the patching and cleaning effect during a worm's propagation.

For a fast spreading worm such as Slammer, it is necessary to have automatic response and mitigation mechanisms. Moore *et al.* [28] discussed the effect of Internet quarantine for containing the propagation of a worm. Williamson [42] proposed a general rate-limiting "throttling" method to greatly constrain infection traffic sent out by infected hosts while not affecting normal traffic. Zou *et al.* [47] presented a feedback dynamic quarantine system for automatic mitigation by borrowing two principles used in the epidemic disease control in the real world: "preemptive quarantine" and "feedback adjustment." Staniford [36] presented automatic worm quarantine for enterprise networks by using CounterMalice devices to separate an enterprise network into many isolated subnetworks. Weaver *et al.* [41] further improved the CounterMalice quarantine by designing hardware-centered quarantine algorithms. Jung *et al.* [18], [19] proposed a "threshold random walk" algorithm to quickly detect and block worm scans based on the excessive illegal scans sent out by worm-infected hosts. EarlyBird in [35] and Autograph in [23] detect and block worm spreading through identifying the common characteristics, such as a common bit-string, among all infection network traffic of a worm. Wu *et al.* [43] proposed a victim counter-based detection algorithm that tracks the increased rate of new infected hosts.

Our early detection system tries to detect the presence of a worm in the global Internet. For worm detection in local networks, Staniford-Chen *et al.* [38] presented GrIDS, which can detect worm-infected hosts in a local network through building the worm's infection graph (based on monitored infection traffic between all hosts); Dagon *et al.* [11] presented a "honeystat" worm detection method by correlating infection statistics provided by local honeypots when a worm tries to infect them. The CounterMalice quarantine device [36] also tries to detect infected hosts in local enterprise networks.

We assume that the IP infrastructure is the current IPv4. If IPv6 replaces IPv4, the vast IP space of the IPv6 would make it futile for a worm to propagate through blindly IP scanning [50]. However, we believe IPv6 will not replace IPv4 in the near future, and worms will continue to use various random scan techniques to spread in the Internet.

III. Worm Propagation Model

A promising approach for modeling and evaluating the behavior of malware is the use of *fluid models.* Fluid models are

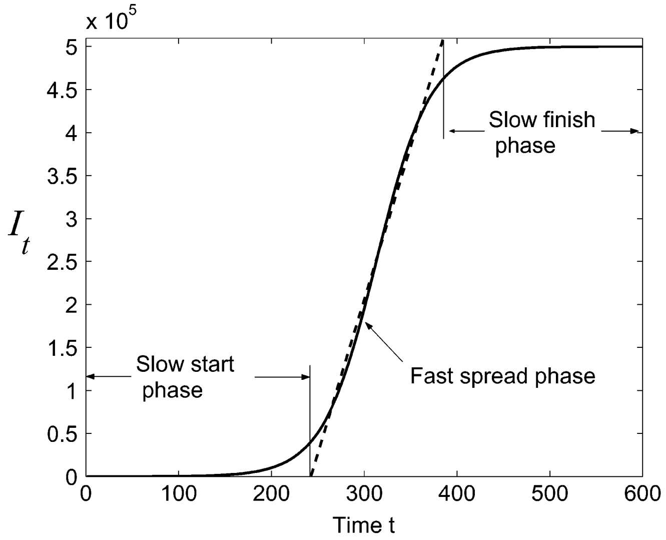


Fig. 1. Worm propagation model.

TABLE I
NOTATIONS IN THIS PAPER

Notation	Definition
N	Number of hosts under consideration
Δ	The length of monitoring interval (time unit in discrete-time model)
I_t	Number of infected hosts at time $t\Delta$
β	Pairwise rate of infection
α	Infection rate per infected host, $\alpha = \beta N$
C_t	Cumulative number of infected hosts monitored by time $t\Delta$
Z_t	Monitored worm scan rate at time $t\Delta$
η	Average scan rate per infected host
p	Probability a worm scan is monitored
R	Variance of observation error of C_t
ν_t, ν'_t, ν''_t	Observation noise in worm models
τ_t	Weight in Kalman filter formula
y_t	Measurement data in Kalman filter
w_t	White noise in measurement y_t at time $t\Delta$
δ	Constant in equation $y_t = \delta I_t + w_t$
MWC	Abbr. of "Malware Warning Center"
$\hat{\alpha}$	Estimated value of α
A^τ	Transpose of a matrix A
$N(\mu, \sigma^2)$	Normal distribution with mean μ and variance σ^2
$E[X]$	Mean value of random variable X

appropriate for a system that consists of a large number of vulnerable hosts, which is the case for Internet-scale worm propagation modeling. In epidemiology research, the simple epidemic model [12] assumes that each host resides in one of two states: susceptible or infected. The model further assumes that once infected by a virus or a worm, a host remains in the infectious state forever. Thus, any host has only one possible state transition: susceptible → infected. The simple epidemic model for a finite population is

$$\frac{dI_t}{dt} = \beta I_t[N - I_t] \tag{1}$$

where I_t is the number of infected hosts at time t, N is the size of the vulnerable population before any of them is infected, and β is called the *pairwise rate of infection* in epidemic studies [12]. At $t = 0$, I_0 hosts are initially infected while the remaining $N - I_0$ hosts are susceptible.

This model captures the basic mechanism of the propagation of a random-scan worm, especially for the initial stage of a worm's propagation when the effect of human counteractions and network congestion is ignorable [46]. A sequential-scan worm (such as Blaster), or a subnet-scan worm (such as Code Red II), propagates differently from a uniform-scan worm. However, through simulation and analysis, [48] showed that the propagation of these worms still closely follows the epidemic model (1).

The epidemic model (1) has its limitations. First, the model assumes that all hosts can directly contact each other, which means it is not suitable for a topological worm (such as Morris [33]) or a mass-mailing e-mail virus [49]. Second, if worm-infected hosts collaborate their infection efforts, such as the divide-and-conquer approach or the permutation scan used by the Warhol worm [37], then the worm's propagation will deviate from the epidemic model.

For the epidemic model (1), Fig. 1 shows the dynamics of I_t as time goes on for one set of parameters. We can roughly partition a worm's propagation into three phases: the slow start phase, the fast spread phase, and the slow finish phase. During the *slow start phase*, since $I_t \ll N$, the number of infected hosts increases exponentially (model (1) becomes $dI_t/dt \approx \beta N I_t$). After many hosts are infected and then participate in infecting others, the worm enters the *fast spread phase* where vulnerable hosts are infected at a fast, near linear speed. When most vulnerable computers have been infected, the worm enters the *slow finish phase* because the few leftover vulnerable computers are difficult for the worm to search out. Our task is to detect the presence of a worm in the Internet in its slow start phase as early as possible.

At the early stage of a worm's propagation, $N - I_t \approx N$. Since we want to detect a worm at its slow start phase, we can accurately model a worm's propagation at this stage by using the exponential growth model:

$$\frac{dI_t}{dt} = \beta N I_t \tag{2}$$

which has the solution

$$I_t = I_0 e^{\beta N t}. \tag{3}$$

In this paper, we use the discrete-time model for worm modeling and early detection. Time is divided into intervals of length Δ, where Δ is the discrete time unit. To simplify the notations, we use "t" as the discrete time index from now on. For example, I_t means the number of infected hosts at the real time $t\Delta$. The discrete-time version of the simple epidemic model (1) can be written as [12]

$$I_t = (1 + \alpha\Delta)I_{t-1} - \beta\Delta I_{t-1}^2 \tag{4}$$

where

$$\alpha = \beta N. \tag{5}$$

We call α the *infection rate* because it is the average number of vulnerable hosts that can be infected per unit of time by one infected host during the early stage of a worm's propagation.

For the exponential worm model (2), we derive an autoregressive (AR) discrete-time model similar to (4):

$$I_t = (1 + \alpha\Delta)I_{t-1} \tag{6}$$

which is called *AR exponential model* in this paper. We can also derive another discrete-time model by taking the logarithm on both sides of the solution (3):

$$lnI_t = t\Delta\alpha + lnI_0 \tag{7}$$

which is called *transformed linear model* in this paper.

It should be mentioned that it is hard to choose an appropriate Δ before we know a worm's propagation speed. We will further discuss this issue in Sections VII and VIII.

Before we go on to discuss how to use the worm models to detect and predict worm propagation, we first present the monitoring system design in Section IV, and discuss data collection issues in Section V.

IV. Monitoring System

In this section, we propose the architecture of a worm monitoring system. The monitoring system aims to provide comprehensive observation data on a worm's activities for the early detection of the worm. The monitoring system consists of a *Malware Warning Center* (MWC) and distributed monitors as shown in Fig. 2.

A. Monitoring System Architecture

There are two kinds of monitors: ingress scan monitors and egress scan monitors. *Ingress scan monitors* are located on gateways or border routers of local networks. They can be the ingress filters on border routers of the local networks, or separated passive network monitors. The goal of an ingress scan monitor is to monitor scan traffic coming into a local network by logging incoming traffic to unused local IP addresses. For management reasons, local network administrators know how addresses inside their networks are allocated; it is relatively easy for them to set up the ingress scan monitor on routers in their local networks. For example, during the Code Red incident on July 19, 2002, a "/8" network at UCSD and two "/16" networks at Lawrence Berkeley Laboratory were used to collect Code Red scan traffic. All port 80 TCP SYN packets coming in to nonexistent IP addresses in these networks were considered to be Code Red scans [27].

An *egress scan monitor* is located at the egress point of a local network. It can be set up as a part of the egress filter on the routers of a local network. The goal of an egress scan monitor is to monitor the outgoing traffic from a network to infer the scan behavior of a potential worm.

Ingress scan monitors listen to the global traffic in the Internet; they are sensors for global worm incidents (called "network telescope" in [29]). However, it is difficult to determine the behavior of each individual infected host from the data collected by ingress scan monitors because such monitors can only capture a small fraction of scans sent out by an infected host. On the other hand, if a computer inside a local network is infected, the egress scan monitor on this network's routers can observe most of the scans sent out by the compromised computer. Therefore, an egress scan monitor is good at observing a worm's scan rate and scan distribution, e.g., uniform scan (such as Code Red), or subnet scan (such as Code Red II and Sasser), or sequential scan (such as Blaster).

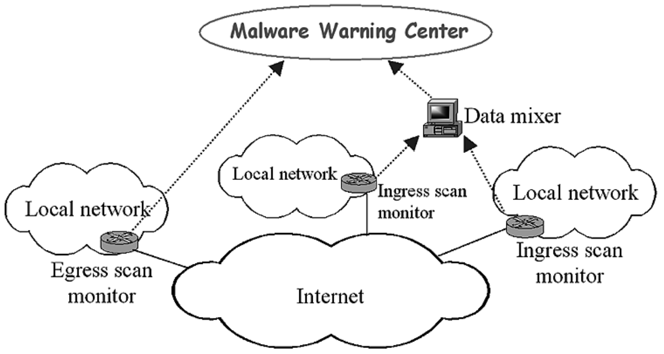


Fig. 2. Generic worm monitoring system.

In order to provide early warning in real time, distributed monitors are required to send observation data to the MWC continuously without significant delay, even when a worm's scan traffic has caused congestion to the Internet. For this reason, a tree-like hierarchy of *data mixers* can be set up between monitors and MWC: MWC is the root; the leaves of the tree are monitors. The monitors nearby a data mixer send observed data to the data mixer. After fusing the data together, the data mixer passes the data to a higher level data mixer or directly to MWC. An example of data fusion is the removal of repetitive IP addresses from the list of infected hosts. However, the tree structure of data mixers creates single points of failure, thus there is a tradeoff in designing this hierarchical structure.

B. Location for Distributed Monitors

Ingress scan monitors on a local network may need to be put on several routers instead of only on the border router because the border router may not know the usage of all IP addresses of this local network. In addition, since worms might choose different destination addresses by using different preferences, such as subnet scanning, we need to use distributed address spaces with different sizes and characteristics to ensure proper coverage. Later on, we show that for monitoring nonuniform scan worms such as Blaster, the IP space covered by a monitoring system should be as distributed as possible.

For egress scan monitors, worms on different infected computers may exhibit different scan behaviors. For example, Slammer's scan rate is constrained by an infected computer's bandwidth [26]. Therefore, we need to set up distributed egress filters to record the scan behaviors of many infected hosts at different locations and in different network environments. In this way, the monitoring system could obtain a comprehensive view of the behaviors of a worm. For example, it can get a better observation of the *average* number of scans an infected host sends out per unit of time.

V. Data Collection and Bias Correction

After setting up a monitoring system, we need to determine what kind of data should be collected. The main task for an egress scan monitor is to determine the behaviors of a worm, such as the worm's average scan rate and scan distribution. Denote η as the "average worm scan rate," which is the *average* number of scans sent out by an infected host in a unit time. Thus, in a monitoring interval Δ, an infected host sends out on average $\eta\Delta$ scans. The ingress scan monitors record two types of data: the number of scans they receive, and the source IP addresses of computers that send scans to them.

If all monitors send observation data to MWC once in every monitoring interval, then MWC obtains the following observation data at each discrete time epoch $t, t = 1, 2, \ldots$:

1) the number of scans monitored in a monitoring interval from discrete time $(t-1)$ to t, denoted by Z_t;
2) the cumulative number of infected hosts observed by the discrete time t, denoted by C_t;
3) a worm's scan distribution;
4) a worm's average scan rate η.

Let us first focus on worms that uniformly scan the Internet. Let p denote the probability that a worm's scan is monitored by a monitoring system. If ingress scan monitors cover m IP addresses, then a worm's scan has the probability $p = m/2^{32}$ to hit the monitoring system. We assume that in the discrete-time model all changes happen right before the discrete time epoch t, then we have

$$E[Z_t] = \eta \Delta p I_{t-1}. \tag{8}$$

In order to detect nonuniform scan worms, it is important to observe a worm's scan distribution since it affects how we should use monitored data in our early detection. For example, if a subnet-scan worm has a higher preference in scanning local "/16" IP space, we can remove these "/16" local scans from monitored data Z_t in order to observe the worm's global scan trend. For a sequential scan worm, as explained later in Section IV, we can first apply a low-pass filter on monitored data Z_t to remove its excessive high-frequency noise before using the Kalman filter for early detection.

An egress scan monitor can observe the scan rates of all its internal infected hosts. If egress scan monitors cover many infected hosts, and if the scan rate of the worm does not vary too much, then we can obtain an accurate estimation of η, the worm's *average* scan rate. However, it is hard for the monitoring system to obtain an accurate estimate of η for a bandwidth-limited worm, such as Slammer or Witty, since the worm's scan rate could vary over several orders of magnitude [26], [34]. In this paper, η is used both in the following "bias correction" and in estimating the vulnerable population size in Section VI. We should keep in mind that both procedures will have more errors when we deal with a bandwidth-limited worm.

A. Correction of Biased Observation C_t

For a uniform-scan worm, each worm scan has a small probability p of being observed by a monitoring system, thus an infected host will send out many scans before one of them is observed by ingress scan monitors. This process can be modeled as a *Bernoulli trial* with a small success probability p. Therefore, the number of infected hosts monitored by the discrete time t, C_t, is not proportional to I_t. This bias has been mentioned in [10] and [29], but neither of them have presented methods to correct the bias. In the following, we present an effective way to obtain an accurate estimate for the number of infected hosts I_t based on C_t and η. Although such a bias correction is not essential to a worm's early detection since we can use monitored data Z_t, it is important for us to know how many computers in the global Internet are really infected.

In the real world, different infected hosts of a worm have different scan rates. To derive the bias correction formula, let us first assume that all infected hosts have the same scan rate η (we will show the effect of removing this assumption in the following simulation). In a monitoring interval Δ, a worm sends out on average $\eta\Delta$ scans, thus the monitoring system has the probability $[1-(1-p)^{\eta\Delta}]$ to observe at least one scan from an infected host in a monitoring interval.

At the discrete time $(t-1)$, the monitoring system has observed C_{t-1} infected hosts among the overall infected ones I_{t-1}. During the next monitoring interval from discrete time $(t-1)$ to t, every host of the as-yet unobserved ones, $I_{t-1} - C_{t-1}$, has the probability $[1-(1-p)^{\eta\Delta}]$ to be observed. Suppose in the discrete-time model, all changes happen right before the discrete time epoch t, then the average number of infected hosts monitored by discrete time t conditioned on C_{t-1} is

$$E[C_t | C_{t-1}] = C_{t-1} + (I_{t-1} - C_{t-1})[1-(1-p)^{\eta\Delta}]. \tag{9}$$

Removing the conditioning on C_{t-1} yields

$$E[C_t] = E[C_{t-1}] + (I_{t-1} - E[C_{t-1}])[1-(1-p)^{\eta\Delta}]. \tag{10}$$

From (10), we can derive the formula for I_t as

$$I_t = \frac{E[C_{t+1}] - (1-p)^{\eta\Delta} E[C_t]}{1-(1-p)^{\eta\Delta}}. \tag{11}$$

Since $E[C_t]$ is unknown in one incident of a worm's propagation, we replace $E[C_t]$ by C_t and derive the estimate of I_t as

$$\hat{I}_t = \frac{C_{t+1} - (1-p)^{\eta\Delta} C_t}{1-(1-p)^{\eta\Delta}}. \tag{12}$$

Now we analyze how the statistical observation error of C_t affects the estimated value of I_t. Without considering nonworm noise, suppose the observation data C_t is

$$C_t = E[C_t] + w_t \tag{13}$$

where the statistical observation error w_t is a white noise with variance R. Substituting (13) into (12) and replacing $E[C_t]$ by I_t from (11) yields

$$\hat{I}_t = I_t + \mu_t \tag{14}$$

where the error μ_t is

$$\mu_t = \frac{w_{t+1} - (1-p)^{\eta\Delta} w_t}{1-(1-p)^{\eta\Delta}}. \tag{15}$$

Since $E[\mu_t] = 0$, the estimated value $\hat{I}_t$ is unbiased (under the assumption that all infected hosts have the same scan rate η). The variance of the error of $\hat{I}_t$ is

$$\mathrm{Var}[\mu_t] = E\left[\mu_t^2\right] = \frac{1 + (1-p)^{2\eta\Delta}}{[1-(1-p)^{\eta\Delta}]^2} R. \tag{16}$$

The equation above shows that $Var[\mu_t]$ is always larger than R, which means the statistical error of observation C_t is amplified by the bias correction formula (12). If ingress scan monitors

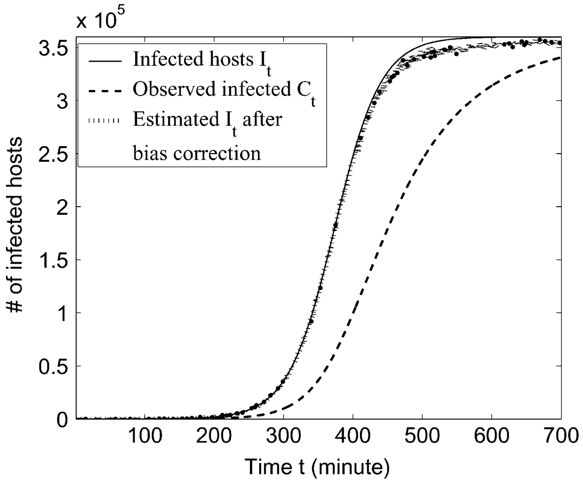


Fig. 3. Estimate $\hat{I}_t$ based on the biased observation data C_t (monitoring 2^{17} IP space).

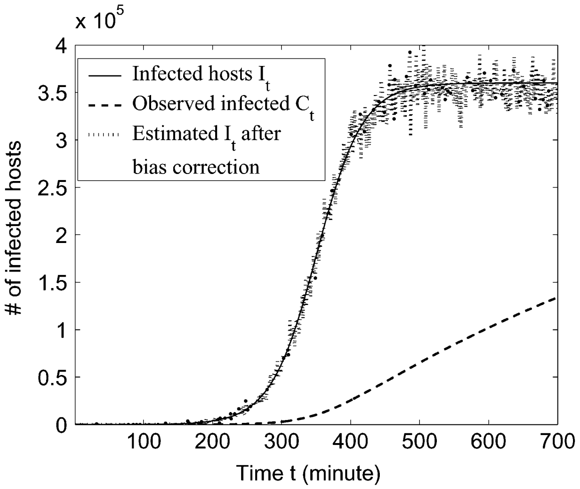


Fig. 4. Estimate $\hat{I}_t$ based on the biased observation data C_t (monitoring 2^{14} IP space).

cover less IP space, p would decrease, then (16) shows that the estimate $\hat{I}_t$ would become noisier.

We simulate Code Red propagation to check the accuracy of the bias correction formula (12). In the simulation, $N = 360\,000$; the monitoring interval Δ is one minute; the average worm scan rate is $\eta = 358$ per minute. The monitoring system covers 2^{17} IP addresses (equal to two Class B networks). Because different infected hosts have different scan rates, we assume each infected host has a scan rate x that is predetermined by the normal distribution $N(\eta, \sigma^2)$, where $\sigma = 100$ in the simulation (x is bounded by $x \geq 1$. We will explain how we choose these parameters in Section VII). The simulation result is shown in Fig. 3.

Fig. 3 shows that the observed number of infected hosts, C_t, deviates substantially from the real value I_t. After the bias correction by using (12), the estimate $\hat{I}_t$ matches I_t well in the simulation before the worm enters the slow finish phase ($\hat{I}_t$ deviates from I_t in the slow finish phase). In deriving the bias correction formula (12), we have assumed that all hosts have the same scan rate η, which is not the case in this simulation. In this simulation, some hosts have very small scan rates; these hosts will take much longer time to hit the monitoring system than others. Thus, in the slow finish phase, many unobserved infected hosts are the ones with very low scan rates. Therefore, during the slow finish phase, the bias correction formula has an error due to the decreasing of the average scan rate η for those unobserved infected hosts. In fact, we have run many other simulations by letting all hosts to have the same scan rate η (i.e., let $\sigma = 0$). In these cases, the $\hat{I}_t$ after bias correction always matches well with I_t without bias.

The bias correction error the appears in a worm's slow finish phase will become larger as the worm-infected hosts have more variable scan rates, especially for bandwidth-limited worms such as Slammer [26] and Witty [34].

Fig. 4 shows the simulation results if the monitoring system only covers 2^{14} IP addresses. The estimate $\hat{I}_t$ after the bias correction is still accurate, but noisier because of the error amplification effect described by (16).

It should be emphasized that the bias correction (12) is derived based on uniform scanning, thus it is accurate for a uniform-scan worm, such as Code Red. For other worms, such as a subnet-scan worm (e.g., Code Red II), or an imperfect uniform-scan worm (e.g., Slammer), the bias correction (12) could possibly produce certain error in its estimation.

The bias correction (12) assumes that we treat a host as infected upon receiving its first illegal scan to our empty IP space. If the nonworm background noise in monitored data is small compared with worm scan traffic, the bias correction formula can still provide a good estimate $\hat{I}_t$. If we want to remove the background noise in the monitored data before using the bias correction formula, we have to wait for some time before estimating $\hat{I}_t$ since we may be able to detect an infected host accurately only after we have received several illegal scans from it [18], [41]. When we detect an infected host, we check our monitored data to find out when it sends the first illegal scan to us, e.g., within the last hour; then, we use the bias correction (12) to estimate the global infected hosts in the Internet an hour ago.

VI. Early Detection and Estimation of Worm Virulence

In this section we present estimation methods based on recursive filtering algorithms (e.g., Kalman filters [4]) for stochastic dynamic systems. At MWC, we recursively estimate the parameter α based on observation data at each monitoring interval in order to detect a worm at its early propagation stage.

Let $y_1, y_2, \ldots, y_t$ be the measurement data used by a Kalman filter estimation algorithm. Suppose the observations have one monitoring interval delay

$$y_t = \delta I_{t-1} + w_t \tag{17}$$

where w_t is the observation error. δ is a constant ratio: if we use Z_t as y_t, then $\delta = \eta \Delta p$ as shown in (8); if we use $\hat{I}_{t-1}$ derived from C_t by the bias correction (12), then $\delta = 1$.

A. Early Detection Based on Kalman Filter Estimation

In Section III, we presented three discrete-time worm models: the epidemic model (4), the AR exponential model (6), and the transformed linear model (7). In this section, we present three Kalman filter estimation algorithms, one for each discrete-time model.

From (17), we have

$$I_{t-1} = y_t/\delta - w_t/\delta. \tag{18}$$

First, we use the simple epidemic model (4). Substituting (18) into the worm model (4) yields an equation describing the relationship between y_t and α, β:

$$y_t = (1+\alpha\Delta)y_{t-1} - \frac{\beta\Delta}{\delta}y_{t-1}^2 + \nu_t \tag{19}$$

where the noise ν_t is

$$\nu_t = w_t - (1+\alpha)w_{t-1} - \beta\Delta\left(w_{t-1}^2 - 2y_{t-1}w_{t-1}\right)/\delta. \tag{20}$$

A recursive least square algorithm for α and β can be cast into a standard Kalman filter format [4], [25]. Let $\hat{\alpha}_t$ and $\hat{\beta}_t$ denote the estimated value of α and β at the discrete time t, respectively. Define the system state as $X_t = \begin{bmatrix} 1+\Delta\alpha \\ -\beta\Delta/\delta \end{bmatrix}$. If we denote $H_t = [y_{t-1}\ y_{t-1}^2]$, then the system is described by

$$\begin{cases} X_t = X_{t-1} \\ y_t = H_t X_t + \nu_t. \end{cases} \tag{21}$$

The Kalman filter in estimating X_t is

$$\begin{cases} H_t = [y_{t-1}\ y_{t-1}^2] \\ K_t = P_{t-1}H_t^\tau/(H_t P_{t-1} H_t^\tau + 1/\tau_t) \\ P_t = (I - K_t H_t)P_{t-1} \\ \hat{X}_t = \hat{X}_{t-1} + K_t(y_t - H_t\hat{X}_{t-1}) \end{cases} \tag{22}$$

where τ_t is the weight of the tth error term in the Least Square (LS) estimation algorithm [25]. We can use it to adjust whether our estimation should rely more on recently monitored data (τ_t increases as t increases) or equally on all monitored data (τ_t is a constant).

ν_t in (20) is a correlated noise. The Kalman filter (22) can be extended to consider such correlated noise to derive unbiased estimates of α and β in theory (such as an *extended Kalman filter* [4]). However, an unbiased Kalman filter introduces additional parameters to estimate, thus the new filter will converge slower than the proposed filter (22). In fact, we have designed an extended Kalman filter and our experiments confirm this conjecture. In this paper, the primary objective is to derive a rough estimate of α as quickly as possible for early detection of a worm. Therefore, it is better to use the simple Kalman filter (22) here.

If we use Z_t as the measurement data y_t in the Kalman filter but do not know δ (e.g., when we do not have data from egress scan monitors), we can still estimate the infection rate α by letting $\delta = 1$. The Kalman filter (22) does not depend on δ in estimating α; the value of δ only affects the estimated value of β.

Now we consider the AR exponential model (6). Substituting (18) into model (6) yields

$$y_t = (1+\Delta\alpha)y_{t-1} + \nu_t' \tag{23}$$

where the noise ν_t' is

$$\nu_t' = w_t - (1+\Delta\alpha)w_{t-1}. \tag{24}$$

Equation (23) has the similar format as (19). Thus, if we change X_t and H_t in the original Kalman filter (22) to $X_t = [1+\Delta\alpha]$ and $H_t = [y_{t-1}]$, we derive a new Kalman filter that is based on the AR exponential model (6).

For the transformed linear model (7), we can derive the formula of y_t as

$$\ln(y_t - w_t) = (t-1)\Delta\alpha + \ln I_0. \tag{25}$$

It is difficult or impossible for us to know when a worm starts spreading, i.e., we do not know the absolute value t. We only know a relative time $t - t_0$ where $t_0 > 0$ is the discrete time when we activate our Kalman filter detection system; the true value of t_0 is not known. It means that in the worm model we can only use variable $t - t_0$ but not t.

If we let $\ln(y_t - w_t) = \ln(y_t) - \nu_t''$, from (25) we can derive the relationship between y_t and a worm's infection rate α as

$$\ln(y_t) = (t-t_0)\Delta\alpha + K + \nu_t'' \tag{26}$$

where

$$K = (t_0 - 1)\Delta\alpha + \ln\delta + \ln I_0 \tag{27}$$

and the noise ν_t'' is

$$\nu_t = \ln(y_t) - \ln(y_t - w_t). \tag{28}$$

When we activate the Kalman filter in our early detection system, $y_t > 1$ and $y_t - w_t > 1$ always hold. From (28) we know that $\mathrm{sign}(\nu_t'') = \mathrm{sign}(w_t)$ and $|\nu_t''| < |w_t|$ because the logarithm function $y = ln(x)$ always increases slower than the function $y = x$ when x increases in the domain $x \in (1, \infty)$. In addition, from (28) we also know that

$$\frac{d|\nu_t''|}{dy_t} = -\frac{|w_t|}{y_t(y_t - w_t)} < 0. \tag{29}$$

Therefore, the noise ν_t'' in (26) decreases its magnitude when y_t increases as time goes on.

When we use the transformed linear model (7) for early detection, the system state vector for the Kalman filter is $X_t = \begin{bmatrix} \Delta\alpha \\ K \end{bmatrix}$. Now $H_t = [t - t_0\ 1]$ and the system is described by

$$\begin{cases} X_t = X_{t-1} \\ \ln(y_t) = H_t X_t + \nu_t''. \end{cases} \tag{30}$$

The Kalman filter in estimating X_t is

$$\begin{cases} H_t = [t - t_0\ 1] \\ K_t = P_{t-1}H_t^\tau/(H_t P_{t-1} H_t^\tau + 1/\tau_t) \\ P_t = (I - K_t H_t)P_{t-1} \\ \hat{X}_t = \hat{X}_{t-1} + K_t[\ln(y_t) - H_t\hat{X}_{t-1}]. \end{cases} \tag{31}$$

B. Estimation of Vulnerable Population Size

For a uniform-scan worm, we present below an effective way to predict the population size N based on the observation data η and the estimate α from Kalman filters above. In this way, we can know how many computers are vulnerable in the Internet when a worm is still in its slow start phase. A uniform-scan worm sends out on average η scans per unit time; each scan has the probability $N/2^{32}$ to hit a host in the population under consideration. Hence, at the beginning when most hosts in the vulnerable population N are still vulnerable, a worm can infect on average $\eta N/2^{32}$ hosts per unit time. (The probability of two scans sent out by a single infected host hitting the same target is negligible). From the definition of infection rate α, we have $\alpha = \eta N/2^{32}$. Therefore, the population N is

$$N = \frac{2^{32}\alpha}{\eta} \tag{32}$$

where the average worm scan rate η is directly estimated from monitored data generated by egress scan monitors. When we use one of the Kalman filters above to estimate α, we can use (32) to estimate N along with the Kalman filter estimation. In this way, the estimation of N has similar convergence properties to the estimation of α from the Kalman filter.

C. Overview of the Steps to Detect a Worm

MWC collects and aggregates reports of worm scans from all distributed monitors once in every monitoring interval in real-time. For each TCP or UDP port, MWC has an alarm threshold for monitored illegitimate scan traffic Z_t. The observed number of scans Z_t, which contains nonworm noise, is below this threshold when there is no global spreading worm. This threshold can be chosen based on observations on normal days when no wide-spreading worm exists in the Internet. If the monitored scan traffic is over the alarm threshold for several consecutive monitoring intervals, e.g., Z_t is over the threshold for three consecutive times, the Kalman filter will be activated. Then MWC begins to record C_t and calculates the average worm scan rate η from the reports of egress scan monitors. Because C_t is a cumulative observation data that could cumulate all nonworm noise, MWC begins to record data C_t only after the Kalman filter is activated. The Kalman filter can either use C_t or Z_t to estimate all the parameters of a worm at discrete time t $(t = 1, 2, 3, \ldots)$.

The recursive estimation will continue until the estimated value of α shows a trend: if the estimate $\hat{\alpha}$ stabilizes and oscillates slightly around a *positive constant* value, we have detected the presence of a worm; if the estimate $\hat{\alpha}$ converges to or oscillates around zero, we believe the surge of illegitimate monitored traffic is caused by nonworm noise.

VII. Simulation Experiments

In this section, we describe the extensive simulations we used to study: 1) how a random-scan worm and a sequential-scan worm propagate, and 2) the performance of our Kalman filter-based early detection system. In addition, we show that the address space covered by the worm monitoring system should be as distributed as possible in order to better monitor nonuniform scan worms, especially a sequential-scan worm such as Blaster.

In our simulation experiments, we do not simply use the epidemic model (1) to numerically generate a worm's propagation curve. Instead, we have programmed discrete-time worm propagation simulators, which can be downloaded from [44], to simulate the detailed scanning behaviors of scans sent out by each infected host during each discrete time interval. In this way, we can accurately simulate the detailed propagation of a worm that uses any kind of scanning strategy.

A. Simulation Settings

We have simulated Code Red [2], SQL Slammer [26], and a sequential-scan worm similar to Blaster [3]. First, we explain how we choose the simulation parameters. In the case of Code Red, more than 359 000 Code Red infected hosts were observed on July 19, 2001 by CAIDA [27]. Thus, in our simulation we set the Code Red vulnerable population $N = 360\,000$. Staniford *et al.* [37] used a different format but the same epidemic model as (1) to model Code Red, where their model's parameter K is actually $K = \beta N = \alpha$ [46]. They determined that $K = 1.8$ for the time scale of one hour. Therefore, for the discrete time unit of one minute in our simulation, $\alpha = 1.8/60 = 0.03$. From (32) we can reversely derive $\eta = 2^{32}\alpha/N = 358$ per minute, i.e., Code Red sends out on average about 358 scans per minute per infected host.

Because different infected hosts have different scan rates, we assume that each infected host has a constant scan rate x, a rate that is independently predetermined by a normal distribution $N(\eta, \sigma^2)$, where $\sigma = 100$ (the scan rate x is bounded by $x \geq 1$). In our simulation, ingress scan monitors cover 2^{20} IP space. We also assume $I_0 = 10$ at the beginning.

Because of the sequential scan used by Blaster, people do not have a good estimation of how many computers were really infected by Blaster within the days following the worm's outbreak. We will explain the reason for this later in our experiments (shown in Fig. 10). In addition, there is no authoritative study of this worm's scan rate η. Therefore, in this paper we simulate and study a sequential-scan worm that has the same "local preference" as Blaster [3], which is called a "Blaster-like" worm in this paper. Since we want to understand how the sequential scan affects a worm's propagation and our early detection system, we give this Blaster-like worm the same parameters as Code Red in order to compare it with Code Red, i.e., we set the Blaster-like worm to have $N = 360\,000$, $\eta = 358$ per minute. Each worm's scan rate x follows normal distribution $N(358\,100^2)$ with the bound $x \geq 1$, and $I_0 = 10$ at the beginning.

For a uniform-scan worm, such as Code Red, the distribution of vulnerable hosts in the Internet will not affect the worm's propagation. However, this distribution may affect the propagation of Blaster [48] because of its sequential scan. Since we do not know the true distribution of vulnerable hosts in the Internet, in our simulations of the Blaster-like worm, we assume vulnerable hosts are uniformly distributed in the IP space defined by BGP routing prefixes, which is less than 30% of the entire IPv4 space [50].

We should choose an appropriate monitoring interval Δ in the Kalman filter estimation. Δ should not be too big in order to obtain enough sampling points in a worm's slow start phase for the Kalman filter. On the other hand, a too small monitoring interval puts more pressure on the MWC data collection, and introduces more monitoring statistical error w_t in (17) (because we can observe fewer worm scans in a smaller monitoring interval). In the discrete-time simulations in this paper, the monitoring interval Δ is set to be one minute for Code Red and the Blaster-like worm. SQL Slammer propagates much faster and can finish infection in about 10 minutes [26]. Hence its monitoring interval should be much shorter in order to catch the dynamics of this worm. For this reason, the monitoring interval for Slammer is set to be one or several seconds (we will further discuss Δ selection in Section VIII).

B. Background Noise Consideration

We need to consider background nonworm noise in our simulations. Fortunately, Goldsmith [14] provided simple data of the background noise for Code Red activities monitored on a "/16" network (covers 2^{16} IP addresses). He recorded TCP port 80 SYN requests from Internet hosts to any unused IP addresses inside his local network. Such data are exactly the monitored data collected by ingress scan monitors in our proposed monitoring system. His monitored data showed that the background noise was small compared to Code Red traffic and the noise did not vary much. If we use normal distribution to model the background noise, then for each hour the number of noise scans follows $N(110.5\ 30^2)$ and the number of source hosts that send noise follows $N(17.4, 3.3^2)$.

We try to hold the statistics of the observed background noise in our experiments: we monitor 2^{20} IP space, which is 16 times larger than what Goldsmith monitored, so the number of noise scans or noise sources should be enlarged by 16 times. We use one minute instead of one hour as the monitoring interval, thus we should decrease the number of noise scans or noise sources by 60. In this way, in our simulations of Code Red and the Blaster-like worm, the noise added into the observation data at each monitoring interval follows $N(29.5, 8^2)$ for Z_t and $N(4.63, 0.893^2)$ for C_t. Of course, this kind of extension of noise is very rough, but it is the best we can do based on the data available. Currently, we are trying to obtain detailed log data on previous worms from other researchers in order to have more realistic experiments.

In the simulation experiments, the alarm threshold for Z_t is set to be two times as large as the mean value of the background noise, i.e., the alarm threshold is $29.5 \times 2 = 59$. The Kalman filter we use in early detection will be activated when the monitored scan traffic Z_t is over the alarm threshold for three consecutive monitoring intervals. In this way, the Kalman filter will not be frequently activated by the surge of background noise traffic in the normal days.

C. Code Red Simulation and Early Detection

We simulate Code Red propagation for 100 simulation runs with the same input parameters but different seeds for random number generator. Fig. 5 shows the number of infected hosts as a function of time for three cases: the average value, the 95th percentile, and the 5th percentile. The curve of 95th percentile means that in 95 out of our 100 simulation runs, Code Red propagates no faster than this curve represents.

This figure shows that a worm propagates slightly differently in different sample runs. The propagation speed difference is mainly caused by a worm's spreading at the beginning, when only several infected hosts scan and attempt to infect others. In fact, we have chosen $I_0 = 1$ and run Code Red propagation for another 100 simulation runs. It shows that Code Red in the $I_0 = 1$ case propagates more variously than the one shown in Fig. 5, where $I_0 = 10$.

For one simulation run of Code Red propagation, Fig. 6 shows the estimation of the worm infection rate α as a function of time by using three Kalman filters based on three discrete-time models: epidemic model (4), AR exponential model (6), and transformed linear model (7), respectively. This figure shows the estimates by using the processed monitored data Z_t after subtracting the average value of background noise from it. We can obtain the average value of noise based on the observations before activating the Kalman filter. We can use either the monitored data Z_t or the data C_t after bias correction (12) to estimate α for Code Red. They provide the similar estimation results [45]. Later, when we study the early detection of the Blaster-like worm, because of its nonuniform scan, we cannot use the bias correction (12) for the monitored data C_t and have to rely on the monitored data Z_t in our early detection. Therefore, in this paper we will only discuss early detection by using the monitored data Z_t.

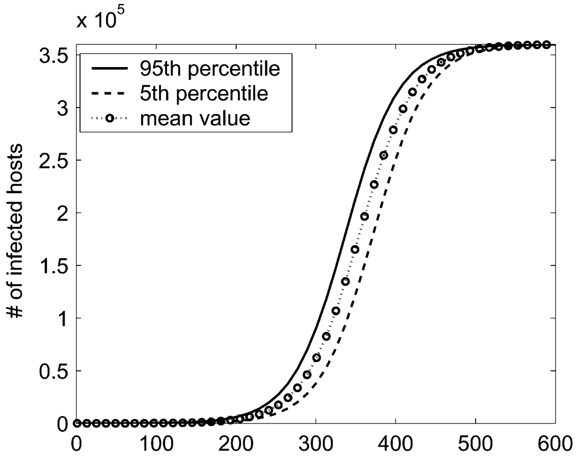


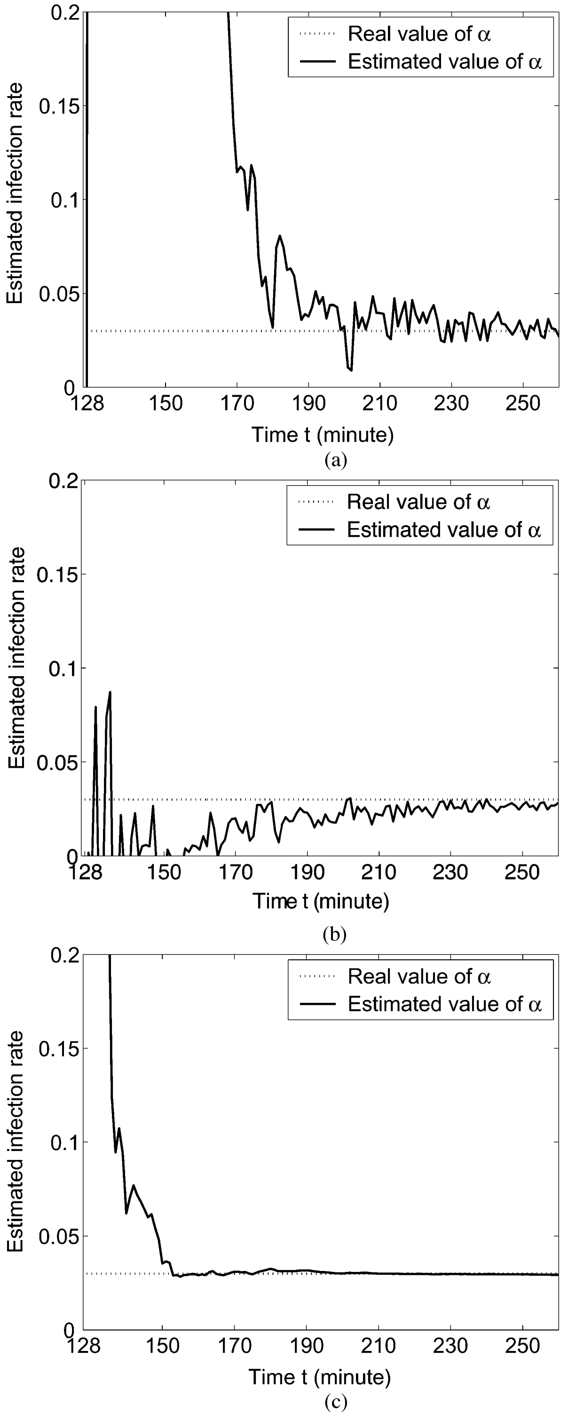
Fig. 5. Code Red propagation and its variability (100 simulation runs).

In this simulation run, Z_t at time 126, 127, and 128 minutes are over the alarm threshold 59, thus the Kalman filter is activated at time 128 minutes. Fig. 6 shows that the Kalman filter estimation based on the transformed linear model provides a much better estimation result than the other two because the noise ν_t'' introduced by the transformed linear model (7) is much smaller than the noise ν_t and ν_t' introduced by the other two models.

The noise ν_t, ν_t', and ν_t'' introduced by these three models are shown in (20), (24), and (28), respectively. We can see that the magnitude of the noise ν_t'' (28) introduced by the transformed linear model decreases as time goes on as shown in (29). On the other hand, the magnitude of the noise ν_t' (24) introduced by the AR exponential model does not change; the magnitude of the noise ν_t (20) introduced by the epidemic model could possibly increase as time goes on (because y_{t-1} in (20) increases as time goes on).

Because of the decreasing noise ν_t'' in the transformed linear model, we select $\tau_t = t$ in the Kalman filter (31) of the transformed linear model in order to put more weight on the newest less noisy observation data. On the other hand, we select $\tau_t \equiv 1$ for both the Kalman filters of the epidemic model and the AR exponential model since their noises, $\nu_t,\ \nu_t'$, do not decrease.

In the Code Red simulation run shown in Fig. 6, the worm infects 0.3% of vulnerable computers in the Internet at time 157 minutes. If we use the transformed linear model in our early detection, Fig. 6(c) shows that the estimate $\hat{\alpha}$ has already stabilized at a positive constant value by that time. Therefore, we can detect the presence of Code Red when it



(a)

(b)

(c)

Fig. 6. Kalman filter estimation of Code Red infection rate α (for one simulation run). (a) Based on epidemic model (4). (b) Based on AR exponential model (6). (c) Based on transformed linear model (7).

has only infected 0.3% of all vulnerable population in the Internet. For the remaining 99 Code Red simulation runs, we have done such early detection by using Kalman filters and have achieved the similar early detection performance. In our previous paper [45], we have shown that the early detection system can achieve a similar detection performance—detect a worm when it infects a similar fraction of the vulnerable population—no matter whether this worm propagates faster or slower in those 100 simulation runs.

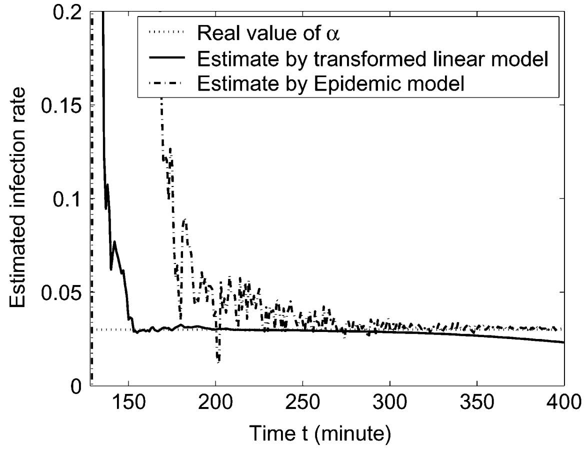


Fig. 7. Long-term Kalman filter estimation.

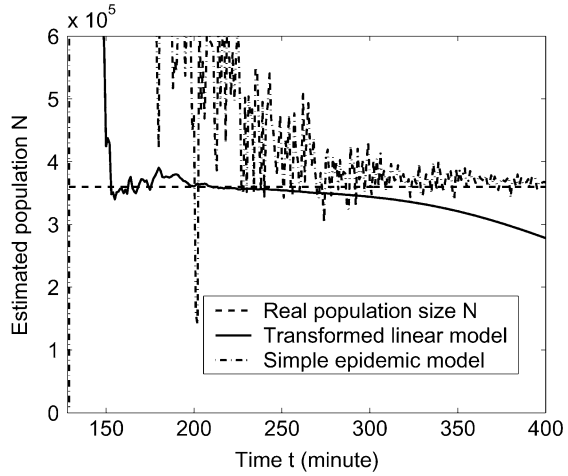


Fig. 8. Estimate of the vulnerable population size N of Code Red.

The Kalman filter (22) (based on epidemic model) is still useful since it is able to estimate worm infection rate α during the whole propagation period of a worm. On the other hand, because the transformed linear model is derived from the exponential-growth model (7), its Kalman filter will underestimate α when the worm enters its "fast spread phase" (as shown in Fig. 1). Fig. 7 shows the estimation results from these two Kalman filters before the worm infects 80% of vulnerable hosts at time 400 minutes. It shows that we should use these two Kalman filters together in the early detection of a worm.

We predict the vulnerable population size N from (32) at each discrete time when we update the estimate of α from Kalman filters. Fig. 8, shows the estimated value of N as a function of time based on the Kalman filters of transformed linear model (31) and epidemic model (32), respectively. Because the estimate $\hat{N}$ is proportional to the estimate $\hat{\alpha}$, this figure has the same pattern as Fig. 7. In a real implementation, we should combine both estimation curves shown in Fig. 8 to predict the vulnerable population size N.

Because Slammer propagates in the same way as Code Red—by uniformly scanning the Internet—its propagation and its early detection are very similar to the methods used for detecting Code Red [45]. (We choose $\eta = 4000$/sec as

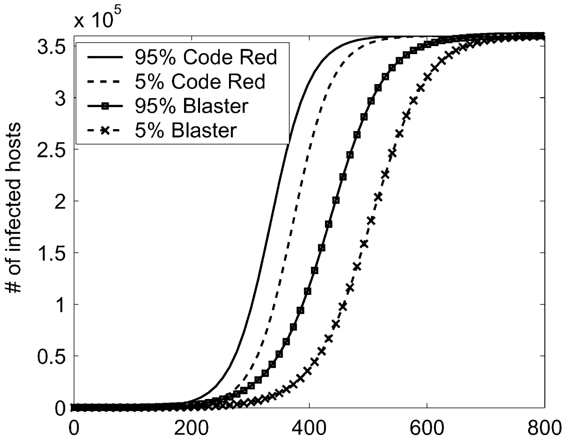


Fig. 9. Worm propagation comparison between Code Red and Blaster-like worm (100 simulation runs for each worm).

explained in [26] and $\delta = 1$ second). Therefore, we do not repeatedly show the early detection of Slammer in this paper.

D. Blaster-Like Worm Simulation and Early Detection

Each Blaster infected host scans the entire IP space sequentially from a selected starting point. To select this starting IP address, each worm copy has a 40% probability to choose the first address of its Class C-size subnet (x.x.x.0), and a 60% probability to choose a completely random IP address [3]. In our simulations, we let the Blaster-like worm to have the same local preference in selecting its starting point.

Since we select the same parameters for simulations of both Code Red and the Blaster-like worm, we can compare them to study how the sequential scan affects a worm's propagation. Again, we run the simulation of the Blaster-like worm for 100 simulation runs. Fig. 9 shows the 95th percentile and 5th percentile of the worm's propagation compared with the previous Code Red simulations shown in Fig. 5.

Even though the simple epidemic model (1) is derived based on uniform scanning, our simulation experiments show that the Blaster-like worm can still be accurately modeled by the simple epidemic model (1), and thus the worm can be modeled by the three discrete-time models presented in this paper. This is consistent with a conclusion in [48], which shows that a sequential-scan worm has the same propagation dynamics as a uniform-scan worm when the vulnerable hosts are uniformly distributed.

However, we should keep in mind that a worm's propagation is in fact a stochastic process; the epidemic model (1) is accurate only when both the number of vulnerable hosts and the number of infected hosts are relatively large. For example, no ordinary differential equation models are suitable to model the very end of a worm's propagation when the worm finishes infecting the last several vulnerable hosts, which can only be modeled accurately by a stochastic model. Since we study an Internet-scale worm's propagation that involves hundreds of thousands or even millions of computers, the epidemic model (1) is a good abstract model for modeling a worm's dynamics except the very beginning and the very end of the worm's propagation.

Fig. 10 shows that the Blaster-like worm propagates slower than Code Red. Zou *et al.* [48] pointed out that this is because the Blaster-like worm selects its starting scanning point with a local preference, not because of its sequential scan mechanism.

Because of its sequential scan, when monitoring the Blaster-like worm, we cannot let the monitoring system cover only one big block of IP address space—such a monitoring system can

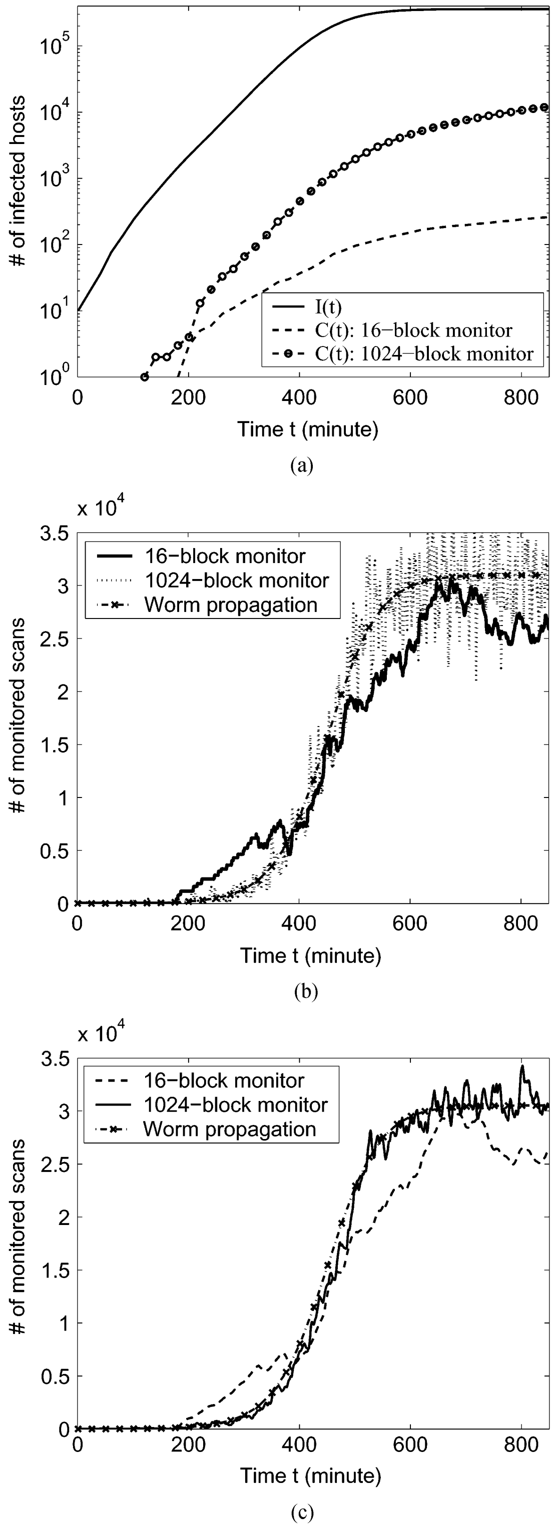


Fig. 10. Blaster-like worm propagation and its monitored data. (a) Worm propagation and observed infected hosts (Y-axis is in logarithm). (b) Monitored data Z_t. (c) Monitored data Z_t after using a low-pass filter.

only observe a very small fraction of infected hosts in the Internet. For example, if a sequential scan worm has the same fast scan rate $\eta = 4000$ per second as Slammer [26], each infected host will take $2^{32}/\eta = 12.4$ *days* to finish scanning the entire IPv4 space. Therefore, most hosts infected by the Blaster-like worm will take days before their scans hit the big block IP address space monitored in such a monitoring system.

For this reason, a good worm monitoring system should cover as distributed as possible an IP address space in the Internet. In this paper, we simulate two monitoring systems. Both monitoring systems cover the same 2^{20} IP addresses (the same as the monitoring system in previous Code Red study), but they consist of a different number of monitored IP blocks: one monitors 16 “/16” networks; the other monitors 1024 “/22” networks. All monitored address blocks in a monitoring system are evenly distributed in the entire IPv4 space. We call these two monitoring systems as “the 16-block monitoring system” and “the 1024-block monitoring system”, respectively.

Fig. 10 shows one simulation run of the Blaster-like worm. Fig. 10(a) shows the number of infected hosts $I(t)$ in the entire Internet as a function of time t. It also shows the cumulative number of observed infected hosts, $C(t)$, from both monitoring systems. Because observed $C(t)$ is very small compared with $I(t)$, we plot this figure by taking logarithmically on the Y-axis.

Fig. 10(a) shows that, during the worm’s propagation period, we can observe less than 0.1% of infected hosts in the Internet from the 16-block monitoring system. Even if we use the 1024-block monitoring system, we can only observe less than 4% of infected hosts in the Internet during the worm’s propagation period. This is the reason why researchers have not derived an accurate estimate of how many computers were really infected by the Blaster-like worm.

Fig. 10(b) shows the monitored data $Z(t)$, the number of worm scans observed within each minute. Compared to the 16-block monitoring system, The 1024-block monitoring system gives noisier observation $Z(t)$. This is because as time goes on, an infected host will enter or leave one of the monitored IP blocks. It happens more frequently in the 1024-block monitoring system than in the 16-block monitoring system.

Although noisier than the data from the 16-block monitoring system, the monitored data from the 1024-block monitoring system represents more accurately the propagation of a sequential-scan worm. From the monitored data sets, we want to know the worm’s propagation pattern in the global Internet, i.e., the curve of I_t shown in Fig. 10(b). Such a growth pattern of I_t is a low frequency signal compared with the high frequency noise presented in the observed data Z_t. Therefore, we can use a low-pass filter to filter out high frequency noise from Z_t without changing the worm’s propagation pattern. Fig. 10(c) shows the observation data Z_t after being filtered by a first-order low-pass filter.[1] This figure clearly shows that the monitored data from the 1024-block monitoring system can better represent the worm’s propagation pattern in the entire Internet.

Based on the filtered monitored data Z_t from the 1024-block monitoring system as shown in Fig. 10(c), we run the Kalman filter estimation based on the transformed linear model. The estimated $\hat{\alpha}$ is shown in Fig. 11 as a function of time. In this simulation run, the Blaster-like worm infects 1.3% of vulnerable population at time 240 minutes, by which time the estimate $\hat{\alpha}$ has already stabilized and oscillated slightly around a positive, constant value. Hence our early detection system can detect the Blaster-like worm before it infects 1.3% of vulnerable population in the Internet.

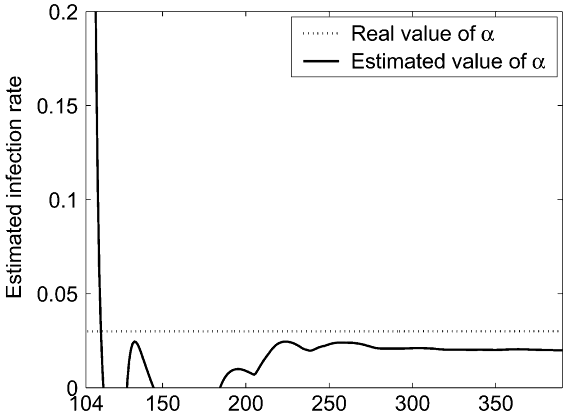


Fig. 11. Kalman filter estimation of worm infection rate α for the Blaster-like worm (based on the transformed linear model and filtered data Z_t from a 1024-block monitoring system).

Worm propagation in other simulation runs of the Blaster-like worm gives results similar to those shown in Figs. 10 and 11, On occasion the 16-block monitoring system provides as good observation as the 1024-block monitoring system. However, the 1024-block monitoring system always provides stable and good observations, while the 16-block monitoring system provides poor observations in many instances.

VIII. Discussion and Future Work

We have used the simple epidemic model (1) and the exponential model (2) for the estimation and prediction. While these models give good results so far, we need to develop more detailed models to reflect a future worm’s dynamics. For example, if a worm spreads through a topology, or spreads by exploiting multiple vulnerabilities, or is a meta-server worm, then its propagation may not follow the models used in this paper.

The monitoring interval Δ is an important parameter in the system design. For a slow-spreading worm, it could be set to be long, but for a fast-spreading worm such as Slammer, the time interval should be in the order of seconds to catch up with the worm’s dynamics. How can we select the appropriated Δ before we know a worm’s presence and its speed? We need to do further research on designing a recursive estimation algorithm that uses adaptive sampling rate. Currently, one way we contemplate is to tag the time stamp with each observed scan. Then at MWC, several estimators run in parallel with different monitoring intervals. From the tagged time stamp the correct C_t or Z_t for every estimator can easily be restored.

It could be useful to develop distributed estimation algorithms so as to reduce the latency and traffic for the report to a central server. Distributed estimators may also reduce the impact of noise when a few monitors experience larger than

[1] Denote by $\hat{Z}_t$ as the Z_t after filtering. The low-pass filter is $\hat{Z}_t = aZ_t + (1-a)\hat{Z}_{t-1}$. We use $a = 0.1$ in Fig. 10(c).

normal noise-to-signal ratios. In addition, we want to use a continuous version of the Kalman filter. This approach would reduce the significance of the monitoring interval selection, and would work nicely with the distributed estimation setting.

The worm detection method presented here assumes that only worm scans can cause exponentially increased traffic to monitors, while other background scan noise cannot. We believe this is a reasonable assumption. If we want to further improve the detection accuracy, however, we can add some other rule sets in the detection system. For example, in order to distinguish a worm attack from a DDoS attack, we can exploit the differences between them: a DDoS attack has one or several targets while a worm's propagation has no specific target.

As mentioned in Section V, the derivation of the bias-correction (12) is based on uniform-scan worms. We need to further study how to accurately estimate the infected population for nonuniform scan worms, especially for a sequential-scan worm like Blaster. In addition, the bias-correction (12) and the estimation formula of vulnerable population size (32) rely on the observation accuracy of η, a worm's average scan rate. As explained at the end of Section IV, it is hard to obtain a good estimate of η for a bandwidth-limited worm. Therefore, we should be cautious when using the above two procedures on a bandwidth-limited worm.

In this paper, we have presented several major issues in designing an Internet Malware Warning Center. However, there are still many challenges in building such a system, such as cooperation mechanism among a large number of communities; the privacy concern in monitored data sharing, and the robustness of the monitoring system itself toward attacks by worms or hackers. How to deal with these issues is out of the scope of this paper.

IX. Conclusion

We have proposed a monitoring and early detection system for Internet worms to provide an accurate triggering signal for mitigation mechanisms in the early stage of a future worm. Such a system is needed in view of the propagation scale and the speed of the past worms. We have been lucky that the previous worms have not been very malicious; the same cannot be said for future worms. Based on the idea of "detecting the *trend*, not the *burst*" of monitored illegitimate scan traffic, we present a "trend detection" methodology to detect the presence of a worm in its early propagation stage by using the Kalman filter and worm propagation models. Our analysis and simulation studies indicate that such a system is feasible, and the trend detection methodology poses many interesting research issues. We hope this paper will generate interest and participation in this topic, and eventually lead to an effective Internet worm monitoring and early detection system.

References

[1] Symantec Corp.: Symantec Early Warning Solutions [Online]. Available: http://enterprisesecurity.symantec.com/SecurityServices/

[2] eEye Digital Security: .ida "Code Red" Worm (2001). [Online]. Available: http://www.eeye.com/html/Research/Advisories/AL20010717.html

[3] eEye Digital Security: Blaster Worm Analysis (2003). [Online]. Available: http://www.eeye.com/html/Research/Advisories/AL20030811.html

[4] B. D. O. Anderson and J. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice Hall, 1979.

[5] D. Anderson, T. Frivold, and A. Valdes, "Next-Generation Intrusion Detection Expert System (Nides): A Summary," SRI International, Tech. Rep. SRI-CSL-95-07, May 1995.

[6] V. H. Berk, R. S. Gray, and G. Bakos, "Using sensor networks and data fusion for early detection of active worms," presented at the SPIE AeroSense Symp., Orlando, FL, 2003.

[7] Cooperative Association for Internet Data Analysis (CAIDA). [Online]. Available: http://www.caida.org

[8] CERT Coordination Center. [Online]. Available: http://www.cert.org

[9] CERT/CC Advisories. [Online]. Available: http://www.cert.org/advisories/

[10] Z. Chen, L. Gao, and K. Kwiat, "Modeling the spread of active worms," in *Proc. IEEE INFOCOM*, Mar. 2003, pp. 1890–1900.

[11] D. Dagon, X. Qin, G. Gu, W. Lee, J. Grizzard, J. Levin, and H. Owen, "Honeystat: Local worm detection using honeypots," in *Proc. 7th Int. Symp. Recent Advances in Intrusion Detection (RAID)*, Sep. 2004, pp. 39–58.

[12] D. J. Daley and J. Gani, *Epidemic Modeling: An Introduction*. Cambridge, U.K.: Cambridge Univ. Press, 1999.

[13] D. Denning, "An intrusion detection model," *IEEE Trans. Software Eng.*, vol. SE-13, no. 2, pp. 222–232, Feb. 1987.

[14] D. Goldsmith. Incidents Maillist: Possible Codered Connection Attempts. [Online]. Available: http://lists.jammed.com/incidents/2001/07/0149.html

[15] S. A. Hofmeyr, S. Forrest, and A. Somayaji, "Intrusion detection using sequences of system calls," *J. Comput. Security*, vol. 6, no. 3, pp. 151–180, 1998.

[16] Honeynet Project. Know Your Enemy: Honeynets. [Online]. Available: http://www.honeynet.org/papers/gen2/index.html

[17] Internet Storm Center. [Online]. Available: http://isc.sans.org/

[18] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan, "Fast portscan detection using sequential hypothesis testing," in *Proc. IEEE Symp. Security and Privacy*, May 2004, pp. 211–225.

[19] J. Jung, S. E. Schechter, and A. W. Berger, "Fast detection of scanning worm infections," in *Proc. 7th Int. Symp. Recent Advances in Intrusion Detection (RAID)*, Sep. 2004, pp. 59–81.

[20] J. O. Kephart, S. R. White, and D. M. Chess, "Computers and epidemiology," *IEEE Spectrum*, vol. 30, no. 5, pp. 20–26, May 1993.

[21] J. O. Kephart and S. R. White, "Directed-graph epidemiological models of computer viruses," in *Proc. IEEE Symp. Security and Privacy*, 1991, pp. 343–359.

[22] J. O. Kephart and S. R. White, "Measuring and modeling computer virus prevalence," in *Proc. IEEE Symp. Security and Privacy*, 1993, pp. 2–15.

[23] H. Kim and B. Karp, "Autograph: Toward automated, distributed worm signature detection," in *Proc. 13th USENIX Security Symp.*, San Diego, CA, Aug. 2004.

[24] W. Lee and S. Stolfo, "A framework for constructing features and models for intrusion detection systems," *ACM Trans. Inf. Syst. Security*, vol. 3, no. 4, pp. 227–261, Nov. 2000.

[25] L. Ljung and T. Soderstrom, *Theory and Practice of Recursive Identification*. Cambridge, MA: MIT Press, 1983.

[26] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the Slammer worm," *IEEE Security and Privacy Mag.*, vol. 1, no. 4, pp. 33–39, Jul. 2003.

[27] D. Moore, C. Shannon, and J. Brown, "Code-Red: A case study on the spread and victims of an Internet worm," in *Proc. 2nd ACM SIGCOMM Workshop on Internet Measurement*, Nov. 2002, pp. 273–284.

[28] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, "Internet quarantine: Requirements for containing self-propagating code," in *Proc. IEEE INFOCOM*, Mar. 2003, pp. 1901–1910.

[29] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, "Network Telescopes," CAIDA, Tech. Rep. TR-2004-04, 2004.

[30] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of Internet background radiation," in *Proc. Internet Measurement Conf. (IMC)*, Oct. 2004, pp. 27–40.

[31] N. Provos, "A virtual honeypot framework," in *Proc. 13th USENIX Security Symp.*, Aug. 2004, pp. 1–14.

[32] SANS Inst. [Online]. Available: http://www.sans.org

[33] D. Seeley, "A tour of the worm," in *Proc. Winter USENIX Conf.*, Jan. 1989, pp. 287–304.

[34] C. Shannon and D. Moore. (2004, Mar.) The Spread of the Witty Worm. [Online]. Available: http://www.caida.org/analysis/security/witty/

[35] S. Singh, C. Estan, G. Varghese, and S. Savage, "Automated worm fingerprinting," in *Proc. 6th ACM/USENIX Symp. Operating System Design and Implementation (OSDI)*, Dec. 2004, pp. 45–60.

[36] S. Staniford, "Containment of scanning worms in enterprise networks," *J. Comput. Security*, to be published.

[37] S. Staniford, V. Paxson, and N. Weaver, "How to own the Internet in your spare time," in *Proc. USENIX Security Symp.*, Aug. 2002, pp. 149–167.

[38] S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, and D. Zerkle, "GrIDS—A graph-based intrusion detection system for large networks," in *Proc. 19th Nat. Information Systems Security Conf.*, Oct. 1996, pp. 361–370.

[39] A. Valdes and K. Skinner, "Adaptive, model-based monitoring for cyber attack detection," in *Proc. 3th Int. Symp. Recent Advances in Intrusion Detection (RAID)*, Oct. 2000, pp. 80–92.

[40] D. Verton. (2003, Oct.) DHS Launches Cybersecurity Monitoring Project. [Online]. Available: http://www.pcworld.com/news/article/0,aid,112764,00.asp

[41] N. Weaver, S. Staniford, and V. Paxson, "Very fast containment of scanning worms," in *Proc. 13th USENIX Security Symp.*, Aug. 2004, pp. 29–44.

[42] M. M. Williamson, "Throttling viruses: Restricting propagation to defeat mobile malicious code," presented at the 18th Annu. Computer Security Applications Conf., Las Vegas, NV, Dec. 2002.

[43] J. Wu, S. Vangala, L. Gao, and K. Kwiat, "An efficient architecture and algorithm for detecting worms with various scan techniques," presented at the 11th Annu. Network and Distributed System Security Symp. (NDSS'04), San Diego, CA, Feb. 2004.

[44] C. C. Zou. (2004, Feb.) Internet Worm Propagation Simulator. [Online]. Available: http://www.cs.ucf.edu/~czou/research/wormSimulation.html

[45] C. C. Zou, L. Gao, W. Gong, and D. Towsley, "Monitoring and early warning for Internet worms," in *Proc. 10th ACM Conf. Computer and Communications Security (CCS'03)*, Washington, DC, Oct. 2003, pp. 190–199.

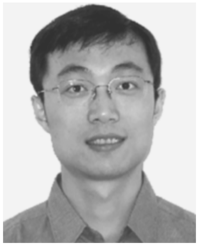
[46] C. C. Zou, W. Gong, and D. Towsley, "Code red worm propagation modeling and analysis," in *Proc. 9th ACM Conf. Computer and Communications Security (CCS'02)*, 2002, pp. 138–147.

[47] C. C. Zou, W. Gong, and D. Towsley, "Worm propagation modeling and analysis under dynamic quarantine defense," in *Proc. ACM CCS Workshop on Rapid Malcode (WORM'03)*, Oct. 2003, pp. 51–60.

[48] C. C. Zou, D. Towsley, and W. Gong, "On the performance of Internet worm scanning strategies," *J. Performance Evaluation*, to be published.

[49] C. C. Zou, D. Towsley, and W. Gong, "Email worm modeling and defense," in *Proc. 13th Int. Conf. Computer Communications and Networks (ICCCN'04)*, Oct. 2004, pp. 409–414.

[50] C. C. Zou, D. Towsley, W. Gong, and S. Cai, "Routing worm: A fast, selective attack worm based on IP address information," in *Proc. 19th Workshop on Principles of Advanced and Distributed Simulation (PADS'05)*, Jun. 2005, pp. 199–206.



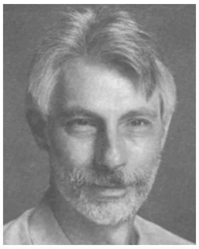
Cliff C. Zou (M'05) received the B.Sc. and M.Sc. degrees in electrical engineering from the University of Science and Technology of China, Hefei, China, in 1996 and 1999, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 2005.

Currently, he is an Assistant Professor in the School of Computer Science, University of Central Florida, Orlando. His research interests include computer and network security, network modeling, and performance evaluation.

Weibo Gong (S'87–M'87–SM'97–F'99) received the Ph.D. degree from Harvard University in 1987.

He has been with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst, since 1987. He is also an Adjunct Professor in the Department of Computer Science at the same university. His major research interests include control and systems methods in communication networks, network security, and network modeling and analysis.

Dr. Gong is a recipient of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL's George Axelby Outstanding paper Award, and the Program Committee Chair for the 43rd IEEE Conference on Decision and Control.



Don Towsley (M'78–SM'93–F'95) received the B.A. degree in physics and the Ph.D. degree in computer science from the University of Texas, Austin, in 1971 and 1975, respectively.

From 1976 to 1985, he was a member of the faculty of the Department of Electrical and Computer Engineering at the University of Massachusetts, Amherst. He is currently a Distinguished Professor at the University of Massachusetts in the Department of Computer Science. He has held visiting positions at IBM T. J. Watson Research Center, Yorktown Heights, NY; Laboratoire MASI, Paris, France; INRIA, Sophia-Antipolis, France; AT&T Labs—Research, Florham Park, NJ; and Microsoft Research Lab, Cambridge, UK. His research interests include networks and performance evaluation.

Dr. Towsley currently serves on the Editorial board of *Journal of the ACM* and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and has previously served on several editorial boards including those of the IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE/ACM TRANSACTIONS ON NETWORKING. He was a Program Co-chair of the joint ACM SIGMETRICS and PERFORMANCE'92 conference and the Performance 2002 conference. He is a member of ACM and ORSA, and Chair of IFIP Working Group 7.3. He has received the 1998 IEEE Communications Society William Bennett Best Paper Award and numerous best conference/workshop paper awards. He has been elected Fellow of both the ACM and IEEE.

Lixin Gao (M'96) received the Ph.D. degree in computer science from the University of Massachusetts, Amherst, in 1996.

She is an Associate Professor of Electrical and Computer Engineering at the University of Massachusetts, Amherst. Her research interests include multimedia networking and Internet routing and security. Between May 1999 and January 2000, she was a visiting researcher at AT&T Research Labs and DIMACS.

Dr. Gao is an Alfred P. Sloan Fellow and received an NSF CAREER Award in 1999. She has served on a number of technical program committees including SIGCOMM2004, SIGMETRICS2003, and INFOCOM2004, and is on the Editorial Board of IEEE/ACM TRANSACTIONS ON NETWORKING.