On the Performance of Internet Worm Scanning Strategies

Cliff Changchun Zou*, Don Towsley†, Weibo Gong*
*Department of Electrical & Computer Engineering
†Department of Computer Science
Univ. Massachusetts, Amherst
Technical Report: TR-03-CSE-07

***Abstract*— In recent years, fast spreading worms have become one of the major threats to the security of the Internet. In order to defend against future worms, it is important to understand how worms propagate and how different scanning strategies affect their propagation. In this paper, we model and analyze worm propagation under various scanning strategies, such as idealized scan, uniform scan, divide-and-conquer scan, local preference scan, sequential scan, target scan, etc. We also analyze and discuss how attackers could optimize their scanning strategies, and provide some guidelines for building up a monitoring infrastructure to defend against future worms.**

I. Introduction

Since the Morris worm in 1988 [6], the security threat posed by worms has steadily increased, especially in the last several years. In 2001, the Code Red and Nimda worms infected hundreds of thousands of computers [7][21], causing millions of dollars loss to our society [24]. The Slammer worm appeared on January 25th, 2003, and quickly spread throughout the Internet. Because of its super fast scan rate, Slammer infected more than 90% of vulnerable computers in the Internet within 10 minutes [8] and generated severe denial of service attacks on many networks across Asia, Europe, and America [25]. Just seven months later, the Blaster worm appeared and spread out quickly in the Internet on August 11th. In the following days, Blaster and its many variants repeatedly attacked the Internet.

Attackers have tried many scanning strategies in recent worms. Code Red and Slammer uniformly scan the entire IPv4 space [8][22]. Blaster sequentially scans the Internet. Code Red II also use a local preference scan in its propagation: Code Red II has a higher probability of scanning an IP address within the same Class B or Class A network than a random address [7]. In its sequential scan, Blaster chooses to sequentially scan from a local IP address with probability 0.4 [26].

We believe that in the future, attackers will continue to implement a variety of scanning strategies to increase their worms' spreading speed and defeat our defenses. In this paper, we mathematically model and analyze various scanning strategies that attackers have already used or may use in the future. Mathematical analysis provides a deep understanding of how different factors affect a worm's propagation. The scanning strategies we analyze include idealized scan, uniform scan, divide-and-conquer scan, local preference scan, sequential scan, target scan, etc. We also combine numerical analysis and simulation experiments in our modelling and analysis.

A better understanding of how various scanning strategies affect a worm's propagation can lead to better defense against future worms. From our analysis, we derive the following conclusions:

- A local preference scan increases a worm's propagation speed when vulnerable hosts are not uniformly distributed. The optimal local preference probability increases when the local scan is on larger subnetworks.
- When vulnerable hosts are uniformly dis-

tributed, the divide-and-conquer scan, the sequential scan, and the uniform scan are equivalent in terms of the total number of infected hosts at any time.
- For a sequential scan worm, using local preference in selecting the starting point slows down the worm's propagation speed.
- For a selective attack worm [17], when the density of vulnerable hosts in the target domain (the ratio of the number of vulnerable hosts over the number of IP addresses in the domain) is higher than in other domains, the worm propagates faster in the target domain if it scans within the target domain than uniformly scans all domains (and vice versa).

We also provide some guidelines in designing our defense system:

- It is crucial to prevent attackers from identifying the IP addresses of a large number of vulnerable hosts, or obtaining address information to dramatically reduce their worm's scanning space.
- A worm monitoring system should cover many well distributed IP blocks in order 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 II surveys related work. In Section III, we introduce two classical simple epidemic models and analyze the underling principles in deriving them. In Section IV, we model and analyze worm propagation under different scanning strategies. Based on our analyses, in Section V we present an important principle in building up a worm monitoring system. In the end, Section VI concludes this paper.

II. Related Work

People have studied modelling and analysis of the propagation of viruses for a long time. Kephart, White and Chess of IBM performed a series of studies from 1991 to 1993 on viral infection based on epidemiology models [3][4][5]. Wang *et al.* presented simulation studies of a simple virus propagation on clustered and tree-like hierarchical networks [11]. Based on the eigenvalues of network graphs, Wang *et al.* presented the epidemic threshold for virus propagation on arbitrary network topologies [12]. Wang *et al.* modelled virus propagation by considering virus infection delay and user vigilance [13].

The Code Red incident on July 2001 [22] stimulated a number of models and analyses of Internet worm propagation. Staniford *et al.* used the "classical simple epidemic model" [2] to model the spread of Code Red right after the Code Red incident [9]. Their model matched well the increasing part of the observation data. Zou *et al.* presented a "two-factor" worm model that considered both the effect of human countermeasures and the effect of the congestion caused by worm scan traffic [15]. Chen *et al.* presented a discrete-time version worm model that considered the patching and cleaning effect during a worm's propagation [1]. In their worm early detection paper, Zou *et al.* presented the relationship between a worm's scan rate, infection rate, scanning space, and the vulnerable population size [16]. Weaver *et al.* [14] presented a taxonomy of computer worms based on several factors: target discovery, carrier, activation, payloads, and attackers.

As researchers understand more how a worm propagates, they identify various ways to make a worm propagate faster as well. Staniford *et al.* presented the "hit-list worm" and "flash worm" [9]. These two worms build a partial or a complete list of IP addresses of vulnerable hosts into worm code, and thus they can dramatically shorten their propagation time. Zou *et al.* presented a "routing worm" that takes advantage of BGP routing prefixes to reduce a worm's scanning space to less than 30% of IPv4 space [17]. In this way, a routing worm propagates more than three times faster than a traditional worm.

In recent years, researchers are paying great attention on how to monitor the Internet for malicious activities. Moore presented the concept of "network telescope" in monitoring Internet abnormal activities and the propagation of a worm [10]. Zou *et al.* used a similar monitoring system to do worm early detection [16].

III. Epidemic Model Introduction

Computer worms are similar to biological viruses in their self-replicating and propagation behaviors. Thus the mathematical techniques developed for

the study of biological infectious diseases can be adapted to the study of computer worm propagation. We briefly introduce two classical deterministic epidemic models: simple epidemic model in homogeneous system and in interacting groups [2], respectively. Our models and analyses in this paper are based primarily on these two models and their underlying principles.

A. Simple epidemic model in a homogeneous system

The simple epidemic model assumes that each host occupies one of two states: *susceptible* or *infectious*. The model also assumes that once a host is infected by a virus, it remains in the infectious state forever. Denote by $I(t)$ the number of infectious hosts at time t; N the number of hosts in the system; thus $N - I(t)$ is the number of susceptible hosts at time t. In a homogeneous system, each host is assumed to have equal probability to contact any other host. In the Internet context, an infected host has equal probability of contacting any other host in the Internet when the worm uniformly scans the Internet. Thus a uniform scan worm can be modelled the same way as an epidemic disease in a homogeneous system. Through analyzing the propagation of a uniform scan worm, we illustrate in the following how to derive the simple epidemic model by using *infinitesimal analysis*.

First, the spreading of an Internet worm or an epidemic disease is in fact a stochastic process. But when considering a large-scale system consisting of a large population N, which is the case for an Internet worm, we can use a *mean value analysis* based on the *law of large number*.

Suppose a uniform scan worm has a scan rate η, which is the number of scans an infected host sends out per unit time. The worm uniformly scans the IP space that has Ω IP addresses. Let us denote δ as the length of a small time interval. During a time interval of length δ, an infected host sends out an average of $\eta\delta$ scans. For a specific IP address in the scanning space Ω, every scan has a probability $1/\Omega$ to hit it. Then, on average an infected host has probability

$$\acute{p} = 1 - (1 - 1/\Omega)^{\eta\delta} \approx \eta\delta/\Omega \tag{1}$$

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

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, from time t to $t + \delta$, an infected host infects on average $[N - I(t)]\acute{p}$ vulnerable hosts. 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 is equal to $I(t) \cdot [N - I(t)]\acute{p}$. Thus at time $t + \delta$, the number of infected hosts should be:

$$I(t + \delta) = I(t) + I(t) \cdot [N - I(t)]\eta\delta/\Omega \tag{2}$$

As $\delta \to 0$, we derive the simple epidemic model:

$$\frac{dI(t)}{dt} = \beta I(t)[N - I(t)] \tag{3}$$

where β is

$$\beta = \frac{\eta}{\Omega} \tag{4}$$

β is called the *pairwise rate of infection* in epidemiology studies [2]. At $t = 0$, $I(0)$ hosts are infectious and the other $[N - I(0)]$ hosts are all susceptible.

The epidemic model (3) has analytical solution [2]:

$$I(t) = \frac{I(0)N}{I(0) + [N - I(0)]e^{-\beta N t}} \tag{5}$$

Suppose a worm takes time T to infect $I(T)$ hosts in the system ($I(T) \leq N$). From (5), the time T is

$$T = -\frac{1}{\beta N} \cdot \ln(\frac{I(0)[N - I(T)]}{I(T)[N - I(0)]}) \tag{6}$$

Zou *et al.* provide the formula [16]:

$$N = 2^{32}\alpha/\eta \tag{7}$$

where $\alpha = \beta N$. They use 2^{32} because they consider a worm that scans the entire IPv4 space. If we replace 2^{32} by Ω and put the $\alpha = \beta N$ in, their formula (7) becomes Equation (4).

B. Simple epidemic model in interacting groups

This model is an extension of (3) to a non-homogeneous system. In this model, the system consists of K groups; each group has population $N_1, N_2, \cdots, N_K$, respectively [2]. Interactions across groups are different from interactions within a group. In place of the pairwise rate of infection

TABLE I

NOTATIONS IN THIS PAPER

Notation	Definition
N	Total number of hosts under consideration
$I(t)$	Total number of infectious hosts at time t
η	Average worm scan rate
Ω	The size of a worm's scanning space
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 in infinitesimal analysis
$\acute{p}$	Probability of a worm scanning a specific address during a small time interval δ
ϵ	Time delay in worm propagation
T	The time when a worm infects $I(T)$ hosts in the system
p	Probability of a local preference scan worm to scan locally
K	Number of "/n" prefixes in the worm scanning space Ω, $\Omega = K2^{(32-n)}$
N_k	Number of vulnerable hosts in the k-th "/n" prefix, $k = 1, 2, \cdots, K$
m	Number of "/n" prefixes that contain vulnerable hosts ($m \leq K$)
$I_k(t)$	Number of infectious hosts in the k-th "/n" prefix at time t, $k = 1, 2, \cdots, K$
Ω_e, Ω_o	Size of worm scanning space in the target (other) domain(s), $\Omega = \Omega_e + \Omega_o$
N_e, N_o	Number of vulnerable hosts in the target (other) domain(s), $N = N_e + N_o$
c_1, c_2	$c_1 = \Omega_e/\Omega$, $c_2 = N_e/N$
$I_e(t), I_o(t)$	Number of infectious hosts in the target (other) domain(s) at time t, $I(t) = I_o(t) + I_e(t)$
$C(t)$	Cumulative number of infected hosts observed by the monitoring system at time t
$Z(t)$	Number of worm scans observed by the monitoring system in a unit time

β, susceptible hosts in the j-th group are subject to infection from infectious hosts in the i-th group at the rate β_{ij} per interacting pair; for $i = j$, the rate is β_{jj}, $i, j = 1, 2, \cdots, K$. Fig. 1 illustrates the model [2].

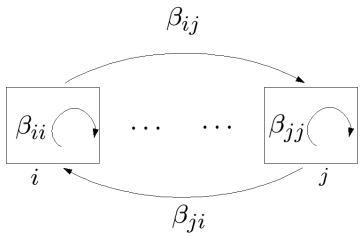


Fig. 1. Pairwise rates of infection in interacting communities $i, j = 1, 2, \cdots, K$

Denote by $I_k(t)$ the number of infectious hosts in each of the groups $k = 1, 2, \cdots, K$, respectively. Then the original model (3) for a homogeneous system can be generalized to be the set of equations

$$\frac{dI_k(t)}{dt} = \beta_{kk} I_k(t)[N_k - I_k(t)] + \sum_{i \neq k} \beta_{ik} I_i(t)[N_k - I_k(t)] \quad (8)$$

for $k = 1, 2, \cdots, K$.

IV. Modeling and Analysis of Worm Scanning Strategies

A. Idealized worm

We first model and analyze two idealized worms, which have the complete IP addresses of all vulnerable hosts in the Internet. We call them "idealized worms" because they are very difficult to be implemented by attackers on the global scale of the Internet. Before releasing an idealized worm, attackers must take great effort to build the address list of all vulnerable hosts. Some computers with special applications, such as web servers or peer-to-peer file sharing computers, advertise their IP addresses. To attack vulnerabilities on these computers, it's possible that attackers could build a list of all vulnerable hosts. However, to attack vulnerable hosts that do not advertise their IP addresses, such as SQL servers, attackers must actively conduct comprehensive scanning to find the addresses of all vulnerable hosts.

In addition to the difficulty of collecting IP addresses, attackers also have to deal with the large payload problem for their idealized worms. For example, the Code Red worm had about $N = 360,000$ vulnerable hosts in the Internet when it spread out [7] — the complete IP address list of all these

vulnerable hosts requires an idealized worm to carry a 1.37MB payload.

1) Perfect worm: We believe the "perfect worm" to be the fastest propagation worm. A perfect worm knows the addresses of all vulnerable hosts in the Internet; all infected hosts fully cooperate with each other such that they will not try to scan and infect an already infected host.

For a perfect worm, no scans are wasted — one scan causes one infection. First, we consider a perfect worm without infection delay, i.e., as soon as an infected host sends out a scan to a vulnerable host, the vulnerable host will be immediately infected and can infect others right away.

In this case, during a small time interval δ, each infected hosts in $I(t)$ sends out $\eta\delta$ scans and infects $\eta\delta$ vulnerable hosts. Since no vulnerable host will be infected twice, at time $t+\delta$, the number of infected hosts will be

$$I(t+\delta) = I(t) + I(t)\cdot\eta\delta \tag{9}$$

Take $\delta \to 0$, we derive the worm propagation model for perfect worm

$$\frac{dI(t)}{dt} = \begin{cases} \eta I(t), & I(t) < N \\ 0, & I(t) = N \end{cases} \tag{10}$$

Suppose a perfect worm begins with $I(0)$ infected hosts. Model (10) has solution:

$$I(t) = \min[I(0)e^{\eta t}, N] \tag{11}$$

To illustrate how fast a perfect worm propagates, we assume that it has the same parameters as the Code Red worm presented in [16], i.e., it has the average scan rate $\eta = 358$ per minute, vulnerable population $N = 360,000$, and initially infected hosts $I(0) = 10$. Then from (11), we know that the perfect worm will infect all vulnerable hosts by the time

$$T = \frac{\ln N - \ln I(0)}{\eta} = 1.758 \text{ seconds}$$

Thus a perfect worm can infect all vulnerable hosts within a couple of seconds. However, in the scenario above, we have not considered various time delays in the worm's propagation: one is the delay for a worm to transfer the worm code to a vulnerable host; another is the delay between when a worm arrived a vulnerable host and the host becomes infectious to others. Since the worm spreads very quickly, these delays cannot be neglected.

Now we analyze a perfect worm with consideration of time delay. Suppose a perfect worm has a delay ϵ, which is the length of time between the time when a worm scan is sent out and the time when the vulnerable host infected by it begins to infect others. In this case, at time $t+\delta$, the number of infectious hosts will be

$$I(t+\delta) = I(t) + I(t-\epsilon)\cdot\eta\delta \tag{12}$$

Take $\delta \to 0$ and we have the worm propagation model

$$\frac{dI(t)}{dt} = \begin{cases} \eta I(t-\epsilon), & I(t) < N \\ 0, & I(t) = N \end{cases} \tag{13}$$

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

We cannot derive an analytical solution for (13), thus we use Matlab Simulink [18] to derive its numerical solution. If we assume the time delay is $\epsilon = 2$ seconds and the worm still has the Code Red worm parameters in previous example, then the worm propagation is shown in Fig. 2, compared with the worm propagation when no time delay is considered.

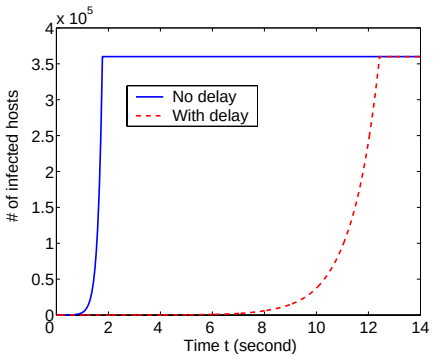


Fig. 2. The propagation of a perfect worm with and without time delay ($N = 360,000$, $\eta = 358$/min, $I(0) = 10$; delay is $\epsilon = 2$ seconds)

2) Flash worm: Staniford *et al.* introduce the "flash worm" [9], which knows the IP addresses of all vulnerable hosts in the Internet and uniformly scans the vulnerable population. The propagation of a flash worm satisfies the epidemic spreading assumptions in a homogeneous system and can be modelled by the simple epidemic model (3). According to (3) and (4), the worm propagation

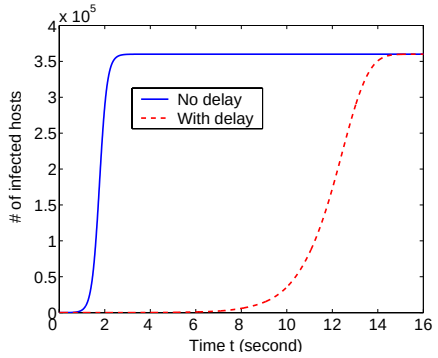


Fig. 3. The propagation of a flash worm with and without time delay ($N = 360,000$, $\eta = 358$/min, $I(0) = 10$; delay is $\epsilon = 2$ seconds)

model for a flash worm is:

$$\frac{dI(t)}{dt} = \frac{\eta}{N} I(t)[N - I(t)] \tag{14}$$

A flash worm has a scanning space of size $\Omega = N$, which is much smaller than the entire IPv4 space scanned by the Code Red worm. Therefore, a flash worm propagates much faster than an ordinary worm that scans the entire IPv4 space.

In the model (14), we have not considered the time delay in the worm's propagation. Without considering delay, from (6) and (14), we know that a flash worm can infect 99% of vulnerable hosts by the time $T = 2.53$ seconds (with the same parameters as the Code Red worm, $N = 360,000$, $\eta = 358$/min, $I(0) = 10$ [16]). For such a fast spreading worm, time delay is significant and should be considered in the worm's propagation model.

Suppose the delay time is ϵ. The worm propagation model for a flash worm when considering time delay becomes

$$\frac{dI(t)}{dt} = \frac{\eta}{N} I(t - \epsilon)[N - I(t)] \tag{15}$$

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

When we assume that the delay is $\epsilon = 2$ seconds, Fig. 3 shows the numerical solution of model (15). The worm infects 99% of the vulnerable population by the time $T = 14.3$ seconds. For comparison, we also show in this figure the worm's propagation when no time delay is considered (described by (14)).

Fig. 2 and 3 show that a flash worm propagates only slightly slower than a perfect worm. These two worms take much longer time to infect the first 10% of vulnerable population than the time to infect the next 80% of vulnerable population. During the time period in infecting the first 10% of vulnerable population, less than 10% of scans waste on already infected hosts in the propagation of a flash worm. Therefore, compared with a flash worm, a perfect worm only slightly increases its propagation speed through cooperation.

B. Uniform scan worm

In this section, we model and analyze several uniform scan worms: the Code Red worm, the "hit-list" worm, the "routing" worm, and the "divide-and-conquer" scan worm.

1) Code Red worm: a worm that scans the entire IPv4 space: When a worm has no knowledge of where vulnerable hosts reside in the Internet, the simplest strategy is to randomly scan the entire IP address space to find targets, which is what the Code Red worm and the Slammer worm did [8][9]. For such a worm, the scanning space is the entire IPv4 address space, i.e., $\Omega = 2^{32}$. Thus from (3) and (4), the worm's propagation follows

$$\frac{dI(t)}{dt} = \frac{\eta}{2^{32}} I(t)[N - I(t)] \tag{16}$$

Comparing (16) with previous (14), the scanning space of a flash worm is much smaller than a uniform scan worm that scans the entire IPv4 space. Thus a flash worm propagates much faster. For this reason, here we do not need to consider time delay in worm propagation since the small delay time is negligible comparing with the worm's spreading speed.

2) Hit-list worm: Staniford *et al.* [9] introduce the "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. During the hit-list scanning phase, a hit-list worm propagates in the same way as a "flash" worm on the list of vulnerable hosts — it can be modelled by (14) after replacing the scanning space N to the size of its hit-list. Therefore, a hit-list worm can infect all vulnerable hosts on its hit-list within several seconds. When a hit-list worm begins to scan the entire Internet after the hit-list scanning phase, it propagates like the Code Red worm and can be modelled by (16). The only difference is that

now the hit-list worm has a large number of initially infected hosts $I(0)$, which is equal to the size of the worm's hit-list.

3) Routing worm: Zou *et al.* [17] introduce the "routing worm", which uses BGP routing prefixes to reduce the worm's scanning space Ω. Based on BGP routing table, they find that currently only about 28.6% of IPv4 addresses are routable [17]. Thus if a worm uses BGP prefixes information, which they call a "BGP routing worm", the worm reduces its scanning space by more than three times. Transforming an ordinary uniform scan worm to a routing worm only changes a worm's scanning space, not its scanning strategy. Therefore, a routing worm can still be modelled by the simple epidemic model (3). In this way, Equation (4) shows that a BGP routing worm could increase its pairwise rate of infection β by $1/0.286 = 3.5$ times.

Suppose when attackers change the original Code Red worm ($N = 360,000$, $\eta = 358/\text{min}$, $I(0) = 10$) into a BGP routing worm and also a hit-list worm, the worm's scan rate does not change, i.e., $\eta = 358/\text{min}$. The BGP routing worm has $I(0) = 10$ as the original Code Red worm; while the hit-list worm has $I(0) = 10,000$ when it has a $10,000$ hit-list. Fig. 4 shows the propagation of the hit-list worm, the BGP routing worm, and the original Code Red worm. We observe that a hit-list worm can infect a large number of vulnerable hosts in a short time because of its hit-list, but it has a slower spreading speed than a BGP routing worm.

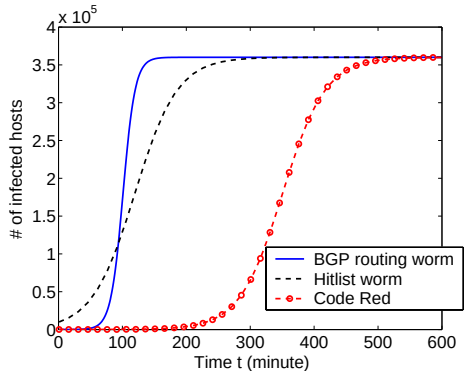


Fig. 4. Worm propagation comparison between the Code Red worm, a BGP routing worm, and a hit-list worm with a $10,000$ hit-list (note that previous "idealized" worms propagate in the time scale of *seconds*, while the three worms here propagate in the time scale of hundreds of *minutes*)

4) 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. We call such a worm a "divide-and-conquer scan worm". In the propagation of such a worm, no two infected hosts will waste their infection power on a same target.

Assume that when a divide-and-conquer scan 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. We make the following assumptions: vulnerable hosts are uniformly distributed in the entire scanning space Ω; no infected host will be removed; each infected host uniformly scans IP addresses in its scanning space; during scanning, an infected host independently chooses an IP address to scan, which means that it may scan the same IP address in its scanning space more than once; initially there is only one infected host in the system.

When a host is infected and begins to scan and infect others, it is the only infected host in its scanning space. At time t, there are $I(t)$ infected hosts in the system; then on average each infected host will be responsible for a scanning space of size $\Omega/I(t) - 1$ (the host will not scan itself), which contains $N/I(t) - 1$ vulnerable hosts. During a small time interval δ, an infected host sends out on average $\eta\delta$ scans. According to (1), the probability that an infected host scans a specific IP address during δ is

$$\acute{p} = \frac{\eta\delta}{\Omega/I(t) - 1} \tag{17}$$

Using the same analysis procedure as in deriving (2), we derive the number of infected hosts at time $t + \delta$:

$$I(t+\delta) = I(t) + I(t) \cdot [\frac{N}{I(t)} - 1] \cdot \acute{p} \tag{18}$$

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

$$\frac{dI(t)}{dt} = \frac{\eta}{\Omega - I(t)} \cdot I(t)[N - I(t)] \tag{19}$$

For Internet worms, the number of vulnerable hosts N is much smaller than Ω. Therefore, $\Omega - I(t) \simeq \Omega$ and (19) becomes

$$\frac{dI(t)}{dt} = \frac{\eta}{\Omega} \cdot I(t)[N - I(t)] = \beta I(t)[N - I(t)] \tag{20}$$

which is exactly the simple epidemic model (3). Summarizing the above arguments and we have

Proposition 1: When vulnerable hosts are uniformly distributed, a "divide-and-conquer" scan worm propagates in the same way as a uniform scan worm and can be modelled by the simple epidemic model (3).

C. Local preference scan worm

Uniform scan is the simplest scanning strategy for a worm to use. However, it is not optimal. This is because the vulnerable hosts in the Internet are not uniformly distributed (at least we know that Internet IP space is not uniformly allocated [19][20]). A worm could increase its spreading speed when it scans more intensively in the IP space where vulnerable hosts are more densely distributed.

Attackers have implemented "local preference scan" in their worms, such as Code Red II [7]: such a worm has a higher probability to scan an IP address within the same Class B or the same Class A subnetwork than a random IP address. Besides the reason mentioned above, another reason for attackers to use local preference scan is because of the presence of firewalls: worm scans may have difficulty reaching a network behind a firewall; however, if one computer inside the firewall is infected, a local preference scan enables it to quickly infect all vulnerable hosts in that local network.

"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 the following, we model and analyze a local preference scan worm that has probability p to uniformly scan IP addresses in its own "/n" prefix subnetwork and probability $(1-p)$ to uniformly scan other IP addresses. A "/n" prefix subnetwork is a network containing all IP addresses that have the same first n bits; thus in the current IPv4 Internet, a "/n" prefix subnetwork contains 2^{32-n} IP addresses.

The analysis in this section can be easily extended to other kinds of local preference scan strategies, such as local preference scanning with several levels of locality (e.g., Code Red II has two-level locality in its local preference scan [7]).

Assume that the worm scanning space Ω consists of K "/n" prefix subnetworks ($\Omega = K2^{32-n}$); each subnetwork has N_k vulnerable hosts, $k = 1, 2, \cdots, K$. Denote by $I_k(t)$ the number of infected hosts in the k-th subnetwork at the time t, β' as the pairwise rate of infection in local scan, and β'' as the pairwise rate of infection in remote scan. Then according to (4), we have

$$\beta' = \frac{p\eta}{2^{32-n}}, \qquad \beta'' = \frac{(1-p)\eta}{(K-1)2^{32-n}} \tag{21}$$

The propagation of a local preference scan worm can be modelled by the epidemic model in interacting groups (8):

$$\frac{dI_k(t)}{dt} = [\beta' I_k(t) + \sum_{j \neq k} \beta'' I_j(t)] \cdot [N_k - I_k(t)] \tag{22}$$

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

1) Local preference scan with identical subnetworks: In its general form, the local preference worm model (22) has no analytical solution. We first analyze the simple case where vulnerable hosts are uniformly distributed and there are the same number of infected hosts in each subnetwork initially, i.e., $I_k(0) = I_0(0)$ and $N_k = N/K$, $k = 1, 2, \cdots, K$. In this case, from (22) we know that the worm propagation in each subnetwork is identical, i.e., $I_k(t) = I_1(t)$, $k = 2, \cdots, K$:

$$\begin{aligned} \frac{dI_k(t)}{dt} &= [\beta' + (K-1)\beta''] \cdot I_k(t)[N_k - I_k(t)] \\ &= \frac{\eta}{2^{32-n}} I_k(t)[N_k - I_k(t)] \end{aligned} \tag{23}$$

Equation (23) shows that the worm's propagation is not affected by the probability p in local preference scan. From each subnetwork's point of view, the worm's propagation on a subnetwork is equivalent to the case where infected hosts only scan their own subnetworks.

From the point of view of the Internet, $I(t) = \sum I_k(t)$, $N = \sum N_k$, $k = 1, 2, \cdots, K$; the entire scanning space is $\Omega = K2^{32-n}$. Thus the worm propagation in the Internet is:

$$\frac{dI(t)}{dt} = K\frac{dI_1(t)}{dt} = \frac{\eta}{\Omega} I(t)[N - I(t)] \tag{24}$$

Comparing (24) with (3) and (4), we see that if the vulnerable hosts are uniformly distributed in the entire scanning space Ω, then a local preference scan worm propagates in the same manner as a uniform scan worm in terms of the total number of infected hosts $I(t)$. Summarize the analysis above and we have

Proposition 2: When vulnerable hosts are uniformly distributed in a worm's scanning space, local preference scan does not help a worm in its propagation speed.

2) Local preference scan with non-uniformly distributed population: Currently, computers are not uniformly distributed within the IPv4 address space. Among all 256 Class A networks in the IPv4 Internet, only 131 Class A IP addresses are allocated by the Internet Assigned Numbers Authority (IANA) [17][20]. This means that there are no computers in the other 125 Class A subnetworks.

Without loss of generality, suppose in the K "/n" prefix subnetworks, 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 [17] to remove those empty subnetworks from the worm's scanning space). Suppose $I_k(0) = I_1(0) > 0, k = 2, 3, \cdots, m$. From (22), the worm propagation on each subnetwork follows

$$\frac{dI_k(t)}{dt} = [\beta' + (m-1)\beta''] \cdot I_k(t)[N_k - I_k(t)] \quad (25)$$

for $k = 1, \cdots, m$.

From the point of view of the Internet, the worm propagation follows

$$\frac{dI(t)}{dt} = m\frac{dI_1(t)}{dt} = \frac{\beta' + (m-1)\beta''}{m} I(t)[N - I(t)] \quad (26)$$

If a local preference scan worm wants to propagate as fast as possible, the worm should select the preference probability p to maximize the pairwise rate of infection $[\beta' + (m-1)\beta'']/m$ in (26). Hence, the optimal preference probability should be $p = 1$. Such a conclusion seems unexpected; but it is reasonable for the assumptions we have used — all those m "/n" subnetworks are assumed to be identical. If $p = 1$, which means a worm only scans its own subnetwork, then no worm scans will be wasted in those $(K-m)$ empty subnetworks. In this way, the worm achieves its fastest spreading speed.

In reality, no subnetwork is exactly the same as the others. A worm cannot just scan locally; remote scan is necessary for the worm to spread out to every part of the entire Internet. Thus if we assume that at the beginning, $I(0) = I_1(0) > 0$ and $I_k(0) = 0, k = 2, 3, \cdots, m$, then a local preference scan worm requires $p < 1$ in order to spread out into other subnetworks — if $p = 1$, the subnetworks that do not have infected hosts initially will never be infected. For this scenario, the worm's propagation in the subnetworks $k = 2, \cdots, m$ are identical, i.e., $I_k(t) \equiv I_2(t), k = 3, \cdots, m$. Hence, the worm propagation on each subnetwork is described by:

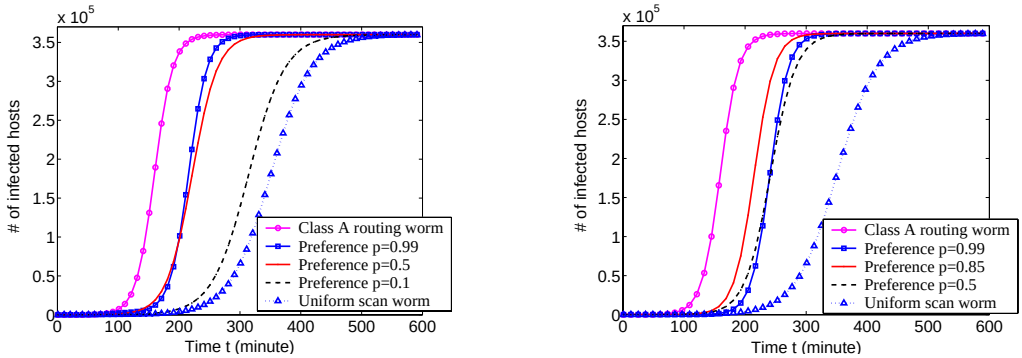
$$\begin{aligned} \frac{dI_1(t)}{dt} &= [\beta' I_1(t) + (m-1)\beta'' I_2(t)][\tfrac{N}{m} - I_1(t)] \\ \frac{dI_k(t)}{dt} &= [\beta'' I_1(t) + (\beta' + m\beta'' - 2\beta'')I_k(t)][\tfrac{N}{m} - I_k(t)] \end{aligned} \quad (27)$$

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

The worm propagation model (27) does not have analytical solution. Thus we use *Matlab Simulink* [18] to solve it for different preference probabilities p. We use the previous Code Red worm parameters in the study here, i.e., $N = 360,000$, $\eta = 358$/min, $I(0) = I_1(0) = 10$. The Code Red II worm used local preference on both Class A networks and on Class B networks [23]. In order to see how the size of the subnetworks in local preference scan affects a worm's propagation, we study two scenarios: in the first scenario each "/n" prefix subnetwork is an Class A network ("/8" prefix); in the second scenario, each "/n" prefix subnetwork is an Class B network ("/16" prefix).

For the first scenario where each subnetwork is a Class A network, $K = 2^8 = 256$. [17] points out that currently only 116 Class A networks are routable according to BGP tables, thus $m = 116$. Based on these parameters, Fig. 5(a) shows $I(t)$ for different preference probabilities p. For comparison, we also show the original Code Red propagation on this figure where the worm uniformly scans the entire IPv4 space. If attackers know that $m = 116$ Class A networks have vulnerable hosts and know their Class A prefixes as mentioned in [17], attackers can implement the "Class A routing worm" presented in [17] to uniformly scan these 116 Class A IP space only. The propagation of such a Class A routing worm is also shown in Fig. 5(a).

For the second scenario where each subnetwork is a Class B network, $K = 2^{16} = 65536$. Since 116 Class A networks have been allocated and each Class A network contains 2^8 Class B networks, thus $m = 116 \cdot 2^8 = 29696$. Based on these parameters,



a. Local preference scan on Class A network level ($K = 256, m = 116$)

b. Local preference scan on Class B network level ($K = 65536, m = 29696$)

Fig. 5. Comparison of a Class A routing worm, a local preference worm, and the original Code Red worm

Fig. 5(b) shows $I(t)$ for different preference probabilities p.

From Fig. 5(a)(b), we observe that when vulnerable hosts are not uniformly distributed, local preference scan increases a worm's propagation speed. If the local preference scan is on Class A network level, Fig. 5(a) shows that the optimal local preference probability should be chosen close to one. On the other hand, if the local preference scan is on Class B network level, Fig. 5(b) shows that the optimal local preference probability should approximately equal to $p = 0.85$: the worm propagates faster when p increases until it reaches $p = 0.85$; after that, increasing p makes the worm propagate slower again.

From Fig. 5, we conclude that the optimal probability p of local preference scan is determined by the locality in the local preference scan. We explain it in the following intuitive way: when only one subnetwork has infected hosts initially, the purpose of remote scan is to spread out worm seeds to every one of the other $(m - 1)$ subnetworks. If a worm uses Class A network in its local preference scan, it needs to spread out worm seeds to at most 256 subnetworks. On the other hand, if the worm uses Class B network in its local scan, it needs to spread out worm seeds to everyone in those $m = 29696$ subnetworks in the previous example. Therefore, a Class B local scan worm needs to take much more effort to spread out worm seeds than a Class A local scan worm.

Summarizing the above analysis and we have

Proposition 3: When vulnerable hosts are not uniformly distributed in a worm's scanning space, comparing with uniform scan, local preference scan increases a worm's propagation speed. The optimal local preference scan probability p increases when the local scan is on larger subnetworks.

The author of Code Red II used the following local scan probabilities [23]: the worm scans the local class A network with $p = 0.5$ and the local Class B network with $p = 0.375$. From the analysis above, it can be seen that the local scan probabilities in Code Red II are much lower than the optimal ones.

Fig. 5 shows that if attackers know the distribution of vulnerable hosts, the routing worm will be the fastest spreading worm by simply removing those empty IP space. The original routing worm presented in [17] uses uniform scan within the IP space defined by BGP prefixes. From the analysis above, we see that in a routing worm, attackers could also implement local preference scan to further increase worm propagation speed.

It should be noted that in our analysis, we have not considered the impact of possible network congestion. If a worm extensively uses local preference scan, the intense worm traffic might cause congestion to local networks and slow down the worm's spreading speed. In this case, the optimal local preference scan probability in such a worm may not be the optimal value in our analysis.

D. Sequential scan worm

Until now we have assumed that worms choose IP addresses randomly. Another scanning strategy is to choose IP addresses sequentially. "Sequential scan" means that a worm scans IP addresses sequentially:

after checking IP address x, the worm continues to check IP address $(x+1)$, or $(x-1)$ if the search direction is reversed. For a sequential scan worm, once a vulnerable host is infected, it first selects a starting IP address to begin with its sequential scan. The Blaster worm is a typical sequential scan worm [26]. Without loss of generality, we assume that a sequential scan worm scans IP addresses additively.

In our previous analysis, we found that local preference scan increases the spread of a random scan worm. In a sequential scan worm, "local preference" has a different meaning: in selecting the starting point for its sequential scan, a worm chooses an IP address close to its own address with higher probability than an IP address far away. For example, for its starting point, the Blaster worm chooses the first address of its Class C subnetwork with a probability 0.4, and chooses a random IP address with a probability 0.6 [26].

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

Proposition 4: 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, in the following we mainly model and analyze a sequential scan worm with a uniformly chosen starting point, which is called a "random sequential scan worm".

First, we analyze the propagation of a sequential scan worm when vulnerable hosts are uniformly distributed. We use the same analysis principles in deriving the simple epidemic model in Section III. Suppose a sequential scan worm has scan rate η, vulnerable population N, and scanning space of size Ω. At time t, $I(t)$ hosts are infected. During the next small time interval δ, one infected host sequentially scans $\eta\delta$ IP addresses. At time t, the density of vulnerable hosts on the entire scanning space is $[N-I(t)]/\Omega$. Then on average, one infected host can infect $\eta\delta[N-I(t)]/\Omega$ vulnerable hosts during the time interval δ. When δ is sufficiently small, the probability of two infected hosts infecting the same vulnerable target during the time interval δ is negligible. Therefore, the number of newly infected hosts during the time interval δ is equal to $I(t)\cdot\eta\delta[N-I(t)]/\Omega$. From (4), we have

$$I(t+\delta) = I(t) + I(t)\cdot\beta[N-I(t)] \qquad (28)$$

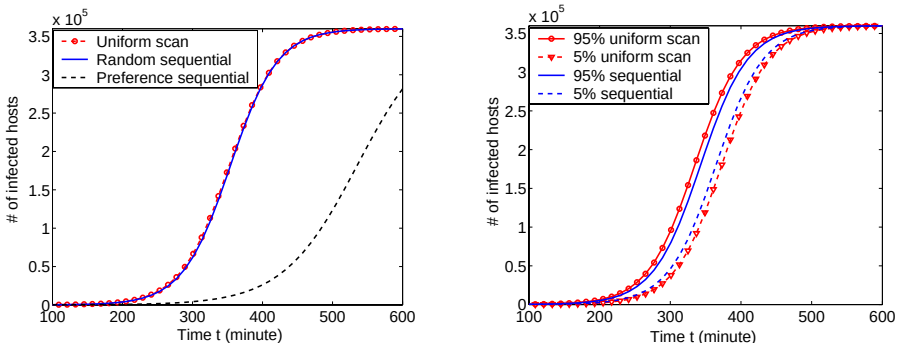
Taking $\delta \to 0$, we can derive the sequential scan worm propagation model — it is identical to the uniform scan worm model (3). Summarizing the analysis above and we have

Proposition 5: If vulnerable hosts are uniformly distributed in a worm's scanning space, a random sequential scan worm has the same propagation speed as a uniform scan worm and can be modelled by the epidemic model (3).

If vulnerable hosts are not uniformly distributed, or a sequential scan worm does not start from a randomly selected IP address, then the accuracy of our analysis and the accuracy of the epidemic model (3) rely on the *law of large number*: some worm copies infect vulnerable hosts more slowly while others infect vulnerable hosts more quickly — such an uneven behavior will average out each other. If the random effect is too severe, such as when all vulnerable hosts are sequentially within one block of IP space, then the epidemic model is a poor model for a sequential scan worm.

In order to verify our analysis, we simulate a "uniform scan worm" (such as the Code Red worm); a "preference sequential scan worm", which chooses the starting point from a local IP address with probability 0.4 (such as the Blaster worm); and a "random sequential scan worm" that uniformly chooses starting point for its sequential scan. For comparison, we use the same Code Red worm parameters, $N = 360,000, I(0) = 10, \eta = 358$/min, for all these three worms.

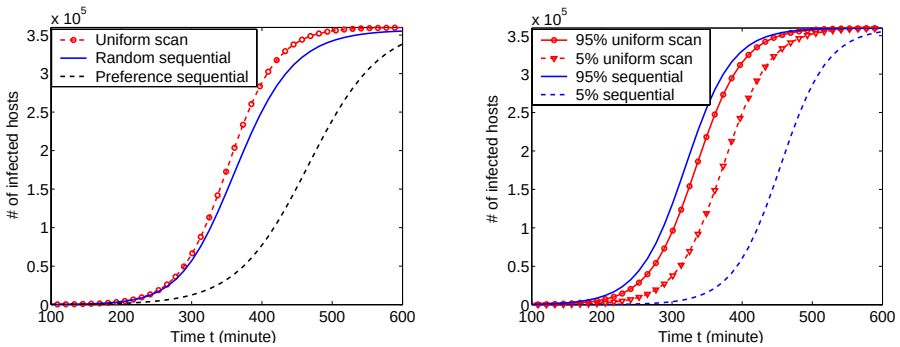
When we assume that vulnerable hosts are uniformly distributed in the entire IPv4 space, Fig. 6 shows the simulation results. For each of those three worms, we run the worm propagation simulation 100 times. Fig. 6(a) shows the mean value $I(t)$ in each worm's propagation; Fig. 6(b) shows the variabilities of worm propagation for the uniform



a. Mean value of worm propagation

b. Variabilities in a uniform scan worm and a random sequential scan worm

Fig. 6. Comparison of a random sequential scan worm, a sequential scan worm with 40% local preference, and a uniform scan worm (vulnerable hosts uniformly distributed in the entire IPv4 space; 100 simulation runs)



a. Mean value of worm propagation

b. Variabilities in a uniform scan worm and a random sequential scan worm

Fig. 7. Comparison of a random sequential scan worm, a sequential scan worm with 40% local preference, and a uniform scan worm (vulnerable hosts uniformly distributed in the address space of BGP prefixes; 100 simulation runs)

scan worm and the random sequential scan worm. The "95%" propagation curve means that a worm propagates no faster than this curve in 95 out of those 100 simulation runs. Therefore, in 90 out of 100 simulation runs, a worm propagates within those two "5%" and "95%" propagation curves.

Fig. 6 agrees with Proposition 5: a sequential scan worm propagates at the same speed as a uniform scan worm when vulnerable hosts are uniformly distributed. In addition, Fig. 6(a) agrees with Proposition 4: For a sequential scan worm, using local preference in selecting the starting point slows down the worm's propagation speed.

In reality, the vulnerable hosts in the Internet are not uniformly distributed. BGP routing tables show that currently about 28.6% of IPv4 addresses are routable [17] — all vulnerable hosts must distribute within the IP space defined by BGP routing prefixes. Since we do not know the true distribution of vulnerable hosts in the Internet, a reasonable approach is to assume that all vulnerable hosts are uniformly distributed in the IP space defined by BGP routing prefixes. For such a simulation scenario, we simulate each of those three worms 100 times again and show the simulation results in Fig. 7, which has the same format as Fig. 6. Fig. 7(a) shows that when vulnerable hosts are not uniformly distributed within the Internet address space, on average a random sequential scan worm propagates slightly slower than a uniform scan worm; and a preference sequential scan worm clearly propagates slower than the other two. Fig. 7(b) shows that the propagation of a sequential scan worm varies considerably because of the non-uniform distribution of vulnerable hosts.

E. Selective attack worm

Routing worms can conduct selective attack [17] based on geographic information of IP addresses.

In a selective attack, attackers only care about how fast their worms propagate in the target domain, not how many vulnerable hosts have been infected in the global Internet. In this section, we model and analyze how a selective attack worm propagates under different scanning strategies.

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 . $N = N_e + N_o$, $\Omega = \Omega_e + \Omega_o$. The worm has scan rate η.

1) Target-only: "Target-only" means that a worm only scans and infects hosts in the target domain. In this case, if the worm uniformly scans the target domain, the worm's propagation follows epidemic model (3) with the pairwise rate of infection η/Ω_e according to (4):

$$\frac{dI_e(t)}{dt} = \frac{\eta}{\Omega_e} I_e(t)[N_e - I_e(t)] \qquad (29)$$

On the other hand, a worm can uniformly scans the entire scanning space. We call such a worm as a "global scan" worm. The question is: which worm propagates faster on 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 $c_1 < c_2$, vulnerable hosts are more densely distributed in the target domain than in other domains. For the global scan worm, $I_e(t) = c_2 I(t)$ because of its uniform scan strategy. For this global scan worm, the number of infected hosts in the target domain follows

$$\frac{dI_e(t)}{dt} = \frac{\eta}{c_2\Omega} I_e(t)[N_e - I_e(t)] \qquad (30)$$

Comparing (29) with (30), we observe that, if $c_2 > c_1$, i.e., the density of vulnerable hosts in the target domain is greater than the density in other domains, the target-only worm propagates faster than the global scan worm in the target domain (and vice versa). Thus we have

Proposition 6: For a selective attack worm, if the density of vulnerable hosts in the target domain is higher than in other domains, the worm propagates faster in the target domain if it scans within the target domain than uniformly scans all domains.

2) Target-global: "Target-global" means that an infected host in the target domain only scans within the target domain, and an infected host in other domains uniformly scans the entire scanning space. The worm's propagation model is:

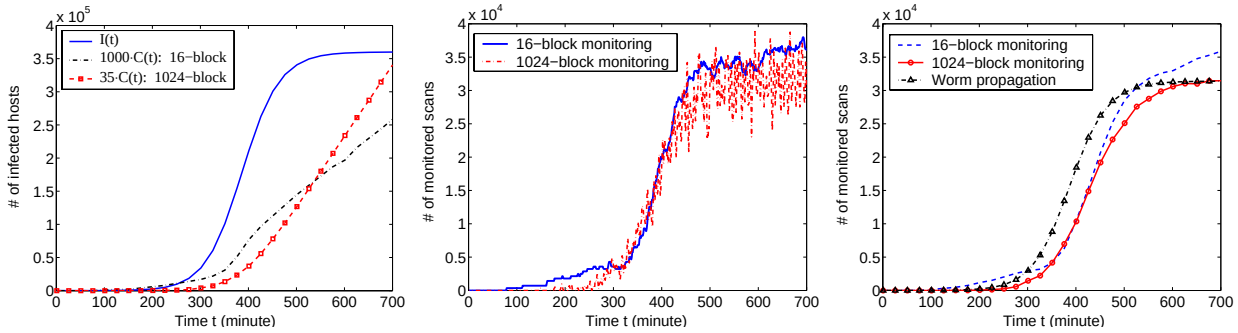
$$\begin{aligned} \frac{dI_e(t)}{dt} &= [\tfrac{\eta}{\Omega_e} I_e(t) + \tfrac{\eta}{\Omega} I_o(t)][N_e - I_e(t)] \qquad (31) \\ \frac{dI_o(t)}{dt} &= \tfrac{\eta}{\Omega} I_o(t)[N_o - I_o(t)] \end{aligned}$$

Comparing (29) with (31), we observe that a target-global worm always propagates faster than a target-only worm in the target domain ($I_e(t)$), no matter how densely the vulnerable hosts are distributed in the target domain. It's easy to explain: compared to a target-only worm, a target-global worm has some extra infected hosts in other domains that could help in infecting more vulnerable hosts in the target domain.

V. Worm Monitoring System Design

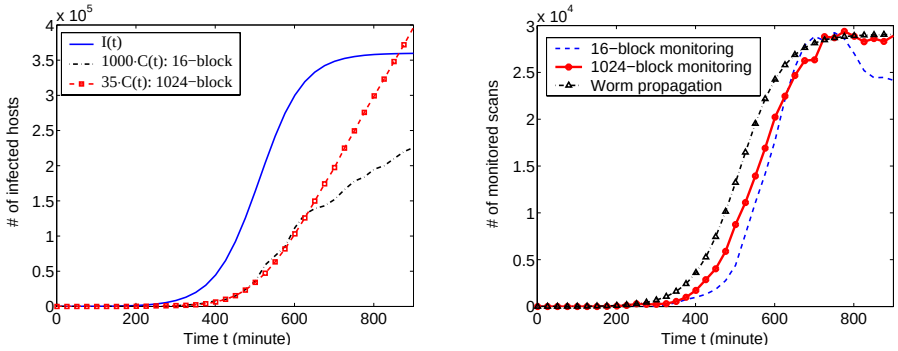
To defend against worm attacks, we first need to set up a worm monitoring infrastructure to monitor and detect the presence of a worm in the Internet. CAIDA has already set up a relatively large-scale network monitoring system by using the "network telescope" concept [10], which is based on several large chunks of IP space. In this section, we show that although such a monitoring system is good at monitoring a uniform scan worm such as Code Red and Slammer, it performs poorly when monitoring a non-uniform scan worm, especially a sequential scan worm such as Blaster.

A worm monitoring system primarily observes two data sets: the number of scans observed in each monitoring time interval, denoted as $Z(t)$; and the cumulative number of infected hosts observed until time t, denoted as $C(t)$. Zou *et al.* [16] show that for uniform scan worms, the worm propagation pattern on the global Internet can be accurately inferred by using either one of these two data sets. However, when a worm does not uniformly scan IP addresses in the Internet, the observed worm traffic does not represent the worm propagation. For example, if a worm uses local preference scan, then a monitor will receive a large number of scans when some hosts close to the monitor are infected. On the other hand, when the same number of infected hosts are far away from the monitor, the monitor can only observe a smaller number of scans. Therefore, monitoring different IP address space will provide



a. Worm propagation and monitored data $C(t)$ b. Original monitored data $Z(t)$ c. Monitored data $Z(t)$ after using a low-pass filter

Fig. 8. Blaster propagation and its monitoring (vulnerable hosts are uniformly distributed in the entire IPv4 space; this is the results for one simulation run)



a. Worm propagation and monitored data $C(t)$ b. Monitored data $Z(t)$ after using a low-pass filter

Fig. 9. Blaster propagation and its monitoring (vulnerable hosts are uniformly distributed in the BGP prefix space, which is less than 30% of the entire IPv4 space; this is the results for one simulation run)

different observation patterns of the same worm's propagation.

Such a monitoring problem becomes even worse when monitoring a sequential scan worm. If the monitoring system only has a few big chunks of IP space (such as several Class A or Class B networks), then, during the worm propagation time period, we can only observe a very small fraction of infected hosts on the global Internet. For example, if the Slammer worm uses sequential scan and its scan rate is $\eta = 4000$ scans per second [8], then an infected host requires $2^{32}/\eta = 12.43$ *days* to scan the entire IPv4 space. Therefore, if the sequential scan worm starts from an IP address far from the monitored IP space, the monitoring system will not observe it for some time. On the other hand, if the worm uniformly scans the Internet and the monitoring system covers two Class B networks (2^{17} IP addresses), then on average we could observe an infected host $2^{(32-17)}/\eta = 8.2$ seconds after it is infected.

In previous simulations of a preference sequential scan worm shown in Fig. 6, we also simulated the monitoring system. As mentioned above, the monitoring system should monitor many distributed chunks of IP addresses. Here we consider two monitoring systems: one monitors 16 blocks of Class B IP space; another monitors 1024 equal-size blocks of IP space. Both monitoring systems monitor the same 2^{20} IP addresses and all monitored address blocks are evenly distributed in the entire IPv4 space. Fig. 8(a) shows the number of infected hosts $I(t)$ in the Internet as a function of time t for one simulation run. It also shows the cumulative number of observed infected hosts, $C(t)$, by both monitoring systems. Because $C(t)$ is very small, in order to show $C(t)$ and $I(t)$ on the same figure, we multiply $C(t)$ by 1000 for the 16-block monitoring system and by 35 for the 1024-block monitoring system. This figure shows that if we use a 16-block

monitoring system, we observe less than 0.1% of infected hosts in the Internet during the worm's propagation time period.

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, infected hosts will enter or leave the monitored IP blocks of a monitoring system — it happens more frequently in the 1024-block monitoring system than in the 16-block monitoring system.

Although it is noisier than the 16-block monitoring system, the observation data from the 1024-block monitoring system represents a sequential scan worm's propagation more accurately. 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(a). Such 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[1] to filter out high frequency noises from $Z(t)$ without changing the worm's propagation pattern. Fig. 8(c) shows the observation data $Z(t)$ after being passed through a first-order low-pass filter. To check if the observation $Z(t)$ can represent the worm's propagation on the entire Internet, we draw the curve of $I(t)$ on this figure too by changing its scale to have the similar value as $Z(t)$. Fig. 8(c) shows that the observation data $Z(t)$ of the 1024-block monitoring system has delay to $I(t)$ but represents well the worm's propagation pattern on the entire Internet.

A more realistic simulation of Blaster worm is the "preference sequential scan worm" shown in Fig. 7, where the vulnerable hosts are assumed to be uniformly distributed in the IP space defined by BGP prefixes. Fig. 9 shows the results of the Blaster worm in one simulation. Fig. 9(a)(b) have the same format and meanings as Fig.8(a)(c), respectively. Because now the vulnerable hosts are not uniformly distributed in the Internet, the observation data is noisier than the data in previous simulation shown in Fig. 8.

[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 $a = 0.02$ in Fig. 8(c) and 9(b).

Worm propagation in other simulation runs give similar results to what shown in Fig. 8 and 9. 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 and we have

Proposition 7: In order to monitor the propagation of a non-uniform scan worm in the Internet, especially the propagation of a sequential scan worm, a worm monitoring system must monitor many well distributed IP blocks.

VI. Conclusion

In this paper, we model and analyze worm propagation when a worm uses different scanning strategies, including idealized scan, hit-list scan, uniform scan, divide-and-conquer scan, local preference scan, target scan, and sequential scan, etc. A better understanding of how various scanning strategies affect a worm's propagation can lead to better defense against future worms.

We show that the local preference scan increases a worm's propagation speed when vulnerable hosts are not uniformly distributed; and the optimal local preference scan probability is determined by the size of the network in local scanning. When vulnerable hosts are uniformly distributed, we prove that the divide-and-conquer scan, the sequential scan, and the uniform scan are equivalent in terms of the total number of infected hosts at any time. For a sequential scan worm, using local preference in selecting the starting point slows down the worm's propagation speed. When conducting "selective attack" [17], a worm propagates faster on the target domain if it propagates both on the target domain and on other domains. The fastest spreading worm is the one that has the complete IP addresses of all vulnerable hosts in the Internet: it can finish its infection task in a matter of seconds regardless whether infected hosts cooperate with each other (perfect worm) or not (flash worm). Fortunately, it is very hard for attackers to construct a complete hit-list of all vulnerable hosts on the global-scale Internet.

A non-uniform scan worm, especially a sequential scan worm, will cause some troubles to a worm

monitoring system as shown in Fig. 8 and Fig. 9. From our analysis and simulation studies, we point out that a worm monitoring system must cover many well distributed IP blocks in order to monitor accurately different kinds of worms. In addition, to infer worm propagation pattern in the Internet from the monitored data, it is suitable to first use a low-pass filter on the monitored data to remove high frequency noises.

VII. Acknowledgement

This work is supported in part by ARO contract DAAD19-01-1-061, the National Science Foundation Grants ANI9980552, and by DARPA contract F30602-00-0554.

References

[1] Z. Chen, L. Gao, and K. Kwiat. Modeling the Spread of Active Worms. *IEEE INFOCOM*, 2003.

[2] D.J. Daley and J. Gani. *Epidemic Modelling: An Introduction*. Cambridge University Press, 1999.

[3] J. O. Kephart and S. R. White. Directed-graph Epidemiological Models of Computer Viruses. In *IEEE Symposimum on Security and Privacy*, 1991.

[4] J. O. Kephart, D. M. Chess, and S. R. White. Computers and Epidemiology. *IEEE Spectrum*, 1993.

[5] J. O. Kephart and S. R. White. Measuring and Modeling Computer Virus Prevalence. In *IEEE Symposimum on Security and Privacy*, 1993.

[6] D. Seeley. A tour of the worm. In *Proceedings of the Winter Usenix Conference*, San Diego, CA, 1989.

[7] D. Moore, C. Shannon, and J. Brown. Code-Red: a case study on the spread and victims of an Internet Worm. In *Proc. ACM/USENIX Internet Measurement Workshop*, France, November, 2002.

[8] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the Slammer Worm. *IEEE Magazine on Security and Privacy*, 1(4):33-39, July 2003.

[9] S. Staniford, V. Paxson, and N. Weaver. How to Own the Internet in Your Spare Time. In *11th Usenix Security Symposium*, San Francisco, August, 2002.

[10] D. Moore. Network Telescopes: Observing Small or Distant Security Events. In *11th USENIX Security Symposium*, 2002.

[11] C. Wang, J. C. Knight and M. C. Elder. On Viral Propagation and the Effect of Immunization. *Proceedings of 16th ACM Annual Computer Applications Conference*, New Orleans, LA, 2000.

[12] Y. Wang, D. Chakrabarti, C. Wang and C. Faloutsos. Epidemic Spreading in Real Networks: An Eigenvalue Viewpoint. *22nd Symposium on Reliable Distributed Computing*, Florence, Italy, Oct. 6-8, 2003.

[13] Y. Wang, C. Wang. Modeling the Effects of Timing Parameters on Virus Propagation. *ACM Workshop on Rapid Malcode*, Washington, DC, Oct. 27, 2003.

[14] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. A Taxonomy of Computer Worms. *ACM Workshop on Rapid Malcode*, Washington, DC, Oct. 27, 2003.

[15] C.C. Zou, W. Gong, and D. Towsley. Code Red Worm Propagation Modeling and Analysis. In *9th ACM Symposium on Computer and Communication Security*, Washington DC, 2002.

[16] C.C. Zou, L. Gao, W. Gong, and D. Towsley. Monitoring and Early Warning for Internet Worms. In *10th ACM Symposium on Computer and Communication Security*, Washington DC, 2003.

[17] C.C. Zou, D. Towsley, W. Gong, and S. Cai. Routing Worm: a Fast, Selective Attack Worm based on IP Address Information. *Univ. Massachusetts Technical Report TR-CSE-03-06*, November, 2003. http://tennis.ecs.umass.edu/ czou/research/routingWorm-techreport.pdf

[18] Mathworks: Simulink. http://www.mathworks.com/products/simulink/

[19] CAIDA. IP v4 Address Space Utilization. 1998. http://www.caida.org/outreach/resources/learn/ipv4space/

[20] Reserved IPv4 addresses. http://www.cidr-report.org/v6/reserved-ipv4.html

[21] CAIDA. Dynamic Graphs of the Nimda worm. http://www.caida.org/dynamic/analysis/security/nimda/

[22] eEye Digital Security. .ida "Code Red" Worm. http://www.eeye.com/html/Research/Advisories/AL20010717.html

[23] eEye Digital Security. CodeRedII Worm Analysis. http://www.eeye.com/html/Research/Advisories/AL20010804.html

[24] USA Today News. The cost of Code Red: $1.2 billion. http://www.usatoday.com/tech/news/2001-08-01-code-red-costs.htm

[25] CNN News. Computer worm grounds flights, blocks ATMs. http://europe.cnn.com/2003/TECH/internet/01/25/internet.attack/

[26] eEye Digital Security. Blaster Worm Analysis. http://www.eeye.com/html/Research/Advisories/AL20030811.html