Understanding Divide-Conquer-Scanning Worms

Yubin Li **Zesheng Chen**

Department of Electrical & Computer Engineering
Florida International University
Miami, FL 33174
E-mail: {yli004, zchen}@fiu.edu

Chao Chen

Department of Engineering
Indiana University - Purdue University Fort Wayne
Fort Wayne, IN 46805
E-mail: chen@engr.ipfw.edu

***Abstract*— Internet worms have been a significant security threat. Divide-conquer scanning is a simple yet effective technique that can potentially be exploited by future Internet epidemics. Therefore, it is imperative that defenders understand the characteristics of divide-conquer-scanning worms and study the countermeasures. In this work, we first provide the intuitions that a divide-conquer-scanning worm can potentially spread faster and stealthier than a traditional random-scanning worm. We then characterize the relationships between the propagation speeds of divide-conquer-scanning worms and the distributions of vulnerable hosts through mathematical analysis and simulations. Specifically, we find that if vulnerable hosts follow a non-uniform distribution such as the Witty-worm victim distribution, divide-conquer scanning can spread a worm much faster than random scanning. We also study empirically the effect of important parameters on the spread of divide-conquer-scanning worms. Furthermore, to counteract such attacks, we discuss the weakness of divide-conquer scanning and study a defense mechanism.**

I. Introduction

Internet worms self-propagate across the Internet by compromising vulnerable hosts and using them to attack other victims. Such malicious attacks have caused enormous damages and posed a significant security threat. For example, the Witty worm infected at least 12,000 hosts in 45 minutes in 2004 [14]; and the Storm worm affected tens of millions of hosts in 2007 [11]. Therefore, Internet worms have been identified as one of the top four security problems and targeted to be eliminated before 2014 [24].

To protect the Internet from worm attacks, we have to study the attacking methods that have been used by existing worms or will potentially be exploited by future worms. A key factor for an efficient worm attack is how a worm finds a target, which is called the *scanning method*. Although most real worms use the simple random-scanning method [3], many advanced worm-scanning strategies have been studied, including localized scanning [2], hitlist scanning [16], permutation scanning [16], routable scanning [18], [22], and importance scanning [4]. Different scanning methods have been designed for different purposes. We find, however, that all scanning strategies have to consider the following three parameters:

- *Scanning rate:* the rate at which a worm sends out scans to find targets. A worm may deliver as many scans as possible, such as the Slammer worm [9]; or dispatch scans slowly to avoid detection, such as the camouflaging worm [20].
- *Scanning probability:* the probability that a worm scans a specific address. A worm may use a uniform scanning method that hits each address equally likely, such as random scanning; or use a biased strategy that prefers scanning a certain range of IP addresses, such as importance scanning.
- *Scanning space:* the IP address space among which an infected host searches for vulnerable hosts. A worm can scan an entire IPv4 address space, such as localized scanning; or probe a routable address space, such as routable scanning. Moreover, different infected hosts may scan different address spaces at the same time.

Many studies on worm-scanning methods have focused on the scanning rate and the scanning probability [20], [19], [18], [21], [12], [4], [5], [17]. The methods that explore the scanning space, however, have been investigated little.

Divide-conquer scanning is a simple strategy that exploits the scanning space and makes different infected hosts probe different scanning spaces. Specifically, an infected host A searches for targets in its scanning space. Once A infects a target B, it would divide its scanning space into halves so that A scans one half and B scans the other half. Divide-conquer scanning is named after the "divide-and-conquer algorithm" that recursively breaks down a problem into two or more sub-problems until these sub-problems become simple enough to be solved directly [7]. Similar to the divide-and-conquer algorithm, the divide-conquer scanning attempts to partition the task of finding targets in a large address space into the sub-tasks of locating victims in a small address sub-space.

Although simple, the divide-conquer scanning exhibits some prominent characteristics:

- *Efficiency:* It attempts to avoid the redundancy that different infected hosts attack the same target. Hence, the scanning is more efficient.
- *Propagation speed:* It can potentially spread a worm much faster than random scanning. We will show this analytically and empirically in the paper.
- *Stealth:* It can propagate an epidemic stealthier than random scanning and avoid the detection of some defense systems such as network telescopes from CAIDA [23]. We will demonstrate this in Section II.

As a result, the divide-conquer scanning can be a powerful

attacking tool for future Internet epidemics.

To the best of our knowledge, only two works have studied divide-conquer-scanning worms. Specifically, divide-conquer scanning was first presented in [18], and was later modeled mathematically in [21]. Both works assume that vulnerable hosts are uniformly distributed and show that under such a condition, a divide-conquer-scanning worm has a similar propagation speed as a random-scanning worm. The real distributions of vulnerable hosts in the Internet, however, have been shown highly uneven [9], [10], [14], [12], [1], [5], [6], [17]. Therefore, it is unclear how fast divide-conquer-scanning worms can spread in the Internet and how defenders can fight against them.

The goal of this work is to better understand the spreading ability and the characteristics of divide-conquer-scanning worms. Our research work makes several contributions:

- We analytically and empirically demonstrate the effect of the vulnerable-host distribution on the spread of divide-conquer-scanning worms. Specifically, if the distribution of vulnerable hosts is not uniform, divide-conquer scanning can spread a worm faster than random scanning. This is because divide-conquer scanning could lead an epidemic to spread towards address sub-spaces with many vulnerable hosts. On the other hand, if the distribution of vulnerable hosts is uniform, divide-conquer scanning is slightly slower than random scanning at the late stage of worm propagation.
- We study empirically the effects of important parameters on the propagation of divide-conquer-scanning worms, such as the number of initially infected hosts (*i.e.,* hitlist), the scanning rate, and the degree of divide and conquer. Specifically, while the hitlist has a limited effect on the propagation speed of divide-conquer-scanning worms, the scanning rate affects the spread significantly. Moreover, if vulnerable hosts follow the distribution of Witty-worm victims, partitioning the address space beyond /8 subnets has little improvement on the spreading speed of a divide-conquer-scanning worm.
- We discuss the weakness of divide-conquer scanning and present a potential countermeasure. Specifically, we point out that removing infected hosts at the early stage has a significant effect on divide-conquer-scanning worms.

The remainder of this paper is structured as follows. Section II motivates the importance of studying divide-conquer scanning. Section III provides a mathematical model on the spread of divide-conquer-scanning worms under the special cases of vulnerable-hosts distributions. Section IV studies divide-conquer scanning through simulations. Next, Section V discusses a potential countermeasure. Finally, Section VI concludes the paper.

II. Motivations

In this section, we first give the background on divide-conquer scanning. We then provide the intuitions why divide-conquer scanning can potentially spread a worm faster and stealthier than random scanning.

TABLE I
NOTATIONS USED IN THIS PAPER.

Notations	Definition or explanation
l	Degree of divide and conquer for DCS
Ω	Size of the scanning space ($\Omega = 2^{32}$)
N	Total number of vulnerable hosts
s	Scanning rate or the number of scans sent by an infected host per unit time
$T(n)$	Average time for a scanning method to infect n hosts at the early stage of worm propagation
I_t	Number of infected hosts at time t

A. Divide-Conquer Scanning

Most real Internet worms use random scanning to locate vulnerable hosts. Random scanning selects target IPv4 addresses uniformly, and each infected host scans an entire IPv4 address space. Comparatively, divide-conquer scanning partitions the IPv4 address space into non-overlapping sub-spaces, and each infected host scans different sub-spaces simultaneously. Specifically, assume that a divide-conquer-scanning worm starts the propagation from an infected host A, which is searching for targets in the IPv4 address space, *i.e.,* $0.0.0.0/0$. Note that at the beginning A behaves identical to random scanning. When A hits a target B, A divides the scanning space into halves so that A would scan $0.0.0.0/1$ and B would scan $128.0.0.0/1$. More generally, if an infected host C is scanning subnet $a.b.c.d/k$ ($0 \leq k < 32$) and then hits a vulnerable host D, C would divide its scanning space so that C scans one half $a.b.c.d/(k+1)$ and D scans the other half $a.b.c.d/k - a.b.c.d/(k+1)$. In this way, the large IPv4 address space is divided into small address sub-spaces or subnets, which are processed by individual infected hosts. Note that once a scanning space is allocated to an infected host, this host will scan this space uniformly until it hits a target.

To bridge random scanning (RS) and divide-conquer scanning (DCS), in this work we consider a generalized version of DCS, called /l DCS, which works as follows:

- If an infected host is scanning subnet $a.b.c.d/k$ and $k < l$, this host would follow the original DCS and would divide its scanning space into halves after it compromises a target;
- otherwise, $k = l$; and the host will not divide its scanning space and will still scan $a.b.c.d/l$ even after infecting other hosts. The new victims compromised by this host will also scan subnet $a.b.c.d/l$.

As a result, RS can be regarded as a special case of /l DCS when $l = 0$, whereas the original DCS is another special case of /l DCS when $l = 32$. Moreover, l reflects the degree of divide and conquer; and a higher value of l indicates a higher degree of divide and conquer. The notations used in this paper are summarized in Table I.

B. Worm Propagation Speed

DCS can potentially spread a worm much faster than RS, depending on the distribution of vulnerable hosts. To show this intuitively, we construct a toy example and compare DCS with RS. Specifically, we consider a discrete-time system. Assume that there are totally N vulnerable hosts, and $N = 2^{16}$. The distribution of vulnerable hosts is extremely uneven so that all vulnerable hosts are contained in one /16 subnet. The worm starts the propagation from an infected host; and each infected host sends out s scans per unit time. We then calculate the *propagation time* $T(n)$, *i.e.,* the average time for a scanning method to infect n vulnerable hosts at the early stage.

First, we consider RS. The probability for a worm scan to hit a vulnerable host is $\frac{N}{\Omega}$, where $\Omega = 2^{32}$. If there are i infected hosts currently, the probability of recruiting a new victim at the next time step is

$$p_{\text{RS}}(i) \approx s \cdot i \cdot \frac{N}{\Omega}. \tag{1}$$

Here, $N \ll \Omega$ and we ignore the case that two or more worm scans hit the same target simultaneously at the early stage of worm propagation. Then, the time for i infected hosts to hit a vulnerable host, $X_{\text{RS}}(i)$, is a random variable and follows the geometric distribution with parameter $p_{\text{RS}}(i)$, *i.e.,*

$$\Pr(X_{\text{RS}}(i) = k) = p_{\text{RS}}(i)\left(1 - p_{\text{RS}}(i)\right)^{k-1}, \quad k = 1, 2, \cdots \tag{2}$$

Therefore, the average time for i infected hosts to recruit the $(i+1)$-th victim is

$$t_{\text{RS}}(i) = \text{E}[X_{\text{RS}}(i)] = \frac{1}{p_{\text{RS}}(i)} = \frac{\Omega}{s \cdot N \cdot i}, \tag{3}$$

which leads to calculate the propagation time of RS

$$T_{\text{RS}}(n) = \sum_{i=1}^{n} t_{\text{RS}}(i) = \sum_{i=1}^{n} \frac{1}{p_{\text{RS}}(i)} = \frac{\Omega}{sN} \sum_{i=1}^{n} \frac{1}{i}. \tag{4}$$

Next, we study /l DCS and assume that $l = 16$. The propagation of a DCS worm is demonstrated in Figure 1, where the shaded area indicates the /16 subnet containing all vulnerable hosts. Now we consider two cases:

- *Case 1:* $n \leq 16$. When there are i vulnerable hosts and $i \leq 16$, only one infected host is scanning the $\frac{\Omega}{2^{i-1}}$ space that contains vulnerable hosts, whereas other infected hosts cannot recruit any victim. Hence, the probability for these i infected hosts to find a vulnerable host in one time step is

$$p_{\text{DCS}}(i) = \frac{sN}{\Omega/2^{i-1}}. \tag{5}$$

  Therefore, when $n \leq 16$, the propagation time of DCS is

$$T_{\text{DCS}}(n) = \sum_{i=1}^{n} \frac{1}{p_{\text{DCS}}(i)} = \frac{\Omega}{sN} \sum_{i=1}^{n} \frac{1}{2^{i-1}}. \tag{6}$$

  Comparing Equation (4) and Equation (6), we find that $T_{\text{DCS}}(n) < T_{\text{RS}}(n)$ when $2 < n \leq 16$.
- *Case 2:* $n > 16$. After infecting 16 hosts, the /16 DCS worm will hit the /16 subnet that contains all vulnerable hosts and will scan this subnet uniformly without further divide and conquer. Since each scan from newly infected hosts would hit a vulnerable host, the scanning method becomes *hitlist scanning* [16], which obviously spreads a worm far faster than RS.

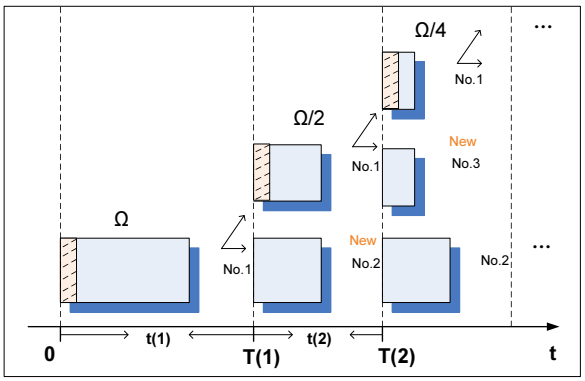


Fig. 1. Illustration of divide-conquer scanning.

Combining two cases, we conclude that DCS can spread a worm faster than RS in our example. Furthermore, the example also shows that DCS could lead a worm to propagate towards a subnet with many vulnerable hosts.

C. Stealth

DCS can propagate a worm stealthier than RS. Take network telescopes as an example. Network telescopes were proposed by CAIDA and monitor a globally routable address space where no active servers or services reside [23]. Hence, most traffic arriving at network telescopes is unwanted or malicious. CAIDA has used an entire /8 subnet as network telescopes and successfully observed the spreading behaviors of several large-scale RS Internet worm attacks such as Code Red [10], Slammer [9], and Witty [14]. The network telescopes used by CAIDA, however, may fail to detect the propagation of DCS worms. Specifically, assume that a subnet $a.b.c.d/k$ is used as network telescopes and contains no vulnerable hosts. Consider a DCS worm that starts from an infected host. Under the best case, the network telescopes can only observe the scans from $k+1$ different infected hosts that scan $a.b.c.d/i$ where $i = 0, 1, \cdots, k$. For example, if $k = 8$ as the real network telescopes used by CAIDA, at most 9 infected hosts would be perceived. Since currently the level of background noise on network telescopes is high [13], it is very difficult to detect the appearance of worms based on the observations of such a small number of infected hosts.

DCS can also weaken the performance of some other detection mechanisms. For example, the systems proposed in [15], [8] make use of destination address dispersion to detect worm appearance. These systems assume that once a worm is released, the distribution of destination addresses will be far more even than typical network traffic that usually has significant clustering. A host infected by DCS worms, however, may scan only a small subnet, instead of the entire

IPv4 address space, and thus lead to the skew distribution of destination addresses. Therefore, the systems in [15], [8] cannot detect DCS worms as easily as RS worms.

III. Mathematical Model

In this section, we extend the analytical active worm propagation (AAWP) model [3] to characterize the spread of DCS worms. Our goal of studying the mathematical model is to provide insights on how DCS worms spread over different vulnerable-host distributions. Specifically, we assume that vulnerable hosts are uniformly distributed in a /m ($m \geq 0$) network. We start from a simple case and then study more complex cases.

A. Uniform in the IPv4 Address Space ($m = 0$)

We first assume that vulnerable hosts are uniformly distributed in the entire IPv4 address space, which has been studied in [18], [21]. Let I_t be the number of infected hosts at the discrete time t ($t \geq 0$). Assume that the worm starts from an infected host (*i.e.,* $I_0 = 1$). Since the distribution of vulnerable hosts is uniform, all I_t infected hosts can be assumed to behave identically, and each of them uses a scanning rate of s to probe a non-overlapping address sub-space with the size of Ω/I_t. Then, the probability that a vulnerable host is hit by at least one worm scan in a unit time is $1-[1-1/(\Omega/I_t)]^s$. Therefore, we can extend the AAWP model to characterize I_{t+1} recursively, *i.e.,*

$$I_{t+1} = I_t + (N - I_t)\left[1 - \left(1 - \frac{I_t}{\Omega}\right)^s\right] \quad (7)$$

$$\approx I_t + \frac{s}{\Omega} I_t (N - I_t), \quad (8)$$

where $\Omega \gg 1$. Note that Equation (7) is identical to the result in [18]. Note that Equation (8) implies that when $m = 0$, the original DCS is equivalent to RS (*i.e.,* /0 DCS) from the perspective of mathematical modeling.

B. Uniform in Half of the IPv4 Address Space ($m = 1$)

Next, we consider that vulnerable hosts are uniformly distributed in half of the IPv4 address space (*e.g.,* 0.0.0.0/1). When $1 \leq I_t < 2$, the initially infected host scans the entire IPv4 address space uniformly, and Equation (7) still holds. When $I_t \geq 2$, one infected host scans one half of the IPv4 address space that contains no vulnerable host (*i.e.,* 128.0.0.0/1); and other $I_t - 1$ infected hosts partition the other half of the IPv4 address space into sub-spaces with the equal size of $\Omega/[2(I_t-1)]$. Then, the probability that a vulnerable host is hit by at least one scan in a unit time is $1-[1-2(I_t-1)/\Omega]^s$. Therefore, I_{t+1} can be described as

$$I_{t+1} = I_t + (N - I_t)\left\{1 - \left[1 - \frac{2(I_t - 1)}{\Omega}\right]^s\right\} \quad (9)$$

$$\approx I_t + \frac{2s}{\Omega}(I_t - 1)(N - I_t). \quad (10)$$

Comparing Equation (10) with Equation (8), we find that a DCS worm can spread faster under the case when $m = 1$ than the case when $m = 0$.

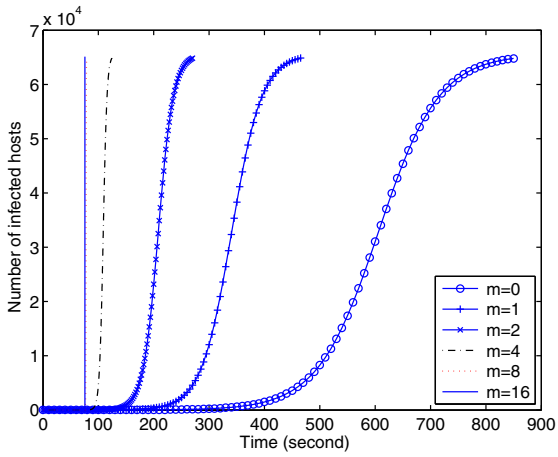


Fig. 2. Effect of vulnerable-host distributions on the spread of DCS worms ($N = 65,536$, $s = 1,200$ /second, and $I_0 = 1$).

C. Uniform in a /m Network

Finally, we study a more general case that vulnerable hosts are uniformly distributed in a /m ($m \geq 0$) network. Consider two situations. The first situation is when $I_t < m + 1$. Specifically, when $i \leq I_t < i + 1$ ($i = 1, 2, \cdots, m$), $i - 1$ infected hosts scan the address sub-spaces that contain no vulnerable host, whereas other $I_t - (i - 1)$ infected hosts probe the address sub-space with the size of $\Omega/2^{i-1}$. Then, the probability that a vulnerable host is hit by at least one scan in a unit time is $1-[1-2^{i-1}(I_t-i+1)/\Omega]^s$. Therefore, I_{t+1} is derived as

$$I_{t+1} = I_t + (N - I_t)\left\{1 - \left[1 - \frac{2^{i-1}(I_t - i + 1)}{\Omega}\right]^s\right\}$$

$$\approx I_t + \frac{2^{i-1}s}{\Omega}(I_t - i + 1)(N - I_t). \quad (11)$$

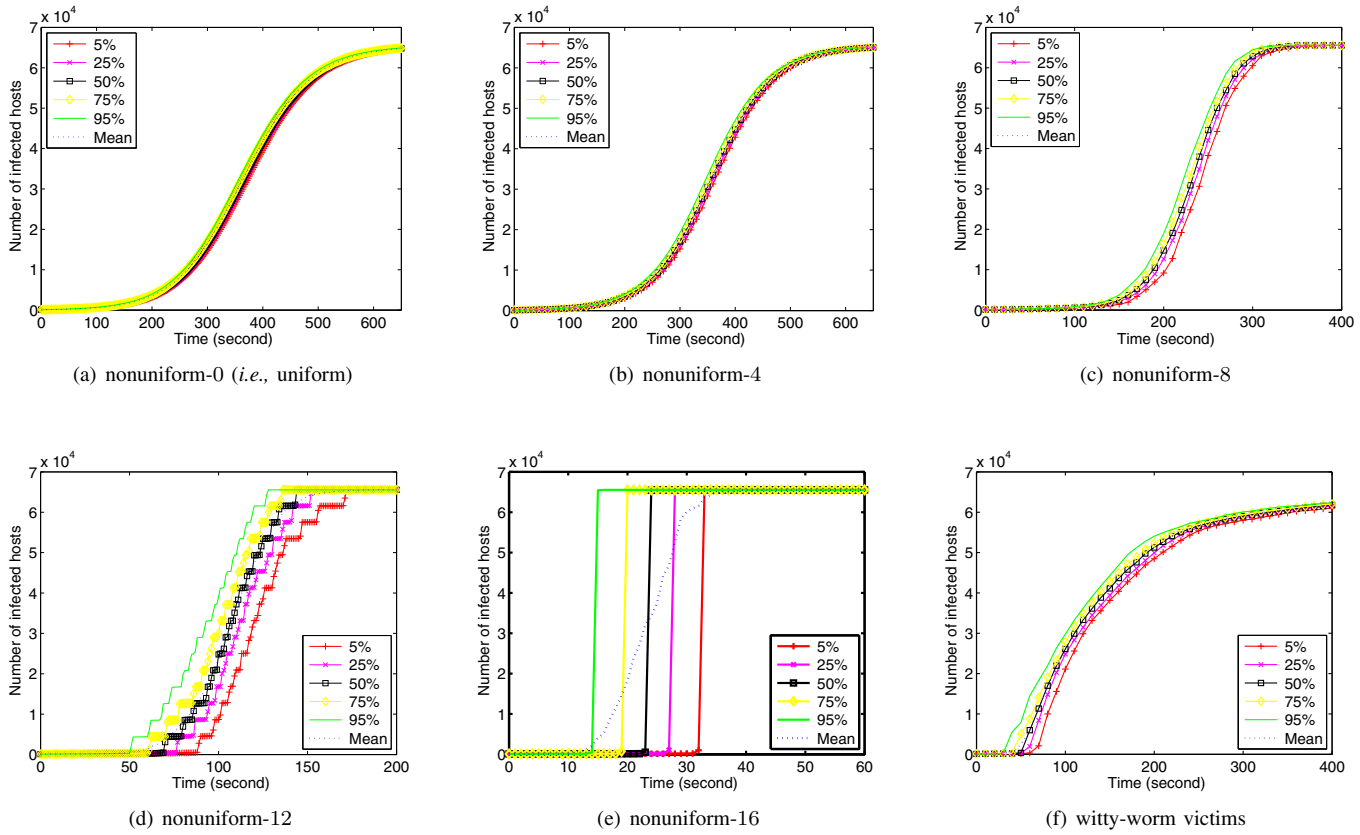
The second situation is when $I_t \geq m+1$. In such a situation, m infected hosts scan the sub-spaces containing no vulnerable host, whereas other $I_t - m$ infected hosts partition the /m subnet into sub-spaces with the equal size of $\Omega/[2^m(I_t-m)]$. Therefore, the spread of DCS worms can be modeled as

$$I_{t+1} = I_t + (N - I_t)\left\{1 - \left[1 - \frac{2^m(I_t - m)}{\Omega}\right]^s\right\}$$

$$\approx I_t + \frac{2^m s}{\Omega}(I_t - m)(N - I_t). \quad (12)$$

Note that Equation (12) implies that when $m = l$, the original DCS is equivalent to /l DCS from the view of modeling. It can be seen that when m is larger, *i.e.,* the distribution of vulnerable hosts is more uneven, the DCS worm can spread faster.

Figure 2 shows the effect of vulnerable-host distributions on the spread of DCS worms by applying the above equations and varying m. Here, a DCS worm starts from an infected host (*i.e.,* $I_0 = 1$) and uses the scanning rate of $1,200$ per second to infect a vulnerable population of 2^{16} ($= 65,536$). When $m = 0$, *i.e.,* the distribution of vulnerable hosts is



(a) nonuniform-0 (*i.e.,* uniform) (b) nonuniform-4 (c) nonuniform-8

(d) nonuniform-12 (e) nonuniform-16 (f) witty-worm victims

Fig. 3. Effect of vulnerable-host distributions on the spread of /16 DCS worms.

uniform, a DCS worm takes 857 seconds to compromise 99% of vulnerable hosts. When m becomes larger, *i.e.,* the distribution of vulnerable hosts becomes more uneven, the DCS worm spends less time to infect the same number of hosts. Specifically, when $m = 1$, 2, 4, 8, and 16, the worm takes only 466, 271, 125, 79, and 76 seconds. Moreover, it is seen that when $m \geq 8$, the worm propagates with a similar speed.

IV. Simulation Study

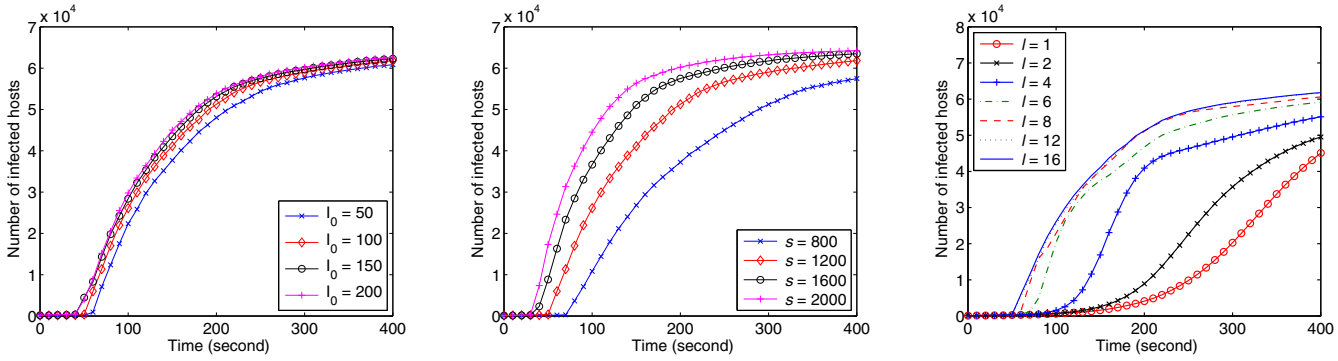
In this section, we apply simulations to study DCS worms. Although the mathematical model in Section III provides direct relationships between the propagation speeds of DCS worms and the distributions of vulnerable hosts under specific cases, simulations on the spread of DCS worms are still necessary for three reasons:

- The model in Section III is built upon simplified assumptions (*e.g.,* infected hosts that scan towards the /m network are assumed to behave identically). But simulations can relax these assumptions and provide more realistic scenarios.
- The model only considers the specific cases of vulnerable-host distributions (*i.e.,* uniform in a /m network). But simulations can study the arbitrary distribution of vulnerable hosts.
- The model only characterizes the average number of infected hosts. But simulations can give both the mean and the variation of the number of infected hosts.

In our simulations, we use a discrete event simulator to imitate the propagation of DCS worms. Our simulator implements each worm scan through a random number generator; and each scenario runs 100 times with different seeds. The default parameter setting for the simulated worm is that the worm starts from 100 initially infected hosts (*i.e.,* $I_0 = 100$) and uses a scanning rate of $1,200$ (*i.e.,* $s = 1,200$) to compromise a vulnerable population of $65,536$ (*i.e.,* $N = 2^{16}$). Each of initially infected hosts scans the IPv4 address space and follows /16 DCS.

A. Vulnerable-Host Distributions

We first study how the distribution of vulnerable hosts affects the spread of /16 DCS worms. To reflect the degree of the unevenness of a distribution, we design the "nonuniform-u" ($u = 0, 1, \cdots, 16$) distribution as follows. The Internet is partitioned into 2^{16} /16 subnets, which is denoted as $a.b.c.d/16$. These 2^{16} subnets are grouped into $a.b.c.d/(16-u)$ subnets, each of which has 2^u /16 subnets. In each group, the first /16 subnet contains 2^u vulnerable hosts, whereas other /16 subnets have no vulnerable host. In this way, the nonuniform-0 distribution denotes a uniform distribution, whereas the nonuniform-16 distribution reflects



(a) Hitlist (b) Scanning rate (c) Degree of divide and conquer

Fig. 4. Effect of parameters on the spread of /16 DCS worms.

an extremely uneven distribution, *i.e.,* all vulnerable hosts concentrate in a /16 subnet. A higher value of u gives a more uneven distribution of vulnerable hosts.

Figure 3 shows the spread of /16 DCS worms over the nonuniform-u distribution (when $u = 0$, 4, 8, 12, and 16) and the distribution of Witty-worm victims [25]. In this figure, the "5%" (or "95%") curve denotes that a worm propagates no slower (or faster) than this curve in 95 out of 100 simulation runs. The similar definition applies for the "25%", "50%", and "75%" curves. The "mean" curve is the average over 100 runs. It can be seen that when u increases, the DCS worm uses less time to compromise all vulnerable hosts. Moreover, the shape of the worm propagation curve differs significantly for different distributions of vulnerable hosts. Specifically, if $u = 0$, 4, and 8, the curve follows the well-known logistic curve [26]. If $u = 12$ and 16, however, the most part of the curve is (nearly) linear. Specifically, when $u = 16$, as pointed out by Section II, after infecting 16 hosts, DCS becomes hitlist scanning, and thus most vulnerable hosts are infected in a very short time. If vulnerable hosts follow the distribution of Witty-worm victims, the DCS worm can compromise most vulnerable hosts with the speed similar to the case when $u = 8$. It is observed, however, that for the distribution of Witty-worm victims, the worm spreads much faster at the early stage, and the propagation takes off and changes to the full speed around 50 seconds. Therefore, a DCS worm can potentially propagate relatively fast, especially at the early stage, based on the realistic distribution of vulnerable hosts.

B. Parameters

Next, we consider how the important parameters affect the propagation of DCS worms, such as the number of initially infected hosts (*i.e.,* hitlist I_0), the scanning rate (*i.e.,* s), and the degree of divide and conquer (*i.e.,* l). Figure 4 demonstrates the spread of /16 DCS worms when these parameters vary. Note that for each scenario, vulnerable hosts follow the distribution of Witty-worm victims; and the curve is the "50%" curve over 100 runs. Specifically, Figure 4(a) shows the spread of /16 DCS worms with different hitlist sizes, a vulnerable population of 2^{16}, and a scanning rate of $1,200$. It is seen

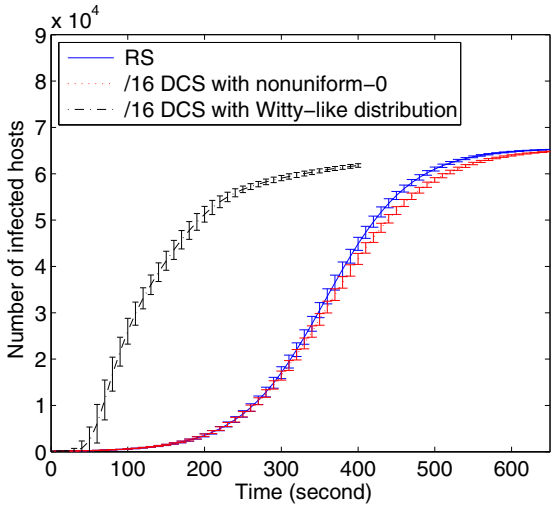


Fig. 5. Comparison between RS and /16 DCS ($N = 65,536$, $s = 1,200$ /second, and $I_0 = 100$).

that the worm propagation speed increases slightly when the hitlist size increases from 50 to 200. Figure 4(b) plots the propagation of /16 DCS worms with different scanning rates, a vulnerable population of 2^{16}, and a hitlist size of 100. It is observed that the worm increases its spreading speed significantly when the scanning rate increases from 800 to $2,000$. Figure 4(c) compares the spreading speeds of DCS worms with the different degrees of divide and conquer, a vulnerable population of 2^{16}, a hitlist size of 100, and a scanning rate of $1,200$. It is obvious that when l increases, the DCS worm spreads faster. Moreover, when $l \geq 8$, the improvement on the propagation speed by increasing l becomes marginal.

C. Comparison with Random Scanning

Finally, we compare DCS with RS in Figure 5. In the figure, the curve shows the mean of 100 runs, whereas the error bar represents the standard deviation over 100 runs. Here, the worm uses a hitlist size of 100 and a scanning rate of $1,200$ to compromise a vulnerable population of 2^{16}. Two distributions of vulnerable hosts are applied to /16 DCS: the nonuniform-0

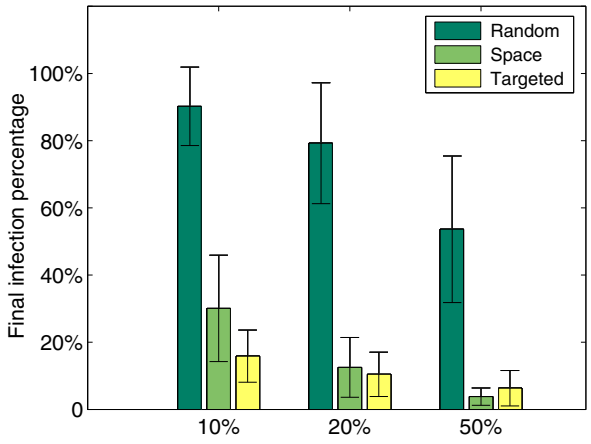


Fig. 6. Effect on /16 DCS worms by removing part of infected hosts at time $t1$ using different strategies ($N = 65,536$, $I_0 = 1$, and $I_{t1} = 100$).

distribution (*i.e.,* uniform distribution) and the distribution of Witty-worm victims. It is seen that when the distribution is uniform, the /16 DCS worm spreads slightly slower than an RS worm at the late stage. This is because for DCS worms, when the infected hosts in a /16 subnet become saturated, the scans from these hosts waste. On the other hand, if vulnerable hosts follow the Witty-like distribution, /16 DCS spreads a worm much faster than RS. Specifically, RS uses 479 seconds to infect 90% of vulnerable hosts, whereas /16 DCS takes only 300 seconds.

V. Countermeasures

How can we defend against DCS worms? In this section, we first discuss the weakness of DCS and then study a simple countermeasure.

A DCS worm assigns different address sub-spaces to different infected hosts so that the worm can spread efficiently and stealthy. On the other hand, the assignment of address sub-spaces leads to the fact that the failures of infected hosts at the early stage make the worm miss scanning a certain range of IP addresses [18]. Comparatively, an RS worm can still recruit all targets under the condition that most (but not all) infected hosts fail at the early stage. Therefore, RS is regarded as a *robust* scanning method, whereas DCS is not. That is, DCS is vulnerable to nodal failures or removals at the early stage.

Hence, we consider a simple countermeasure as follows. Once the appearance of DCS worms is detected, remove part of infected hosts immediately. Specifically, we study three different removing strategies:

- *Random:* Remove infected hosts randomly.
- *Space:* Remove infected hosts that scan the largest address sub-spaces.
- *Targeted:* Remove infected hosts that scan address sub-spaces containing the largest number of vulnerable hosts.

Figure 6 shows the effect of removing part of infected hosts at the early stage using random, space, and targeted strategies. Specifically, the /16 DCS worm attacks a vulnerable population of $65,536$, starting from an infected host (*i.e.,* $I_0 = 1$). We assume that at time $t1$ when 100 hosts are compromised (*i.e.,* $I_{t1} = 100$), the worm is detected, and then 10%, 20%, and 50% of infected hosts are removed. We calculate the percentage of vulnerable hosts that can be infected eventually, *i.e.,*, the final infection percentage. The bar line in the figure is the average of 100 runs, whereas the error-bar represents the standard deviation over 100 runs. It is seen that if the random strategy is used, the percentage of vulnerable hosts that cannot be infected eventually is roughly the same as the percentage of infected hosts removed at time $t1$. Therefore, even a simple removing strategy has a significant effect on DCS worms. More advanced strategies (*i.e.,* space and targeted) have a more significant influence on protecting vulnerable hosts. For example, when 10% infected hosts are removed at time $t1$, space and targeted strategies can reduce the final infection percentage to 30.08% and 15.85%, respectively. When the percentage of removed infected hosts increases to 50%, space and targeted strategies further reduce the final infection percentage to 3.82% and 6.32%, respectively. Figure 6 also shows that the targeted strategy is not always better than the space strategy. This is because we study a /16 DCS worm, instead of the original DCS worm. Once the worm hits a /16 subnet, it will not further divide the /16 subnet. Thus, some infected hosts may scan the same /16 subnet simultaneously. Moreover, the infected hosts at time $t1$ tend to concentrate in /16 subnets containing many vulnerable hosts. As a result, to protect the vulnerable hosts in a dense /16 subnet, all infected hosts that scan this subnet have to be removed. The targeted strategy, however, may not fulfill such a task in some cases.

On the other hand, attackers may strengthen DCS worms by adding the scanning redundancy to avoid the issue of the single-point failure. For example, instead of starting from one infected host, the worm can spread from 100 initially infected hosts that all scan the IPv4 address space (*e.g.,* $I_0 = 100$). In our future work, we would study the countermeasures against such worms.

VI. Conclusions

In this paper, we attempt to better understand the characteristics of DCS worms and the potential countermeasure through both analysis and simulations. We have shown intuitively that a DCS worm can propagate both faster and stealthier than a traditional RS worm through toy examples. We have also demonstrated analytically and empirically that DCS can spread a worm much faster than RS if the vulnerable hosts follow a non-uniform distribution such as the Witty-like distribution. To counteract DCS worms, we have exploited one weakness of DCS and studied the impact of using different strategies to remove some infected hosts at the early stage on the worms.

As part of our on-going work, we plan to develop other effective defense mechanisms against DCS worms. Moreover, we will study future intelligent worms that exploit all three

parameters (*e.g.,* scanning rate, scanning probability, and scanning space) in an optimal way.

REFERENCES

[1] P. Barford, R. Nowak, R. Willett, and V. Yegneswaran, "Toward a model for sources of Internet background radiation," in *Proc. of the Passive and Active Measurement Conference (PAM'06)*, Mar. 2006.

[2] Z. Chen, C. Chen, and C. Ji, "Understanding localized-scanning worms," in *Proc. of 26th IEEE International Performance Computing and Communications Conference (IPCCC'07)*, New Orleans, LA, Apr. 2007, pp. 186-193.

[3] Z. Chen, L. Gao, and K. Kwiat, "Modeling the spread of active worms," in *Proc. of INFOCOM'03*, vol. 3, San Francisco, CA, Apr. 2003, pp. 1890-1900.

[4] Z. Chen and C. Ji, "Optimal worm-scanning method using vulnerable-host distributions," *International Journal of Security and Networks: Special Issue on Computer and Network Security*, vol. 2, no. 1/2, 2007.

[5] Z. Chen and C. Ji, "Measuring network-aware worm spreading ability," in *Proc. of INFOCOM'07*, Anchorage, AK, May 2007.

[6] Z. Chen, C. Ji, and P. Barford, "Spatial-temporal characteristics of malicious sources," in *Proc. of INFOCOM'08 Mini-Conference*, Phoenix, AZ, Apr. 2008.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*. The MIT Press and McGraw-Hill, 2002.

[8] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. of ACM SIGCOMM'05*, Philadelphia, PA, Aug. 2005.

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

[10] D. Moore, C. Shannon, and J. Brown, "Code-Red: a case study on the spread and victims of an Internet worm," in *ACM SIGCOMM/USENIX Internet Measurement Workshop*, Marseille, France, Nov. 2002.

[11] P. Porras, H. Saidi, and V. Yegneswaran, "A multi-perspective analysis of the Storm (Peacomm) Worm," *SRI Technical Report*, Nov. 2007.

[12] M. A. Rajab, F. Monrose, and A. Terzis, "On the effectiveness of distributed worm monitoring," in *Proc. of the 14th USENIX Security Symposium (Security'05)*, Baltimore, MD, Aug. 2005, pp. 225-237.

[13] D. W. Richardson, S. D. Gribble, and E. D. Lazowska, "The limits of global scanning worm detectors in the presence of background noise," in *Proc. of ACM Workshop on Rapid Malcode (WORM'05)*, Fairfax, VA, Nov. 2005, pp. 60-70.

[14] C. Shannon and D. Moore, "The spread of the Witty worm," *IEEE Security and Privacy*, vol. 2, no 4, Jul-Aug 2004, pp. 46-50.

[15] S. Singh, C. Estan, G. Varghese, and S. Savage, "Automated worm fingerprinting," in *Proc. of the 6th ACM/USENIX Symposium on Operating System Design and Implementation (OSDI'04)*, San Francisco, CA, Dec. 2004, pp. 45-60.

[16] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the Internet in your spare time," in *Proc. of the 11th USENIX Security Symposium (Security'02)*, San Francisco, CA, Aug. 2002.

[17] M. Vojnovic, V. Gupta, T. Karagiannis, and C. Gkantsidis, "Sampling strategies for epidemic-style information dissemination," in *Proc. of INFOCOM'08*, Phoenix, AZ, Apr. 2008.

[18] J. Xia, S. Vangala, J. Wu, L. Gao, and K. Kwiat, "Effective worm detection for various scan techniques," *Journal of Computer Security*, vol. 14, no. 4, 2006, pp. 359-387.

[19] W. Yu, X. Wang, D. Xuan, and D. Lee, "Effective detection of active smart worms with varying scan rate," in *Proc. of IEEE Communications Society/CreateNet International Conference on Security and Privacy in Communication Networks (SecureComm'06)*, Aug. 2006.

[20] W. Yu, X. Wang, D. Xuan, and W. Zhao, "On detecting camouflaging worm," in *Proc. of Annual Computer Security Applications Conference (ACSAC'06)*, Dec. 2006.

[21] C. C. Zou, D. Towsley, and W. Gong, "On the performance of Internet worm scanning strategies," *Elsevier Journal of Performance Evaluation*, vol. 63. no. 7, July 2006, pp. 700-723.

[22] C. C. Zou, D. Towsley, W. Gong, and S. Cai, "Advanced routing worm and its security challenges," *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 82, no. 1, 2006, pp.75-85.

[23] CAIDA, "Network telescope," http://www.caida.org/research/security/telescope/.

[24] Computing Research Association, "Grand research challenges in information security & assurance," http://www.cra.org/Activities/grand.challenges/security/home.html.

[25] The CAIDA Dataset on the Witty Worm - March 19-24, 2004, Colleen Shannon and David Moore, http://www.caida.org/data/passive/witty_worm_dataset.xml. Support for the Witty Worm Dataset and the UCSD Network Telescope are provided by Cisco Systems, Limelight Networks, the US Department of Homeland Security, the National Science Foundation, DARPA, Digital Envoy, and CAIDA Members.

[26] Wikipedia, "Logistic function," http://en.wikipedia.org/wiki/Logistic_function.