Available online at www.sciencedirect.com
SCIENCE DIRECT®
Performance Evaluation 63 (2006) 700–723
PERFORMANCE EVALUATION
An International Journal
www.elsevier.com/locate/peva

On the performance of Internet worm scanning strategies

Cliff C. Zou[a,*], Don Towsley[b], Weibo Gong[c]

[a] *School of Computer Science, University of Central Florida, FL, United States*
[b] *Deptartment of Computer Science, University of Massachusetts, Amherst, MA, United States*
[c] *Department of Electrical & Computer Engineering, University of Massachusetts, Amherst, MA, United States*

Received 6 September 2004; received in revised form 25 July 2005
Available online 8 September 2005

Abstract

In recent years, fast spreading worms, such as Code Red, Slammer, Blaster and Sasser, have become one of the major threats to the security of the Internet. In order to defend against future worms, it is important to first understand how worms propagate and how different scanning strategies affect worm propagation dynamics. In this paper, we systematically model and analyze worm propagation under various scanning strategies, such as uniform scan, routing scan, hit-list scan, cooperative scan, local preference scan, sequential scan, divide-and-conquer scan, target scan, etc. We also provide an analytical model to accurately model Witty worm's destructive behavior. By using the same modeling framework, we reveal the underlying similarity and relationship between different worm scanning strategies. In addition, based on our simulation and analysis of Blaster worm propagation and monitoring, we provide a guideline for building a better worm monitoring infrastructure.

Keywords: Worm modeling; Worm scanning strategy; Network security; Network monitoring

1. Introduction

Computer "worms" are programs that self-propagate across a network exploiting security or policy flaws in widely used services [30]. In recent years, two major classes of worms, "*scan-based worms*" and "*email worms*", have attacked us frequently. "*Email worms*" propagate through emails and compromise computers when email users execute worm email attachments or simply view worm emails, they require

* Corresponding author.
E-mail address: czou@cs.ucf.edu (C.C. Zou).

0166-5316/$ – see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.peva.2005.07.032

Nomenclature

c_1, c_2	$c_1 = \Omega_e/\Omega$, $c_2 = N_e/N$
$C(t)$	cumulative number of infected hosts observed by a monitoring system at time t
$d(t)$	density of vulnerable hosts in the unscanned IP space for a cooperative scan worm
$D(t)$	number of infected hosts that are destroyed by Witty worm by time t
$I(t)$	number of infected hosts at time t
$I_e(t)$, $I_o(t)$	number of infected hosts in the target (other) domain(s) at time t, $I(t) = I_o(t) + I_e(t)$
$I_k(t)$	number of infectious hosts in the k-th "$/n$" prefix network at time t, $k = 1, 2, \ldots, K$
K	number of "$/n$" prefix networks in the worm scanning space Ω, $\Omega = K2^{(32-n)}$
m	number of "$/n$" prefix networks that contain vulnerable hosts ($m \leq K$)
N	total number of vulnerable hosts in the Internet before worm infection
N_e, N_o	number of initially vulnerable hosts in the target (other) domain(s), $N = N_e + N_o$
N_k	number of initially vulnerable hosts in the k-th "$/n$" prefix network, $k = 1, 2, \ldots, K$
p	probability of a local preference scan worm to scan locally
q	probability of a worm scanning a specific address in a time interval δ, $q = \eta\delta/\Omega$
$Z(t)$	number of worm scans observed by a monitoring system in a unit time at time t

Greek letters

β	pairwise rate of infection in worm propagation model, $\beta = \eta/\Omega$
β', β''	pairwise rate of infection in local(remote) scan for a local preference scan worm
δ	the small time interval used in infinitesimal analysis
ϵ	time delay in worm propagation (considered in idealized worms)
η	a worm's average scan rate
Ω	number of IP addresses contained in a worm's scanning space
Ω_e, Ω_o	size of worm scanning space in the target (other) domain(s) for a selective attack worm, $\Omega = \Omega_e + \Omega_o$
λ	average destruction rate of Witty worm

human interference to propagate and thus propagate relatively slowly [30]. On the other hand, "*scan-based worms*" propagate by generating IP addresses to scan and directly compromise any vulnerable target computer, they need no human activation and, thus could propagate much faster than email worms [26,30]. For example, Slammer in January 2003 infected more than 90% of vulnerable computers in the Internet within just 10 min [18]. Recent well-known worms, Code Red [19], Code Red II [19], Slammer [18], Blaster [8] and Sasser [10], are all scan-based worms. In this paper, we concentrate our study on scan-based worms.

Attackers have tried many scanning strategies in previous worms. Code Red and Slammer uniformly scanned the entire IPv4 space [7,18]. Code Red II used a local preference scan: it had a higher probability to scan an address within the same "/16" or "/8" network than a random address [19]. Blaster sequentially scanned the Internet and chose its sequential-scan starting point from a local address with probability 0.4 [8].

Until now, worm attacking techniques are more advanced than defense techniques. Many scanning strategies have been implemented by previous worms but we have not fully understood them yet. Like earthquake modeling or tornado modeling, a good Internet worm model gives us deep understanding of the dynamics of worms, helps us to generate effective early warning and defense mechanisms, and provides the simulation basis for accurately evaluating the performance of various defense systems. In this paper, we mathematically model and analyze various scanning strategies that attackers have already used or may use in their future worms.

From our analysis, we derive the following understandings of worm scanning strategies:

- Cooperation among infected hosts for scanning does not significantly increase a worm's spreading speed.
- A computer infected by Witty worm has a crash time that is exponentially distributed; the mean value of the crash time of an infected computer is proportional to the computer's hard disk volume and inversely proportional to its network bandwidth.
- A local preference scan increases a worm's propagation speed when vulnerable hosts are not uniformly distributed. The optimal local-scan probability increases when the size of the local-scan network increases.
- When vulnerable hosts are uniformly distributed, divide-and-conquer scan, sequential scan and uniform scan are equivalent in terms of a worm's propagation speed.
- A "flash worm" [26] using uniform scan is an optimal spreading worm converged both from a "hit-list worm" [26] and from a "routing worm" [34]. A flash worm that makes sure no IP address is scanned more than once is the fastest spreading worm in terms of worm scanning strategy.
- For a sequential scan worm, such as Blaster [8], using local preference in selecting its starting point slows down the worm's propagation speed.
- For a selective attack worm, such as a routing worm [34], when the density of vulnerable hosts in the target domain is higher than the density in other domains, the worm propagates faster in the target domain if it only scans the target domain instead of all domains (and vice versa).

We also provide two guidelines in defense against worm attacks:

- It is crucial to prevent attackers from easily identifying the IP addresses of a large number of potentially vulnerable hosts through a public service (i.e., do not advertise IP addresses if possible), or obtaining address allocation information to dramatically reduce a worm's scanning space.
- The address space covered by a monitoring system should be as distributed as possible to accurately monitor the propagation of a non-uniform scan worm, especially a sequential scan worm, such as Blaster.

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 introduces the propagation model for uniform scan worms, which is the modeling basis for this paper. Then Section 4 models and analyzes worm propagation under various scanning strategies. Section 5 provides the analytical model for modeling the unique destructive behavior of Witty worm. Through studying the monitoring of Blaster worm, Section 6 provides a guideline on how to set up a comprehensive Internet worm monitoring system. In the end, Section 7 concludes this paper.

2. Related work

The security implications of self-propagating codes were first studied by Cohen in 1987 [5]. The Morris worm in 1988 [24], the ancestor for contemporary Internet worms, was the first widely spreading self-propagating code. After the Morris worm, there were few studies of computer worms, since no severe worm incident happened for more than 10 years until the Code Red outbreak in 2001 [7].

From 1991 to 1993, Kephart, White and Chess of IBM performed a series of studies on viral infection based on epidemiology models [13–15]. Then the Code Red incident in July 2001 [7] stimulated a number of models and analyses of Internet worm propagation. Staniford et al. [26] used the "simple epidemic model" [6] to model the spread of Code Red right after the Code Red incident and also presented several theoretical worms. Zou et al. [33] presented a "two-factor" worm model that considered both the effect of human counter measures and the effect of the congestion caused by worm scan traffic. Chen et al. [4] presented a discrete-time version worm model and also studied local preference scanning. Zou et al. [32] used Kalman filter to early detect an Internet worm based on the worm's exponential growth trend at its early stage. Nicol and co-workers [17,21] modeled how various "good" worms could help defend against worm attacks. Staniford [25] studied enterprise network worm quarantine based on epidemic modelling. Weaver et al. [30] presented a taxonomy of computer worms based on several factors: target discovery, carrier, activation, payloads and attackers. Weaver et al. studied how to model and simulate worm propagation accurately in a down-scaled networking environment.

The Slammer worm in January 2003 [18] clearly showed that we must consider the network congestion caused by the propagation of a worm in worm modeling. Wagner et al. [28] presented an Internet worm simulator considering network bandwidth and latency. Serazzi and Zanero [23] considered the link bandwidth of Autonomous Systems (ASes) in worm modeling based on simple epidemic model. Kesidis et al. [16] used coupled Kermack–McKendrick models to model bandwidth-limited worms.

As researchers understand more how a worm propagates, they identify various ways to make a worm propagate faster as well. Staniford et al. [26] presented a "hit-list worm" and a "flash worm", which build a list of IP addresses of vulnerable hosts into worm code, and thus shorten their propagation time. Wu et al. [31] presented a "routable scan" worm that increases its propagation speed by scanning BGP routing prefixes instead of the entire IPv4 space. Zou et al. [34] presented a "routing worm" that is similar to the above "routable scan" worm and also presented the "selective attack" idea and potential damage from a routing worm.

In recent years, researchers are paying more attention on monitoring the Internet for malicious activities. Moore et al. [20] presented the concept of "network telescope" to monitor Internet abnormal traffic to unused IP space. Zou et al. [32] used a similar monitoring system for early detection of a worm outbreak. Berk et al. [2] proposed to collect ICMP "Destination Unreachable" messages generated by routers for packets to unused IP addresses, which in fact monitors the same traffic as the above "network telescope". Through active traffic response, "Honeynet" [11] and "Honeyd" [22] can gather more detailed information of Internet malicious traffic.

A close work to ours is the simulation studies of various worm scanning strategies conducted by Vogt [27]. However, his work is entirely based on simulation experiments without mathematical analysis and modeling.

3. Modeling basis: uniform scan worm model

First, we briefly describe how a scan-based worm propagates: for a TCP vulnerability, a worm first sets up a TCP connection with a target host on the vulnerable TCP port, then sends exploiting code to compromise the target (such as Code Red and Blaster); for a UDP vulnerability, a worm directly sends exploiting code to a target on the vulnerable UDP port (such as Slammer). Computers behind a NAT device use one single public IP address for communication with the outside Internet, and hence we treat them as one single host in our Internet worm modeling.

3.1. Uniform scan worm model

The basis of our Internet worm modeling is the classical epidemic model [6]. The system is assumed to be a "*homogeneous system*", any infectious host has the equal probability to infect any susceptible host in the system. Once a host is infected by a disease, it is assumed to remain in the infectious state forever. Denote by $I(t)$ is the number of infectious hosts at time t and N is the total number of susceptible hosts in the system before an epidemic spreads out. Thus, $[N - I(t)]$ is the number of susceptible hosts at time t. The epidemic model for a homogeneous system is [6]:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \beta I(t)[N - I(t)] \tag{1}$$

where β is called the *pairwise rate of infection* in epidemiology studies, it represents "*infection intensity*" from infectious hosts $I(t)$ to susceptible hosts $[N - I(t)]$. At $t = 0$, $I(0)$ hosts are infectious and the other $[N - I(0)]$ hosts are all susceptible.

For readers' convenience, see nomenclature lists all notations used in this paper.

It should be noted that N is the total number of *vulnerable* hosts in the system before a disease or a worm spreads out ($I(0)$ of them are initially infected). For Internet worms, we do not consider invulnerable computers in the modeling. For example, we use $N = 360,000$ for Code Red, since it infected around 360,000 computers on July 19th 2001 [19], even though millions of Windows web servers were on-line on that day.

A "*uniform scan worm*" is a worm that uniformly picks IP addresses in its scanning space to scan and compromise, which can be modeled by (1). However, we need to find a model described by a worm's parameters that have concrete physical meanings, not just by an abstract value β. In the following, we derive the uniform scan worm model based on *"infinitesimal analysis"* and *"mean value analysis"*.

Suppose, a uniform scan worm has an average scan rate η, which is the average number of scans an infected host sends out per unit time. Denote by δ as the length of a small time interval. Thus, during δ time interval, an infected host sends out an average of $\eta\delta$ scans. Suppose, the worm uniformly scans the IP space that has Ω addresses; then every scan has a probability of $1/\Omega$ to hit any one IP address in this scanning space. Hence, on average an infected host has a probability

$$q = 1 - \left(\frac{1-1}{\Omega}\right)^{\eta\delta} \approx \frac{\eta\delta}{\Omega} \tag{2}$$

to hit a specific IP address in the scanning space Ω during a small time interval δ. The approximation in (2) is accurate, since $1/\Omega \ll 1$.

At time t, there are $[N - I(t)]$ vulnerable hosts in the system. From time t to $t + \delta$, the probability that two scans sent out by an infected host hit the same vulnerable host is negligible when δ is sufficiently small. Therefore, an infected host infects on average $[N - I(t)]q$ vulnerable hosts from time t to $t + \delta$. When δ is sufficiently small, the probability of two infected hosts infecting the same vulnerable host during the time interval δ is also negligible. Therefore, the number of newly infected hosts during the time interval δ equals $I(t)[N - I(t)]q$. Put (2) in and we derive the number of infected hosts at time $t + \delta$:

$$I(t + \delta) = I(t) + I(t)[N - I(t)]\frac{\eta\delta}{\Omega} \tag{3}$$

Note that we do not exclude the possibility that two instances of a worm scan the same target during a time interval δ. Such a probability is a second-order function of δ, and hence quickly goes to zero as the time interval δ goes to zero. Therefore, Eq. (3) is accurate when δ is very small (which is the essence of infinitesimal analysis).

Taking $\delta \to 0$, we derive the worm model for uniform scan worms:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\eta}{\Omega} I(t)[N - I(t)] \tag{4}$$

It is identical to the epidemic model (1) where β corresponds to

$$\beta = \frac{\eta}{\Omega} \tag{5}$$

Staniford et al. [26] presented a "random constant spread" (RCS) model to model the propagation of a uniform scan worm:

$$\frac{\mathrm{d}a(t)}{\mathrm{d}t} = Ka(t)[1 - a(t)] \tag{6}$$

where $a(t)$ is the proportion of vulnerable machines that have been infected by time t. Therefore, $a(t) = I(t)/N$. This RCS model is, in fact, equivalent to the epidemic model (1) where its parameter $K = \beta N$. Our contribution in this section is to derive a uniform scan worm model (4) that is described directly by a worm's propagation parameters η and Ω.

3.2. Modeling assumption and justification

It should be noted that the uniform scan worm model (4) models the propagation of a worm in an "ideal" network condition. It does not consider the two major factors affecting a worm's spreading as mentioned in [33]: human counteraction and network congestion. For more accurate worm modeling, the human counteraction is a major factor to consider in modeling a slow spreading worm, while the network congestion is a major factor to consider in modeling a fast spreading worm.

We do not, however, consider human counteraction and network congestion for most of the models in this paper. A worm's propagation in an ideal network condition exhibits the worm's essential propagation properties. Considering those two factors in modeling will make the modeling much more complex, and hence make it unclear how different worm scanning mechanisms affect a worm's propagation. When reading this paper and interpreting the modeling results, a reader should keep in mind of this modeling assumption and limitation.

4. Modeling and analysis of worm scanning strategies

4.1. Uniform scan worm and its variants

In this section, we first model well-known uniform scan worms, "Code Red" [19] and "Slammer" [18], then model their several possible variants: "hit-list worm", "routing worm" and "divide-and-conquer" scan worm.

4.1.1. Uniform scan worms that scan the entire IPv4 space

When a worm has no knowledge of where vulnerable hosts reside in the Internet, the simplest strategy is to uniformly scan the entire IP address space to find targets, which is what Code Red and Slammer did [18,26]. For such a worm, the scanning space is the entire IPv4 address space, i.e., $\Omega = 2^{32}$. Therefore, such a worm can be modeled by the uniform scan worm model (4) with $\Omega = 2^{32}$.

4.1.2. Hit-list worm

Staniford et al. [26] introduce a "*hit-list worm*", which has an IP address list of some vulnerable hosts in the Internet. A hit-list worm first scans and infects all vulnerable hosts on the hit-list, then randomly scans the entire Internet to infect others. A hit-list worm can infect all vulnerable hosts on its hit-list within just several seconds [26]. When a hit-list worm begins to scan the entire Internet after finishing its hit-list, it propagates just like an ordinary uniform scan worm. If we ignore the first several seconds used in compromising vulnerable hosts in its hit-list, a hit-list worm can be modelled by (4) with $\Omega = 2^{32}$ and a large number of initially infected hosts $I(0)$, which is equal to the size of the worm's hit-list.

4.1.3. Routing worm

In our previous paper [34], we have introduced a "*routing worm*", which uses BGP routing prefixes to reduce the worm's scanning space Ω. Based on BGP routing table, they find that currently about 28.6% of IPv4 addresses are routable. Thus, if a worm uses BGP prefixes information, which is called a "BGP routing worm", the worm reduces its scanning space by more than three times. When a BGP routing worm uniformly scans the BGP routable space, it can be modeled by the uniform scan worm model (4) by replacing $\Omega = 0.286 \times 2^{32}$.

A BGP routing worm has a big BGP prefix payload (more than 100 kB) that could possibly cause congestion and slow down the propagation of the worm. Thus, we introduce a "/8 routing worm", which only scans 116 "/8" prefix networks that contain all BGP routable addresses. In this way, a /8 routing worm only needs to carry a 116-byte payload and can be modeled by (4) with $\Omega = 0.453 \times 2^{32}$.

Because of its tiny payload, a /8 routing worm might be used by attackers in their future "bandwidth-limited worm", which is a worm that fully uses the link bandwidth of an infected host to send out infection traffic. For example, Slammer is a bandwidth-limited worm with an average scan rate $\eta = 4000$ scans/s at its early stage [18]. Each UDP infection packet sent out by Slammer is 404 bytes [18]. If attackers transform Slammer into a /8 routing worm, which is referred to as a "*routing Slammer worm*", its UDP scan packet would be 520 bytes. In this case, the "routing Slammer worm" would have an average scan rate $\eta = 1,616,000/520 = 3108$ s (since the average bandwidth of a Slammer infected host is $4000 \times 404 = 1,616,000$ bytes/s). Fig. 1 shows the worm propagation of the original Slammer and the new "routing Slammer worm" as functions of time (the other parameters are $N = 100,000$, $I(0) = 10$, the same as [32]).

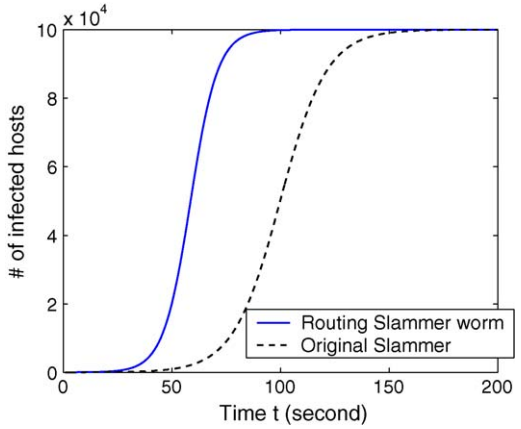


Fig. 1. Worm propagation of Slammer and "routing Slammer worm" ($N = 100,000$, $\eta = 4000$ s, $I(0) = 10$).

Note that because of limited network bandwidth, the real Slammer only propagated according to the curve in Fig. 1 at its early stage [18], the worm quickly dropped its spreading speed as the Internet was congested by worm traffic sent out from a large number of infected hosts.

4.1.4. Comparison of Code Red, a hit-list worm and routing worms

In the following, we use the same parameters for Code Red as what used in [32], i.e., $N = 360,000$, $\eta = 358$/min, $I(0) = 10$. Suppose when attackers change the original Code Red into a BGP routing worm and also a hit-list worm, the worm's scan rate does not change, i.e., $\eta = 358$ min. The BGP routing worm has $I(0) = 10$ as the original Code Red, while the hit-list worm is assumed to have a 10,000 hit-list, i.e., $I(0) = 10,000$. Fig. 2 shows the propagation of the hit-list worm, the BGP routing worm, and the original Code Red. Compared with a BGP routing worm, a hit-list worm can infect a much larger number of vulnerable hosts in a short time because of its hit- list, but it has a slower infection speed because of its larger scanning space Ω.

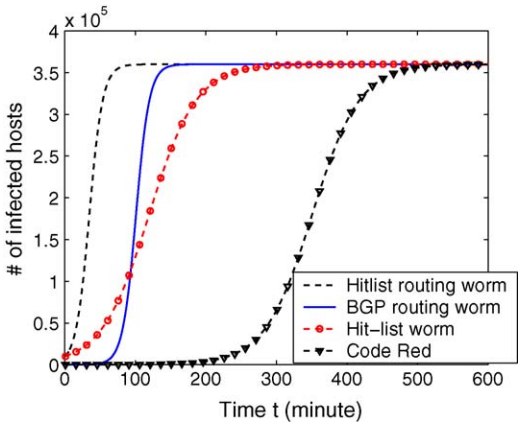


Fig. 2. Worm propagation of Code Red, BGP routing worm, hit-list worm and hit-list routing worm.

Since a hit-list worm and a routing worm use two different approaches to increase their propagation speed, they can be easily combined together to generate a new worm called a "*hit-list routing worm*". The propagation of such a worm is also shown in Fig. 2. This figure shows that a hit-list routing worm has both advantages of a hit-list worm and a routing worm, it only needs less than 50 min to finish infection instead of the 500 min used by the original Code Red.

The famous *Warhol* worm presented in [26] is a hit-list worm that uses "permutation scan" instead of uniform scan when it scans the entire IPv4 space. Permutation scan provides a form of coordination among infected hosts [26], and hence the Warhol worm propagates faster than a uniform scan hit-list worm after a large fraction of the vulnerable population has been infected (as shown in Fig. 3 in [29]). When the BGP routing worm and Code Red implement the same permutation scan as the Warhol worm instead of uniform scan, these three worms will still have the similar propagation relationship as what shown in Fig. 2 by replacing the hit-list worm with the Warhol worm. However, they cannot be modeled by the uniform scan worm model (4) anymore because of their permutation scan.

A routing worm increases its speed by reducing its scanning space; a hit-list worm increases its speed by knowing addresses of a large number of vulnerable hosts. Based on above experiments and analyses, we derive a guideline on worm defense.

Proposition 1. *It is crucial to prevent attackers from easily identifying the IP addresses of a large number of potentially vulnerable hosts through a public service (such as Google search), or obtaining address allocation information to dramatically reduce a worm's scanning space.*

This guideline tells us that: (1) we should not advertise addresses of computers unless it is absolutely necessary (such as web servers), otherwise attackers can easily get the addresses of a large number of hosts through a public service, such as the *Google* search service[1] and (2) BGP routing table is inherently public, and hence we cannot prevent attackers from exploiting it.[2] We should, however, prevent attackers from further reducing their worm's scanning space.

Of course, attackers can always generate a target list by slowly scanning the Internet to build the hit-list for their hit-list worm, it is inherent in the nature of the Internet. However, such a hit-list building process might take some time and also might alert many people of the incoming worm attack. The first half of the Proposition 1 tells us not to make it easy for attackers to build a hit-list worm.

4.1.5. Divide-and-conquer scan worm

A uniform scan worm can use a "divide-and-conquer" approach to allow different infected hosts to scan and infect vulnerable hosts on different parts of IP space, which is referred to as a "divide-and-conquer scan worm". This scanning strategy might be used by a routing worm or a hit-list worm to reduce the worm payload by carrying a small part of routing prefixes or hit-list in each worm copy.

Now we model such a divide-and-conquer scan worm. Assume that when the worm infects a target, it passes half of its scanning space to the target (the space passed to the target includes the target host), and then continues to scan the remaining half of its original scanning space. Suppose each infected host uniformly scans addresses in its scanning space without remembering what addresses have already been scanned, which means that a same address might be scanned more than once, this is the assumption

[1] For example, Santy worm used Google search engine to find web servers that have phpBB software in use [9].

[2] Because of the BGP protocol, any one of the major BGP routers around the world has (almost) complete information of all BGP routing prefixes.

used in deriving the uniform scan worm model (4). We also assume that vulnerable hosts are uniformly distributed in the worm's scanning space Ω (e.g., the entire IPv4 space or the BGP routable space); each of those $I(0)$ initially infected hosts is responsible for one scanning space with $\Omega/I(0)$ addresses.

Under the above divide-and-conquer process, each infected host is the only infected one in its scanning space. Thus, at time t the whole scanning space Ω is divided into $I(t)$ blocks: each infected host is responsible for one scanning space with the average size of $[\Omega/I(t) - 1]$ (minus 1 because an infected host will not scan itself). Since vulnerable hosts are uniformly distributed and the worm uniformly picks addresses to infect, at any time the remaining vulnerable hosts are also uniformly distributed. Thus, at time t, each infected host's scanning space contains on average $[N/I(t) - 1]$ vulnerable hosts (minus 1 because of the infected host itself).

According to (2), the probability an infected host scans a specific IP address during a time interval δ is $q = \eta\delta/[\Omega/I(t) - 1]$. Using the same infinitesimal analysis procedure as in deriving (3), we get:

$$I(t + \delta) = I(t) + I(t)\left[\frac{N}{I(t)} - 1\right] q \tag{7}$$

Taking $\delta \to 0$, we derive the propagation model of the divide-and-conquer scan worm:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\eta}{\Omega - I(t)} I(t)[N - I(t)] \approx \frac{\eta}{\Omega} I(t)[N - I(t)] \tag{8}$$

which is exactly the uniform scan worm model (4). In (8), $\Omega - I(t) \simeq \Omega$ because an Internet-scale worm has $I(t) \leq N \ll \Omega$.

Summarizing the above arguments yields.

Proposition 2. *If vulnerable hosts are uniformly distributed in the worm's scanning space, a "divide-and-conquer" scan worm has the same propagation speed as a uniform scan worm and can be modeled by (4).*

Note that Wu et al. [31] discussed a similar "divide-and-conquer" scan worm. However, they assumed that no IP address would be scanned twice. This is the reason why their "divide-and-conquer" scan worm propagates exponentially [31]. Such a scanning strategy requires that an infected host either needs to record what IP addresses it has already scanned, or needs some perfect cooperative mechanism. Therefore, we believe the divide-and-conquer scan we have modeled above is more realistic.

4.2. Idealized worm

In this section, we study two idealized worms: "*cooperative scan worm*" and "*flash worm*". We call them "idealized worms" because they are theoretical worms that are hard to be implemented by attackers on the global scale of the Internet.

4.2.1. Cooperative scan worm

A "*cooperative scan worm*" is a worm that lets all infected hosts to cooperate with each other such that no IP address would be scanned more than once. As described above, the "divide-and-conquer" scan worm in [31] satisfies the condition and thus is one kind cooperative scan worm.

Now we model a general cooperative scan worm that randomly scans its unscanned IP space; all infected hosts are assumed to have set up a perfect communication channel such that no IP address

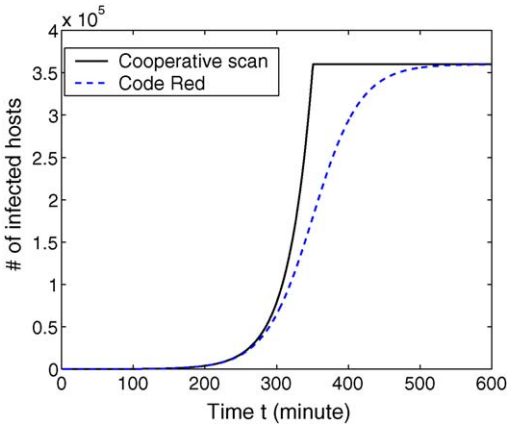


Fig. 3. Worm propagation comparison between a "cooperative scan worm" and original Code Red.

is scanned more than once. Denote $\mathrm{d}(t)$ as the density of vulnerable hosts in the unscanned IP space by time t. At beginning $t = 0$, $N' = N - I(0)$ hosts are vulnerable and the worm's scanning space has $\Omega' = \Omega - I(0)$ unscanned addresses. Thus, $d(0) = N'/\Omega'$. When the worm infects another vulnerable host at time t_0, on average Ω'/N' IP addresses have been scanned by all infected hosts. Thus, at time t_0 the density $\mathrm{d}(t_0) = [N' - 1]/[\Omega' - \Omega'/N'] = N'/\Omega'$. Carry on this process and we derive that at any time t, $\mathrm{d}(t) \equiv N'/\Omega' \simeq N/\Omega$ (when $I(0) \ll N$). From time t to time $t + \delta$, $I(t)$ infected hosts send out on average $\eta\delta I(t)$ scans, which in turn infect on average $\eta\delta I(t)d(t)$ hosts. Therefore, $I(t + \delta) = I(t) + \eta\delta I(t)N/\Omega$. Taking $\delta \to 0$, we derive the propagation model of a cooperative scan worm:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \begin{cases} \frac{\eta}{\Omega} NI(t), & I(t) < N \\ 0, & I(t) = N \end{cases} \tag{9}$$

When Code Red is changed to a cooperative scan worm, Fig. 3 shows its propagation in comparison with the original Code Red. Comparing this figure with Fig. 2, we can see that the "cooperative scan" strategy does not increase a worm's propagation speed as much as the "hit-list" or the "routing" scanning strategies. This is because at the early stage of a worm's propagation, few scans from a uniform scan worm are wasted on already scanned IP addresses. Thus, cooperative scan has little effect at the early stage of a worm's propagation.

The cooperative scanning strategy discussed above is an optimal cooperation between all worm copies, which is hard to be implemented by a worm. The "permutation scan" presented in "*Warhol worm*" [26], which can be easily implemented by a worm, is a sub-optimal cooperation. Thus, the propagation curve of a permutation scan worm lies in between those two curves shown in Fig. 3.

Summarizing this analysis yields.

Proposition 3. *An optimal cooperative scanning strategy increases a worm's propagation speed by maintaining the worm's exponential growth until finishing the infection process. However, it does not increase the speed "significantly", since it does not increase the exponential growth rate at the early stage of the worm's propagation (not as effective as the hit-list strategy or the routing scanning strategy).*

This proposition provides a fundamental understanding of cooperation in a distributed system. It shows that instead of designing various complex cooperative mechanisms, it is more effective for

attackers to design a better infection and scanning strategy (such as the hit-list scan or the routing scan).

4.2.2. Flash worm

Staniford et al. [26] introduced a "*flash worm*", which knows the IP addresses of all vulnerable hosts in the Internet and uses divide-and-conquer to scan the address list. If such an approach makes sure that no IP address is scanned more than once, the flash worm is a special cooperative scan worm that has a scanning space with the size $\Omega = N$. A "flash worm" is hard to be deployed by attackers because: (1) it needs attackers to scan the whole IPv4 Internet to collect a complete address list of all potentially vulnerable hosts and (2) to split the large address-list payload among worm copies, it requires a complicated divide-and-conquer algorithm that is robust to the removal of infected hosts.

A flash worm propagates much faster than an ordinary worm that scans the entire IPv4 space because of its scanning space $\Omega = N \ll 2^{32}$. For this reason, in the flash worm modeling we need to consider the time delay caused by the infection process of a vulnerable host. Denote ϵ as the time delay, which is the time interval from the time when a worm scan is sent out to the time when the vulnerable host infected by the scan begins to send out worm scans. Based on model (9), we derive the flash worm propagation model:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \begin{cases} \eta I(t-\epsilon), & I(t) < N \\ 0, & I(t) = N \end{cases} \tag{10}$$

where $I(t-\epsilon) = 0, \forall t < \epsilon$.

If a flash worm uniformly scans the address list of all vulnerable hosts, we call it as a "*uniform-scan flash worm*", which can be modeled by the model (4) with the worm's scanning space $\Omega = N$. After considering the propagation delay ϵ, the uniform-scan flash worm is modeled by:

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\eta}{N} I(t-\epsilon)[N - I(t)] \tag{11}$$

where $I(t-\epsilon) = 0, \forall t < \epsilon$.

In fact, a uniform-scan flash worm can be derived when maximizing the spreading speed of either a hit-list worm or a routing worm, the size of a flash worm's scanning space, $\Omega = N$, is the optimal value for the size of both the hit-list in a hit-list worm and the scanning space of a routing worm.

Fig. 4 shows $I(t)$ as a function of time for the flash worm and the uniform-scan flash worm ($\epsilon = 2$ s). To compare with previous experiments, here we use the same Code Red parameters, i.e., $N = 360,000$, $\eta = 358$ min, $I(0) = 10$. Note that these flash worms finish infection within 20 s, while Code Red finishes infection around 500 min as shown in Fig. 2 (because of flash worms' scanning space $\Omega = N \ll 2^{32}$).

Summarizing the above analysis yields.

Proposition 4. *A flash worm that uses uniform scan is an optimal spreading worm converged both from hit-list worm and from routing worm. A flash worm that conducts cooperative scan (i.e., no IP address is scanned more than once) is the fastest spreading worm in terms of worm scanning strategy.*

4.3. Local preference scan worm

Uniform scan is the simplest scanning strategy for a worm to use. However, it is not optimal since vulnerable hosts in the Internet are not uniformly distributed. Intuitively, a worm could increase its

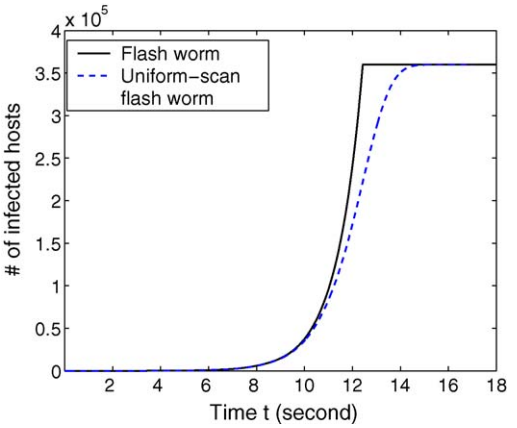


Fig. 4. Propagation of flash worms. (Note that the time scale is *second* instead of *minute* in Figs. 2 and 3.)

spreading speed when it scans more intensively in the IP space where vulnerable hosts are more densely distributed.

For this reason, attackers have implemented "*local preference scan*" in their worms, such as in Code Red II [19]: it has a higher probability to scan an IP address within the same "/16" or "/8" network than a random IP address. In general, "local preference scan" is the scanning strategy where an infected host scans IP addresses close to its address with a higher probability than addresses farther away.

In this paper, we model and analyze a local preference scan worm that has probability p to uniformly scan addresses in its own "$/n$" prefix network and probability $(1-p)$ to uniformly scan other addresses. A "$/n$" prefix network is a network containing all IP addresses that have the same first n bits. Thus, in the current IPv4 Internet, a "$/n$" prefix network contains 2^{32-n} IP addresses. Assume that the worm scanning space Ω consists of K "$/n$" prefix networks ($\Omega = K2^{32-n}$), each network has N_k vulnerable hosts, initially $k = 1, 2, \ldots, K$. Denote by $I_k(t)$ the number of infected hosts in the kth network at the time t; β' and β'' as the pairwise rates of infection in local scan and remote scan, respectively. Then according to (5), we have

$$\beta' = \frac{p\eta}{2^{32-n}}, \qquad \beta'' = \frac{(1-p)\eta}{(K-1)2^{32-n}} \tag{12}$$

Based on the epidemic model (1), the propagation of a local preference scan worm can be modeled by:

$$\frac{\mathrm{d}I_k(t)}{\mathrm{d}t} = \left[\beta' I_k(t) + \sum_{j\neq k} \beta'' I_j(t)\right] [N_k - I_k(t)] \tag{13}$$

with initial conditions $I_k(0)$ for $k = 1, \ldots, K$.

In the current Internet, computers are not uniformly distributed within the IPv4 address space. For example, currently about 116 out of the total 256 "/8" networks of the IPv4 space contain routable IP addresses [34]. Without loss of generality, suppose in K "$/n$" prefix networks, only the first m networks ($m < K$) have uniformly distributed vulnerable hosts, i.e., $N_1 = \cdots = N_m = N/m$, $N_{m+1} = \cdots = N_K = 0$. However, attackers do not know which "$/n$" prefix networks are empty (otherwise, attackers can use the "routing worm" idea [34] to remove those empty networks from the worm's scanning space). From

(13), the worm propagation on each network follows ($k = 1, \ldots, m$):

$$\frac{\mathrm{d}I_k(t)}{\mathrm{d}t} = [\beta' + (m-1)\beta'']I_k(t)[N_k - I_k(t)] \tag{14}$$

Suppose $I_k(0) = I_1(0) > 0$, $k = 2, 3, \ldots, m$. From the entire Internet point of view, the worm follows

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = m\frac{\mathrm{d}I_1(t)}{\mathrm{d}t} = \frac{\beta' + (m-1)\beta''}{m} I(t)[N - I(t)] \tag{15}$$

A local preference scan worm propagates fastest when it chooses the optimal value $p = 1$ to maximize the pairwise rate of infection $[\beta' + (m-1)\beta'']/m$ in (15). Such a conclusion seems unexpected, but it is reasonable for the assumptions we have used, all m "$/n$" networks are assumed to be identical. When $p = 1$, no worm scans will be wasted in the other $(K - m)$ empty "$/n$" networks.

In reality, no network is exactly the same as others. Remote scan is necessary for a worm to spread out to every part of the Internet. When we assume that initially $I(0) = I_1(0) > 0$ and $I_k(0) = 0$, $k = 2, 3, \ldots, m$, then a local preference scan worm requires $p < 1$ in order to spread out into other networks. For this scenario, $I_k(t) \equiv I_2(t)$, $k = 3, \ldots, m$. Hence, the worm propagation on each prefix network is described by:

$$\begin{aligned}
\frac{\mathrm{d}I_1(t)}{\mathrm{d}t} &= [\beta' I_1(t) + (m-1)\beta'' I_2(t)]\left[\frac{N}{m} - I_1(t)\right] \\
\frac{\mathrm{d}I_k(t)}{\mathrm{d}t} &= [\beta'' I_1(t) + (\beta' + m\beta'' - 2\beta'')I_k(t)]\left[\frac{N}{m} - I_k(t)\right]
\end{aligned} \tag{16}$$

for $k = 2, 3, \ldots, m$. From the entire Internet point of view, $I(t) = I_1(t) + (m-1)I_2(t)$.

Serazzi and Zanero [23] presented a "compartment-based" model that is similar to (16). The compartment-based model tries to explain the worm infection process between Autonomous Systems; here, we use the model (16) to explain and model the propagation of a local preference scan worm. In fact, these two models are essentially the same, both of them try to model the worm propagation within many interacting groups.

We use Matlab Simulink [12] to solve the model above under different preference probabilities p. We use the previous Code Red parameters in the study here, i.e., $N = 360,000$, $\eta = 358$ min, $I(0) = I_1(0) = 10$. We consider two scenarios to study the effect of the size of "$/n$" prefix networks: one considering local scan on "/8" networks and another considering local scan on "/16" networks.

For the first scenario where each prefix network is "/8", $K = 2^8 = 256$. [34] points out that currently around 116 "/8" networks are routable, thus $m = 116$. Based on these parameters, Fig. 5(a) shows $I(t)$ under different preference probabilities p. For comparison, we also show the original Code Red propagation on this figure (the one labelled as "uniform scan worm"). If attackers know that $m = 116$ "/8" networks have vulnerable hosts and know their prefixes, they can implement the "/8 routing worm" [34] to uniformly scan the m "/8" networks IP space only. The propagation of such a worm is also shown in Fig. 5(a).

For the second scenario where each network is "/16", $K = 2^{16} = 65536$. Since 116 "/8" networks have been allocated and each one of them contains 2^8 "/16" networks, we assume $m = 116 \times 2^8 = 29,696$. Based on these parameters, Fig. 5(b) shows $I(t)$ under different preference probabilities p.

Fig. 5 shows that when vulnerable hosts are not uniformly distributed, local preference scan increases a worm's propagation speed. When local scan is on "/16" prefix networks, the optimal value of p is about 0.85; on the other hand, it is close to one when local scan is on "/8" prefix networks.

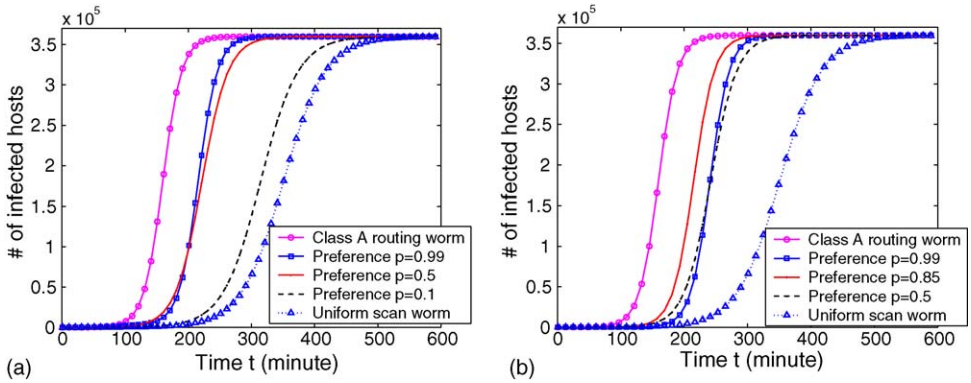


Fig. 5. Comparison of Code Red, a /8 routing worm, a local preference worm with different preference probabilities p. (a) Local preference scan on "/8" network level ($K = 256$, $m = 116$). (b) Local preference scan on "/16" network level ($K = 65,536$, $m = 29,696$).

Therefore, the optimal value of p is determined by the locality in local scan. We explain it in an intuitive way: when x "$/n$" networks have infected hosts initially, the purpose of remote scans is to spread out worm seeds to every one of the other $(m - x)$ "$/n$" networks. If a worm uses "/8" network in its local preference scan, it needs to spread out worm seeds to at most $m = 116$ networks. On the other hand, using "/16" network in its local scan, a worm needs to spread out worm seeds to everyone in those $m = 29,696$ networks. Therefore, a "/16" local scan worm needs much more effort to spread out worm seeds than a "/8" local scan worm.

Summarizing the above analysis yields.

Proposition 5. *When vulnerable hosts are not uniformly distributed in a worm's scanning space, local preference scan increases a worm's propagation speed. The optimal local preference scan probability p increases as the size of local scan networks increases.*

Note that in this analysis, we have not considered the impact of possible network congestion and the impact of different network connection speed. If a worm extensively uses local preference scan, the intense local scan traffic might cause congestion to local networks and slow down the worm's overall spreading speed. In this case, the optimal local scan probability p should be smaller than our derived theoretical optimal value to avoid local congestion. On the other hand, a local preference scan worm enjoys the benefit of better network connections and smaller values of Round-Trip-Time (RTT) in its local scanning. So if there is no congestion issue, then the optimal local scan probability p should be a bit bigger than the theoretical optimal value derived in our analysis.

Besides the speed concern, other advantages of using local preference scan include: (1) infect more hosts in local networks that are either behind firewalls or have private IP addresses and (2) decrease the number of scans that across organizational perimeters (where advanced intrusion detection systems are more likely to be deployed) to avoid detection.

Chen et al. [4] modeled local preference scan- based on their discrete-time worm propagation model. However, they assumed that vulnerable hosts are evenly distributed in every subnet, which is not the case for the current Internet. In our analysis, we have used a more realistic assumption that vulnerable hosts are uniformly distributed in BGP routable subnetworks.

4.4. Sequential scan worm

Until now we have assumed that worms scan IP addresses not in any order. A very different scanning strategy is to scan IP addresses *sequentially*. Once a vulnerable host is infected by a "*sequential scan worm*", it begins to sequentially scan from a starting IP address selected by the worm. Blaster is a typical sequential scan worm [8]. Without loss of generality, we assume that a sequential scan worm scans IP addresses additively.

In our previous analysis, we find that local preference scan increases the spreading speed of a random scan worm. For a sequential scan worm, "local preference" means when a worm chooses its starting point, it chooses an address close to its own one with a higher probability than an address far away. We call such a worm as a "*preference sequential scan worm*". Blaster is such a worm: it chooses its starting point locally as the first address of its Class C subnetwork with a probability 0.4 [8].

Now we analyze the effect of such a local preference scan on a worm's propagation. When an infected host (parent) finds and infects a vulnerable host (child) that has address x, the parent will keep going on to scan IP addresses $x+1, x+2, \ldots$. If the child infected host uses local preference to select its starting point, it is more likely to repeat its parent's scanning trail, i.e., repeatedly scans IP addresses $x+1, x+2, \ldots$ that have already been scanned by its parent. Therefore, the local preference strategy wastes most of the infection power of infected hosts that have chosen local IP addresses to start scanning.

Summarizing the analysis above yields.

Proposition 6. *For a sequential scan worm, using local preference in selecting the worm's starting point slows down the worm's propagation speed.*

For this reason, we mainly model and analyze a sequential scan worm with a uniformly chosen starting point, which is referred to as a "*non-preference sequential scan worm*". First, we analyze the propagation of such a worm when vulnerable hosts are uniformly distributed. Suppose at time t, $I(t)$ hosts are infected by a non-preference sequential scan worm; and the density of vulnerable hosts in the worm's scanning space is $[N - I(t)]/\Omega$. During the next small time interval δ, each infected host sequentially scans $\eta\delta$ IP addresses and thus infects on average $\eta\delta[N - I(t)]/\Omega$ vulnerable hosts. When δ is sufficiently small, the probability of two infected hosts infecting the same vulnerable target is negligible. Thus, we have

$$I(t+\delta) = I(t) + \frac{\eta\delta}{\Omega} I(t)[N - I(t)] \tag{17}$$

Taking $\delta \to 0$, we derive the same propagation model as the uniform scan worm model (4).

Summarizing the analysis above yields.

Proposition 7. *If vulnerable hosts are uniformly distributed in the worm's scanning space, a sequential scan worm that uniformly selects its starting point has the same propagation speed as a uniform scan worm and can be modeled by* (4).

To verify our analysis, we simulate a "uniform scan worm" (Code Red); a "preference sequential scan worm" that chooses its starting point locally with probability 0.4 (Blaster) and a "non-preference sequential scan worm". For comparison, we use the same Code Red parameters, $N = 360,000$, $I(0) = 10$, $\eta = 358$ min, for all these three worms. Considering that different infected hosts have different scan rates, we use the same simulation method as in [32]: every host has a scan rate x while x follows a normal distribution $N(358, 100^2)$.

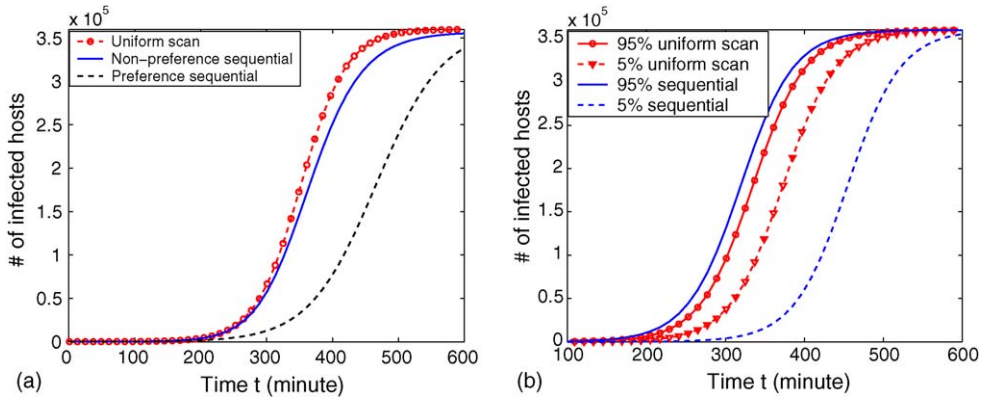


Fig. 6. Comparison of a non-preference sequential scan worm, a preference sequential scan worm with 40% local preference, and a uniform scan worm (vulnerable hosts uniformly distributed in the BGP routable space; 100 simulation runs). (a) Mean value of worm propagation and (b) variabilities in worm propagation.

In this experiment, we need to consider the distribution of vulnerable hosts in the Internet. Since we do not know the real distribution, a reasonable approach is to assume that all vulnerable hosts are uniformly distributed in the routable space defined by BGP routing prefixes [34]. We simulate each of the above three worms 100 times and show the results in Fig. 6. Fig. 6(a) plots the mean value of $I(t)$ of each worm over 100 simulation runs; Fig. 6(b) shows the variability of worm propagation for the uniform scan worm and the non-preference sequential scan worm. The "95%" propagation curve means that a worm propagates no faster than this curve in 95 out of 100 simulation runs. Therefore, in 90 out of 100 simulation runs, a worm propagates within the boundary defined by the "5%" and "95%" propagation curves.

Fig. 6(a) agrees with Proposition 6: for a sequential scan worm, using local preference in selecting its starting point significantly slows down the worm's propagation speed. Fig. 6(b) shows that the propagation of a sequential scan worm varies considerably because of the non-uniform distribution of vulnerable hosts.

To verify Proposition 7, we run the above 100-simulation experiment again with the assumption that vulnerable hosts are uniformly distributed in the entire IPv4 space. This time the experiment shows that the average $I(t)$ (over 100 simulation runs) of the non-preference sequential scan worm is identical to the uniform scan worm, which verifies Proposition 7. In Fig. 6(a), because of the non-uniform distribution of vulnerable hosts, the non-preference sequential scan worm propagates slightly slower than the uniform scan worm.

4.5. Selective attack worm

Attackers may not want their worm to infect or destroy all compromised computers in the Internet. In a selective attack, attackers only care about how fast a worm propagates in the target domain, not how many vulnerable hosts have been infected in the global Internet. Suppose the target domain has N_e vulnerable hosts and a scanning space of size Ω_e; the other domains have N_o vulnerable hosts and a scanning space of size Ω_o. Thus, overall $N = N_e + N_o$, $\Omega = \Omega_e + \Omega_o$. Denote by $I_e(t)$ and $I_o(t)$ the number of infected hosts at time t in the target domain and in the other domains, respectively.

"*Target-only*" scanning strategy means that a selective attack worm only scans and infects vulnerable hosts in the target domain. In this case, if the worm uniformly scans the target domain, the worm's

propagation follows the uniform scan worm model (4) by replacing the scanning space Ω to Ω_e:

$$\frac{dI_e(t)}{dt} = \frac{\eta}{\Omega_e} I_e(t)[N_e - I_e(t)] \tag{18}$$

On the other hand, a selective attack worm can uniformly scans the entire scanning space Ω. We call such a worm as a "*global scan worm*", which is modeled by the uniform scan worm model (4). The question is: which worm propagates faster in the target domain, the target-only worm or the global scan worm?

Assume that $c_1 = \Omega_e/\Omega$ and $c_2 = N_e/N$; if $c_2 > c_1$, vulnerable hosts are more densely distributed in the target domain than in other domains. The global scan worm has $I_e(t) = c_2 I(t)$ because of its uniform scan strategy. Substituting this equation into model (4), we derive:

$$\frac{dI_e(t)}{dt} = \frac{\eta}{c_2\Omega} I_e(t)[N_e - I_e(t)] \tag{19}$$

Comparing (18) with (19), we observe that, if $c_2 > c_1$, i.e., vulnerable hosts are more densely distributed in the target domain than in other domains, then the target-only worm propagates faster than the global scan worm in the target domain (and vice versa).

Summarizing the above analysis yields.

Proposition 8. *For a selective attack worm, if vulnerable hosts are more densely distributed in the target domain than in other domains, the worm propagates faster in the target domain if it only scans the IP space of the target domain instead of all domains.*

5. Modeling destructive worm: Witty

Most previous worms are "benevolent" worms that do not destroy any resource on compromised computers. However, recent "*Witty*" worm (appeared on March 20, 2004) is the first widely spread worm that carries a destructive payload [3]. We call such a worm as a "*destructive worm*". A destructive worm wants to destroy as many computers as possible, thus it should destroy a compromised computer as soon as possible to prevent people from having time to clean the computer. However, destroying a compromised computer might make the computer unable to send out worm packets to infect others, and thus, prevent the worm from spreading out quickly. Therefore, a destructive worm usually needs to make a trade-off between destroying a computer and holding an infected computer for propagation.

After sending out 20,000 infection packets, Witty worm writes 65 K data to a random point of hard disk on a compromised computer and then repeats this process until the computer is crashed due to the random destruction of hard disk [3]. Since the destructive data has fixed size and is written to a random point of hard disk, each hard disk destructive writing has a small but constant probability q to crash a compromised computer, where q is determined by the computer's operating system and hard disk volume. Suppose a compromised computer is crashed when the worm writes X times destructive data, then X follows "*geometric distribution*". Define "*destruction time*" as the time interval from compromising a computer to the crashing of the computer due to Witty's destructive data. Suppose a compromised computer takes a constant time T to send out 20,000 infection packets due to its constant network bandwidth, then the destruction time follows geometric distribution, too. Because $q \ll 1$, the

destruction time can be roughly modeled by a continuous "*exponential distribution*" with a rate q/T. Due to differences in network bandwidth and hard disk volume, different Witty compromised computers have exponential distributed destruction time with different rates, we denote the average "destruction rate" as λ.

Denote $D(t)$ as the number of crashed computers due to the destruction of Witty worm. Suppose there are $I(t)$ infectious hosts at time t. In the next small time interval δ, each infected host has the small probability $\lambda\delta$ to be crashed, regardless of how long the host has been infected due to the memoryless exponential distribution of destruction time. Thus, on average $\lambda\delta I(t)$ infected hosts will be crashed and removed from $I(t)$ from time t to time $t+\delta$ (added to $D(t)$). Taking $\delta \to 0$ and from the uniform scan worm model (4), we derive the worm propagation model for Witty worm (which is identical to the Kermack–Mckendrick epidemic model [6]):

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t} = \frac{\eta}{\Omega} I(t)[N - I(t) - D(t)] - \frac{\mathrm{d}D(t)}{\mathrm{d}t}, \quad \frac{\mathrm{d}D(t)}{\mathrm{d}t} = \lambda I(t) \tag{20}$$

We obtain a set of Witty monitored data from the "Internet Motion Sensor" (IMS) in University of Michigan [1]. The data shows the number of Witty scans per 1000 s observed by IMS on three "/24" blackhole networks, it can represent the number of infectious hosts $I(t)$ in the entire Internet since Witty uniformly scanned the Internet. In the modeling, $\Omega = 2^{32}$ since Witty scanned the entire IPv4 space. Witty infected about 12,000 machines with around 110 initially infected hosts [3], thus in our model (20), we choose $N = 12,000$, $I(0) = 110$. To match the model (20) with the monitored worm trace, we choose the worm's average scan rate $\eta = 1200$ s and the average destructive rate $\lambda = 0.000025$ s, this value of λ means that on average a Witty infected computer would be crashed $1/\lambda = 11.1$ h after it was compromised.

Fig. 7(a) shows the number of infectious hosts $I(t)$ derived from model (20) compared with the monitored trace within the first 24 h of the worm's outbreak. It shows that our model (20) could model well the propagation of Witty worm.

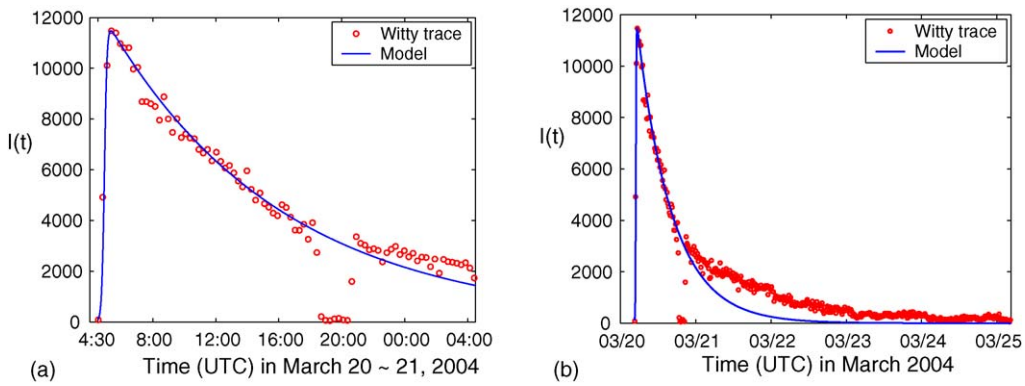


Fig. 7. Witty worm modeling compared with real monitored trace. (a) Infectious population in the first 24 h and (b) long-time infectious population.

However, if we consider the monitored worm trace over several days, we find that, as shown in Fig. 7(b), Witty died out slower than our model (20) predicts. This is because as time went on, Witty infectious computers had a decreasing *average* destruction rate λ, not a constant rate as used in model (20). Due to hard disk volume and network bandwidth differences, different Witty infected computers had very different destruction time. After Witty infected almost all vulnerable computers in the Internet within 1 h [3], compromised computers with larger destruction rates were crashed first, making the remaining infectious computers to have a decreasing average destruction rate λ as time went on, the decreasing λ made the Witty infectious population $I(t)$ dropped slower than the model (20) predicts.

There are two other possible reasons for the slow decay of Witty infected population in the long term: (1) some new vulnerable hosts gradually came on online in those several days and got infected and (2) some infected hosts were counted multiple times due to the changes of their DHCP-assigned IP addresses. However, without packet-level monitored data, we cannot analyze the impact of these two factors. Currently, we are trying to obtain such detailed monitored data from other researchers in order to conduct further analysis.

Summarizing the above analysis yields.

Proposition 9. *A computer infected by Witty worm has a crash time that is exponentially distributed; the mean value of the crash time of an infected computer is proportional to the computer's hard disk volume and inversely proportional to its network bandwidth.*

6. Worm monitoring system design

For worm defense, we first need to set up a worm monitoring infrastructure to monitor and early detect the presence of a worm in the Internet. CAIDA has set up a large-scale network monitoring system by using the "network telescope" concept [20], which covers several large chunks of IP space. Although such a monitoring system is good at monitoring a uniform scan worm such as Code Red and Slammer, it performs poorly for a non-uniform scan worm, especially a sequential scan worm, such as Blaster. For example, Slammer is a uniform scan worm with $\eta = 4000$ scans/s [18]. If a monitoring system covers one big chunk of IP block consisting of 2^{17} addresses (two Class B networks), then on average an infected host can be observed $2^{(32-17)}/\eta = 8.2$ s after it is infected.[3] On the other hand, if Slammer randomly chooses a starting point to sequentially scan the Internet, on average an infected host requires $2^{32}/(2\eta) = 6.2$ days to be observed by the monitoring system (an infected host on average needs to scan half of IPv4 space to hit the monitored IP block). Therefore, if an infected host sequentially scans the Internet by starting far from any monitored address, the monitoring system will not be able to observe it for a long time.

In previous Blaster simulation, the "preference sequential" worm shown in Fig. 6(a), we have also simulated two monitoring systems: one monitors 16 blocks of Class B IP space; another monitors 1024 equal-size blocks of IP space of size 2^{10}. Both monitoring systems monitor the same number of IP addresses (2^{20}) and all monitored address blocks are evenly distributed in the entire IPv4 space.

A worm monitoring system primarily observes two data sets [32]: the number of scans observed in each monitoring time interval, denoted by $Z(t)$ and the cumulative number of infected hosts observed by time t, denoted by $C(t)$. Fig. 8(a) shows the number of infected hosts $I(t)$ in the Internet

[3] Because of uniform scan, each worm scan has probability $1/2^{(32-17)}$ to hit the monitoring system. The hitting event forms a *Bernoulli trial*, and hence, an infected host needs to send out on average $2^{(32-17)}$ scans to hit the monitoring system once.

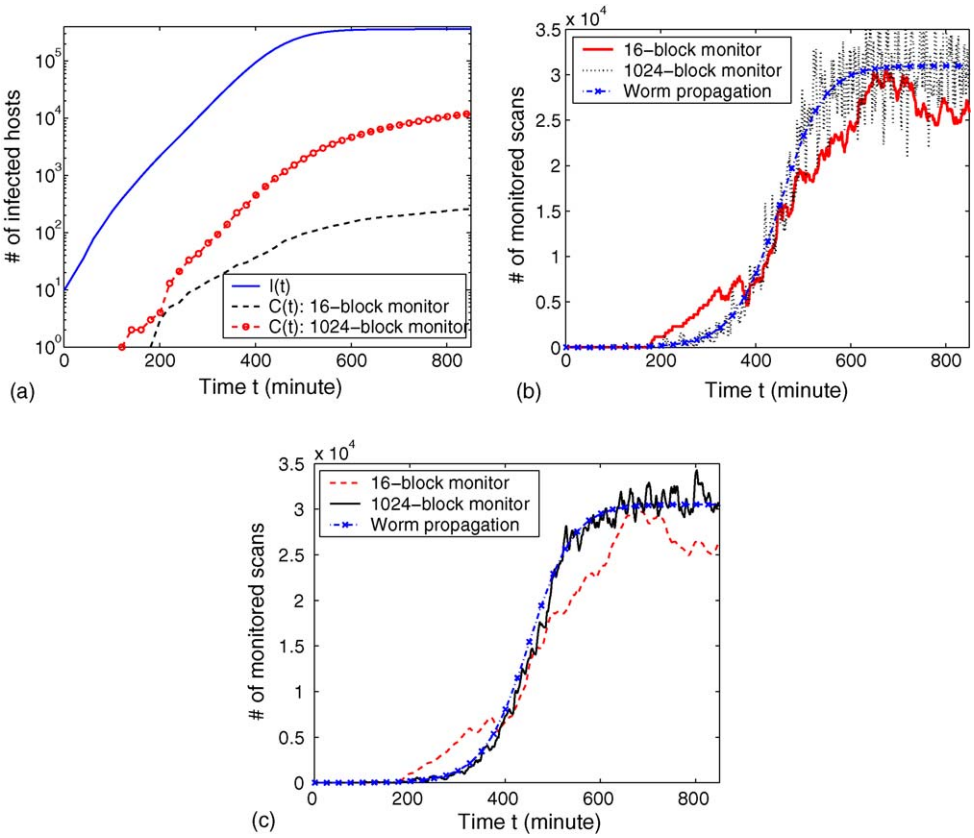


Fig. 8. Blaster propagation and its monitoring (vulnerable hosts are uniformly distributed in BGP routable space; this figure shows the results of one simulation run). (a) Worm propagation and monitored infected $C(t)$ (Y-axis is log format). (b) Monitored data $Z(t)$ compared with worm propagation. (c) Monitored data $Z(t)$ after using a low-pass filter.

as a function of time t for one simulation run. It also shows $C(t)$ from both monitoring systems. Because observed $C(t)$ is very small compared with $I(t)$, we plot this figure by taking logarithm on Y-axis. This figure shows that we can observe less than 0.1% of infected hosts in the Internet from the 16-block monitoring system during the worm's propagation period. Even if we use a 1024-block monitoring system, we can only observe less than 4% of infected hosts. This is the reason why until now researchers have not derived an accurate estimate of how many computers were really infected by Blaster worm.

Fig. 8(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 more frequently in the 1024-block monitoring system than in the other one.

From the monitored data sets, we want to know the worm propagation pattern on the global Internet, i.e., the curve of $I(t)$ shown in Fig. 8(b). Such a growth pattern of $I(t)$ is a low frequency signal compared with the high frequency noise contained in $Z(t)$. Therefore, we can use a low-pass filter to filter out noise

from $Z(t)$ without changing the worm's propagation pattern. Fig. 8(c) shows the observation data $Z(t)$ after filtered by a first-order low-pass filter.[4] Fig. 8(c) shows that the observation data $Z(t)$ from the 1024-block monitoring system represents better a worm's propagation pattern than the 16-block monitoring system.

Worm propagation in other Blaster simulation runs give similar results to what shown in Fig. 8. On occasion the 16-block monitoring system provides as good observation as the 1024-block monitoring system. However, the 1024-block monitoring system provides stable observations in all simulation runs, while the 16-block monitoring system provides very poor observations in many instances.

Summarizing the analysis above yields.

Proposition 10. *In order to monitor the propagation of a non-uniform scan worm in the Internet, especially the propagation of a sequential scan worm, the address space covered by a monitoring system should be as distributed as possible.*

7. Conclusion

Like earthquake modeling or tornado modeling, a good Internet worm model can: (1) give us deep understanding of the dynamics of a worm; (2) provide the simulation basis for accurately evaluating the performance of various worm defense systems and (3) help us to generate effective early warning (the early warning system in [32] is based on worm model) and provide accurate worm damage prediction.

For these purposes, based on a uniform modeling framework, we model and analyze many different worms that use various scanning strategies, including uniform scan, hit-list scan, routing scan, local preference scan, cooperative scan, sequential scan, divide-and-conquer scan, target scan, destructive scan, etc. Although most conclusions drawn in this paper are intuitively clear, we prove them through sound mathematical models that have the same underlying principles. The analysis in this paper makes it clear how different scanning strategies are related with each other. We hope this paper could provide a solid framework on Internet worm modeling and help us to better understand and defend against future Internet worm attacks.

Acknowledgements

We gratefully thank researchers in University of Michigan "Internet Motion Sensor" for providing us their monitoring data on Witty worm propagation. This work was supported in part by ARO contract DAAD19-01-1-0610, NSF Grant EEC-0313747, EIA-0080119, ANI-0085848 and CNS-0325868.

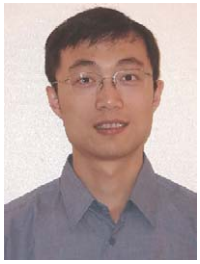
References

[1] University of Michigan Internet Motion Sensor, http://ims.eecs.umich.edu/.

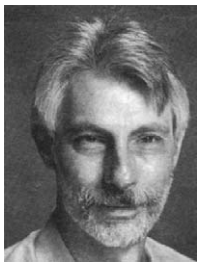
[4] Denote by $\hat{Z}(t)$ as the $Z(t)$ after filtering. The low-pass filter is $\hat{Z}(t) = aZ(t) + (1-a)\hat{Z}(t-1)$. We use $a = 0.1$ in Fig. 8(c).

[2] V.H. Berk, R.S. Gray, G. Bakos, Using sensor networks and data fusion for early detection of active worms, Proceedings of the SPIE AeroSense, 2003.
[3] CAIDA. The spread of the witty worm. http://www.caida.org/analysis/security/witty, March 2004.
[4] Z. Chen, L. Gao, K. Kwiat, Modeling the spread of active worms, Proceedings of the IEEE INFOCOM, 2003 March.
[5] F. Cohen, Computer viruses: theory and experiments, Comput. Secur. 6 (February(1)) (1987).
[6] D.J. Daley, J. Gani, Epidemic Modeling: An Introduction, Cambridge University Press, 1999.
[7] eEye Digital Security, ida "Code Red" worm. http://www.eeye.com/html/Research/Advisories/AL20010717.html, 2001.
[8] eEye Digital Security, Blaster worm analysis. http://www.eeye.com/html/Research/Advisories/AL20030811.html, 2003.
[9] F-Secure, F-secure virus descriptions: Santy. http://www.f-secure.com/v-descs/santy_a.shtml, 2004.
[10] LURHQ Threat Intelligence Group, Sasser worm analysis. http://www.lurhq.com/sasser.html, May 2004.
[11] Honeynet Project, know your enemy: Honeynets. http://project.honeynet.org/papers/honeynet.
[12] Mathworks Inc., Simulink. http://www.mathworks.com/products/simulink.
[13] J.O. Kephart, D.M. Chess, S.R. White, Computers and epidemiology, IEEE Spectr. 30 (May(5)) (1993).
[14] J.O. Kephart, S.R. White, Directed-graph epidemiological models of computer viruses, Proceedings of IEEE Symposium on Security and Privacy, 1991, pp. 343–359.
[15] J.O. Kephart, S.R. White, Measuring and modeling computer virus prevalence, Proceedings of IEEE Symposium on Security and Privacy, 1993.
[16] G. Kesidis, I. Hamadeh, S. Jiwasurat, Coupled kermack-mckendrick models for randomly scanning and bandwidth-saturating internet worms, Proceedings of 3rd International Workshop on QoS in Multiservice IP Networks (QoS-IP), 2005 February.
[17] M. Liljenstam, D. Nicol, Comparing passive and active worm defenses, Proceedings of the 1st International Conference on Quantitative Evaluation of Systems (QEST), 2004 September.
[18] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver, Inside the Slammer worm, IEEE Mag. Secur. Privacy 1 (July(4)) (2003).
[19] D. Moore, C. Shannon, J. Brown, Code-Red: a case study on the spread and victims of an Internet worm, Proceedings of the Second ACM SIGCOMM Workshop on Internet Measurement, 2002 November.
[20] D. Moore, C. Shannon, G.M. Voelker, S. Savage, Network Telescopes: Technical Report. Technical Report TR-2004–04, CAIDA, 2004.
[21] D. Nicol, M. Liljenstam, Models of active worm defenses, Proceedings of the IPSI Studenica Conference, 2004 June.
[22] N. Provos, A virtual honeypot framework, Proceedings of 13th USENIX Security Symposium, 2004 August.
[23] G. Serazzi, S. Zanero, Computer virus propagation models, Proceedings of 11th IEEE/ACM Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), 2003 October.
[24] E. Spafford, The internet worm: crisis and aftermath, Commun. ACM 32 (June(6)) (1989).
[25] S. Staniford, Containment of scanning worms in enterprise networks, J. Comput. Secur. (2003).
[26] S. Staniford, V. Paxson, N. Weaver, How to own the Internet in your spare time, Proceedings of USENIX Security Symposium, 2002 August.
[27] T. Vogt, Simulating and optimising worm propagation algorithms. http://web.lemuria.org/security/WormPropagation.pdf, September 2003.
[28] A. Wagner, T. Dubendorfer, B. Plattner, R. Hiestand, Experiences with worm propagation simulations, Proceedings of ACM CCS Workshop on Rapid Malcode (WORM'03), 2003 October.
[29] N. Weaver, Warhol worms: The potential for very fast internet plagues. http://www.cs.berkeley.edu/~nweaver/warhol.html, 2001.
[30] N. Weaver, V. Paxson, S. Staniford, R. Cunningham, A taxonomy of computer worms, Proceedings of ACM CCS Workshop on Rapid Malcode (WORM'03), 2003 October.
[31] J. Wu, S. Vangala, L. Gao, K. Kwiat, An efficient architecture and algorithm for detecting worms with various scan techniques, Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS'04), 2004 February.
[32] C.C. Zou, L. Gao, W. Gong, D. Towsley, Monitoring and early warning for Internet worms, Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03), 2003 October.
[33] C.C. Zou, W. Gong, D. Towsley, Code Red worm propagation modeling and analysis, Proceedings of 9th ACM Conference on Computer and Communications Security (CCS'02), 2002 October.

[34] C.C. Zou, D. Towsley, W. Gong, S. Cai, Routing worm: A fast, selective attack worm based on IP address information, Proceedings of 19th ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation (PADS), 2005 June.



Cliff C. Zou received his PhD degree in electrical and computer engineering from University of Massachusetts, Amherst in 2005, his BSc degree and MSc degree in electrical engineering from University of Science and Technology of China, Hefei, China in 1996 and 1999, respectively. Currently, he is an assistant professor in the School of Computer Science in the University of Central Florida. His research interests include computer and network security, network modeling and performance evaluation.



Don Towsley (M'78-SM'93-F'95) holds a BA in physics (1971) and a PhD in computer science (1975) from University of Texas. 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.

He 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. Last, he has been elected fellow of both the ACM and IEEE.

Weibo Gong (S'87″CM'87″CSM'97″CF'99) received his PhD degree from Harvard University in 1987, and have been with the Department of Electrical and Computer Engineering, University of Massachusetts, Amherst since then. He is also an adjunct professor in the Department of Computer Science at the same campus. His major research interests include control and systems methods in communication networks, network security and network modeling and analysis.

He is a receipient of the IEEE Transactions on Automatic Control's George Axelby Outstanding paper award, an IEEE Fellow and the Program Committee Chair for the 43rd IEEE Conference on Decision and Control.