Modeling and Automated Containment of Worms

Sarah Sellke, Ness B. Shroff, and Saurabh Bagchi
School of Electrical and Computer Engineering
Purdue University
{ssellke, shroff, sbagchi}@ecn.purdue.edu *

Abstract

Self-propagating codes, called worms*, such as Code Red, Nimda, and Slammer, have drawn significant attention due to their enormous adverse impact on the Internet. There is a great interest in the research community in modeling the spread of worms and in providing adequate defense mechanisms against them.*

In this paper, we present a (stochastic) branching process *model for characterizing the propagation of Internet worms. This model leads to the development of an automatic worm containment strategy that prevents the spread of worms beyond its early stages. Specifically, using the branching process model, we are able to (1) provide a precise condition that determines whether the worm will eventually die out and (2) provdide the probability that the total number of hosts that the worm infects will be below a certain level. We use these insights to develop a simple automatic worm containment scheme, which is demonstrated, through simulations and real trace data, to be both effective and non-intrusive.*

Keywords: Internet scanning worms, stochastic worm modeling, branching process model, early phase propagation, automatic worm containment.

1. Introduction

The Internet has become critically important to the financial viability of the national and global economy. Meanwhile, we are witnessing an upsurge in the incidents of malicious code in the form of computer viruses and worms. One class of such malicious code, known as worms, spreads itself without human intervention by using a scanning strategy to find vulnerable hosts to infect. Code Red, SQL Slammer, and Sasser are some of the more famous examples of worms that have caused considerable damage. Network worms have the potential to infect many vulnerable hosts on the Internet before human countermeasures take place. The aggressive scanning traffic generated by the infected hosts have caused network congestion, equipment failure, and blocking of physical facilities such as subway stations, 911 call centers, etc. As a representative example, consider the Code Red worm version 2 that exploited a buffer overflow vulnerability in the Microsoft IIS web servers. It was released on July 19th, 2001 and over a period of less than 14 hours infected more than 359,000 machines. The cost of the epidemic, including subsequent strains of Code Red is estimated by Computer Economics to be $2.6 billion [22]. While Code Red was particularly virulent in its economic impact (e.g., see [2, 11]) it provides an indication of the magnitude of the damage that can be inflicted by such worms. Thus, there is a need to carefully characterize the spread of worms and develop efficient strategies for worm containment.

In the current literature, three broad classes of strategies have been identified for mitigating the risks of worms.
(i) *Prevention:* This involves improving the security and heterogeneity of software on the Internet and automatically checking hosts for vulnerabilities worms could exploit, and patching them before a worm incident happens; (ii) *Treatment:* This involves eliminating the vulnerability exploited by the worm after the incident has become known and removing the worm from the host itself; (iii) *Containment:* This involves blocking or slowing down the communication between infected and uninfected hosts. These three strategies complement each other and in this paper, our focus will be on developing an effective containment strategy.

The goal of our research is to provide a model for the propagation of random scanning worms and the corresponding development of automatic containment mechanisms that prevent the spread of worms beyond its early stages.

Several early worm warning and detection systems have been proposed [10, 20, 21]. Most models of worm propagation are based on deterministic epidemic models [3, 15, 19]. They are acceptable for modeling worm propagation when the number of infected hosts is large. However, it is gen-

*This work is partially supported by the National Science Foundation grant 0335247-ANI and an NSF Graduate Fellowship.

erally accepted that they are inadequate to model the early phase of worm propagation accurately because the number of infected hosts earlier on is very small [10]. The reason is that epidemic models capture only expected or mean behavior, while not being able to capture the variability around this mean, which could be especially dramatic during the early phase of worm propagation. While stochastic epidemic models can be used to model this early phase, they are generally too complex to provide useful analytical solutions.

In this paper, we propose a branching process model for the early phase of worm propagation. We consider the generation-wise evolution of worms, with the hosts that are infected at the beginning of the propagation forming generation zero and a host in generation n infecting hosts that will be said to belong to generation $n+1$. According to the branching process model, each individual in generation n independently produces a random number of individuals in generation $n+1$, according to a fixed probability distribution that does not vary from individual to individual. Our model allows us to better understand the worm spreading dynamics for worms of arbitrary scanning rate, including stealth worms that may turn themselves off at times.

Using this model, we find that it is the total number of scans that any infected host attempts, and not the more restrictive scanning rate, that determines whether the worms can spread. The total number of scans M is over a period we call the *containment cycle*. As we will illustrate in this paper, the containment cycle is a relatively long period of time, on the order of weeks, and can be determined based on the host's normal scanning characteristics. In practice, the value of M is a large number that prevents worm spreading without interfering with legitimate traffic. Note that this is fundamentally different from rate limiting schemes because we are not bounding instantaneous scanning rates. Instead, the limiting is done over a large window of time that can be adapted to host characteristics. Further, this limiting value is derived from our branching process model and can be tuned to enforce a more (or less) rapid termination of worm spread. Using **Code Red** as an example, we show that if we restrict the total scans per host to $M = 10000$, with a high probability (0.99), the total number of infected hosts on the Internet will be less than 360, which according to [11], corresponded to only about 0.1% of the total vulnerable hosts at the time of the outbreak.

The main contributions of the paper can be summarized as follows. We provide a means to accurately model the early phase of propagation of uniform scanning worms. Our model provides us with a mechanism for containing worm spread without needing to detect whether a host is infected. This scheme is non-intrusive in terms of its impact on legitimate traffic. Our model and containment scheme are validated through analysis, simulation, and real traffic statistics.

The rest of the paper is organized as follows, In section 2, we review relevant research on network worms. In Section 3, we present our *branching process* model with corresponding analytical results on the spread of the infection. In Section 4, we describe an automatic worm containment scheme. In Section 5, we provide numerical results that validate our model and confirm the effectiveness of our containment scheme. In Section 6, we summarize our contributions, provide some discussion, and directions for future work.

2. Related Work

As mentioned in the introduction, deterministic epidemic models have been used to study worm propagation [15, 19]. For illustration, consider the two factor worm model proposed by Zou *et al.* [19]:

$$\frac{dI(t)}{dt} = \beta(t)[V - R(t) - I(t) - Q(t)]I(t) - \frac{dR(t)}{dt}, \quad (1)$$

where V is the total number of susceptible hosts on the Internet, and $I(t), R(t), Q(t)$ represent the number of infectious hosts, the number of removed hosts from the infectious population, and the number of removed hosts from the susceptible population at time t, respectively. The parameter $\beta(t)$ is the infection rate at time t and reflects the impact of the Internet traffic on the worm propagation. The parameters $R(t)$ and $Q(t)$ reflect the human countermeasures in patching.

When there is no patching and when the infection rate is constant, the two factor model equation is the random constant spread model (RCS) proposed by Staniford *et al.* [15]:

$$\frac{dI(t)}{dt} = \beta I(t)(V - I(t))$$

These types of models are suitable for when there are a large number of infected hosts. However, during the early stage of the worm propagation, the number of infected hosts is small and these deterministic models may not accurately characterize the spread of worms. Nonetheless, most existing models for Internet worms are based on deterministic epidemic models.

Early worm detection systems have been proposed by several researchers. Zou *et al.* use a Kalman filter [20] to detect the worms. The Kalman filter is used to detect the presence of a worm by detecting the trend, not the rate, of the observed illegitimate scan traffic. The filter is used to separate worm traffic from background non-worm scan traffic. Liljenstam *et al.* and Bert develop an early worm detection system called DIB:S/TRAFEN in which a select group of routers forward ICMP T-3 packets to the analysis station. It is shown in [10] that the total number of participating routers can be small, but these routers must be

distributed across a significant fraction of the Internet address space to ensure timely and accurate worm detection. With optimized deployments, it is shown that the system can detect the Code Red worm when there are only 0.03% vulnerable hosts infected. They develop a worm simulation model that is used for generating worm traffic for evaluating the DIB:S/TRAFEN detection system. Their simulation model uses a combination of the deterministic epidemic model and a general stochastic epidemic model to model the effect of large scale worm attacks. They found the stochastic epidemic model is useful for modeling the early stage of the worm spread. However, the complexity of the general stochastic epidemic model does not yield to closed form analysis.

Rate-control based countermeasures, such as *Virus throttling* by Williamson [17] have been shown to be successful in detecting and slowing down fast scanning worms. Wong *et al.* [18] studied the effect of rate control on suppressing the spread of the worms when this mechanism is deployed at various points (e.g., host, LAN, and core router) of the network. The rate control is effective in slowing down fast worms but is not effective against slow scanning worms. In addition, the limit on the rate must be carefully tuned in order to let the normal traffic through.

Zou *et al.* propose and analyze a dynamic quarantine scheme of Internet worms [21]. They assume that the underlying worm detection system has a certain false alarm rate. Their dynamic quarantine system confines all hosts that have triggered the alarm, and automatically releases them after a short time. They found that this scheme can slow down the worm spread but cannot guarantee containment.

Moore *et al.* [13] examined the reaction time required to contain the Internet scale worms using countermeasures such as blacklisting the infected hosts and content filtering at routers. Their study concluded that to effectively contain Internet worms, it is necessary to take actions early, within minutes of the worm outbreak.

The goal of our research is to provide a model for the propagation of random scanning worms, which can lead to the development of automatic containment mechanisms that prevent the spread of worms beyond its early stages. We use a stochastic model called the branching process model [6, 14] to characterize the spread of worms. Using the branching process model, we show how to guarantee an extinction probability[1] of 1 by appropriately limiting the total number of scans per host M in a containment cycle. More importantly, we are also able to probabilistically bound the total number of hosts that are infected for a given value of M.

We develop an automatic worm containment strategy based on the insights gained from our model. The main idea is to limit the total number of distinct IP addresses contacted (denoted the limit as M)) per host over a long period of time (weeks or even months). The value for M does not need to be carefully tuned as in the traditional rate control mechanism, because our theoretical results suggest M can be much larger than the normal network activities and can still effectively contain the worms. Our containment scheme can effectively contain both fast scan worms and slow scan worms *without knowing the worm signature in advance or needing to detect the worm.*

3. Branching Process Model for Random Scanning Worms

We now present the branching process model we use to characterize the propagation of random scanning worms. Scanning worms are those that generate a list of random IP addresses to scan from an infected host. The uniform scanning worms are those in which the addresses are chosen completely randomly while preference scanning worms weight the probability of choosing an address from different parts of the network differently. In this paper, we will focus on the uniform scanning worms, and discuss possible extensions for preference scanning worms. In our model, a vulnerable host is assumed to be in one of three states: *susceptible, infected, and removed*. A susceptible host is one that is vulnerable to being infected by the worm. The term *vulnerable* is used synonymously here. An infected host generates a list of random IP address to scan. If a susceptible host is found among the scans, it will become infected. A removed host is one which has been removed from the list of hosts that can be infected.

We use V to denote the total number of vulnerable hosts. The probability of successfully finding a vulnerable host in one scan is $p = \frac{V}{2^{32}}$, for 2^{32}, which is the size of current IPv4 address space. We call p the density of the vulnerable hosts, or *vulnerability density* for short. The value of p also measures how widespread a vulnerability is. There were approximately 360,000 vulnerable hosts during the Code Red outbreak [11][2]. In this case, the vulnerability density p is only 8.5×10^{-5}.

We assume that the total scans of an infectious host is no more than M, i.e, we put *an upper bound* of M on the number of times an infected host can scan in its containment cycle. We characterize the values M can take to achieve an extinction probability of one. We also provide the probability distribution of the total number of infected hosts as a

[1]The extinction probability is the probability that a worm spread eventually stops. It is formally defined in section III.

[2]We are making the approximation in the paper that the total number of infected hosts over the entire course of the outbreak equals the number of vulnerable hosts at the beginning. This is a lower bound estimate of the number of vulnerable hosts.

function of this parameter M. To that end, we first describe our branching process model.

3.1. Galton-Watson Branching Process

The Galton-Watson Branching process[3] is a Markov process that models a population in which each individual in generation n independently produces some random number of individuals in generation $n+1$, according to a fixed probability distribution, that does not vary from individual to individual [6, 14].

All infected hosts can be classified into generations in the following manner. The initial infected hosts belong to the $0-th$ generation. All hosts that are directly infected by the initial infected hosts are the 1^{st} generation hosts, regardless of when they are infected. In general, an infected host H_b is an $(n+1)$-th generation host if it is infected *directly* by a host H_a from the n-th generation. H_b is also called an offspring of H_a. All infected hosts form a tree if we draw a link between a host and its offspring. Figure 1 illustrates the notion of generation-wise evolution. In this model, there is no direct relationship between generation and time. A host in a higher generation may precede a host in a lower generation, as host D (generation 2) precedes host B (generation 1) in Figure 1 ($t(D) < t(B)$).

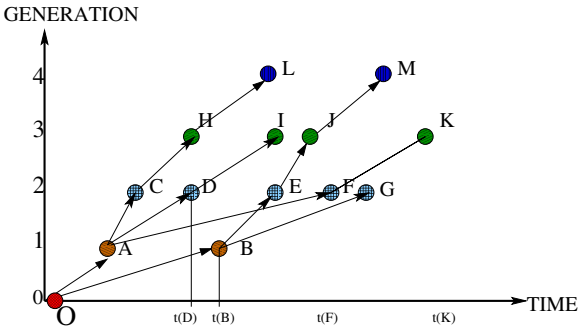

Figure 1. Generation wise evolution in a tree structure, O is the initial infected host. O has two offsprings host A and B.

Figure 2 illustrates the Code Red propagation in the early stage showing the growth of the number of infected hosts of the first 6 generations.

Let ξ be a random variable, representing the offsprings of (i.e., the number of vulnerable hosts infected by) one infected host scanning M times. During the initial phase of the worm propagation, ξ is a $binomial(M,p)$ random variable, where p is the vulnerability density. Hence,

$$P\{\xi = k\} = \binom{M}{k} p^k (1-p)^{M-k}. \quad (2)$$

[3]Branching process models have already been successfully used in modeling the spread of infectious diseases in the early phase of the outbreak.

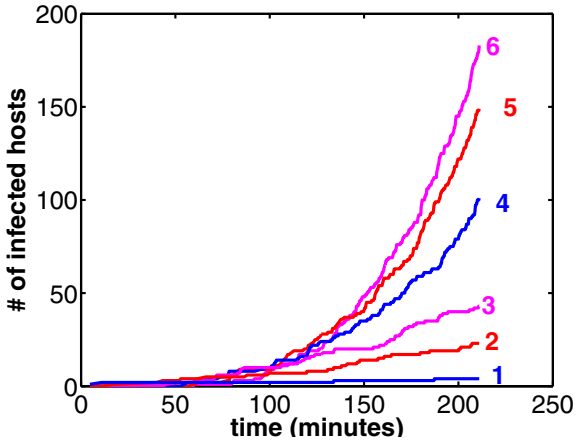

Figure 2. Growth of the infected hosts in generations. The generation number is by the growth curve.

During the early phase of the propagation, the vulnerability density remains constant since the number of infected hosts is much smaller than the number of vulnerable hosts in the population. Let I_n be the number of infected hosts in the n-th generation. I_0 is the number of initial hosts that are infected. During the early phase of the worm propagation, each infected host in the n^{th} generation infects a random number of vulnerable hosts independent of one another according to the same probability distribution. These newly infected hosts are the $(n+1)^{th}$ generation hosts. Let $\xi_k^{(n)}$ denote the number of hosts infected by the k^{th} infected host in the n^{th} generation. The number of infected hosts in the $(n+1)^{th}$ generation can be expressed as $I_{n+1} = \sum_{k=1}^{I_n} \xi_k^{(n)}$, where $\xi_k^{(n)}$ are independent $binomial(M,p)$ random variables.

During the initial worm epidemic, each infected host produces offsprings independently and according to the same probability distribution as in Equation (2). Therefore, the spread of infected hosts in each generation $\{I_n, n \geq 0\}$ forms a branching process. For convenience, we provide a list of the symbols used in our model and their corresponding explanation in Table I.

We next use the branching process model to answer questions on how the worm propagates as a function of the total number of allowable scans M.

Symbol	Explanation
V	The size of the vulnerable hosts
p	$p = \frac{V}{2^{32}}$, the density of the vulnerable hosts
M	The total number of scans by each infected host
ξ	Random number of offsprings generated by each infected host
$\xi_k^{(n)}$	Number of offsprings produced by k^{th} host in n^{th} generation
I_0	Number of initial infected hosts
I_n	Number of n^{th} generation infected hosts
I	Total number of all infected hosts $[I = \sum_{n=0}^{\infty} I_n]$
π	Extinction probability
P_n	Extinction probability at n^{th} generation

Table 1. Notations

3.2. Extinction Probability for Scanning Worms

For simplicity of illustration, we assume that the initial number of infected hosts is 1. The results can be readily generalized to an arbitrary number of initial infected hosts.

Let $\mu = E\xi$ be the mean number of offsprings per infected host. Let π denote the probability that the population dies out eventually, also known as the *extinction probability*. It is defined as

$$\pi = P\{\text{worm dies out}\} = P\{I_n = 0, \text{ for some } n\} \quad (3)$$

In the case of network worms, the extinction probability measures the likelihood of the worm spread dying out after a certain number of generations. When $\pi = 1$, we are certain that the infections from the worm cannot be spread for an arbitrarily large number of generations. The following *Proposition* provides the necessary and sufficient condition for extinction.

Proposition 1 *Let the density of the vulnerable hosts be p and the total number of scans per host be M. Then $\pi = 1$ if and only if $M \leq \frac{1}{p}$.*

Proof: The spread of the worm in its early stage form a branching process, where each infected host independently produces a random number of offsprings. Let ξ be the random variable representing the number of offsprings produced by each infected host. Since the total number of scans per infected host is M, ξ is a Binomial random variable with probability density given by Equation (2) and mean $E(\xi) = Mp$.

According to Theorem 4.5.1 in [14], the extinction probability of a branching process is 1 if and only if the expected number of offsprings from each individual is no more than 1. That is, $\pi = 1$ if and only if $E(\xi) \leq 1$.

Therefore, $\pi = 1$ if and only if $M \leq \frac{1}{p}$. ■

The practical implication of the above proposition is that if we limit the total number of scans per host to be no more than $1/p$, the worm spread will eventually be contained.

Using *Code Red* and *SQL Slammer* as examples, Proposition 1 implies that if the total scans per host is less than 11,930 and 35,791 respectively, the worms would eventually die out. (V=360,000 for Code Red, and V=120,000 for SQL Slammer are used in this calculation.)

This suggests that the limit on the total scans per host may not be restrictivein practice. The value of M corresponds to the number of unique addresses that can be contacted and therefore, the restriction on M is not expected to significantly interfere with normal user activities. This is borne out by actual data for traffic originated by hosts at the Lawrence Berkeley National Laboratory presented in Section 4 and Figure 6. Further, note that this restriction is over a period of time denoted by the *containment cycle*, which could be on the order of weeks or months. We will discuss this further in Section 4.

We can also compute the extinction probability at each generation, denoted by $P_n = P\{I_n = 0\}$. Observe that P_n is non-decreasing in n and can be calculated by using the probability generating function of ξ, defined as $\phi(s) = E[s^{\xi}]$. Since ξ is a binomial random variable with parameters (M, p), it follows from elementary probability theory that $\phi(s) = (ps + (1-p))^M$. Let $\phi_n(s)$ be the probability generating function of I_n, then $\phi_n(s) = \sum_{k=0}^{\infty} s^k P\{I_n = k\}$. Now, clearly, $P_n = Pr\{I_n = 0\} = \phi_n(0)$.

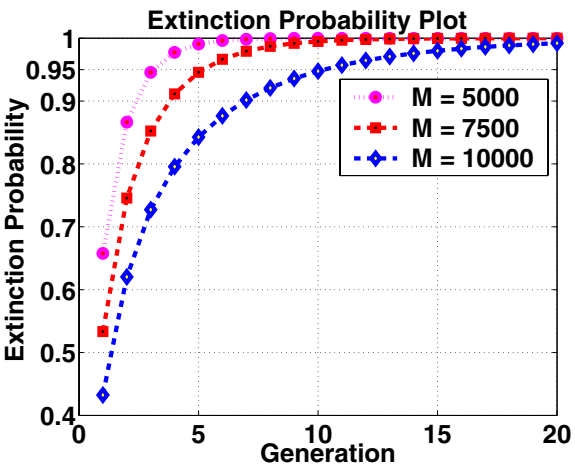

Figure 3. Extinction Probability at each generation for the Code Red worm

It is shown in [6] that $\phi_{n+1}(s) = \phi_n(\phi(s))$ for $n \geq 1$, $\phi_0(s) = s^{I_0}$, and $\phi_1(s) = [\phi(s)]^{I_0}$, where I_0 is the number of initial infected hosts. Using this formula and the fact that

$P_n = \phi_n(0)$, we can calculate the extinction probability at each generation P_n.

In Figure 3, we plot the extinction probability P_n for the Code Red worm for three different values of M. We use a vulnerable host size of 360,000 with a single initial infected host. Since the value of M is smaller than the theoretically computed threshold in all cases, the worm is guaranteed to die out. As shown in the figure, the smaller the value of M, the quicker (in generations) the worm dies out.

3.3. Probability Distribution of Total Number of Infected Hosts

While the probability of extinction gives us a bound on maximum number of allowable scans per host, the true effectiveness of a worm containment strategy is measured by *how many hosts have been infected before the worm finally dies out*. We next provide a probability density function for the the total infections during when the total scans per hosts is below $1/p$.

The total infections, denoted by I, is the sum of the infections in all generations ($I = \sum_{n=0}^{\infty} I_n$). Our objective is to provide a simple closed form equation that accurately characterizes $P\{I = k\}$, the probability that the total number of hosts infected is k, for a given value of M.

We consider any uniform scanning worm with I_0 initial infected hosts. We allow all hosts to scan $M \leq 1/p$ times, where the vulnerability density is p, and the total number of infected hosts is $I = \sum_{n=0}^{\infty} I_n$. As shown earlier, $\{I_n\}$ is a branching process. The infected hosts independently infect a random number of vulnerable hosts that obeys the same probability distribution as ξ. Since the total number of scans per infected host is M, then ξ, is a binomial random variable $B(M, p)$. Further, since p is typically small in practice (e.g., $p \approx 8.5 \times 10^{-4}$ for Code Red), and M is typically large, ξ can be accurately approximated by a Poisson random variable with mean $\lambda = Mp$ [14]. Hence, the probability density function for ξ is given by:

$$P\{\xi = k\} \approx e^{-\lambda} \frac{(\lambda)^k}{k!}.$$

It then follows from [4], that the total progeny of the branching process has a Borel-Tanner distribution, i.e.,

$$P\{I = k\} = \frac{I_0}{k(k - I_0)!} (k\lambda)^{(k-I_0)} e^{-k\lambda}, \; k = I_0, I_0+1, \cdots \tag{4}$$

where $\lambda = Mp$. The mean and variance of I are given by [4]:

$$E(I) = \frac{I_0}{1-\lambda} \quad VAR(I) = \frac{I_0}{(1-\lambda)^3}$$

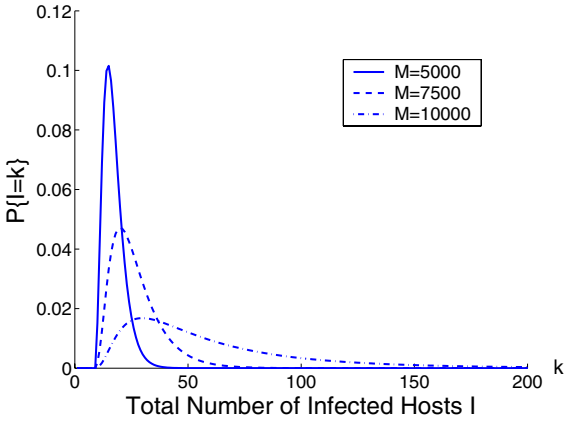

Figure 4. Probability Density of I

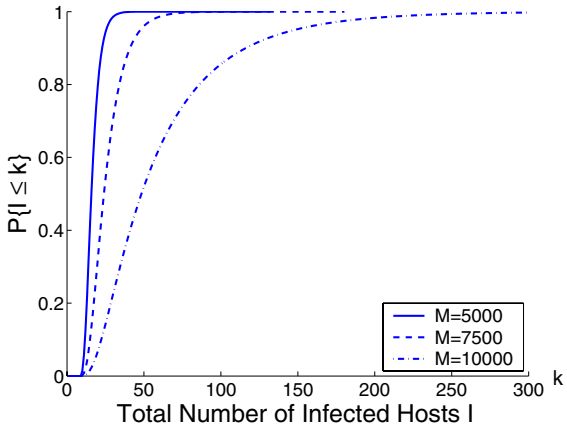

Figure 5. Cumulative Distribution of I

The above probability density function is very useful in calculating statistics of interest. For example, consider Code Red with 10 initial infected hosts, also used in [21]. If $M = 10000$ ($Mp = 0.83$), with probability 0.99 the worm will be contained to less than 360 infected hosts, which is 0.1% of the total vulnerable population.

Figure 4 shows the plot of the probability density function of I for three different values of M for Code Red with 10 initial infections. Figure 5 plots the cumulative probability distribution of I for three different values of M. As we can see from Figure 5, with probability 0.95, Code Red will not spread to more than $150, 50$, and 27 total infected hosts if the values of M are chosen to be $10000, 7500$, and 5000, respectively.

We also consider the SQL Slammer worm with 10 initial infected hosts. If we use the same value for M ($M = 10000$), $P\{I < 20\} > 0.97$. i.e. with high probability, no more than 20 vulnerable hosts (or 10 additional vulnerable hosts) will be infected. This corresponds to 0.008% of the total vulnerable population. If we further reduce M to

5000, $P\{I > 14\} < 0.97$, i.e. with high probability, no more than 4 additional vulnerable hosts will be infected.

Now, we compare this result to existing worm detection systems [10], which provide detection when approximately 0.03% (Code Red) and 0.005% (slammer) of the susceptible hosts are infected. This performance is achieved by careful selection of the routers at which worm detection mechanisms are put in place. With our scheme, when M is kept below a pre-defined threshold, with very high probability, the infection will not be allowed to spread that widely. Further, our results also hold for slow worms, which most other detection techniques have trouble detecting.

Based on our results, we now develop an automatic worm containment system that can inhibit the spread of the worms.

4. Automated Worm Containment System

The results in Section 3 provide us with a blueprint of a *host based* worm containment strategy. The containment system is based on the idea of restricting the total number of scans to unique IP addresses by any host. We assume that we can estimate or bound the percentage of infected hosts in our system. Our proposed automated worm containment strategy has the following steps.

1. Let M be the total number of unique IP addresses (scans) that a host can contact in a *containment cycle*. At the beginning of each new containment cycle, set a counter that counts the number of unique IP addresses for each host to be zero.

2. Increment this counter for each host when it scans a new IP address.

3. If a host reaches its scan limit before the end of the containment cycle it is removed and goes through a heavy duty checking process to ensure that it is free of infection before being allowed back into the system. When allowed back into the system, its counter is reset to zero.

4. Hosts are thoroughly checked for infection at the end of a containment cycle (one by one to limit the disruption to the network) and their counters reset to zero.

Choose M based on the probability that the total number of infected hosts given by Equation (4) is less than some acceptable value ϵ. Further, the containment cycle would be obtained through a learning process. Initially choose a containment cycle of a fixed but relatively long duration, e.g., a month. Since the value of M that we can allow is fairly large (on the order of thousands, as indicated by analysis with SQL Slammer and Code Red) we don't expect that normal hosts will be impeded by such a restriction. We can then increase (reduce) the duration of the containment cycle depending on the observed activity of scans by correctly operating hosts.

We use the 30 day trace of wide-area TCP connections (LBL-CONN-7) [24] originating from 1645 hosts in the Lawrence Berkeley Laboratory to analyze the growth of the number of unique destination IP addresses per host (this is clean data over a period when there was no known worm traffic in the network). Our study indicates that 97% of hosts contacted less than 100 distinct destination IP addresses during this period. Only six hosts contacted more than 1000 distinct IP addresses and the most active host has contacted approximately 4000 unique IP addresses. Figure 6 shows the growth trend of the total unique destination IP addresses for these six most active hosts. If our containment system is used with the containment cycle to be one month and M is set to be 5000, none of the above hosts will trigger alarm. As shown in section 3, with high probability the total infections caused by Code Red will be under 27 hosts when $M = 5000$. This suggests that our containment system is not likely to interfere significantly with normal traffic, yet containing the spread of the worms.

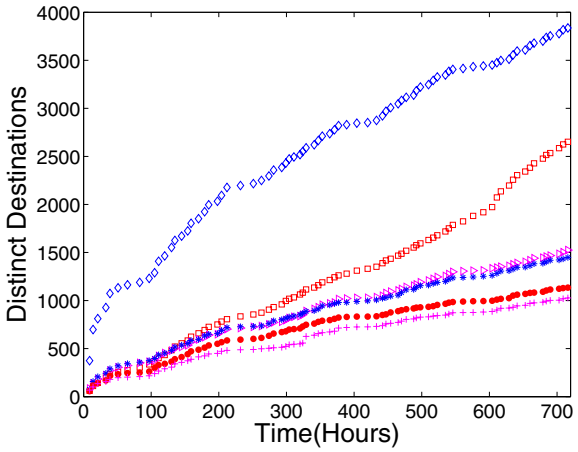

Figure 6. Number of Distinct IP contacted over 30 days for the six most active hosts

The containment cycle can also be adaptive and dependent on the scanning rate of a host. If the number of scans originating from a host is getting close to the threshold, say it reaches a certain fraction f of the threshold, then the host goes through a complete checking process. The advantage of this worm containment system is that it does not depend on diagnosis of infected hosts over small time-granularities. It is also effective in preventing an Internet scale worm outbreak because the total number of infected hosts is extremely low, as shown in the examples in the pre-

vious section.

Traditional rate based techniques attempt to limit the spread of worms by limiting the scanning rate. The limit imposed must be carefully tuned so as not to interfere with normal traffic. For example, the *rate throttling* technique [17] limits the host scan rate to 1 scan per second. The rate limiter can inhibit the spread of fast worms without interfering with normal user activities. However, slow scanning worms with scanning rate below 1 Hz and stealth worms that may turn themselves off at times will elude detection and spread slowly.

Our worm containment system, on the contrary, can contain fast worms, slow worms, and stealth worms. The fast scanning worms will reach the limit on M sooner, while the slow worms will reach this limit after a longer period of time. As long as the host is disinfected before the threshold is reached, the worm cannot spread in the Internet.

A comparison with network based containment systems is also valid. In such systems, e.g., DIB:S/TRAFEN [23], the mechanism has to be implemented on a carefully chosen set of network routers. Such mechanisms have been shown to be effective in detecting the worm spread at small levels of infection, e.g., 0.03% for Code Red and 0.005% for SQL/Slammer [10]. However, our scheme has the advantage that it is host based and therefore easier to deploy, while showing comparable or better containment in terms of fraction of susceptible hosts infected. In our scheme, if end hosts are unwilling or unable to deploy the scan limiting policy, it can also be done at edge routers by aggregating the total number of allowable scans for all the hosts connected to the router. This however provides less fine-grained control on the traffic.

We next provide numerical results to illustrate the effectiveness of our model and containment strategy.

5. Simulation Results

We use a discrete event simulator to simulate a uniform scanning worm with our defense strategy. In our simulator, each vulnerable host is assigned a IPv4 address randomly, and each will be in one of the three states: *susceptible, infected*, and *removed*. A host is in *removed* state if it has sent M scans. The *infected* hosts independently generate random IP addresses to find the victims. If the random IP address matches any of the IP addresses of the hosts in the *susceptible* state, the susceptible host will become infected.

In our simulation for Code Red, we used $V = 360,000$ for the vulnerable population size and $I_0 = 10$ for the number of initial infected hosts. We used $M = 10,000$ that is below the threshold required for worm extinction. In this case, $p = 0.83 \times 10^{-5}$ and $\lambda = Mp = 0.83$.

As discussed earlier, the total number of infected hosts I measures how well a worm is contained. We ran this simulation with $M = 10,000$ for a 1000 times and collected the values of I. Figure 7 shows the relative frequency of I from our simulations and the probability density function of I obtained from our theoretical results in Section 3. Figure 8 shows the relative *cumulative* frequency of I from our simulations and the *cumulative density function* of I from the theoretical analysis.

The simulation results validate the accuracy of our model and the effectiveness of our containment strategy. Figures 7 and 8 demonstrate that our simulation results match closely with the theoretical results from Section 3. We can see from Figure 8 that with high probability (0.95), the total number of infected hosts is held below 150 hosts. As mentioned in Section 3, one can reduce the spread of infection by further reducing the value of M.

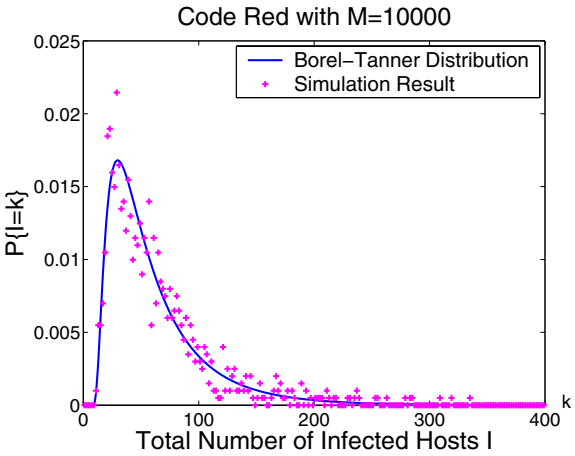

Figure 7. Probability density for total number of infected hosts

Figures 9 and 10 show two sample paths of the Code Red propagation when our containment strategy is utilized. We used a scan rate of 6 scans/second for Code Red for the purpose of illustrating worm propagation and containment with respect to time. This scan rate is taken from the empirical study in [11]. In one scenario depicted in Figure 9, there are a total of approximately 300 hosts infected. However, the active number of infected hosts (number infected - number removed) is held below 30 at all times. This is due to our countermeasure that when a host scans M times, it is removed. The worm ceased spreading after all infected hosts were removed. Figure 10 shows another scenario when there are 55 total infected hosts. In this scenario, the removal process quickly catches the infection process, so that the worm dies out rapidly. Using formula provided in section 3, when $M = 10000$ ($\lambda = 0.83$), E(I) = 58 and var(I) = 2035 (std=45). The variation is large and can be significant in the early stage, which will have significant effect in modeling the latter stage of growth of the

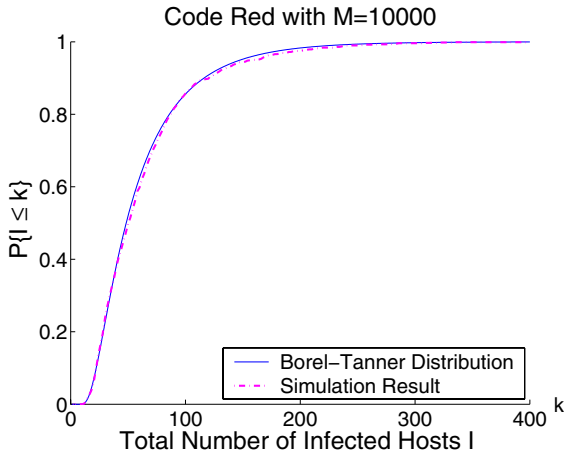

Figure 8. Cumulative Distribution of total number of infected hosts

worm. Stochastic models need to be used to capture this variation.

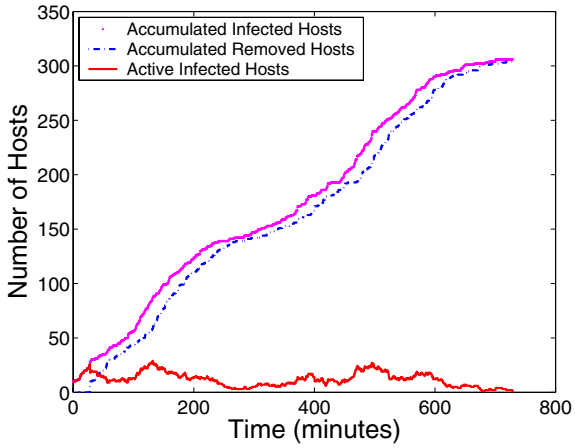

Figure 9. A Sample Path (Code Red)

We also ran our simulations with SQL Slammer parameters. Here we used V=120,000 as used in [10], $I_0 = 10$, and $M = 10,000$. The experiment with Slammer show close match between predicted and observed values and this is borne out by Figures 11 and 12. The worm containment scheme contains the infection to below 20 hosts (i.e., only 10 newly infected hosts) with a very high probability.

6. Conclusion

In this paper, we have studied the problem of combating Internet worms. To that end, we have developed a branching process model to characterize the propagation of Internet worms. Unlike deterministic epidemic models studied

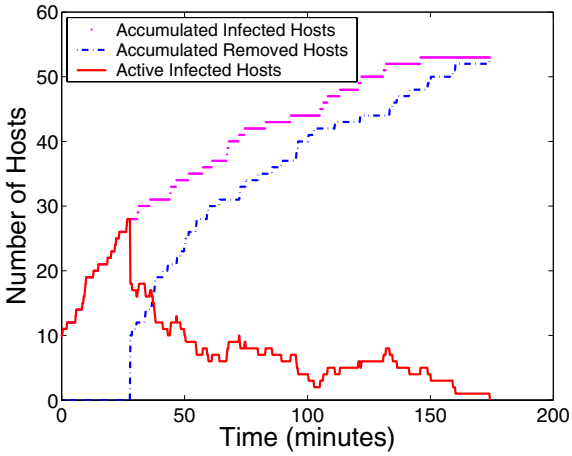

Figure 10. A Sample Path (Code Red)

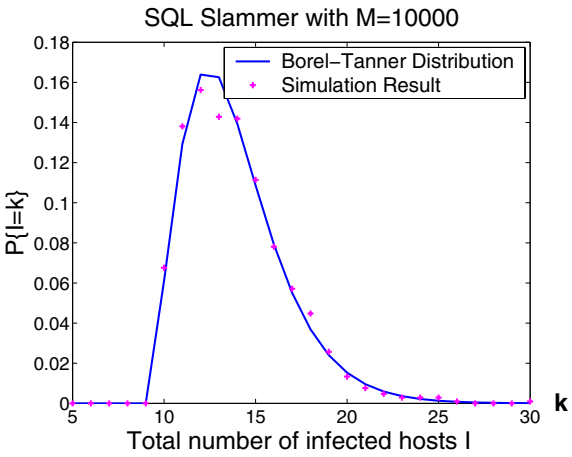

Figure 11. Probability density for total number of infected hosts

in the literature, this model allows us to characterize the early phase of worm propagation. Using the branching process model we are able to provide a precise bound M on the total number of scans that determines whether the worm will eventually die out. Further, from our model we also obtain the probability that the total number of hosts that the worm infects is below a certain level, as a function of the bound M. The insights gained from analyzing this model also allows us to develop an effective and automatic worm containment strategy. The accuracy of our model is higher when the spread of the worm is smaller. This is why we develop a defense mechanism that does not allow the spread of the worm beyond a certain level. Our strategy can effectively contain both fast scan worms and slow scan worms *without knowing the worm signature in advance or needing to explicitly detect the worm*. We show via simulations and

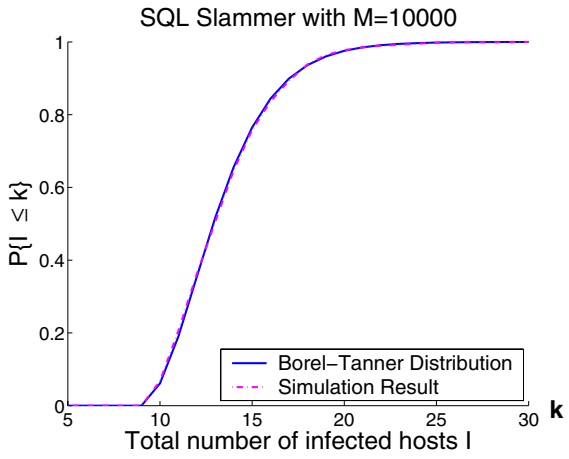

Figure 12. Cumulative distribution for total number of infected hosts

real trace data that the containment strategy is both effective and non-intrusive.

The focus of this paper has been on modeling and containment of uniform scan worms. The containment scheme may require universal deployment to be most effective. An important problem to consider is how to extend such a strategy to model and combat preferential scanning worms and allow for incremental deployment. We will focus on these areas in our future work.

References

[1] CAIDA, "CAIDA Analysis of Code-Red,," *http://www.caida.org/analysis/security/code-red/*.

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

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

[4] P. C. Consul, "Generalized Poisson Distributions, Properties and Applications," *STATISTICS: textboods and monographs*, vol 99, *Marcel Dekker, Inc*.

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

[6] S. Karlin and H. M. Taylor, "A First Course in Stochastic Processes, Second Edition," *Academic Press*, 1975.

[7] J.O. Kephart and S. R. White, "Directed-graph Epidemiological Models of Computer Viruses," *Proceedings of the IEEE Symposium on Security and Privacy*, 1991.

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

[9] J.O. Kephart and S. R. White, "Measuring and Modeling Computer Virus Prevalence," *Proceedings of the IEEE Symposium on Security and Privacy*, 1993.

[10] M. Liljenstam, D. M. Nicol, V. H. Berk, and R. S. Gray, "Simulating Realistic Network Worm Traffic for Worm Warning System Design and Testing," *WORM'03*, pp.24–33, Oct. 2003.

[11] D. Moore, C. Shannon, and J. Brown, "Code-Red: A Case Study on The Spread and Victims of an Internet Worm," *Proc. 2nd ACM Internet Measurement Workshop*, ACM Press, 2002.

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

[13] D. Moore, C. Shannon, G. M. Voelker and S. Savage, "Internet Quarantine: Requirements for Containing Self-Propagating Code," *Proc. INFOCOM 2003*.

[14] S. Ross "Stochastic Processes, 2nd edition" *John Wiley & Sons, Inc*, 1996.

[15] S. Staniford, V. Paxson, and N. Weaver, "How to Own the Internet in Your Spare Time," *Proc. 11th USENIX Security Symposium* (SEC 02), Usenix Assoc., 2002, pp. 149–167.

[16] S. Staniford, "Containment of Scanning Worms in Enterprise Networks," *Journal of Computer Security*, to appear

[17] M. M. Williamson "Throttling Viruses: Restricting Propagation to Defeat Malicious Mobile Code," *Proceedings of ACSAC Security Conference*, 2002.

[18] C. Wong, C. Wang, D. Song, S. Bielski, G. R.Ganger "Dynamic Quarantine of Internet Worms," *The International Conference on Dependable Systems and Networks (DSN-2004)*, 2004.

[19] C. C. Zou, W. Gong and D. Towsley, "Code Red Worm Propagation Modeling and Analysis," *In 9th ACM Symposium on Computer and Communication Security*, pp. 138–147, 2002.

[20] 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*, pp.190–199, 2003.

[21] C. C. Zou, W. Gong, and D. Towsley, "Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense," *WORM'03*, pp. 51–60, 2003.

[22] Computer Economics "2001 Economic Impact of Malicious Code Attacks," *http://www.computereconomics.com/cei/press/pr92101.html*.

[23] Vincent H. Berk, Robert S. Gray, and George Bakos, "Using Sensor Networks and Data Fusion for Early Detection of Active Worms," *In Proceedings of AeroSense 2003: SPIE's 17th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls,* Orlando, Florida, April 2003.

[24] LBL-CONN-7 "Thirty Days' Wide-Area TCP Connections," *http://ita.ee.lbl.gov/html/contrib/LBL-CONN-7.html*.

[25] H. W. Hethcote, "The Mathematics of Infectious Diseases," *In SIAM Review, vol. 42, no. 4, pp. 599-653,* 2000.