Peer-to-Peer System-based Active Worm Attacks: Modeling, Analysis and Defense

Wei Yu, Sriram Chellappan, Xun Wang and Dong Xuan [*]

Abstract

Active worms continue to pose major threats to the security of today's Internet. This is due to the ability of active worms to automatically propagate themselves and compromise hosts in the Internet. Due to the recent surge of Peer-to-Peer (P2P) systems with large numbers of users and rich connectivity, P2P systems can be a potential vehicle for the attacker to achieve rapid worm propagation in the Internet. In this paper, we tackle this issue by modeling and analyzing active worm propagation on top of P2P systems, and designing effective defense strategies within P2P systems to suppress worm propagation. In particular; 1) we define two P2P-based active worm attack models: an offline P2P-based hit-list attack model and an online P2P-based attack model; 2) we conduct a detailed analysis on the impacts of worm propagation on top of P2P-based systems, and study the sensitivity of worm propagation to various P2P system and attack related parameters; 3) finally, we propose defense strategies within the P2P system to combat worms. Based on extensive numerical analysis and simulation data, we demonstrate that P2P-based active worm attacks can significantly enhance worm propagation, and important P2P related parameters (system size, topology degree, host vulnerability etc.) have significant impacts on worm spread. We also find that our proposed defense strategies can effectively combat worms by rapidly detecting and immunizing infected hosts.

Keywords: P2P systems, Active worm attacks, Internet security.

1 Introduction

Active worms pose major threats to the security of today's Internet. Being self-propagating and self-replicating, active worms can propagate in an automated fashion without human intervention from infected hosts to other

[*] Wei Yu (corresponding author) is with the Department of Computer Science, Texas A&M University, College Station, TX 77843. E-mail: weiyu@cs.tamu.edu. Sriram Chellappan is the Department of Computer Science at Missouri University of Science and Technology, Rolla, MO 65409. Email: chellaps@mst.edu. Xun Wang and Dong Xuan are with the Department of Computer Science and Engineering, The Ohio-State University, Columbus, OH 43210. E-mail: {wangxu, xuan}@cse.ohio-state.edu.

vulnerable hosts in a network. Propagation of active worms in the Internet enables one to control thousands of hosts, launch distributed denial of service attacks, access confidential information, destroy valuable data etc [1, 2, 3, 4]. Due to the recent surge of many popular peer-to-peer (P2P) systems with a large number of users and connectivity, P2P systems can potentially be a powerful vehicle for attackers to launch active worms and achieve rapid worm propagation in the Internet. In this paper, we tackle this issue by modeling and analyzing active worm propagation on top of P2P systems, and designing effective defense strategies within the P2P system to suppress worm propagation.

Active worms have been persistent threats to the Internet, especially during the last several years. In 2001, the Code-Red worm infected over $350,000$ hosts in less than 14 hours and caused more than $\$1.2$ billion in economic damages in the first 10 days of its propagation [5]. A host of other worms have also been reported in the past [6, 7, 8]. Traditional worms predominantly adopt the random-based scan approach to propagate, where once a host is infected, the worm randomly scans the Internet IP address space to find new vulnerable hosts. The downside in this strategy is that, since many IP addresses in the Internet are not being used by any valid host, propagation of worms is relatively slow. A more powerful worm attack strategy is the hit-list strategy, which collects a list of IP addresses prior to the attack to improve success rate of infection [9]. The downside of this strategy is in the generation of the hit-list. Systems in well-known IP addresses are more secured these days, making the generation of an effective hit-list non-trivial.

P2P computing has been becoming an active area for Internet-scale resource sharing and cooperation. The recent surge of P2P applications can be observed by following statistical data collected on Nov 3, 2004: there are a total of $2,256,612$ users in the FastTrack P2P system, $2,401,835$ users in the eDonkey P2P system, $1,258,775$ users in the Warez P2P system, and $600,926$ users in the Gnutella P2P system [10]. These numbers are still increasing. We believe that P2P systems, being highly popular can become a powerful vehicle for attackers to rapidly propagate worms in the Internet. Some incidents in the recent past have indicated this. Examples are the Igloo and MyDoom worms that spread across KaZaA P2P system through P2P file sharing [11, 12]. P2P-based worm propagation can be one of the best facilitators for Internet worm attacks due to the following reasons; 1) compromising P2P systems with a large number of registered hosts can rapidly accelerate Internet worm propagation, as hosts in P2P systems are real and active; 2) since hosts in P2P systems maintain a certain number of neighbors for routing purposes, worm infected hosts in the P2P system can easily propagate the worm to their neighbors, which continue the worm propagation to other hosts and so on; 3) some hosts in P2P systems may have insecure network and system environments (e.g., home networks); 4) P2P hosts install various application specific software, and any vulnerability in such software can enhance their risk of being infected.

Our goal in this paper is to quantitatively understand the impacts of worm propagation on top of P2P systems, and design defenses within the P2P system to combat worm propagation. The highlights of this paper are:

1. We formally define two P2P-based worm attack models: an offline P2P-based hit-list attack model, and an online P2P-based attack model. We identify the important P2P system-related and attack-related parameters in the modeling of the attacks.

2. We conduct a detailed analysis to analyze the performance of the attack models (in terms of the number of hosts infected), and the impacts of various P2P system-related and attack-related parameters on attack effects. Our analysis and experimental data demonstrate that P2P-based worm attacks can significantly worsen attack effects. The parameters including P2P size, topology degree, host vulnerability etc. have important impacts on attack effects. We also observe that attack effects are more pronounced in the case of unstructured P2P systems compared to structured P2P systems.

3. We design and evaluate defense strategies within the P2P system to combat worms. Our strategy consists of P2P hosts performing two tasks: worm detection and rapid immunization. To detect worms, we incorporate the methodologies of trend-based and threshold-based worm detection schemes. We prove that P2P-based worm attacks have clearly identifiable exponential propagation trends, which enables rapid and accurate worm detection within P2P systems. For immunization, we incorporate the methodology of active immunization-based schemes, and analyze its effectiveness in rapidly immunizing hosts in P2P systems. Our experimental data clearly demonstrate the effectiveness of our defense strategies in rapidly detecting worm attacks and reducing the number of infected hosts. We observe that the trend-based scheme performs favorably compared to the threshold-based scheme in terms of both detection time and detection accuracy. We also observe that active immunization-based schemes can rapidly suppress worm propagation and contain their spread.

The rest of the paper is organized as follows. In Section 2, we define our P2P-based worm attack models, and identify the important modeling parameters. In Section 3, we formally analyze the P2P-based worm attack models. In Section 4, we present our defense strategies within the P2P system to combat worm attacks. In Section 5, we report results of performance evaluations. Related work is discussed in Section 6. Lastly, we conclude our paper with some final remarks in Section 7.

2 Modeling P2P-based Active Worm Attacks

2.1 P2P-based Attack Models

An active worm is a program that propagates across hosts in a network by exploiting their security flaws. Active worms are similar to biological viruses in their self-replicating and propagating behavior. In general, there are two stages in an active worm attack: 1) scanning the network to select victim hosts; 2) infecting the victim after discovering its vulnerability. Infected hosts further propagate the worm to other vulnerable victims and so on. In the above two stages there are three key factors that decide worm propagation speed: 1) how fast the worm can scan other hosts in the network; 2) the probability of the worm to scan a real host; and 3) vulnerability of the scanned host. The first factor is modeled as the scan rate S, which is the number of hosts per unit time that a worm infected host can scan. The scan rate S is a property of the worm itself, and is independent of the victims it attacks. However, the second and third factors are victim dependent. We remark that not all addresses in the Internet are

Table 1. Notations in this paper

Notation	Definition
T	Total IP addresses in the Internet
S	Scan rate (number of hosts per unit time that the worm can simultaneously scan)
M_0	Initial number of infected hosts
P_1	Probability of IP addresses used by hosts
P_2	Probability of real host in the Internet which is vulnerable to worm infection
u	Probability of a P2P host being up (up-time/total attack time) with probability $\in [0, 1]$
m	Total number of P2P hosts m (including hosts joining and leaving system). The effective P2P system size is $m * u$
θ	Mean topology degree of the structured P2P system
σ	Power law degree of the unstructured P2P system
r_j	Topology degree of host j in the P2P system
ω	Mean value of topology degree for the unstructured P2P system with power law distribution
P_3	Probability of hosts in the P2P system to be vulnerable to worm infection
P_d	The probability of defense host to successfully generate alarms
P_v	Probability of a P2P volunteer itself to take part in worm defense
P_s	The anti-worm immunization successful probability
g	The defense region size
L	The immunization rate for the anti-worm defense
H_1	The threshold value for the threshold-based worm detection scheme
$M(i)$	Number of infected hosts at step i in the Internet ($M(0) = M_0$)
$N(i)$	Number of vulnerable hosts at step i in the Internet ($N(0) = T * P_1 * P_2$)
$E(i)$	Number of newly infected hosts added at step i in the Internet ($E(0) = 0$)
$M^n(i), M^s(i)$	$M^n(i)$ and $M^s(i)$ are no. of infected hosts in *Non*-P2P system and in *Super*-P2P system at step i, respectively
$N^n(i), N^s(i)$	$N^n(i)$ and $N^s(i)$ are no. of vulnerable hosts in *Non*-P2P system and in *Super*-P2P system at step i, respectively
$E^n(i), E^s(i)$	$E^n(i)$ and $E^s(i)$ are no. of newly infected hosts in *Non*-P2P system and in *Super*-P2P system at step i, respectively

used. Recent studies have shown that only 24 % of addresses in the Internet space are used by active hosts [13]. Thus, a significant number of scans launched by worm actually hit many such non-existent hosts. Nevertheless, when propagating on P2P systems, scans can be more accurate, since P2P systems have a large number of real and active hosts with rich connectivity to other P2P hosts. The third factor, namely vulnerability of victim hosts is quite high in the case of P2P systems as most P2P hosts are untrusted and unvalidated during the entry into the P2P system. The last two factors are the reasons, why the threat of worm propagation on P2P systems attains significance.

In the following, we present our P2P-based worm attack models. We first present a baseline attack model namely Pure Random Scan (PRS), where the worm randomly scans the network to identify victims. We then present two P2P-based attack models that propagate on P2P systems to achieve very rapid propagation.

Pure Random Scan: A basic attack model adopted by many worms during victim selection, is pure random scan [7, 14]. In this model, worm infected hosts do not have any prior vulnerability knowledge or active/inactive information of other hosts. The worm infected host randomly selects IP addresses of victims from the global Internet IP address space and launches the attack to those addresses. When the new host is infected, it continuously attacks the Internet via the same method. The main shortcoming in this approach is that many IP addresses in the Internet are not being used by any valid host. Thus, many scans are wasted when targeting non-existing hosts. In the following, we present two representative and highly effective P2P system-based active worm attack models.

Offline P2P-based Hit-list Scan (OPHLS): In this model, the large population of users in P2P systems is the first target for the attacker. In this model, the attacker collects IP address information of the P2P system offline. We denote this as the hit-list of the attacker. Obtaining the hit-list can be achieved by various methods, such as using P2P-based Crawler tools [15]. In this attack model, there are two phases: in the first phase (called the P2P system attack phase), all newly infected hosts continuously attack the hit-list until all hosts in the hit-list have been scanned. In the second phase, all infected hosts continue to attack the Internet via PRS. Algorithm 1 describes the OPHLS attack model.

Algorithm 1 OPHLS - Offline P2P-based Hit-list Scan

```
Require: node $i$ is the worm infected host in the P2P system with scan rate $S$, and hit-list $L_p$
while $L_p$ is not empty do
    Select a set $V$ consisted of $S$ victims from $L_p$ and launch the attack to all victims in $V$
    $L_p = L_p - V$
end while
Attack the rest of the Internet via Pure Random Scan
```

Online P2P-based Scan (OPS): In this model, the rich connectivity of P2P systems will be utilized by worms during propagation. After a worm infected host joins the P2P system, the host immediately launches the attack on its P2P neighbors as a high priority. We illustrate this with an example. Let A_1 be a worm infected host in the P2P system with scan rate $S = 5$ (i.e., the worm can scan 5 hosts per unit time). Let A_1 have three P2P neighbors B_1,

B_2, and B_3. In the OPS model, A_1 will attack B_1, B_2, and B_3 (using 60% of its attack capability), and attack the rest of the Internet with its remaining capability (40%) via PRS. Assuming that B_2 and B_3 are vulnerable hosts and are infected, these two hosts will continuously attack their P2P neighbors and the Internet via PRS in a similar manner. At this point, since A_1 has no new P2P hosts to scan, it will use 100% of its attack capability to attack the Internet via PRS. All infected hosts will follow this rule to propagate the worm further. Algorithm 2 describes the OPS attack model.

Active worms exploit connectivity in a network to self-propagate. P2P systems in the Internet have large number of users, rich connectivity, and host vulnerability. In both our P2P-based attack models, worms exploit these effectively. This translates to rapid worm propagation and infection in our attack models (which we quantitatively analyze subsequently), thus highlighting the threats posed by P2P system-based worm attacks.

Algorithm 2 OPS - Online P2P-based Scan

```
Require: node $i$ be the worm infected host in P2P system with scan rate $S$
 1: Find all P2P neighbors of node $i$. Let $q$ neighbors of node $i$ be $G = \{h_1, h_2, \ldots, h_q\}$
 2: while $G$ is not empty do
 3:   if $S \geq q$ then
 4:     Scan and launch attacks on the $q$ P2P neighbors
 5:     Use the $S - q$ scan capability to attack the Internet via Pure Random Scan
 6:     $G = NULL$
 7:   else
 8:     Scan $S$ neighbors in $G$, i.e., $h_1, h_2, \ldots, h_S$
 9:     $G = G - \{h_1, h_2, \ldots, h_S\}$
10:   end if
11: end while
12: Attack the rest of the Internet via Pure Random Scan
```

Note that there are two types of P2P systems: structured and unstructured. The key difference between these two types of systems is the connectivity of hosts in the systems. In the OPHLS model, since the attacker predetermines the hit-list before attacks, Algorithm 1 is the same in both types of systems. In the OPS model, the number of P2P neighbors that the worm can scan depends on the connectivity of the P2P system, which is different in structured and unstructured systems. This difference is observed in the OPS model (Algorithm 2) at step 1. The result q (number of neighbors) returned after execution of step 1 is different for structured and unstructured systems. However, irrespective of number of neighbors, the basic work-flow of Algorithm 2 remains the same for both structured and unstructured systems. We will highlight this issue further in the next sub-section.

2.2 Model Parameters

The parameters in modeling worm propagation on P2P systems can be broadly classified into two types: attack related and P2P system related. The critical attack related parameters are the scan rate (S) defined in Section 2.1, and the initial number of hosts that are infected (denoted as $M(0)$) prior to the spread of the worm. Intuitively, larger values of S and $M(0)$ mean more potent attacks.

The P2P-based attack models defined above exploit features of P2P systems. In the following, we identify three critical P2P system parameters that are related to the propagation of worms in our attack models.

1) P2P system size: We define the system size as the total number of users in the P2P system. While there can be multiple P2P systems in reality, for analysis purposes, we assume that there is one *Super*-P2P systems that theoretically encompasses all P2P hosts in the Internet. The size of the *Super*-P2P system is denoted as m. The remaining hosts in the Internet are considered to be a part of the *Non*-P2P system.

2) P2P Structured/ Unstructured topology: There are two types of P2P systems depending on their topology: structured P2P and unstructured P2P systems. Structured P2P systems such as CAN [16], Chord [17] are those in which nodes organize themselves in an orderly topology, while unstructured P2P systems, such as Gnutella [18], Freenet [19] are the ones in which nodes organize themselves randomly. In structured P2P systems (where the topology is fixed), all P2P nodes maintain the similar number of neighbors (denoted as topology degree) for efficient routing. For example, a host in d-dimensional CAN maintains $2 * d$ neighbors [16]. Conversely, in unstructured P2P systems like Gnutella [18], Freenet [19] (that are randomly organized) the number of neighbors is not equal among the hosts (i.e., the topology degree is different). Routing here is mostly through breadth-first or depth-first searches.

The topology degree has an important impact on the performance of the online P2P-based (OPS) worm attack model. We denote average topology degree of structured P2P systems as θ. For unstructured P2P systems, the topology degree for each P2P host is quite different. We model the topology degree for unstructured P2P systems as follows. A nontrivial development related to complex networks discovered that for most large networks, including the Internet, metabolic, protein networks, social networks and email systems (whose topologies are random), the distribution of host topology degree follows the power-law distribution. Based on previous studies in this realm [18, 19, 20], we model the topology of the unstructured P2P systems using the power law distribution. In power law theory, the spread in the number of edges of diverse network hosts is characterized by the degree distribution $P(k)$, which gives the probability that a randomly selected host has exactly k edges. We consider the distribution as follows [21]:

$$P(k) = C_1 \frac{\omega}{k^{\sigma}}, \tag{1}$$

where ω is the mean value of topology degree, C_1 is a constant for a given ω [21], and $\sigma \in [1, 8]$ denotes the power law degree.

3) P2P host vulnerability: Hosts in P2P systems are real computers, and can be located in less protected environments (homes, schools, public locations etc). P2P hosts install various P2P application specific software, and any vulnerability in such P2P software can enhance their risk of being infected. Some software are in fact bundled with Spyware, further increasing the chances of backdoor entry into P2P hosts further increasing their vulnerability. In fact, the likelihood of having an exploitable vulnerability of P2P hosts is very high, as demonstrated by recent studies. Examples include the Igloo and MyDoom worms that were part of application software [11, 12]; the security leak in Emule (a part of eDonkey software) that permitted a remote attacker to execute arbitrary code on the victim machines [22] etc. In this paper, we model vulnerability using the notion of probability. Specifically, we denote P_3 as the probability that a host in the *Super*-P2P system is vulnerable to worm infection, and P_2 as the probability that a host in the *Non*-P2P system is vulnerable to worm infection.

To enhance readability, we add Table 1 listing all parameters and notations in this paper.

3 Analyzing P2P-based Active Worm Attacks

3.1 Preliminaries

Before we describe our analysis, we state the assumptions made. While there are multiple P2P systems in reality, for analysis purposes, we assume that there is one *Super*-P2P system that theoretically encompasses all P2P hosts in the Internet. All other hosts in the Internet are considered to be in the *Non*-P2P system. Initially, there are a certain number of infected hosts in the *Super*-P2P system (denoted by $M^s(0)$), and a certain number of infected hosts in the *Non*-P2P system (denoted by $M^n(0)$). For characterizing worm spread, we adopt the epidemic dynamic model for disease propagation for our analysis [23], which assumes that each host is in one of the following states: immune, vulnerable or infected. Modeling active worm propagation using the epidemic dynamic model has been done in [7, 14, 24]. In this model, an *immune* host is one that cannot be infected by a worm. A *vulnerable* host becomes an *infected* host after being successfully infected by a worm. In order to model the dynamics of both offline and online P2P-based active worm attacks, we enhance the traditional epidemic modeling via an average case and discrete recursive analysis to approximate P2P-based worm propagation, similar to the methods used in [25, 26].

In the following, we present our analysis on P2P-based worm propagation. Our basic metric is the number of newly infected hosts at each step i, denoted by $E(i)$ under each attack model. Our rationale to obtain $E(i)$ is to derive it from the number of existing uninfected vulnerable hosts $N(i-1)$ at step $i-1$, the number of existing infected hosts $M(i-1)$ at step $i-1$, and the propagation method of the worms in our attack models. Note that $E^s(i)$ denotes the number of newly infected hosts in the *Super*-P2P system and $E^n(i)$ denotes the number of newly infected hosts in the *Non*-P2P system at step i.

3.2 Analyzing Offline P2P-based Hit-list Attack Model

In the OPHLS attack model, the attacker first collects the IP addresses of the hosts in the *Super*-P2P system, and builds a hit-list of potential victims. Recall that the worm attack in OPHLS includes two stages. In the first stage, all the attack scans are performed on the *Super*-P2P system. When all the hosts in the hit-list have been scanned, the attack enters the second stage and all worm infected hosts start to attack the *Non*-P2P system using pure random scan approach. Theorem 1 gives the number of newly infected hosts as a result of OPHLS attack model. Please refer to Appendix A for detailed proof.

Theorem 1 *In the OPHLS attack model, with $M^s(i)$ and $N^s(i)$ at step i in the* Super-*P2P system, the next tick will have*

$$E^s(i+1) = \begin{cases} N^s(i)[1-(1-\frac{1}{m*u})^{S*M^s(i)}], & M^s(i) \leq m*u*P_3; \\ 0, & M^s(i) > m*u*P_3; \end{cases} \tag{2}$$

newly infected hosts, where $M^s(0) = M_0, N^s(0) = m*u*P_3$.

With $M^n(i)$ and $N^n(i)$ at the step i in the Non-*P2P system, the next tick will have*

$$E^n(i+1) = \begin{cases} 0, & M^s(i) \leq m*u*P_3; \\ N^n(i)[1-(1-\frac{1}{T})^{(S*M^n(i)+S*M^s(K))}] & M^s(i) > m*u*P_3; \end{cases} \tag{3}$$

newly infected hosts, where $M^n(0) = 0, N^n(K) = T*P_1*P_2 - m*u*P_3$,$M^n(K) = M^s(K-1)$, ($K = \min(i)$ for all i satisfying $M^s(i) > m*u*P_3$).

With Theorem 1, we can determine the number of infected hosts in both the *Super*-P2P system and the *Non*-P2P system at step i ($M(i)$), and the total number of vulnerable hosts in both *Super*-P2P system and the *Non*-P2P system at step i ($N(i)$). We have the following recursive formulas:

$$\begin{aligned} M^n(i+1) &= M^n(i) + E^n(i+1), \quad & N^n(i+1) &= N^n(i) - E^n(i+1), \\ M^s(i+1) &= M^s(i) + E^s(i+1), \quad & N^s(i+1) &= N^s(i) - E^s(i+1), \\ M(i) &= M^s(i) + M^n(i), \quad & N(i) &= N^s(i) + N^n(i). \end{aligned} \tag{4}$$

From the above discussions, we make the following observations. From Formulas (2), (3) and (4), we see that as effective system size increases (an increase in m or u or both) $E^s(i)$ and $M^s(i)$ increases. The total number of infected hosts $M(i)$ consequently increases, meaning rapid worm propagation. These observations will be validated by our performance evaluation results in Section 5.

3.3 Analyzing Online P2P-based Attack Model

We now analyze the number of newly infected hosts in the OPS attack model. Recall from Section 2.2 that while the workflow of the OPS attack model is the same for structured and unstructured P2P systems, the propagation impacts are different due to the difference in topology of the two systems. Similar to the methods used in [25], we recursively derive the newly added infection hosts in the P2P system by considering the P2P host topology degree and probability of the P2P host being infected by its P2P neighbors. We only show our results in the following. Please refer to Appendix B for detailed proof.

In the Online OPS attack model, we have the following theorem.

Theorem 2 *In the OPS attack model, with $M^s(i)$ and $N^s(i)$ at step i in the* Super-*P2P system, the next tick will have*

$$E^s(i+1) = N^s(i)[1-(1-\frac{1}{m*u})^{((\sum_{j=1}^{E^s(i)} \min(r_j,S))+\frac{m*u}{T}*S*M^n(i))}] \tag{5}$$

newly infected hosts in the Super-*P2P system, where r_j is the topology degree for host j, which is one of $E^s(i)$ newly infected hosts at step i. We have $M^s(0) = M_0, N^s(0) = m*u*P_3$.*

With $M^n(i)$ and $N^n(i)$ at step i in the Non-*P2P system, the next tick will have*

$$E^n(i+1) = N^n(i)[1-(1-\frac{1}{T})^{(S*(M^n(i)+M^s(i))-\sum_{j=1}^{E^s(i)}(\min(r_j,S))-\frac{m*u}{T}*S*M^n(i))}] \tag{6}$$

newly infected hosts in the Non-*P2P system, where $M^n(0) = 0, N^n(0) = T*P_1*P_2 - m*u*P_3$.*

In Theorem 2,$M^s(i), N^s(i), M^n(i), N^n(i), M(i)$ and $N(i)$ in the OPS attack model can be calculated similar to the recursive method in Formula (4). Note that r_j in Formula (5) and (6) represents the topology degree of a single P2P host. In structured P2P systems, the node identifier can be chosen at random and also they can be highly heterogenous. Consequently, we model structured P2P systems as following normal distribution (the mean value of r_j is θ) [27]. For the unstructured P2P system, r_j follows power law distribution defined in Formula (1). From Formulas (5) and (6), we have the following observations: With an increase in P2P topology mean degree θ (for the structured P2P system) or an increase in the mean topology degree ω (for unstructured P2P system), r_j increases with high probability. Then causes $E^s(i)$ to increase and $E^n(i)$ to decrease. The increase in $E^s(i)$ here is larger than the decrease in $E^n(i)$. Thus, $E(i)$ $(= E^s(i) + E^n(i))$ will increase and cause the total number of infected hosts $M(i)$ to increase, causing more rapid worm propagation. These observations will be confirmed by our performance evaluation results in Section 5.

4 Defending Against P2P-based Active Worm Attacks

4.1 Defense Rationale

We now discuss the issue of defending against active worm attacks. We first discuss our defense rationale, followed by worm detection and anti-worm response.

To enable rapid and accurate worm detection, it is imperative that we need to analyze the P2P traffic in multiple locations to detect suspicious traffic due to worms. Monitoring traffic towards a single P2P node is not enough to detect worms. We propose a worm defense approach within the P2P system to monitor the P2P system behavior at different locations for worm detection. As P2P-based worm attacks target hosts in the P2P system, our approach relies on volunteer P2P defense hosts (also called defense hosts) performing worm detection and immunization tasks to combat worms. Such volunteer P2P hosts can be provided with incentives commensurate with their contributions to worm defense. Such incentives naturally fall within the purview of enhancing sharing and cooperation, and will help recruit many other volunteer hosts further enhancing the effectiveness of defense. The expected functionality of the volunteer hosts will be software installation to detect worms, install immunization patches, storage space for monitoring attack traffic etc. We discuss our approach in further detail below.

Our defense framework is typically composed of a control center and a number of volunteer defense hosts that report to the control center [28]. The control center and defense hosts cooperatively perform worm detection and anti-worm response. The control center can be a system deployed node, or it can be a stable P2P node itself. The defense hosts analyze the broad classes of worm related vulnerabilities pioneered by host-based worm detection work in [29, 30]. Such detection techniques include analyzing hacking of vulnerable programs, execution of malicious code injected from the network etc. The hosts then report local alarm with summarized information to the control center. It may happen that defense hosts can incorrectly identify worm alarms. Also, defense hosts independently cannot detect a system wide attack. Thus, we propose to let the control center take decisions regarding successful worm detection (after obtaining inputs from detection hosts). Once a detection is made, the control center will initiate rapid anti-worm response (immunization commands) to immunize hosts.

In our framework, the control center and defense hosts collaborate to conduct two major functions: worm detection and immunization. To perform a system level worm detection, we incorporate the methodologies of threshold-based and trend-based worm detection schemes. The threshold-based scheme is one anomaly detection method to detect the presence of a worm [31, 32], where if the number of received alarms exceeds a threshold, a worm attack is identified as happening. The trend-based scheme [33] takes a different approach. It adopts the principle of “detecting dynamic traffic trends”, and has also been used in stock analysis, weather forecast etc for future predictions. The challenging issue in the trend-based detection scheme, is determine accurate trends for P2P-based worm attacks. We will address this issue further, based on our analysis results in the previous section. For the anti-worm response, we apply the principle of active-immunization in adaptive immune systems in biology and analyze its effectiveness in immunizing hosts within the P2P system. Finally, we provide discussions on how to build such a defense system that is scalable and efficient in accommodating the large number of P2P hosts for rapid worm detection and immunization.

4.2 Worm Attack Detection

In our framework, a local worm detection component is installed at each volunteer P2P defense host. The component in each host measures the host traffic activities to detect worms, and hosts report worm detection alarms to the control center if they believe an abnormal scan has occurred. This can be done by analyzing the broad classes of worm attack related vulnerabilities (observing abnormalities due to malicious code execution, hacking of vulnerable programs, observing abnormal scanning rates [34]. Once a set of detection worm related alarms are received by the control center, the control center determines that whether a worm attack has actually occurred in the system. We discuss two schemes that the control center can use for worm detection below.

The threshold-based scheme has been widely used in anomaly intrusion detection field [35]. In this scheme, the control center detects a worm attack happening based on *number* of P2P defense hosts generating alarms. During worm detection, the control center configures a detection sampling window and the control center calculates the number of alarming P2P defense hosts in the configured sampling window. That is, the control center determines that a worm attack is taking place if the number of P2P defense hosts generating alarms is greater than or equal to a threshold value (denoted by H_1) with the same alarm signature from defense hosts. While the threshold-based scheme is simple and easy to apply, one drawback is its applicability to situations of high false alarm rates due to its relying on the simple count of alarms in detection. We study the sensitivity of worm detection to the threshold value and false alarm rates in Section 5.

The second scheme is the trend-based scheme. This scheme reduces the false alarm problem by detecting dynamic worm propagation trends. Here also, the number of worm alarms generated is the input. However, rather than just the number of alarms, the trend of the alarms is used to detect worm attacks. Specifically, the control center configures a large detection sliding window, which consists of a number of smaller continuous detection sampling windows. The control center calculates the number of alarms generated by P2P defense hosts in each sampling window. It then conducts a trend analysis based on the recorded time sequence data of the alarms to determine if the trend follows already established trends for worm attacks. Clearly, in order to apply the trend-based scheme in detecting the P2P-based worm attacks, we need to determine the trends for P2P-based worm attacks. In the following, we discuss how to use our analysis results in the previous section to get trends for OPHLS and OPSS attacks. From our analysis and Theorem 1 in Section 3.2 for the OPHLS model, in the early stages of worm propagation, we have $M^s(i) \ll N^s(i)$ and

$$\begin{aligned} M^s(i+1) &= M^s(i) + N^s(i)[1-(1-\frac{1}{m*u})^{(S*M^s(i))}] \\ &\simeq M^s(i) + (m*u*P_3)\frac{S*M^s(i)}{m*u} \\ &\simeq (1+P_3*S)M^s(i). \end{aligned} \quad (7)$$

Given the worm scan rate (S) and P2P system vulnerability degree (P_3), $M^s(i+1)$ equals to $M^s(i)$ times a constant. Thus, the number of infected hosts $M^s(i)$ in the OPHLS attack model constantly shows a positive exponentially increasing trend.

However, not all P2P hosts will volunteer to participate in the defense system. Thus $M^s(i)$ in Formula (7) (which considered all hosts participating in defense) is not the exact number of alarms observed by controller center. In the following, we establish the relationship between the number of observed alarms and $M^s(i)$. Let us denote the total number of alarms observed by the control center till step i as $M^s_*(i)$, and the total number of newly received alarms by the control center at step i as $E^s_*(i)$. We denote P_v as the volunteer probability, and is defined as the probability that a P2P host volunteers itself for defense tasks. Using a similar approach in Section 3 to calculate number of newly infected hosts, we can calculate the average number of worm alarms at step i. For example, in case of OPHLS strategy each worm infected host generates S scans. Thus, at least one scan from a worm infected host (that generates S scans simultaneously) will be detected by P2P defense hosts with probability $P_x = 1-(1-P_v)^S$. There are a total of $(M^s(i) - M^s_*(i))$ unobserved scans as yet. Denoting P_d as the probability that a defense host can successfully generate alarm to the control center, the expected number of newly added alarms at step $i+1$ can be calculated as $P_x(M^s(i) - M_*^s(i))P_d$. Thus, we have $M_*^s(i+1) = M_*^s(i) + (M^s(i) - M_*^s(i))[1-(1-P_v)^S]P_d$. Using simple substitutions, the above formula can be represented as

$$M_*^s(i+1) + (c_1 - 1)M_*^s(i) = c_1 M^s(i), \tag{8}$$

where $c_1 = [1-(1-P_v)^S]P_d$ is a constant. From Formula (7) and (8), we can calculate $M_*^s(i)$ as follows.

$$M_*^s(i) = \frac{b}{a - c_2}(a^i - c_2^i), \tag{9}$$

where $c_2 = 1 - c_1 \in [0,1], b = c_1 * M^s(0) > 0, a = (1 + P_3 * S) > 1$. From above Formula, we can see that $M_*^s(i)$ shows trend of positive and exponential increase.

Similarly, we can determine the trend for worm propagation OPS attack model based on our analysis and Theorem 2 in Section 3.3. This also shows an exponential trend. From above analysis, the derived trends for P2P-based worm attacks provide us with a foundation for applying trend-based scheme to detect P2P-based worm attacks. We will investigate and compare the trend-based scheme with the threshold-based scheme in detail further in Section 5. In the following, we discuss anti-worm responses initiated by the control center to immunize vulnerable/ infected hosts after successful worm detection.

4.3 Anti-Worm Response

Once a detection has been made by the control center, the next step is anti-worm responses to immunize hosts. We apply an active-immunization based strategy similar to technologies like Microsoft shield that generates filters or patches that protect hosts from infection [36], IBM worm killer that uses signaling to remotely conduct host virus clean [24], and proactively patching hosts or remote immunization [37]. Active-immunization has been practically adopted in recent years quite successfully like counter-worms *Welchia* [38] and *CRClean* [39]. After detecting worm attack, the control center sends out similar anti-worm commands to coordinate the defense hosts to immunize infected/ vulnerable hosts in the P2P system.

We now analyze the effectiveness of active-immunization in our P2P defense framework. After a worm attack is detected in the P2P system, anti-worm response will be initiated by the control center. This response is anti-worm reaction commands from the control center to P2P defense hosts, which perform the immunization tasks (at an immunization rate L) following from the commands. We assume that a P2P host becomes immune with probability P_s after executing the commands. For the OPHLS attack model, there are $M^s(i)$ infected hosts in the *Super*-P2P system. The average system infection degree (as the percentage of the vulnerable hosts that has been infected in the *Super*-P2P system) can be calculated as $\frac{M^s(i)}{m*u*P_3}$. With an immunization rate L and effective defense host number $m * u * P$, the number of vulnerable hosts which have not been infected, and can be immunized by anti-worm reaction at step i is $m*u*P_v*P_s*L*(1-\frac{M^s(i)}{m*u*P_3})$. Thus, considering corresponding worm detection schemes discussed in IV-B with worm detection delay D^*, $N^s(i+1)$ (in Formula (4)) with immunization is given by

$$N^s(i+1) = \begin{cases} N^s(i) - E^s(i+1), & i \leq D^*; \\ N^s(i) - E^s(i+1) - m*u*P_v*P_s*L*(1 - \frac{M^s(i)}{m*u*P_3}) & i \geq D^*. \end{cases} \tag{10}$$

Similarly, the number of worm infected hosts which will be immunized by anti-worm response is $m*u*P_v*P_s*L\frac{M^s(i)}{m*u*P_3}$. Thus, $M^s(i+1)$ (in Formula (4)) with immunization is given by

$$M^s(i+1) = \begin{cases} M^s(i) + E^s(i+1), & i \leq D^*; \\ M^s(i) - E^s(i+1) - m*u*P_v*P_s*L*\frac{M^s(i)}{m*u*P_3} & i \geq D^*. \end{cases} \tag{11}$$

A similar procedure can derive results for OPS attack model. Above formulas provides analytical observations: An increase in P2P host defense volunteer probability P_v and smaller detection time D^* result in decreased $M^s(i)$ and $N^s(i)$. As a result, the overall worm propagation slows down. For larger values of L and P_s, worm propagation will be slower. In Section 4, we will show both the analytical and simulation results for anti-worm response along with worm detection schemes, which contribute the worm detection delay D^*.

4.4 Discussions

For efficient defenses, we need multiple hosts performing worm detection and response in a distributed fashion. Nevertheless, the large number of P2P hosts have to be efficiently managed, and the host dynamics have to handled in a scalable manner in our framework. In the following, we discuss a multi-layered and hierarchical region based defense system. That is, the P2P system is divided into a number of defense regions. In each region, hosts are arranged in a hierarchical manner in multiple layers. The degree and quality of layering can be contingent on heterogeneity of hosts (in terms of storage, processing power etc). The advantage in this approach is that efficient fusion of data and information can be carried out across layers (in the lower layers) and within regions, and responses from the higher layers can be disseminated efficiently.

To facilitate the construction process, a certain number of geographically distributed, stable and powerful P2P nodes or system deployed nodes are chosen to be region leaders (or control centers). Each region leader broadcasts its identity to nearby P2P neighbors (say within g P2P hops). Thus, the defense region size g denotes a region with a group of P2P defense hosts within g P2P hops from the region leader. When a new defense host joins the P2P system, it finds out nearby region leader from other P2P neighbors and joins the region. To compensate for unexpected node failures, we also propose to have back-up leaders per region for better fault-tolerant operation.

5 Performance Evaluation

In the following, we report results of our experiments in evaluating the impacts of P2P-based worm attacks, the sensitivity of worm propagation to various attacks and P2P system parameters, and the performance of our worm defense in containing worm propagation.

5.1 Evaluation Methodology

1) Evaluation Metrics. We broadly classify our metrics into two types: attack related and defense related. Our attack related metric in this paper is the *infection ratio over time*, which quantifies the speed of worm propagation. Specifically, it is the ratio of the total number of infected hosts to the number of vulnerable hosts over time. The higher this value is, better is the attack performance. The hosts include those in the *Super*-P2P and the *Non*-P2P systems. There are two defense related metrics we evaluate here. The first is the worm detection time, defined as the time taken to successfully detect a worm attack to the P2P system. Obviously, if the detection time is smaller, the defense performance is better. The second metric is once again the ratio of the number of infected hosts to the number of vulnerable hosts over time. However, this parameter here is measured with our defense in place. Ideally, we expect the ratio to go down with defense.

2) Evaluation Systems. We represent our evaluation systems using a tuple of the form $\langle SYS, ATT, DE \rangle$. In particular, SYS represents system parameters made up of $(T, P_1, P_2, P_3, \theta, \sigma, \omega, m, u)$. ATT represents the attack parameters made up of (WA, S, M_0), where $WA \in \{PRS, OPHLS, OPSS, OPUS\}$ identifies the attack model, where for convenience, we denote OPSS as the Online P2P-based scan attack model for the structured P2P system, and denote OPUS as the Online P2P-based scan attack model for the unstructured P2P system. For the structured P2P system, we randomly generate the topology degree for the structured P2P system with mean θ and variance with 2 in our simulation. DE represents the defense parameters made up of $\langle WB, D, F, P_d, P_v, P_s, g, L, H_1 \rangle$, where $WB \in \{OPHLS\_DE, OPSS\_DE, OPUS\_DE\}$ means defense is in place for the corresponding attack models, $D \in \langle$ Trend-based detection($D1$), Threshold-based detection($D2$) $\rangle$. Note that all other parameters have the same definition as in Table 1. In the analytical and simulation evaluation, the default parameters are $SYS = (2^{29}, 0.25, 0.3, 0.3, 4, 3, \ 4, 10000, 0.7)$, $ATT = (*, 6, 1)$, and $DE = (*, *, *, 0.92, *, 0.8, *, 2, *)$ unless otherwise stated. We point out that in our simulations here, we consider hosts in the P2P systems which have the same vulnerability as other hosts in the Internet. In reality, hosts in the P2P system are likely to be more vulnerable (due to file sharing, downloading malicious software etc.) As such, the impacts of worm propagation on P2P systems may be far worse than what we show here in our simulations. The term $*$ means that the corresponding parameters are variables in our performance evaluations. We use both numerical analysis and discrete time simulation to obtain performance data. Our results are averaged by 200 simulation runs with the same input parameters but different seeds for random number generator.

In the following, we report our performance results. Due to space limitations, we only present a limited number of cases here. However, we found that the conclusions we draw generally hold for many other cases we have evaluated. The data obtained from analytical derivations in Section 3 and 4 are represented by dotted lines, while data obtained from simulations are represented by solid lines. In all data reported, the data obtained from our analytical results is in very good agreement with our simulation data.

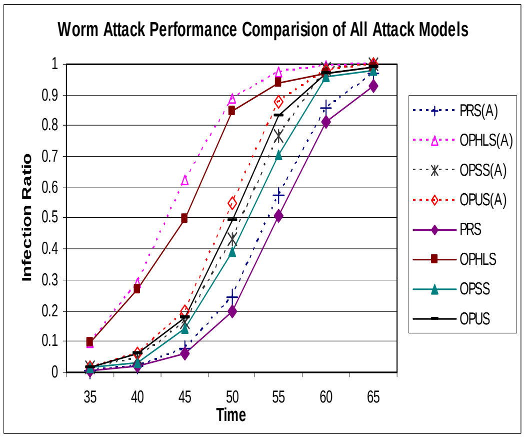

Figure 1. Performance Comparison of All Attack models

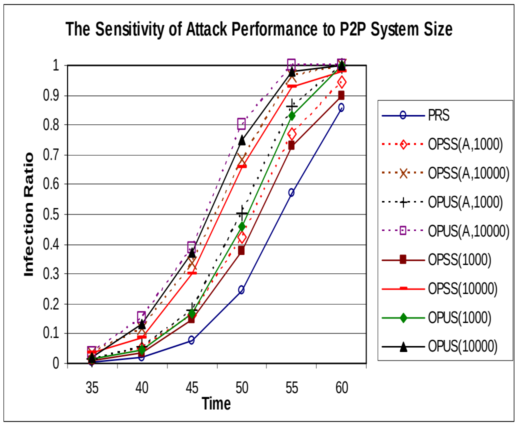

Figure 2. Attack Performance vs. P2P System Size

5.2 P2P-based Worm Attack Performance Results

We first compare different attack models to see their impacts. We then study the sensitivity of worm propagation for different attack models to important P2P system parameters. All data shown start at time 35, as the infection ratio is very small (close to 0) in the time interval $[0, 35]$ due to very small number of infected hosts compared to the large number of vulnerable hosts initially.

1) Performance Comparison of All Attack models. Figure 1 shows the data on the performance of various attack models over time. The parameters are $\langle SYS, ATT, \emptyset \rangle$, where SYS and ATT have default values. The attack models are shown in the legend in Figure 1. We report both analytical and simulations data. The term A in the legend (e.g., $OPSS(A)$) represents the data obtained for the corresponding attack model using our *analytical derivations*. For the case where the attack model is just mentioned in the legend, the data shown is the *simulation data* for the corresponding attack model. From Figure 1, we can make the following observations: a) The P2P-based attack models overall outperform the PRS attack model. For example, in the fast propagation phase of the worm (from simulation time 40 to 60), P2P-based attack models can achieve a minimum of $100\% - 300\%$ performance increase over the PRS attack model, highlighting the impacts of P2P based worm propagation. b) The OPHLS attack model achieves the fastest propagation compared to online-based attack models. The reason is that IP addresses of all P2P hosts (attack hit-list) are obtained prior to attacks in the OPHLS model, which enables rapid worm propagation. In our simulations, we do not consider the time taken for the worm attacker to obtain the P2P hit-list. In this paper, we study impacts during the phase of worm *propagation* on P2P systems. The phase where the hit-list is obtained is orthogonal to worm propagation. The attacker does not pursue any attack during this phase. The attacker can off-line obtain the hit-list by means of P2P crawler tools, information published on web sites, etc. The time during collection may take days, weeks, or even months, depending on the attackers approach, resources, P2P host dynamics etc. It is too difficult, if not impossible to model the phase of hit-list collection, and is not our focus here.

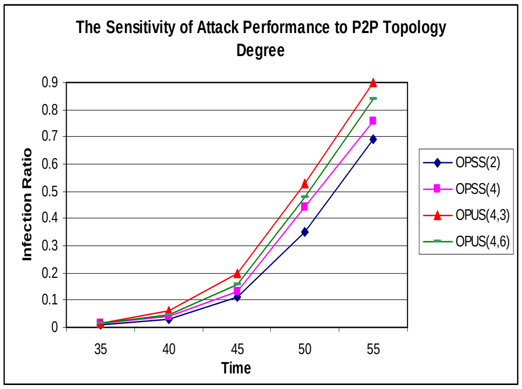

Figure 3. Sensitivity of Attack Performance to P2P Topology Degree

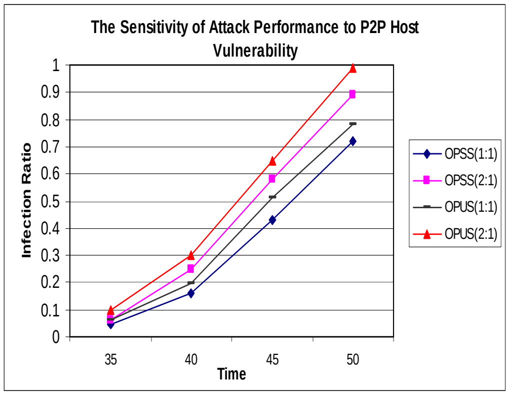

Figure 4. Attack Performance vs. P2P Host Vulnerability

2) Attack Performance vs. P2P System Size. Figure 2 shows the sensitivity of infection ratio over time under two P2P system sizes (number of hosts in the *Super*-P2P system). The two sizes evaluated are $m = 1000$ and $m = 10000$. The parameters are $\langle SYS, ATT, \emptyset \rangle$, where $SYS = (2^{29}, 0.25, 0.3, 0.3, 4, 3, 4, *, 0.7)$, $ATT = (*, 6, 1)$. In the legend, $OPSS(\#)$ denotes attack model OPSS with P2P system size $\#$ and $OPUS(\#)$ denotes the attack model OPUS with P2P system size $\#$. From this figure, we make the following observations: When P2P system size increases, the attack performance becomes consistently better for all attack models. This is because, when the P2P system size increases, the probability that a scan hits the vulnerable hosts increases, consequently increasing the number of infections.

3) Attack Performance vs. P2P Topology Degree. Recall from our analysis in Section 3, that demonstrated the importance of structured/ unstructured property of P2P systems on worm propagation (which is function of topology degree). Figure 3 shows the data on sensitivity of attack performance over time under different topology degrees. The parameters are $\langle SYS, ATT, \emptyset \rangle$, where $SYS = (2^{29}, 0.25, 0.3, 0.3, *, *,$ $, 10000, 0.7)$ and $ATT = (*, 6, 1)$. In the legend, OPSS($\#$) denotes OPSS attack model with topology degree $\#$, and OPUS (ω, σ) denotes the OPUS attack model with power law parameters ω and σ. From this figure, we make the following observations: a) For the structured P2P-based attack model, an increase in topology degree achieves better worm propagation. This is because, an increase in topology degree means an increase in connectivity. This enables the attacker to identify more hosts, consequently speeding up worm propagation. b) We observe that for the same mean topology degree (equal to 4) in the structured and unstructured P2P systems, the infected ratio increases in unstructured systems. This is because of the power law property in unstructured P2P systems. In power law property, with large topology degrees, hosts have much larger connectivity. Consequently, worms will be able to scan a large number of P2P neighbors, increasing rapid worm propagation. c) For the unstructured P2P system with the fixed mean value of topology degree ω, a lower power law degree (σ) achieves better attack performance. The reason can be explained: due to the large tail property of the power law distribution, a smaller value of σ means that the probability of a host having k neighbors ($P(k)$) is high (from Formula (1)). Thus, due

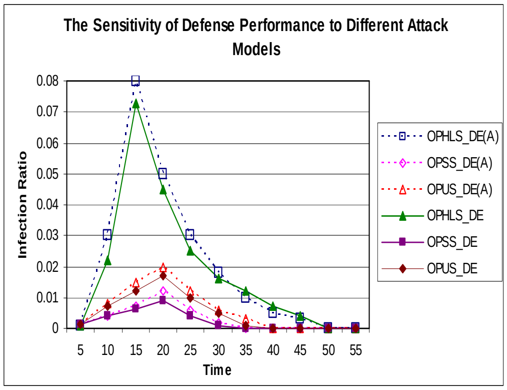

Figure 5. Defense Performance vs. Different Attack Models

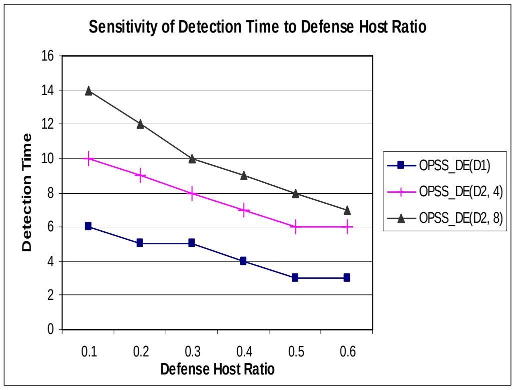

Figure 6. Sensitivity of Detection time to Defense Host Ratio

to the increased connectivity, the infection ratio is high when σ is small.

4) Attack Performance vs. P2P host vulnerability. Figure 4 shows data on the sensitivity of the attack performance to host vulnerabilities in both structured and unstructured P2P systems. The parameters are $\langle SYS, ATT, \emptyset \rangle$, where $SYS = (2^{29}, 0.25, 0.3, *, 4, 4, 3, 10000, 0.7)$ and $ATT = (*, 6, 1)$. Here $P_3 : P_2$ is selected as $1 : 1$ and $2 : 1$. From this figure, we make the following observations. With the increase in vulnerability of P2P hosts, better attack performance is achieved consistently for all P2P-based attack models. The result matches our expectation: larger host vulnerability means infection is easier, which increases worm propagation speed.

To summarize our observations at this point, we can see that P2P-based active worm attacks are highly potent in their propagation compared to traditional random scan attacks. The topology degree plays a critical role in speed of worm propagation. Unstructured P2P systems being randomly organized is more vulnerable to worms compared to structured P2P systems.

5.3 Defense Performance Results

We now study the performance of our defense strategies in containing worm spread. Specifically, we study the time taken to make a successful worm detection and its sensitivity to important defense parameters, and the reduction in infection ratio with our defense in place. The defense system is region-based as described in Section 4.4. That is, the entire *Super*-P2P system is divided into regions, with each region having a control center and a number of volunteer hosts. For the trend-based detection, the control center will report worm attack after observing exponential trend in alarms received. For the threshold-based detection, the control center will report worm attack after the number of received alarms is over threshold H_1. To reduce impact of background noise, we adopt exponential smoothing (exponentially smaller weights are assigned to old observations) to preprocess alarms. In all the figures, SYS and ATT have default values.

1) Defense Performance vs. Different Attack models. Figure 5 shows the defense performance in containing the worm spread over time under different attack models. In the legend, $OPHLS\_DE(A)$ represents the results

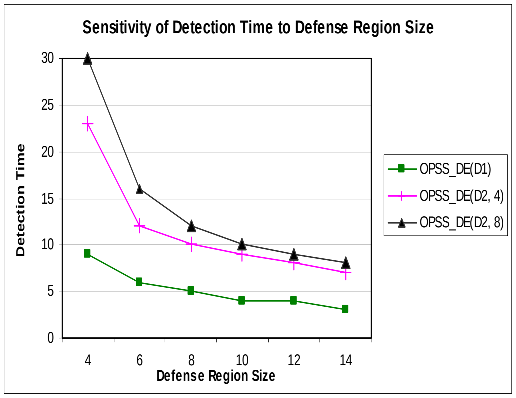

Figure 7. Detection Time vs. Defense Region Size

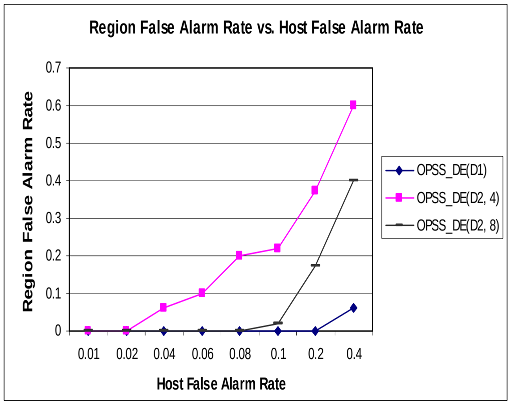

Figure 8. Region False Alarm Rate vs. Defense Host False Alarm Rate

obtained from our *analytical derivations* (in Formulas (10) and (11)) and $OPHLS\_DE$ denotes results obtained using *simulations* for the *OPHLS* attack model (similar is the case for attack models *OPSS* and *OPUS*). The defense parameters are $DE = (*, D1, 0.05, 0.92, 0.2, 0.8, 8, 2, \emptyset)$. We remark that $D1$ represents the trend-based detection. From this figure, we make the following observations: Our defense strategies effectively contains the worm propagation in the P2P system for all P2P-based attack models. The result matches our expectation - our defense strategies can rapidly detect the worm attack and immunize the number of both worm infected hosts and vulnerable hosts during the worm attack, which significantly contains the worm propagation in the P2P system.

2) Detection Time vs. Defense Host Ratio and Defense Region Size. Figures 6 and 7 shows the data on sensitivity of time to detect a successful worm attack to defense host ratio and defense region size, respectively in the system with presence of trend or threshold-based detection scheme. The defense host ratio (same as the probability P_v of a P2P host becoming a volunteer) defines the probability of P2P hosts participating in the defense (detection and immunization). The defense region size g denotes a region with a group of P2P defense hosts within g P2P hops from the region leader. The defense parameters are $DE = (OPSS\_DE, *, 0.05, 0.92, *, 0.8, *, 2, *)$. In the legend, $OPSS\_DE(\#, \&)$ represents the defense system corresponding to attack model OPSS, where $\#$ denotes the detection schemes (trend-based ($D1$) or threshold-based ($D2$)) and $\&$ denotes the threshold value H_1 (for $D2$). From Figures 6 and 7, we make the following observations: a) Larger defense host ratio and larger region sizes achieve better worm detection performance for both types of detection schemes. This demonstrates the advantages of more hosts, which enable faster detection of worms. b) The performance of the trend-based detection scheme is less sensitive to both defense host ratio and region size compared to the threshold-based detection scheme. The reason is because, since trend-based scheme discovers the trend of P2P-based worm attacks, it is more robust and less sensitive to both the defense host ratio and region size. c). We also observe that threshold-based detection is very sensitive on the threshold value H_1, since H_1 alone determines the performance of the threshold-based detection scheme.

3) Detection Time vs. Defense Host False Alarm Rate. The detection components at the detection hosts are prone to false worm alarms. Figure 8 shows the data on the sensitivity of incorrect detections at the control center of regions to host false alarm rates. The false alarm rate is the rate at which P2P hosts generate false worm detection alarms. The defense parameters are $DE = (OPSS\_DE, *, 0.05, 0.92, 0.2, 0.8, 8, 2, *)$. In the legend, $OPSS\_DE(\#, \&)$ represents the defending on OPSS attack model, where $\#$ denotes the type of detection schemes (trend (D1) or threshold based(D2)) and $\&$ denotes the threshold value H_1 (for $D2$). From this figure, we make following observations: Trend-based detection scheme generates less false alarm rates than the threshold-based detection scheme, since the trend-based detection detects the presence of a worm by the trend, not the burst, of the observed attack events. For the threshold-based detection scheme, even slight increases in false alarm rates at hosts, will cause significant increase in the false alarm rate at control center. In the case of large traffic among P2P nodes, hosts are likely to generate false alarms. To enable more accurate detection and avoid false alarms, we prefer the Trend-based detection scheme for defending P2P-based worm attacks.

6 Related Work

Our work in this paper focuses on active worm propagation and P2P system security. In this section, we will give important related work on these areas.

Active worm attacks are the most common threats in the Internet and is not new. There have consistently been worm propagations on the Internet since the Morris worm appeared in 1988. The security threat posed by active worms has steadily increased, especially in the last several years. One well known instance is Code-Red version 2 worm that was able to infect over more than $350,000$ IIS web servers in less than 14 hours on July 2001 [5]. On September 2001, Nimbda achieved very successful attack damage due to improved contagion schemes [6]. On January 2003, the Slammer worm presented a new attack record - it infected nearly $75,000$ Microsoft SQL server in less than 10 minutes [7]. There are evidences showing infected computers are being rented out as Botnets for creating an entire black-market industry for renting, trading, and managing owned computers, leading to economic incentives for attackers [1].

Much work has been done in analyzing and modeling viruses/ worms. For example, Kephat *et al.* in [40, 24] modeled worm propagation using an epidemiology model. Chen *et al.* in [26] analyzed worm propagation using a discrete-time model. Moore *et al.* in [7] modeled the “Slammer” worm propagation based on the analysis of the infected systems. Garetto *et al.* in [41] investigated the modeling of malware spreading dynamics. Zou *et al.* in [14] modeled “Code-Red” worm propagation, Ma *et al.* in [42] modeled a “self-stopping” worm that propagates rapidly until a large fraction of the vulnerable computers has been compromised, and then globally halts. In [43, 44], modeling worm propagation through Email and instant messaging system is also discussed. However, worm propagation through Email and instant messaging systems is quite different from worm propagation in P2P systems. For example, Email worm propagation depends on the user’s Email checking times, storage capacity, possibility of opening worm attachments etc., unlike worm propagation in P2P systems. In [45], the possibility of worms propagating on top of P2P systems is also presented, but detailed discussions on P2P-based worm propagation are absent. Additionally, in order to increase propagation efficiency, worms may use other strategies such as using a local network to increase the propagation efficiency [26, 6], using DNS, network topology and routing information to identify active computers instead of randomly scanning IP addresses [31]. Worms may also split the target IP address space during propagation in order to avoid duplicate scans [31]. Worms may also gather intelligence by other means, *e.g.*, actively learning the distribution of vulnerable hosts [46] and finding targets through search engines [47].

There are two types of systems for worm detection: host-based detection and network-based detection. Many host-based worm detection schemes are proposed in the literature [48, 49, 50, 51], which mainly focus on detecting worms via software anomalies. For example, Wang *et al.* [49] proposed a packet vaccine mechanism that randomizes address-like strings in packet payloads to carry out fast exploit detection, vulnerability diagnosis, and signature generation. Gao *et al*. [50] presented an approach for detecting anomalous behavior of executing processes based on the insights that processes running the same executable should behave similarly in response to a common input. Ideally, security vulnerabilities must be prevented to begin with, a problem which must addressed by the programming language community on end-hosts. However, while vulnerabilities exist and pose threats of large-scale damage, it is critical to also study the network-based detection, as this paper does, to detect wide-spreading worms.

As a complimentary approach to detect worm attacks, many network-based worm detection schemes are proposed in the literature, many of which focus on detecting worms via network traffic analysis. For example, Wu *et al.* in [32] developed the victim counter-based worm detection scheme based on the fast increase feature of worm propagation traffic. Jung *et al*. in [52] developed a threshold-based detection algorithm to identify anomalous scan traffic generated by a computer. Zou *et al.* in [33] presented a trend-based detection scheme to examine the exponentially-increasing scan-traffic pattern. Silicon Defense developed the CounterMaliceworm defense solution [53] to proactively identify and automatically block worm activity in an internal network. Zou *et al.* in [54] presented a dynamic defense system automatically containing all hosts that have set the alarm and release them after a short time. Chen *al.* in [55] studied a scan traffic rate-based countermeasure to suppress the spread of the worms. Lakhina *et al*. [56] proposed a scheme to examine other features of scan traffic, such as the distribution of destination addresses. Yu *et al*. [57] proposed a worm-detection scheme, which utilizes the attack-target distribution and robust statistical features in conjunction with dynamic decision adaptation to detect worm attacks. Their work also presented a comparatively complete space of traffic detection schemes and conducted extensive performance evaluation. Perdisci *et al.* [58] studied worms that attempt to “take on” new payload patterns to avoid detection.

Note that our work in this paper, focuses on modeling P2P-based worm propagation and defending P2P systems against worms, which distinguishes our contributions from the above.

P2P systems as large scale overlay systems have been widely deployed in the Internet to provide various services, i.e., file sharing, network storage, content delivery, etc. Without authority or policing, malicious overlay nodes typically do not worry about being tracked down or assume responsibility for actions they commit. There are several current events which highlight the problem of P2P-based attacks on Internet security. For example, security vendor Symantex reported that the number of attacks over P2P systems quadrupled from Jan. 2003 to June 2003 [59]. Some incidents show that worms can attack the P2P system. Examples are the worm attack on Instant message system [59], Igloo and MyDoom worm spreading over KaZaA P2P system [11, 12]. In recent study by ICSA lab, a division of Tru Security, it was reported that 45% of files downloaded through KaZaA contains malware, which can make things even worse. Thus, critical concerns on large P2P overlay systems and Internet security has been raised in general [22].

P2P system security including threats due to malicious participants [60] and DoS (denial of service) attacks [61, 62] have been studied extensively. In recent years, the worm related threats in the P2P systems have been given attention. For example, Chen *et al.* in [63] carried out the simulation study, driven by a P2P file-sharing workload mode, to investigate the non-scanning P2P worms (including passive worms that hide themselves in malicious files and trick users into downloading and opening them; proactive worms that propagate with legitimate network activities and even automatically discover and infect computers). Benevenuto *et al.* in [64] studied the passive propagation in file-sharing P2P systems via the content pollution. Their worm is motivated by the fact that dissemination of polluted content in a P2P system has the detrimental effect of reducing content availability, and ultimately, decreasing the confidence of users in such systems. Khiat *et al.* in [65] provided an overview of worms propagation over the P2P systems and suggested the design of a detection and mitigation system. Freitas *et al.* in [66] presented a defensive approach for containing P2P worm propagation based on the fact that some overlay nodes may not have common vulnerabilities, due to their platform diversity. By properly reorganizing the overlay graph, this can lead to the containment of P2P-assisted worms in small islands of nodes with common vulnerabilities that only have knowledge of themselves or nodes running on distinct platforms. Ding *et al.* in [67] proposed a dynamic trust management scheme based upon localized trust evaluation and alert propagation which prevent innocent peers from downloading malwares from the infected peers. Zou *et al.* in [68] discusses threats and defenses of Peer-to-Peer worm propagation. In contrast to the above works, our work conducts detailed modeling/ analysis of the worm aspects of propagation, and our defense strategy considers the aggregation of detection information to reduce false alarm rates and increase reliability of detection. Our work covers a wider spectrum including, more worm attack schemes (offline/online), different P2P systems (structured/unstructured) and more comprehensive worm detection and defense schemes.

7 Final Remarks

In this paper, we modeled and analyzed P2P-based active worm propagation, and designed effective defense strategies against them. We first defined two P2P-based attack models: an offline P2P-based hit-list attack model (*OPHLS*) and an online P2P-based attack model (*OPS*), and modeled their propagation. We then conducted a detailed analysis to study the impacts of P2P-based active worm propagation. We finally investigated worm detection and immunization strategies within the P2P system to rapidly detect worms and immunize hosts.

We conducted extensive performance evaluation to further study P2P-based worm attack and defense. Our worm attack performance data clearly showed that P2P-based worm attacks can significantly enhance the worm propagation effects. We studied the sensitivity of worm propagation to important P2P system and attack related parameters. Our observation showed that P2P size, topology degree, host vulnerability etc. have important impacts on attack effects. We observed that attack effects are more pronounced in the case of unstructured P2P systems compared to structured P2P systems. Our worm defense performance data showed that our defense strategies (with detection and immunization) could effectively contain worm spread. We observed that the trend-based scheme performs favorably compared to the threshold-based scheme in terms of both detection time and detection accuracy.

P2P systems are gaining rapid popularity in the Internet. We believe that P2P-based active worm attacks are very dangerous threats for rapid worm propagation and infection. Understanding worm propagation on P2P systems and defending against them are critical to security of the Internet. To the best of our knowledge, ours is the first work to study this issue in thorough detail and design effective defense strategies to suppress P2P-based worm attacks.

References

[1] R. Naraine, *Botnet Hunters Search for Command and Control Servers*, `http://www.eweek.com/article2/0,1759,1829347,00.asp`.

[2] *W32/MyDoom.B Virus*, `http://www.us-cert.gov/cas/techalerts/TA04-028A.html`.

[3] *W32.Sircam.Worm@mm*, `http://www.symantec.com/avcenter/venc/data/w32.sircam.worm@mm.html`.

[4] *Worm.ExploreZip*, `http://www.symantec.com/avcenter/venc/data/worm.explore.zip.html`.

[5] D. Moore, C. Shannon, and J. Brown, "Code-red: a case study on the spread and victims of an internet worm," in *Proceedings of 2-th Internet Measurement Workshop (IMW)*, Marseille, France, November 2002.

[6] CAIDA, *Dynamic Graphs of the Nimda Worm*, `http://www.caida.org/dynamic/analysis/security/nimda`.

[7] D. Moore, V. Paxson, and S. Savage, "Inside the slammer worm," *IEEE Magazine of Security and Privacy*, vol. 1, no. 4, pp. 33–39, 2003.

[8] CERT, *CERT/CC advisories*, `http://www.cert.org/advisories/`.

[9] N. C. Weaver, *A Warhol Worm: An Internet Plague in 15 Minutes*, `http://www.cs.berkeley.edu/~nweaver/warhol.old.html`.

[10] Slyck, *Slyck News*, `http://www.slyck.com`.

[11] Mydoom, *Mydoom*, `http://www.f-secure.com/tools`.

[12] J. Leyden, *Worm Spreads Through KaZaA Network*, `http://www.theregister.co.uk/2002/08/22/worm_spreads_through_kazaa_network/`, August 2002.

[13] A. Zeitoun and S. Jamin, "Rapid exploration of internet live address space using optimal discovery path," in *Proceedings of IEEE Globlecom (Next Generation Networks and Internet)*, San Francisco, CA, December 2003.

[14] C. Zou, W. B Gong, and D. Towsley, "Code red worm propagation modeling and analysis," in *Proceedings of the 9-th ACM Conference on Computer and Communication Security (CCS)*, Washington DC, November 2002.

[15] D. Z. Yazti and T. Folias, "Quantitative analysis of the gnutella network traffic," in *TR-CS-89*, Dept. of Computer Science, University of California, Riverside, August 2002.

[16] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content addressable network," in *Proceedings of ACM SIGCOMM 2001*, San Deigo, CA, August 2001.

[17] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of ACM SIGCOMM 2001*, San Deigo, CA, August 2001.

[18] M. Ripeanu and I. Foster, "Mapping the gnutella network: Macroscopic properties of large-scale peer-to-peer systems," in *Proceedings of 1-th International Workshop on Peer-to-Peer Systems (IPTPS)*, Cambridge, MA, March 2002.

[19] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in power-law networks," *Physical Review E*, vol. 046135, no. 64, 2001.

[20] P. Silvey and L. Hurwitz, "Adapting peer-to-peer topologies to improve system performance," in *Proceedings of the Hawaii International Conference on System Sciences*, Hawaii, Janary 2004.

[21] L. A. Adamic and B. A. Huberman, *Zipf's Law and the Internet*, Journal of Glohometrics, 2002.

[22] McGill, *Introduction to P2P Security*, `http://www.mcgill.ca/ncs/products/security/p2p/`.

[23] R. M. Anderson and R. M. May, *Infectious Diseases of Humans: Dynamics and Control*, Oxford University Press, Oxford, 1991.

[24] J. O. Kephart and S. R. White, "Measuring and modeling computer virus prevalence," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 1993.

[25] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, 2003.

[26] Z. S. Chen, L. X. Gao, and K. Kwiat, "Modeling the spread of active worms," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 2003.

[27] A. Rao, K. Lakshminarayanan, S. Surana R. Karp, and I. Stoica, "Load balancing in structured p2p systems," in *Proceedings of the 2-th International Workshop on Peer-to-Peer Systems (IPTPS'03)*, Berkeley, CA, February 2003.

[28] M. A. Rajab, F. Monrose, and A. Terzis, "On the effectiveness of distributed worm monitoring," in *Proceedings of the 14-th USENIX Security Synposium (SECURITY)*, Baltimore, MD, August 2005.

[29] J. Newsome and D. Song, "Dynamic taint analysis: Automatic detection and generation of software exploit attcks," in *Proceedings of the 12-th Annual Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2005.

[30] Cisco System Inc., *Deploying Host-Based Intrusion Detection*, `www.cisco.com/application/pdf/en/us/guest/netsol/ns376/c649/cdccont/0900aecd800ebd24.pdf`.

[31] S. Venkataraman, D. Song, P. Gibbons, and A. Blum, "New streaming algorithms for superspreader detection," in *Proceedings of the 12-th IEEE Network and Distributed Systems Security Symposium (NDSS)*, San Diego, CA, Febrary 2005.

[32] J. Wu, S. Vangala, and L. X. Gao, "An effective architecture and algorithm for detecting worms with various scan techniques," in *Proceedings of IEEE Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, February 2004.

[33] C. Zou, W. B. Gong, D. Towsley, and L. X. Gao, "Monitoring and early detection for internet worms," in *Proceedings of the 10-th ACM Conference on Computer and Communication Security (CCS)*, Washington, DC, October 2003.

[34] Cisco System Inc., *Cisco Security Agent and the Zotob Worm*, `http://www.cisco.com/en/US/products/sw/secursw/ps5057/prod/bulletin0900aecd8031193a.html`.

[35] D. E. Denning, "An intrusion detection model," *IEEE Transactions on Software Engineering*, vol. 2, no. 13, pp. 222–232, 1987.

[36] H. J. Wang, C. X. Guo, D. R. Simmon, and A. Zugenmaier, "Shield vulnerability-driven network filters for preventing known vulnerability exploits," in *Proceedings of ACM SIGCOMM 2004*, Portland, OR, August 2004.

[37] M. Liljenstam and D. M. Nicol, "Comparing passive and active worm defenses," in *Proceedings of the 2004 Conference on Quantitative Evaluation of Systems*, Enschede, Netherlands, September 2004.

[38] J. Leyden, *Blaster Variant Offers 'Fix' for Pox-ridden Pcs*, `http://www.theregister.com/2003/08/19/`, August 2003.

[39] M. Kern, *Re: Codegreen Beta Release*, `http://ww.securityfocus.com/archive/82/211462`, September 2001.

[40] J. O. Kephard and S. R. White, "Directed-graph epidemiological models of computer virus," in *Proceedings of 1991 Computer Society Symposium on Research in Security and Privacy (S&P)*, Oakland, CA, May 1991.

[41] M. Garetto, W. B. Gong, and D. Towsley, "Modeling malware spreading dynamics," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, San Francisco, CA, March 2003.

[42] J. Ma, G. M. Voelker, and S. Savage, "Self-stopping worms," in *Proceedings of the ACM Workshop on Rapid Malcode (WORM)*, Washington D.C, November 2005.

[43] C. Zou, Don Towsley, and Weibo Gong, "Email worm modeling and defense," in *Proceedings of the 13-th International Conference on Computer Communications and Networks (ICCCN)*, Chicago, IL, October 2004.

[44] M. Mannan and P. C. Oorschot, "Instant messaging worms, analysis and countermeasures," in *Proceedings of the 3-rd Workshop on Rapid Malcode (WORM)*, Fairfax, VA, Novermber 2005.

[45] S. Staniford, V. Paxson, and N. Weaver, "How to own the internet in your spare time," in *Proceedings of the 11th USENIX Security Symposium (SECURITY)*, San Francisco, CA, August 2002.

[46] C. S. Chen and C. Y. Ji, "A self-learning worm using importance scanning," in *Proceedings of the 1th ACM CCS Workshop on Rapid Malcode (WORM)*, Fairfax, VA, November 2005.

[47] N. Provos, J. McClain, and K. Wang, "Search worms," in *Proceedings of the 1th ACM CCS Workshop on Rapid Malcode (WORM)*, Fairfax, VA, November 2006.

[48] D. Wagner and D. Dean, "Intrusion detection via static analysis," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 2001.

[49] X. F. Wang, Z. Li, J. Xu, M. Reiter, C. Kil, and J. Choi, "Packet vaccine: Black-box exploit detection and signature generation," in *Proceedings of the 13th ACM Conference on Computer and Communication Security (CCS)*, Alexandria, VA, October/November 2006.

[50] D. Gao, M. Reiter, and D. Song, "Behavioral distance for intrusion detection," in *Proceedings of Symposium on Recent Advance in Intrusion Detection (RAID)*, Seattle, WA, September 1999.

[51] H. H Feng, J. T. Giffin, Y. Huang, S. Jha, W. Lee, and B. P. Miller, "Formalizing sensitivity in static analysis for intrusion detection," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 2004.

[52] J. Jun, S. E. Schecher, and A. W. Berger, "Fast detection of scanning worm," in *Proceedings of the 7-th International Symposium on Recent Advances in Intrusion Detection (RAID)*, French Riviera, France, September 2004.

[53] S. Staniford, "Containment of scanning worms in enterprise networks," in *Journal of Computer Security*, 2003.

[54] C. Zou, W. Gong, and D. Towsley, "Worm propagation modeling and analysis under dynamic quarantine defense," in *Proceedings of the 1-th ACM CCS Workshop on Rapid Malcode (WORM)*, Washington DC, October 2003.

[55] S. G. Chen and Y. Tang, "Slowing down internet worms," in *Proceeding of the 24-th International Conference on Distributed Computing Systems (ICDCS)*, Tokyo, Japan, March 2004.

[56] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distribution," in *Proceedings of ACM SIGCOMM*, Philadelphia, PA, August 2005.

[57] W. Yu, X. Wang, D. Xuan, and D. Lee, "Effective detection of active worms with varying scan rate," in *Proceedings of IEEE International Conference on Security and Privacy in Communication Networks (SECURECOMM)*, Baltimore, MD, August 2006.

[58] R. Perdisci, O. Kolesnikov, P. Fogla, M. Sharif, and W. Lee, "Polymorphic blending attacks," in *Proceedings of the 15th USENIX Security Symposium (SECURITY)*, Vancouver, B.C., August 2006.

[59] Erlanger, *IM and P2P Security*, `http://www.pcmag.com/article/0,4149,14335184.asp`, February 2004.

[60] E. Sit and R. Morris, "Security considerations for peer-to-peer distributed hash tables," in *Proceedings of International Workshop on Peer-to-Peer Systems (IPTPS)*, Berkeley, CA, March 2002.

[61] N. Daswani and H. G. Molina, "Query-flood dos attack in gnutella," in *Proceedings of the 9-th ACM conference on Computer and Communications Security (CCS)*, Washington, DC, November 2002.

[62] E. Athanasopoulos, K. G. Anagnostakis, and E. P. Markatos, "Misusing unstructured p2p systems to perform dos attacks: The network that never forgets," in *Proceedings of the 4-th International Conference on Applied Cryptography and Network Security (ACNS)*, Singapore, June 2006.

[63] G. Chen and R. S. Gray, "Simulating non-scanning worms on peer-to-peer networks," in *Proceedings of the 1st International Conference on Scalable Information Systems (InfoScale)*, Hong Kong, P. R. China, May 2006.

[64] F. Benevenuto, C. Costa, M. Vasconcelos, V. Almeida, J. Almeida, and M. Mowbray, "Impact of peer incentive on the dissemination of polluted content," in *Proceedings of the 2006 ACM Symposium on Applied Computing*, Dijon, France, May 2006.

[65] N. Khiat, Y. Charlinet, and N. Agoulmine, "The emerging threat of peer-to-peer worms," in *Proceedings of IEEE / Ist Workshop on Monitoring, Attack Detection and Mitigation*, Tuebingen, Germany, September 2006.

[66] F. Freitas, R. Rodrigues, C. Ribeiro, P. Ferreira, and L. Rodrigues, "Tverme: Worm containment in peer-to-peer overlays," in *Proceedings of the 6th International Workshop on Peer-to-Peer Systems*, Sellevae, WA, February 2007.

[67] X. Ding, W. Yu, and Y. Pan, "A dynamic trust management scheme to mitigate malware proliferation in p2p networks," in *Proceedings of IEEE International Conference on Communication (ICC)*, Beijing, P.R. China, May 2008.

[68] L. D. Zhou, L. T. Zhang, F. Mcsherry, N. Immorlica, M. Costa, and S. Chien, "A first look at peer-to-peer worms: Threats and defenses," in *Proceedings of the 4-th International Workshop on Peer-To-Peer Systems (IPTPS'05)*, Ithaca, NY, February 2005.

Appendix A: Proof of Theorem 1

Proof 1 *We first derive the result for the* Super-*P2P system, and then derive the result for the* Non-*P2P system.*

(i) $E^s(i+1)$ for the *Super*-P2P system. *In the* Super-*P2P system, there are* $N^s(0) = m * u * P_3$ *vulnerable hosts and total* $m * u$ *hosts. We need to prove that*

$$E^s(i+1) = N^s(i)[1 - (1 - \frac{1}{m * u})^J] \quad (12)$$

for J *scans. We will use induction to derive the Formula (12). When* $J = 1$ *(one scan), there are* $N^s(i)$ *vulnerable hosts at step* i*. One scan adds* $\frac{N^s(i)}{m*u} = N^s(i)[1 - (1 - \frac{1}{m*u})^1]$ *newly infected hosts. Therefore for* J *scans, we assume that the newly added infected hosts can be derived by*

$$E^s((i+1)|J) = N^s(i)[1 - (1 - \frac{1}{m * u})^J]. \quad (13)$$

Then, the $J + 1^{th}$ *scan can be divided into two steps: the first* J *scans and the last scan. For the last scan, there are two possibilities: adding a newly infected host and not adding a newly infected host. For convenience, we introduce a variable* Y *here. If the last scan hits a vulnerable hosts, we let* $Y = 1$*. Otherwise, we let* $Y = 0$*. We can now calculate the newly added vulnerable hosts with* $J + 1$ *scans as follows.*

$$\begin{aligned} E^s((i+1)|J+1) &= (E^s((i+1)|J) + 1)P(Y = 1) + E^s(i+1|J)P(Y = 0) \\ &= E^s((i+1)|J) + P(Y = 1) \end{aligned}$$

$$\begin{aligned} &= E^s((i+1)|J) + \frac{(N^s(i) - E^s((i+1)|J)}{m*u} \\ &= N^s(i)[1-(1-\frac{1}{m*u})^{J+1}]. \end{aligned}$$

As there are $S * M^s(i)$ scans launched to attack the Super-*P2P system at step i, we have*

$$E^s((i+1)|(J = S * M^s(i))) = N^s(i)[1-(1-\frac{1}{m*u})^{(S*M^s(i))}]. \tag{14}$$

When $M^s(i) > m * u * P_3$, all P2P vulnerable hosts have been infected, all attack resources are used to attack the Non-*P2P system. There are no newly infected hosts in the* Super-*P2P system. Thus, we have $E^s(i+1) = 0$. Then*

$$E^s(i+1) = \begin{cases} N^s(i)[1-(1-\frac{1}{m*u})^{S*M^s(i)}], & M^s(i) \leq m*u*P_3; \\ 0, & M^s(i) > m*u*P_3; \end{cases} \tag{15}$$

which is the number of newly infected hosts at step $i+1$ in the Super-*P2P system. We will now prove the second result in Theorem 1 (for the* Non-*P2P system).*

(ii) $E^n(i+1)$ for the *Non*-P2P system. *In the OPHLS attack model, after attacking the hit-list in the* Super-*P2P system, all infected hosts continuously attack the* Non-*P2P system. Clearly, the initial step K to start attacking the* Non-*P2P system is determined by,*

$$K = \min(i) \quad \forall i \qquad M^s(i) > m*u*P_3. \tag{16}$$

Considering that the effective number of hosts in the Super-*P2P system is $m*u$ and the number of vulnerable hosts in the* Super-*P2P system is $m*u*P_3$, the number of vulnerable hosts that have not been scanned at step K (calculated by Formula (16)) in the* Non-*P2P system is $N^n(K) = T*P_1*P_2 - m*u*P_3$. As such, the total number of scans to attack the* Non-*P2P system at step i $(i \geq K)$ is $S*M^n(i) + S*M^s(K)$. For 1 scan, there are $N^n(i)$ vulnerable hosts at step i $(i \geq K)$. One scan adds $\frac{N^n(i)}{T} = N^n(i)[1-(1-\frac{1}{T})^1]$ newly infected hosts. Similar to the derivation of (14), we have*

$$E^n(i+1) = \begin{cases} 0, & M^s(i) \leq m*u*P_3; \\ N^n(i)[1-(1-\frac{1}{T})^{(S*M^n(i)+S*M^s(K))}], & M^s(i) > m*u*P_3; \end{cases} \tag{17}$$

Appendix B: Proof of Theorem 2

Proof 2 *In the OPS model, worms launch attack to both* Super-*P2P and* Non-*P2P system simultaneously. In the* Super-*P2P system, since there are $N^s(i)$ vulnerable hosts and $M^s(i)$ infected hosts at step i, 1 scan for each infected host adds $\frac{N^s(i)}{m*u} = N^s(i)[1-(1-\frac{1}{m*u})^1]$ newly infected hosts at step i. Since there are $E^s(i)$ infected hosts at step i and each infected host j has topology degree r_j, each infected host can simultaneously generate at most $\min(r_j, S)$ scans to scan other P2P hosts. As there are $E^s(i)$ worm infected hosts in the* Super-*P2P system, the total number of scans to attack* Super-*P2P system at step i is $\sum_{j=1}^{E^s(i)} \min(r_j, S)$. As there are $M^n(i)$ worm infected hosts in* Non-*P2P system which launch $S*M^n(i)$ scans at step i. Considering the P2P size $m*u$, there* *are* $\frac{m*u}{T}*S*M^n(i)$ *scans from the hosts in* Non-*P2P system which are launched to the* Super-*P2P system. Using a similar method used to prove Theorem* 1, *we have*

$$E^s(i+1) = N^s(i)[1-(1-\frac{1}{m*u})^{((\sum_{j=1}^{E^s(i)}(\min(r_j,S)))+\frac{m*u}{T}*S*M^n(i))}]. \tag{18}$$

Using a similar method used to prove Theorem 1, we have

$$E^s(i+1) = N^s(i)[1-(1-\frac{1}{m*u})^{(\min(\theta,S)*E^s(i)+\frac{m*u}{T}*S*M^n(i))}]. \tag{19}$$

For the Non-*P2P system, since there are* $N^n(i)$ *vulnerable hosts and* $M^n(i)$ *infected hosts at step* i*, 1 scan adds* $\frac{N^s(i)}{m*u} = N^s(i)[1-(1-\frac{1}{T})^1]$ *newly infected hosts at step* i. *There are a total of* $M^n(i)+M^s(i)$ *worm infected hosts in the whole Internet and* $\sum_{j=1}^{E^s(i)}\min(r_j,S)+\frac{m*u}{T}*S*M^n(i)$ *scans are launched to the* Super-*P2P system at step* i. *Thus, the total number of scans launched for the* Non-*P2P system at step* i *is* $S*(M^n(i)+M^s(i)) - \sum_{j=1}^{E^s(i)}\min(r_j,S)-\frac{m*u}{T}*S*M^n(i)$. *Follow the similar method in the proof of Theorem 1, we have*

$$E^n(i+1) = N^n(i)[1-(1-\frac{1}{T})^{(S*(M^n(i)+M^s(i))-(\sum_{j=1}^{E^s(i)}\min(r_j,S))-\frac{m*u}{T}*S*M^n(i))}]. \tag{20}$$