# Self-Disciplinary Worms and Countermeasures: Modeling and Analysis

Wei Yu, Nan Zhang, Xinwen Fu, and Wei Zhao

***Abstract*— In this paper, we address issues related to the modeling, analysis, and countermeasures of worm attacks on the Internet. Most previous work assumed that a worm always propagates itself at the highest possible speed. Some newly developed worms (e.g., "Atak" worm) contradict this assumption by deliberately reducing the propagation speed in order to avoid detection. As such, we study a new class of worms, referred to as *self-disciplinary worms*. These worms adapt their propagation patterns in order to reduce the probability of detection, and to eventually infect more computers. We demonstrate that existing worm detection schemes based on traffic volume and variance cannot effectively defend against these self-disciplinary worms. To develop proper countermeasures, we introduce a game-theoretic formulation to model the interaction between the worm propagator and the defender. We show that an effective integration of multiple countermeasure schemes (e.g., worm detection and forensics analysis) is critical for defending against self-disciplinary worms. We propose different integrated schemes for fighting different self-disciplinary worms, and evaluate their performance via real-world traffic data.**

***Index Terms*— Worm, Game Theory, Anomaly Detection**

## I. Introduction

Worm is a malicious software program that propagates to other computers on the Internet by remotely exploiting vulnerabilities in these computers. Worm attack is considered a dangerous threat to the Internet. There have been many cases of Internet worm attacks, such as the "Code-Red" worm in 2001 [1], the "Slammer" worm in 2003 [2], and the "Witty/Sasser" worms in 2004 [3]. All these worms caused significant damage. For example, the "Code-Red" worm infected more than 350,000 computers in less than 14 hours by exploiting the buffer-overflow vulnerability of Microsoft's Internet Information Services (IIS) 4.0/5.0, causing more than $1,200,000,000 in damage.

Generally speaking, a worm propagator has two objectives. One is to infect as many computers as possible within a given period of time. The other is to avoid being detected and punished by the defenders. After infecting a number of computers without being detected, the worm propagator can remotely control the infected computers and use them as stepping-stones to launch further attacks (e.g., distributed denial-of-service (DDoS) [4], phishing [5], and spyware [6]). Recent studies showed the existence of a black market for trading/renting compromised computers (as "bots") for malicious purposes [7]–[9], providing further economic incentives for worm attacks. Exist study also showed the possibility of a "super-botnet," which coordinates independent botnets for attacks of unprecedented scale [10]. From an attack perspective, "super-botnets" would be extremely versatile and resistant to countermeasures. Consequently, research on worm-attack modeling and defense is vital to computer and network security.

Most existing work (explicitly or tacitly) assumes that worms constantly propagate at the highest possible speed. Interestingly, some recently developed worms contradict this assumption by intentionally reducing their propagation speed to avoid detection. For example, the "Atak" worm [11] and the "self-stopping" worm [12] circumvent detection by hibernating (i.e., stop propagating) with a pre-determined period. If such a worm successfully avoids (or delays) detection, it will eventually infect more computers and result in more damage.

In order to address the rising threats from these worms, in this paper, we define a new class of worms called *self-disciplinary worms*. A self-disciplinary worm adapts its propagation patterns to defensive countermeasures, aiming to avoid or delay detection and, ultimately to infect more computers.

Specifically, we partition self-disciplinary worms into two categories, namely static and dynamic self-disciplinary worms. *Static self-disciplinary worms* are those that intelligently select a propagation speed at the initial time of attack but nevertheless maintain the same strategy during the attack session. On the other hand, a *dynamic self-disciplinary worm* may dynamically adjust its propagation speed during the attack session.

Based on the models of static and dynamic self-disciplinary worms, we propose and evaluate countermeasures against such worms. In particular, we demonstrate that the integration of multiple defensive schemes is critical for the defense against self-disciplinary worms. We consider two existing defensive schemes, threshold-based [13], [14] and trace-back [15], [16], and a novel one introduced in the paper, the spectrum-based scheme. Using game-theoretic analysis, we show that an integration of the first two schemes is effective against static self-disciplinary worms, while defense against dynamic ones requires the integration of all three schemes. Table I summarize the results.

To the best of our knowledge, this paper is the first to study worms that adaptively reduce their propagation speed to (eventually) infect more computers. This paper is also the first to formally study the integration of various worm-defense strategies and to analyze the interaction between the defender

Wei Yu is with the Department of Computer and Information Sciences, Towson University, Towson, MD 21252. E-mail: wyu@towson.edu. Nan Zhang is with the Department of Computer Science, The George Washington University, Washington, DC 20052. Email: nzhang10@gwu.edu. Xinwen Fu is with the Department of Computer Science, University of Massachusetts Lowell. Email: xinwenfu@cs.uml.edu. Wei Zhao is with the University of Macau, Av. Padre Toms Pereira Taipa Macau, China. Email: zhao8686@gmail.com.

TABLE I
PERFORMANCE OF DEFENSIVE SCHEMES

$S_1$: Threshold-based; $S_2$: Trace-back; $S_3$: Spectrum-based; $\surd$: Effective; $\times$: Ineffective

| | $S_1$ | $S_1 + S_2$ | $S_1 + S_2 + S_3$ |
|---|---|---|---|
| Traditional worm | $\surd$ | $\surd$ | $\surd$ |
| Static self-disciplinary worm | $\times$ | $\surd$ | $\surd$ |
| Dynamic self-disciplinary worm | $\times$ | $\times$ | $\surd$ |

and the worm propagator in a game-theoretic fashion.

The rest of the paper is organized as follows: We present the classifications of worms and countermeasures in Section II. We also introduce the classification of self-disciplinary worms in this section. In Section III, we consider two baseline scenarios, where a static self-disciplinary worm propagates either freely without defensive countermeasures, or under an existing threshold-based scheme. Our analysis shows that a self-disciplinary worm can lead to significant damage even with the presence of existing threshold-based schemes. To address the threats from self-disciplinary worms, we introduce a game-theoretic formulation of the system in Section IV, and discuss the integrated defense against static and dynamic self-disciplinary worms in Sections V and VI, respectively. We present the simulation results of our countermeasures in Section VII. We extend our theoretical analysis with new utility function in Section VIII. The related work is reviewed in Section IX, followed by final remarks in Section X.

## II. CLASSIFICATION OF WORMS AND COUNTERMEASURES

In this section, we present an overview of worm propagation models and defensive strategies. In particular, we first briefly review the model for traditional worms, and then define a novel model for self-disciplinary worms. After that, we introduce a taxonomy of defensive strategies against worm propagation.

### A. Worm Propagation: Traditional and Self-Disciplinary Worms

*1) Traditional Worms:* A traditional worm behaves similar to a biological virus, in terms of its self-propagating nature [1], [2]. Worm propagation on the Internet is an iterative process that usually starts with a computer known as the worm propagator. In order for a worm to propagate itself, it must be able to identify computers with exploitable vulnerabilities. Since the attacker does not have complete information about the locations of vulnerable computers on the Internet, a commonly used strategy for identifying the vulnerable computers is Pure Random Scan (PRS) [2], [17], in which a worm scans random IP addresses to identify vulnerable computers. Most previous studies [1], [2] (explicitly or tacitly) make a *max-speed assumption* on worm propagation. That is, every worm-infected computer constantly performs the maximum possible number of scans (on other computers) per unit of time. We denote such maximum scan rate by $S$ scans/unit of time.

*2) Definition of Self-Disciplinary Worms:* With defensive systems in place, worms have consequently evolved and become more sophisticated than the traditional worms mentioned above. In particular, some worms deliberately reduce their propagation speed to avoid detection [11], [12]. In particular, the "Atak" worm attempts to remain hidden by sleeping (suspending scans) when it suspects it is under detection. The "self-stopping" worm rapidly propagates until a fraction of the vulnerable computers has been compromised, and then globally halts. In this paper, we study these new, smarter worms. Specifically, we remove the max-speed assumption, and consider *self-disciplinary* worms which manipulate their propagation growth rate in order to avoid or delay detection.

In general, a self-disciplinary worm is a generalization of traditional worms with a scan rate of less than $S$. Recall that $S$ is the maximum possible number of scans that the infected computer can launch per unit of time. In particular, we consider a worm which controls its propagation speed by a parameter $p(t)$ such that the *average* number of scans each infected computer performs at time $t$ is $p(t) \cdot S$. We refer to $p(t)$ as the *propagation growth rate* at time $t$. Note that a traditional worm with the max-speed assumption has $p(t) = 1$ for all $t$. The "Atak" worm [11] and the "self-stopping" worm [12] are the special cases of self-disciplinary worms, where $p(t)$ is changed between 0 and 1.

In our analysis, we assume $p(t)$ to be a global variable. In practice, to maintain the same $p(t)$ for all worm-infected computers (to the gratitude of minute), the worm propagator may leverage the automatic Internet-time-synchronization feature of modern operating systems (e.g., Windows XP).

*3) Propagation Scheme of Self-Disciplinary Worms:* In an ideal situation, if $p(t)$ is extremely small, a self-disciplinary worm may propagate for a very long time without being detected. In practice, however, such forever-propagation might not be possible because the worm will ultimately be detected by host/software-based detection methods and the vulnerability exploited by the worm will be fixed through software updates within a certain amount of time [18]–[22]. To reflect this fact, we introduce a *time limit* $t_{\mathrm{E}}$, such that a worm will always be detected after propagating for $t_{\mathrm{E}}$ units of time. With this parameter, the objective of worm propagation is to infect as many computers as possible by time $t_{\mathrm{E}}$.

*4) Classification of Self-Disciplinary Worms:* Note that a self-disciplinary worm can either use a constant $p(t)$ for the duration of worm propagation, or deliberately change $p(t)$ over time. We consider both cases in this paper. In particular, we call the self-disciplinary worms with constant $p(t)$ as a *static self-disciplinary worm.* If a self-disciplinary worm has $p(t)$ changed over time $t$, we call it a *dynamic self-disciplinary worm.* For static self-disciplinary worms, we use $p$ to denote the constant value of $p(t)$. Table II depicts our classification of worms. As we can see, a self-disciplinary worm is either static or dynamic.

TABLE II
CLASSIFICATION OF WIDE-SPREADING WORMS

| Name | Property |
|---|---|
| Traditional worm | $p(t) = 1$ |
| Static self-disciplinary worm | $0 < p(t) = c < 1$ |
| Dynamic self-disciplinary worm | $p(t)$ varies over time |

Note that each type of worm has advantages and disadvantages. Static self-disciplinary worms are easier to implement and, as we will show later, are already very effective against existing defensive mechanisms. The dynamic ones, on the other hand, require most infected computers to be roughly synchronized, such that they could compute the amount of time elapsed since the start of propagation and determine $p(t)$ correspondingly. Nonetheless, dynamic self-disciplinary worms may outperform the static self-disciplinary worms in terms of infecting computers and avoiding detection. The "Atak" worm [11] and the "self-stopping" worm [12] mentioned in Section I are special cases of dynamic self-disciplinary worms, as their propagation growth rates are changing between 0 and 1 over time.

### *B. Countermeasures: Detection and Trace-Back*

In this subsection, we review two types of existing defensive countermeasures: *worm detection* and *trace-back*. Worm detection focuses on the detection of propagating worms on the Internet. Trace-back schemes, on the other hand, aim to identify the origin of worm propagation, in order to take appropriate legal steps to punish the worm propagator. Trace-back is triggered after a propagating worm is detected.

Various schemes have been proposed for each countermeasure. We now briefly review these schemes and provide simple abstractions which will form the basis of our investigation on the effectiveness of detection/trace-back schemes and their combination. We will study the concrete algorithms in the experiments presented in Section VII.

*1) Worm Detection:* A major effort for detecting worm propagation has been the Internet Threat Monitoring (ITM) system. An ITM system consists of one centralized data center and a number of monitors, which are distributed across the Internet at hosts, routers, and firewalls, etc. Each monitor is responsible for monitoring suspicious traffic (e.g., scans to unoccupied IP addresses or ports) and reporting them to the data center. The data center then analyzes the collected traffic logs and detects worm attacks. Although the IP address space directly monitored by an ITM system is much smaller than the entire Internet [23]–[25], the collected logs can be considered as random samples of the Internet traffic, and therefore can provide critical insights for detecting worm attacks.

The majority part of this paper focuses on a simple abstraction of *threshold-based detection on traffic volume*, which is used by many ITM systems. That is, the data center issues a worm alert if and only if the number of suspicious scans per unit of time, that are generated by infected computers, exceeds a pre-determined threshold [14]. We denote such threshold as $T_{\mathrm{R}}$ scans/unit of time. Note that the data center must carefully choose $T_{\mathrm{R}}$ to make a proper tradeoff between detections and false alarms.

*2) Trace-back:* Another defensive countermeasure is trace-back, which enables law enforcement agencies to identify the original worm propagators and punish them [15], [26], [27]. A trace-back scheme typically involves a number of routers which monitor all through-traffic and store traffic logs in a storage server. When a "trace-back" order is given, the traffic logs (e.g., flow-level recorded logged by the networks) are post-mortem analyzed in order to identify the origins of the worm propagator.

For example, Xie [15] proposed a random-walk-based approach which, provided there are enough traffic logs, can determine both a number of suspect computers responsible for originating a worm propagation and the attack flows that make up the initial attack stages. The basic idea is to exploit the "wide tree" shape of a worm propagation (emanating from the source) by performing random "moon-walks" backward in time along paths of flows. Correlating the repeated walks reveals the initial causal flows, thereby aiding in identifying the source of a worm. Ahmad *et al.* [16] studied a technique which integrates both end-host system logs and network monitoring logs to link worm propagation to its source.

The existing trace-back schemes differ in terms of their inputs (i.e., information collected for trace-back), outputs (i.e., information of suspicious computers), and techniques for post-mortem analysis. Nonetheless, we observe the following common properties of all existing trace-back schemes:

- The storage server only has the capacity to store traffic logs for a limited time period because of the large amount of traffic transmitted through the routers.
- Due to this limitation, the stored traffic logs usually do not contain enough information to *uniquely* identify the exact worm-origin computer. Instead, the trace-back techniques may only provide a list of suspicious computers (and accompany each of them with its probability of being the worm origin). For example, a number of suspect computers could be identified by the random-walk based algorithm in [15] as potential origins of the worm propagation.
- As such, the defender (e.g., law enforcement) may have to engage some further investigation (e.g., check the end-host traffic traces of suspicious computers) to correctly identify the worm origin. Clearly, the number of suspicious computers must be reasonably small in order to enable such investigation in practice.

Based on the two observations, we consider an abstraction of trace-back schemes as follows: Let $t_{\mathrm{B}}$ be the maximum length of time (i.e., "window size") during which the traffic logs will be retained. That is, when the trace-back order is issued at time $t$, the earliest traffic log the defender has access to is at time $\max(0, t - t_{\mathrm{B}})$. If $t < t_{\mathrm{B}}$, then the trace-back can successfully identify the original worm propagator (note that there is always worm propagation activities to observe in the traffic log because the trace-back order is only issued when a worm detection is made). If $t \geq t_{\mathrm{B}}$, then at time $t - t_{\mathrm{B}}$ the worm has already compromised $f(t - t_{\mathrm{B}})$ ($f(t - t_{\mathrm{B}}) \geq 1$) computers, between which the defender cannot distinguish. As such the defender may have to generate a suspicious list of $f(t - t_{\mathrm{B}})$ computers (because the defender has no way to distinguish between them). Let $m$ be the maximum possible number of suspicious computers to allow further investigation. As we can see, the defender can successfully identify (and punish) the worm propagator if and only if

$$t < t_{\mathrm{B}} \text{ or } f(t - t_{\mathrm{B}}) \leq m. \tag{1}$$

## III. Baseline Scenarios for Propagation of Self-Disciplinary Worms

In this section, we analyze two baseline scenarios where a static self-disciplinary worm 1) freely propagates without any defensive countermeasure, and 2) propagates with the presence of existing threshold-based schemes. The results show that the threshold-based scheme, by itself, is ineffective against self-disciplinary worms. The analysis also forms the basis for our analysis of more complicated systems in the next two sections.

### A. Table of Notations

Table III lists all the notations that are used throughout the paper. Some of these concepts will be introduced in the latter part of the paper.

TABLE III

Table of Notations

| Notation | Definition |
|---|---|
| $S$ | maximum scans/unit of time |
| $N$ | total number of vulnerable hosts |
| $v$ | percentage of monitored IP addresses |
| $V$ | total number of IP addresses |
| $\beta$ | pair-wise propagation rate, $\beta = S/V$ |
| $t$ | units of time elapsed since start of propagation |
| $t_{\mathrm{D}}$ | time of detection |
| $t_{\mathrm{E}}$ | maximum propagation length |
| $t_{\mathrm{B}}$ | windows size of trace-back |
| $p(t)$ | propagation growth rate |
| $p$ | $p(t)$ (as a constant) for a static self-disciplinary worm |
| $f(t)$ | number of infected hosts at time $t$ |
| $T_{\mathrm{R}}$ | threshold for detection |
| $m$ | maximum number of hosts for manual check in trace-back |
| $E(\cdot)$ | Expected value |

### B. Case 1: No Defense

Let $f(t)$ be the number of infected computers in the baseline system at time $t$, where $t$ is the amount of time elapsed since the start of worm propagation. Apparently,

$$f(0) = 1. \tag{2}$$

Equation (3) shows the relationship between $f(t)$ and other system parameters. The detailed derivation of Equation (3) can be found in Appendix A.

$$f(t) = \frac{N \cdot e^{\beta \cdot p \cdot N \cdot t}}{e^{\beta \cdot p \cdot N \cdot t} + N - 1} \approx \frac{N \cdot e^{\beta \cdot p \cdot N \cdot t}}{e^{\beta \cdot p \cdot N \cdot t} + N}, \tag{3}$$

where $V$ and $N$ be the total number of IP addresses and vulnerable computers on the Internet, respectively, $p$ is the propagation growth rate, $\beta = S/V$ is commonly referred to as the *pair-wise propagation rate* in the epidemic research literature [28].

Based on (3), we make the following observations:

- $f(t)$ is an increasing function of $t$. Also, $f(t)$ increases when $\beta$, $N$, or $p$ increases.
- When $t$ is sufficiently small such that $e^{\beta \cdot p \cdot N \cdot t} \ll N$, we have
$$f(t) \approx e^{\beta \cdot p \cdot N \cdot t}. \tag{4}$$
That is, when a worm is in its initial propagation phase, the number of infected computers increases exponentially over time $t$.
- On the other hand, when $t$ is sufficiently large, $f(t) = N$. This indicates that when no defense system exists, eventually all vulnerable computers will be infected.
- Except for a new parameter "$p$", our result is identical to the result in [28]. We nevertheless present the derivation process in this paper to help our readers understand the physical meaning of the equation and its solution.
- Consider the extension of our baseline system to include the detection scheme. Let $t_D$ be the time when detection is made. Then, our formula in Appendix A will correctly represent the number of infected computers as long as $t \le t_D$.
- While we derive $f(t)$ for static self-disciplinary worms, the derivation can be useful for the dynamic self-disciplinary worms as well. From the derivation process in Appendix A, if we replace $p$ by $p(t)$ in (17), the differential equation still holds. Unfortunately, the analytical solution in (3) requires that $p$ be constant, and thus cannot be directly applied to dynamic self-disciplinary worms.

### C. Case 2: With Existing Threshold-Based Detection

We now analyze the propagation of a static self-disciplinary worm with the presence of a threshold-based detection scheme. Let $v$ be the ratio of computers on the Internet that are monitored by the ITM system. Let $f_{\mathrm{N}}(t)$ be the number of infected computers when there is no defensive mechanism in place (i.e., $f(t)$ derived in (3)). Recall that $t_{\mathrm{E}}$ is the maximum propagation length and $N$ is the total number of vulnerable hosts. We have the following theorem.

*Theorem 1:* For a detection scheme with threshold $T_{\mathrm{R}}$, the number of computers infected by a static self-disciplinary worm with propagation growth rate $p$ is $m_{\mathrm{S}}^{\mathrm{T}}(p) = \min(\frac{T_{\mathrm{R}}}{p \cdot v}, f_{\mathrm{N}}(t_{\mathrm{E}}))$, which satisfies

$$\max_{p \in [0,1]} m_{\mathrm{S}}^{\mathrm{T}}(p) \ge \frac{N \cdot e^{\beta T_{\mathrm{R}} t_{\mathrm{E}}/v}}{e^{\beta T_{\mathrm{R}} t_{\mathrm{E}}/v} + N}. \tag{5}$$

*Proof:* (Sketch) Recall that the static self-disciplinary worm will use the constant propagation growth rate $p$. If $f(t_{\mathrm{E}}) \cdot p \cdot v \ge T_{\mathrm{R}}$, the worm detection will be flagged by threshold-based detection. Thus, the number of computers infected by a static self-disciplinary worm is $\min(\frac{T_{\mathrm{R}}}{p \cdot v}, f_{\mathrm{N}}(t_{\mathrm{E}}))$. The lower bound in (5) is taken when $p \approx \frac{T_{\mathrm{R}}}{N \cdot v} + \frac{T_{\mathrm{R}}}{v} \cdot e^{\frac{T_{\mathrm{R}} \beta t}{v}}$. ■

As we can see from the theorem, when the threshold-based scheme is the only available defensive measure, the worm propagator can significantly increase the number of infected computers by reducing $p$ to delay the detection (e.g., until $t_{\mathrm{E}}$).

We now show a simulation results which illustrates that the threshold-based scheme, by itself, is ineffective against static self-disciplinary worms. We consider two existing detection schemes: the mean threshold detection scheme (i.e., issues an alert when the average volume of illegal traffic captured exceeds a threshold) [14] and the variance threshold scheme (based on the variance of illegal traffic captured) [13].

Figure 1 shows the number of computers that a static self-disciplinary worm can eventually infect when the mean-threshold and variance-threshold detection mechanisms [13], [14] are employed, respectively. The parameter settings are as

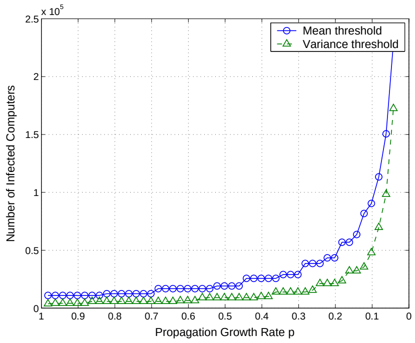

Fig. 1. Number of infected computers vs. Propagation growth rate

follows: the number of vulnerable computers on the Internet is 350,000, which is close to the number of computers vulnerable to the "Code-Red" worm [29]; the maximum tolerable false positive rate is $1\%$; the propagation growth rate varies between $0.03$ and $1$.

An interesting observation from Figure 1 is that a worm can actually infect more computers when $p$ is smaller. For example, when $p = 0.1$, the number of infected computer will be $90,000$ and $50,000$ when mean and variance threshold schemes are used, respectively. Clearly, when a static self-disciplinary worm chooses a propagation growth rate less than $1$, the existing threshold-based detection schemes become less effective. Since static self-disciplinary worms are special cases of the dynamic self-disciplinary, the existing threshold-based defense is also ineffective against dynamic self-disciplinary worms.

## IV. A Game-Theoretic Formulation For Defense Against Self-Disciplinary Worms

As we demonstrated in the above section, under the existing threshold-based detection, a self-disciplinary worm can infect a large number of computers without being detected. In order to effectively defend against self-disciplinary worms, we will introduce an integration of detection and traceback schemes to intimidate the worm propagators to abandon their attacks. In this section, we first present a game-theoretic formulation for the interactions between a worm propagator and the defender.

### *A. Game between Defender and Worm Propagator: Basic Concepts*

We model the interactions between the defender and a worm propagator as a two-player non-cooperative game. The worm propagator and the defender are the two players $P_1$ and $P_2$, respectively. Let $S_i$ be the strategy space of $P_i$, which includes all possible strategies of it. Let the utility function of $P_i$ be $u_i : S_1 \times S_2 \rightarrow \Re$, which reflects the player's objective.

The game is non-cooperative in that the two players are in opposition and are unlikely to make any binding agreement when choosing their strategies [30]. We consider the cases where both players are rational. That is, each player chooses the strategy which maximizes the expected value of its utility function.

### *B. Strategies*

The strategy of the defender is to select the input parameters for detection and trace-back. However, recall from Section II-B that the trace-back parameters $t_{\rm B}$ and $m$ are determined by the storage capacity and human resources of the trace-back algorithm for further investigating a number of suspect computers, and cannot be arbitrarily changed by the defender. Thus, the strategy of the defender is limited to determine the threshold $T_{\rm R}$ for worm detection. That is, its strategy set consists of all possible values of $T_{\rm R} \geq 0$.

The strategy of the worm propagator is to determine the propagation growth rate $p$. Recall from Section II-A that the worm propagator can choose to either use a constant $p$ or to vary $p$ over time. Thus, the strategy set of a static self-disciplinary worm is consisted of all possible values of $p \in [0,1]$. The strategy set of a dynamic self-disciplinary worm consists of all possible functions $p(\cdot)$ which map time $t \in [0, t_{\rm E}]$ to a real value in $[0,1]$.

Note that the defensive strategy may be known by the worm propagator. Thus, the worm propagator may determine its strategy $p$ (or $p(\cdot)$) based on its knowledge of $T_{\rm R}$.

### *C. Utility Functions*

The utility function measures the benefit (or loss if $< 0$) gained by a player and reflects the objectives of the player.

*1) Worm Propagator:* A worm propagator has two objectives:

- to maximize the number of infected computers.
- to avoid being traced back (and punished for its malicious activities).

Correspondingly, we define the utility function of the worm propagator as: $u_{\rm A} = \frac{f(t_{\rm D})}{N} - \beta_{\mathcal{A}} \cdot g$, where $f(t_{\rm D})$ is the number of hosts the worm infects when it is detected, $N$ is the total number of vulnerable hosts on the Internet,

$$g = \begin{cases} 1, & \text{if the propagator is traced back and punished;} \\ 0, & \text{otherwise,} \end{cases}$$

and $\beta_{\mathcal{A}}$ measures the (relative) penalty the worm propagator receives from being traced back and punished.

Clearly, the value of $\beta_{\mathcal{A}}$ varies between different worm propagators. Nonetheless, it is a common belief that most worm propagators on the Internet consider the penalty of being traced back to be substantially greater than the benefits gained from worm propagation [15], [31]. In this case, we have $\beta_{\mathcal{A}} \gg 1$.

*2) Defender:* The defender also has two objectives:

- to minimize the number of infected computers.
- to control the *false positive rate*, i.e., the number of alarms falsely triggered (per time slot) when there is no worm propagation.

Clearly, the worm detection system becomes useless if it generates an excessive number of false positives. In this paper, we impose a pre-determined (upper-bound) threshold on the false positive rate as the *maximum tolerable false positive rate* $\delta$ (false alarms/second). Naturally, to make the detection system feasible in real practice, there should be $\delta \ll 1$. The defender will receive the ultimate penalty if this threshold is violated.

Formally, we define the utility function for the defender as:

$$u_{\mathrm{D}} = \begin{cases} -\infty, & \text{if false alarm rate} > \delta; \\ -f(t_{\mathrm{D}})/N, & \text{otherwise.} \end{cases}$$

As we can see, the defender will never choose a strategy which violates the threshold $\delta$, because, otherwise simply choosing no defense would yield better utility. When the threshold is satisfied, the loss of the defender is the percentage of vulnerable hosts infected by the worm.

## V. Integration of Multiple Defensive Scheme to Counteract Static Self-Disciplinary Worms

Based on the game-theoretic formulation, we present our integrated defense against static self-disciplinary worms (i.e., constant $p \in [0, 1)$).

### *A. Basic Ideas of Integrating Threshold-Based Detection and Trace-Back Schemes*

The rationale behind integrating threshold-based and trace-back schemes for defense against static self-disciplinary worms can be stated as follows: When both threshold-based and trace-back schemes are in place, if the worm propagator chooses to increase the propagation speed $p$, it also increases the probability of being detected and reduces the duration of propagation. If the worm propagator chooses to reduce $p$, the worm will propagate slower and perhaps infect only a small number of computers at time $t_{\mathrm{E}}$. Such a small number of infected computers may enable the trace-back of the worm propagator.

Therefore, the propagator of a static self-disciplinary worm faces a dilemma, in that 1) reducing the propagation speed will cause it to be traced back; or 2) increasing the propagation speed will cause it to be detected. In the following, we will discuss how the defender can exploit such a dilemma to force a worm propagator to cease its attack.

### *B. Performance Analysis*

The main result for integrating threshold-based detection with trace-back is the following theorem. Recall that $T_{\mathrm{R}}^0$ is the maximum value of the defense strategy $T_{\mathrm{R}}$ which satisfies the maximum tolerable false positive rate $\delta$. Let $p = p_{\mathrm{E}}$ be the solution for $\frac{N \cdot e^{\beta \cdot p \cdot N \cdot t_{\mathrm{E}}}}{e^{\beta \cdot p \cdot N \cdot t_{\mathrm{E}}} + N - 1} \cdot p \cdot v = T_{\mathrm{R}}^0$.

*Theorem 2:* When the worm propagator propagates a static self-disciplinary worm in the system and the defender uses an integration of threshold-based and trace-back schemes, the Nash equilibrium of the game is as follows:

- When $t_{\mathrm{B}} \geq t_{\mathrm{E}}\left(1 - \frac{1}{\log T_{\mathrm{R}}^0} \log \frac{m \cdot N}{N - m}\right) \approx t_{\mathrm{E}}\left(1 - \frac{\log m}{\log T_{\mathrm{R}}^0}\right)$, the worm propagator chooses not to propagate the worm (i.e., $p = 0$). The defender chooses $T_{\mathrm{R}} = T_{\mathrm{R}}^0$.
- Otherwise, the worm propagator chooses $p = p_{\mathrm{E}}$. The defender chooses $T_{\mathrm{R}} = T_{\mathrm{R}}^0$.

*Proof:* We prove the theorem by showing that no player can benefit by unilaterally changing its strategy. We first consider the case where $t_{\mathrm{B}} \geq t_{\mathrm{E}}(1 - \log m / \log T_{\mathrm{R}}^0)$. Apparently, the defender has already reached the maximum possible $u_{\mathrm{D}} = 0$ and cannot benefit by changing its strategy. For the worm propagator, suppose that it changes the propagation strategy to $p = p_1 > 0$. Consider $f(t_{\mathrm{D}} - t_{\mathrm{B}})$, the number of infected computers at time $t_{\mathrm{D}} - t_{\mathrm{B}}$. Let $f_{\mathrm{E}}(t)$ be the function of the number of infected computers when $p = p_{\mathrm{E}}$. We have $f(t_{\mathrm{D}} - t_{\mathrm{B}}) < f_{\mathrm{E}}(t_{\mathrm{E}} - t_{\mathrm{B}}) = \frac{N(e^{\beta \cdot p_{\mathrm{E}} \cdot N t_{\mathrm{E}}})^{1 - \frac{t_{\mathrm{B}}}{t_{\mathrm{E}}}}}{(e^{\beta \cdot p_{\mathrm{E}} \cdot N t_{\mathrm{E}}})^{1 - \frac{t_{\mathrm{B}}}{t_{\mathrm{E}}}} + N}$. Since $t_{\mathrm{B}}/t_{\mathrm{E}} \geq (1 - \log m / \log T_{\mathrm{R}}^0)$, with some mathematical manipulation, we have $f(t_{\mathrm{D}} - t_{\mathrm{B}}) < f_{\mathrm{E}}(t_{\mathrm{E}} - t_{\mathrm{B}}) \leq m$. As such, if the worm propagator changes its strategy to $p > 0$, the defender can always use the forensic analysis scheme to trace-back to the worm propagator with probability of at least 50%. That is, $u_{\mathrm{A}}$ will become $-\infty$. Thus, if $t_{\mathrm{B}} \geq t_{\mathrm{E}}(1 - \log m / \log T_{\mathrm{R}}^0)$, the worm propagator will not change its strategy unilaterally.

When $t_{\mathrm{B}} < t_{\mathrm{E}}(1 - \log m / \log T_{\mathrm{R}}^0)$, the game is exactly the same as the one discussed in Theorem 1, and thus follows the same Nash equilibrium. ■

As we can see from the theorem, there are two possible outcomes of worm propagation:

- Outcome 1. If the trace-back interval $t_{\mathrm{B}}$ is longer than $t_{\mathrm{E}}(1 - \log m / \log T_{\mathrm{R}}^0)$, the integration of threshold-based detection and trace-back will force the worm propagator *not* to propagate the worm.
- Outcome 2. If the trace-back interval $t_{\mathrm{B}}$ is shorter than $t_{\mathrm{E}}(1 - \log m / \log T_{\mathrm{R}}^0)$, the worm will propagate in the same way as we discussed in Section III-C.

The key observation in the above theorem is as follows: With both the threshold-based and trace-back schemes in place, if the worm propagator chooses a larger $p$, it will be detected earlier, and the number of infected computers at time $t_{\mathrm{D}} - t_{\mathrm{B}}$ will be smaller. If the worm propagator chooses a smaller $p$ to delay the detection until $t_{\mathrm{E}}$, the worm will propagate slower and the number of infected computers at time $t_{\mathrm{E}} - t_{\mathrm{B}}$ will still be very small. If the trace-back interval $t_{\mathrm{B}}$ exceeds a threshold such that $f(t_{\mathrm{D}} - t_{\mathrm{B}}) \leq m$ in both cases, then the worm propagator will be forced not to propagate the worm because, otherwise, it will be traced back and punished (i.e., $u_{\mathrm{A}} = -\infty$). However, if $t_{\mathrm{B}}$ does not reach this threshold, the outcome is the same as that of the threshold-based detection scheme alone.

We now analyze which outcome is likely to occur if the trace-back schemes can be widely deployed in practice. In particular, we find that $t_{\mathrm{B}}$ is most likely longer than $t_{\mathrm{E}}(1 - \log m / \log T_{\mathrm{R}}^0)$ in real systems: Consider the system setting specified in Section III-C. In addition, we set $m = 700$. This makes it feasible for the defender to manually check such a small number of computers in order to correctly identify and punish the worm propagator. Due to the theorem, no worm infection will occur if the trace-back interval $t_{\mathrm{B}}$ is longer than 1.81 days. Based on the real-world estimation of trace-back cost [31], deploying a trace-back scheme with interval of 1.81 days requires a one-time cost of approximately \$216,000 per Internet service provider (ISP). Compared with the maintenance cost of ISP, the cost of trace-back is fairly moderate. Thus, an integration of the threshold-based and trace-back schemes can effectively defend again static self-disciplinary as well as traditional worms.

## VI. Integration of Multiple Defensive Schemes to Counteract Dynamic Self-Disciplinary Worms

In this section, we consider a system with a dynamic self-disciplinary worm, which changes its propagation growth rate over time to better adapt to the countermeasures. We first show that the integration of threshold-based and trace-back schemes are no longer effective against dynamic self-disciplinary worms. After that, we introduce a new defensive scheme, called the *spectrum-based scheme*. We demonstrate that an integration of all three schemes can effectively defend against dynamic self-disciplinary worms.

### A. Ineffectiveness of Only Integrating Threshold-Based and Trace-Back Schemes

The main reason why only integrating threshold-based and trace-back schemes is ineffective against dynamic self-disciplinary worms can be stated as follows: To avoid being traced back, a dynamic self-disciplinary worm can propagate itself at full speed in the initial stage of worm propagation until it infects $m$ computers, say at time $t_{\mathrm{A}}$. Since $m$ is usually a small number, the worm is unlikely to be detected by the threshold-scheme at this time. Then, the worm can reduce its propagate speed to delay detection by the threshold-based scheme until at least $t_{\mathrm{A}} + t_{\mathrm{B}}$, which makes the trace-back scheme useless. Since the threshold-based scheme is ineffective against self-disciplinary worms by itself, the defender cannot eliminate worm propagation.

The formal result is presented in the following theorem:

*Theorem 3:* When the worm propagator propagates a dynamic self-disciplinary worm in the system and the defender uses an integration of threshold-based and trace-back schemes, the Nash equilibrium of the game is as follows:

- When $t_{\mathrm{B}} \geq t_{\mathrm{E}} - \frac{\log m}{N \cdot \beta} \approx t_{\mathrm{E}}$, the worm propagator chooses not to propagate the worm (i.e., $p(t) \equiv 0$). The defender chooses $T_{\mathrm{R}} = T_{\mathrm{R}}^0$.
- Otherwise, the worm propagator chooses $p(t) = \min(1, T_{\mathrm{R}}^0/f(t))$ for every $t \in [0, t_{\mathrm{E}}]$. The defender chooses $T_{\mathrm{R}} = T_{\mathrm{R}}^0$.

Please refer to Appendix B for the proof of the theorem. As we can see from the theorem, the threats posed by the trace-back scheme are significantly weakened when the worm is dynamically self-disciplinary. In particular, the possible outcomes of worm propagation become:

- Outcome 1. When the trace-back interval exceeds a very large threshold $t_{\mathrm{E}} - \log m/(N \cdot \beta) \approx t_{\mathrm{E}}$, the worm propagator will be forced not to propagate the worm.
- Outcome 2. When the trace-back interval is lower than the threshold, however, the worm will propagate to more computers than what a static self-disciplinary worm can infect in a system with the threshold-based scheme only.

We now analyze which outcome is likely to occur in practice. In particular, we use examples to demonstrate that the lower bound on trace-back interval $t_{\mathrm{B}}$ in Outcome 1 is very difficult to achieve in real systems: Again, consider the system setting in Sections III-C and Section V. Due to Theorem 3, no worm propagation will occur if and only if the trace-back interval is longer 4.8 days (i.e., $t_{\mathrm{B}} \approx 4.8$ days). According to the estimation of trace-back cost [31], this will incur a one-time cost of at least $\$2,430,000$ per ISP, which is more than 10 times the cost for defending against static self-disciplinary worms, and is apparently unaffordable by many ISPs in practice. Thus, the lower bound on $t_{\mathrm{B}}$ derived in the theorem is unachievable in practice. As such, an integration of the threshold-based and trace-back schemes cannot effectively defend against dynamic self-disciplinary worms.

### B. Spectrum-Based Scheme

A key observation from Theorem 3 is that a dynamic self-disciplinary worm can use full-speed propagation in the beginning to avoid being traced back. To counteract these worms, we introduce a spectrum-based detection scheme to restrict the propagation growth rate of a worm at the initial stage of propagation.

The basic idea of the spectrum-based scheme can be stated as follows: Note that if a worm adopts a high propagation growth rate (e.g., $p(t) = 1$) at the beginning of propagation, the worm-scan traffic will exhibit a highly visible pattern (i.e., trend of exponential increase) when compared with the network background traffic. The objective of spectrum-based detection is to extract such a pattern (as signal) from the normal network traffic (as noise). The idea of using spectrum-based approaches to identify signal from noise has been widely used in the literature of signal processing [32], and has been shown to be capable of distinguishing a signal from noise, even when the signal-to-noise ratio is low.

For the purpose of this paper, we consider the background traffic as white noise in the theoretical analysis. Nevertheless, in our simulation study in Section VII, we use the real-world port-scan traffic logs provided by the SANs Internet storm center (ISC) as the background traffic. Note that our scheme is also ready to work with other cases of Internet background traffic. For example, consider cases where network congestion could possibly cause a high low-frequency component in frequency domain of traffic [33], we can leverage existing work on traffic congestion detection and diagnosis to filter such false alarms of raising worm attack events, maintaining the high worm detection performance [33]–[35].

In particular, we analyze the frequencies contained in the sampled time-series data of scan traffic volume, which is collected by the detection control center. If there is no worm propagation on the network, the background traffic volume, as white noise, should have equal (expected) strengths on all frequency components (i.e., from low to high frequency). If a worm is propagating, however, there will be a relatively abrupt change point and a strong low-frequency component in the frequency domain, because of the continuous and exponential growth of worm-generated traffic volume (which can be considered as having a very relatively long period). Thus, the spectrum-based scheme detects worm propagation by identifying low-frequency components with high power spectrum.

Formally, let $r(t)$ be the traffic volume collected at time $t$. At time $t_0$, the control center has collected a time-series data set $\{r(0), r(1), \ldots, r(t_0)\}$. We transform the time-series data to the frequency domain using the discrete Fourier transform [32] as follows: for all integer $k \in [0, t_0]$, $s(k) = \sum_{n=0}^{t_0} r(n) \cdot e^{-\frac{2\pi i}{t_0+1}kn}$,, where $s(k)$ are the transformed frequency component corresponding to period $2\pi k/(t_0+1)$, and $i$ is the imaginary unit. If $r(t)$ is consisted of white noise only, the expected complex modulus of $s(k)$ (i.e., $|s(k)|$) should be the same for all $k \in [0, t_0]$. Nonetheless, when a worm is propagating, the expected $|s(k)|$ for lower frequencies (i.e., large $k$) will be larger than higher frequencies. Thus, in order to detect worm propagation, we need to measure the differences between $|s(k)|$ for difference frequency ranges.

In particular, we can use a widely adopted measure in pattern recognition called Spectral Flatness Measure (SFM) [36], which is defined as the ratio between the geometric mean and the arithmetic mean of $s(k)$.

$$\mathrm{SFM} = \frac{\left[\prod_{k=0}^{t_0} s(k)\right]^{\frac{1}{t_0+1}}}{\frac{1}{t_0+1}\sum_{k=0}^{t_0} s(k)}, \tag{6}$$

Generally speaking, the smaller SFM is, the more difference there is between $s(k)$ at different frequency ranges [36], and thus the more likely it is that a worm is propagating on the network. As such, our spectrum-based detection scheme issues an alert when the value of SFM is smaller than or equal to a pre-determined threshold $T_{\mathrm{M}}$. Note that the greater $T_{\mathrm{M}}$ is, the more false alarms will be generated by the spectrum-based approach. Thus, the defender must specify the value of $T_{\mathrm{M}}$ (along with $T_{\mathrm{R}}$ for the threshold-based scheme) based on the maximum tolerable false alarm rate $\delta$.

Since the value of SFM decreases when the worm propagator adopts a higher growth rate for a longer period of time, we assume, for the sake of simplicity that at time $t_0$, $\mathrm{SFM} \leq T_{\mathrm{M}}$ if and only if the worm uses $p(t) > p_{\mathrm{M}}$ for a (cumulated) period longer than $\gamma_{\mathrm{M}} \cdot t_0$ time slots ($p_{\mathrm{M}}, \gamma_{\mathrm{M}} \in [0,1]$). The values of $p_{\mathrm{M}}$ and $\gamma_{\mathrm{M}}$ depend on the defender-specified threshold $T_{\mathrm{M}}$. The larger $T_{\mathrm{M}}$ is, the smaller $p_{\mathrm{M}}$ and $\gamma_{\mathrm{M}}$ will be.

Note that this spectrum-based scheme can be easily integrated with the threshold-based and trace-back schemes in Section II-B. In particular, the control center will perform both the threshold-based scheme and the spectrum-based scheme based on collected data, and issue an alert if either scheme generates an alarm. After detecting a propagating worm, the control center initiates the trace-back process automatically.

### *C. Integration of Threshold-Based, Trace-Back, and Spectrum-Based Schemes*

We now show that an integration of the threshold-based, trace-back, and spectrum-based schemes can effectively defend against the propagation of dynamic self-disciplinary worms. In particular, we prove that if the trace-back interval $t_{\mathrm{B}}$ is longer than a (reasonable) threshold, the game will reach Nash equilibrium in the case where the worm propagator will be forced *not* to propagate any (static or dynamic) self-disciplinary worm. Note that with the introduction of the spectrum-based scheme, the strategy set of the defender includes the determination of not only the volume threshold $T_{\mathrm{R}}$ but also the SFM threshold $T_{\mathrm{M}}$. The strategy set of the worm propagator remains the same. As we mentioned in Section VI-B, the false positive rate $\Lambda$ now depends on both $T_{\mathrm{R}}$ and $T_{\mathrm{M}}$.

Let $T_{\mathrm{M}}^0$ be the maximum threshold for the false positive rate to satisfy $\Lambda \leq \delta$ when $T_{\mathrm{R}} = \infty$. Let $p_{\mathrm{M}}^0$ and $\gamma_{\mathrm{M}}^0$ be the corresponding values of $p_{\mathrm{M}}$ and $\gamma_{\mathrm{M}}$ when $T_{\mathrm{M}} = T_{\mathrm{M}}^0$. Suppose that $f_{\mathrm{M}}^0(t)$ is the number of infected computers at time $t$ when no defender exists in the system, and the worm propagator uses

$$p(t) = \begin{cases} 1, & \text{with probability } \gamma_{\mathrm{M}}^0; \\ p_{\mathrm{M}}^0, & \text{with probability } 1-\gamma_{\mathrm{M}}^0. \end{cases} \tag{7}$$

for all $t \in [0, t_{\mathrm{E}}]$. We have the following theorem.

*Theorem 4:* When the worm propagator propagates a dynamic self-disciplinary worm in the system and the defender uses an integration of the threshold-based, trace-back, and spectrum-based schemes, the Nash equilibrium of the game is as follows:

- When $f_{\mathrm{M}}^0(t_{\mathrm{E}} - t_{\mathrm{B}}) \leq m$, the worm propagator chooses not to propagate the worm (i.e., $p(t) \equiv 0$). The defender chooses $T_{\mathrm{R}} = \infty$ and $T_{\mathrm{M}} = T_{\mathrm{M}}^0$.
- Otherwise, the worm propagator chooses

  $$p(t) = \begin{cases} \min(1, T_{\mathrm{R}}/f(t)), & \text{with prob. } \gamma_{\mathrm{M}}; \\ \min(p_{\mathrm{M}}, T_{\mathrm{R}}/f(t)), & \text{with prob. } 1-\gamma_{\mathrm{M}}. \end{cases} \tag{8}$$

  The defender chooses the integration of $T_{\mathrm{R}}$ and $T_{\mathrm{M}}$ that 1) minimizes $f(t_{\mathrm{D}})$ when the worm uses the above strategy, and 2) satisfies $\Lambda \leq \delta$.

Please refer to Appendix C for the proof of the theorem. Due to the theorem, with the integration of all three schemes, there are two possible outcomes of worm propagation:

- Outcome 1. When $t_{\mathrm{B}}$ is greater than the derived threshold (i.e., satisfies $f_{\mathrm{M}}^0(t_{\mathrm{E}} - t_{\mathrm{B}}) \leq m$), the trace-back and spectrum-based schemes will force the worm propagator not to propagate the worm.
- Outcome 2. When $t_{\mathrm{B}}$ does not satisfy the condition, the trace-back scheme poses no threat to the worm propagator. In this case, it is the threshold-based and spectrum-based schemes that force the worm propagator to reduce $p(t)$ to a reasonable level as specified in the theorem.

We now analyze which outcome is likely to occur in practice based on practical examples. In particular, we demonstrate that the derived threshold on the trace-back interval $t_{\mathrm{B}}$ in Outcome 1 is reasonable in many practical systems: We use the same system setting as the one used in Sections III-C, V, and VI-A. Based on the simulation results, there is $T_{\mathrm{M}}^0 = 72,000$, $p_{\mathrm{M}} = 0.22$ and $\gamma_{\mathrm{M}}^0 = 0.5$. Due to the theorem, the worm propagator will not propagate the worm as long as $t_{\mathrm{B}} > 1.8$ days. As we mentioned in Section V, this trace-back interval is reasonable in practice. Thus, the integration of all three schemes can effectively defend again dynamic self-disciplinary worms in the system, as shown in Table I.

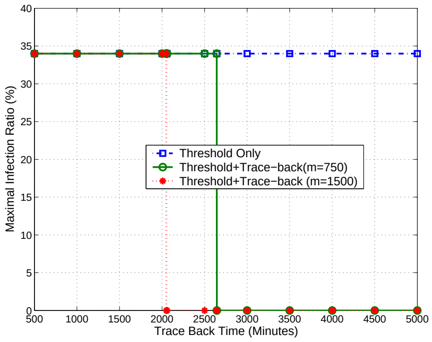

Fig. 2. Maximum infection rate for static self-disciplinary worms

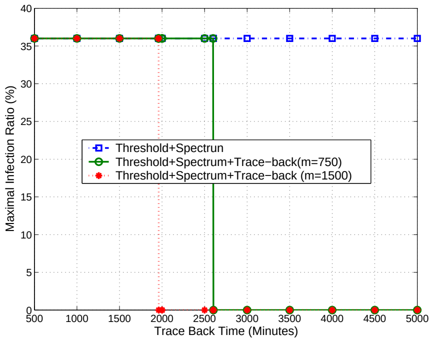

Fig. 3. Maximum infection rate for dynamic self-disciplinary worms

## VII. Simulation Results

In this section, we present the simulation results of systems with static and dynamic self-disciplinary worms. In particular, we conduct the simulation on a combination of real-world background scan traffic and simulated worm-generated traffic.

For the background scan traffic, we use the real-world DShield logs dataset provided by the SANs Internet storm center (ISC) [37]. The dataset contains more than 80 million scan records, with a size of over 80GB. All scan records are captured between January 1, 2005 and January 15, 2005. Each record includes the source IP address, destination IP address, destination port number, and time stamp of a monitored scan.

With the real-world scan traces serving as the background traffic, we add simulated worm generated traffic as follows: We use the same system setting as the one specified in Section III-C: The number of vulnerable computers on the Internet is $350,000$. The total number of IP addresses is $4.3 \times 10^9$. The scan rate of worm propagation is 358 scans/minute. The maximum false positive rate is $2\%$. The maximum propagation time is $t_\mathrm{E} = 5$ days. We conduct the simulation based on various trace-back parameters, with $m \in [750, 5000]$ and the maximum trace-back interval $t_\mathrm{B}$ ranging from $1,400$ to $7,000$ minutes.

We measure the performance of our countermeasures by the maximum infection rate when the worm propagator chooses the optimal strategy of propagation growth rate as specified in the Nash equilibrium. The maximum infection rate is defined as the ratio of the number of infected computers to the total number of vulnerable computers at the moment when the worm is detected, or at time $t_\mathrm{E}$, whichever comes first.

We present the simulation results of our countermeasures on static self-disciplinary and dynamic self-disciplinary worms, respectively. For static self-disciplinary worms, we measure the performance of an integration of the threshold-based and trace-back schemes. We also compare the results with previous approaches that use threshold-based scheme only [14]. The simulation results are shown in Figure 2. As we can see from this figure, when the trace-back interval $t_\mathrm{B}$ is longer than $1.40$ days when $m = 750$ or $1.81$ days when $m = 1500$, the worm propagator will be forced to not propagate the worm. As we discussed in Section V, such trace-back interval is reasonable in practice. Thus, an integration of the threshold-based and trace-back schemes can defend against static self-disciplinary worms effectively. On the other hand, if only the threshold-based scheme is available, the number of infected computers is more than $119,300$ ($34.1\%$ of all vulnerable computers). As we can see, the threshold-based scheme cannot defend against static self-disciplinary worms effectively by itself.

For dynamic self-disciplinary worms, we measure the performance of an integration of threshold-based, trace-back, and spectrum-based schemes. We also compare the results with previous approaches that use the threshold-based scheme only [14] plus the spectrum-based scheme shown in Section VI.B. The simulation results are shown in Figure 3. As we can see from this figure, when the trace-back interval $t_\mathrm{B}$ is longer than $1.32$ days when $m = 750$ or $1.76$ days when $m = 1500$, the worm propagator will be forced to not propagate the worm. As we discussed in Section V, such trace-back interval is reasonable in practice. Thus, an integration of all three schemes can effectively defend against dynamic self-disciplinary worms.

In Figure 4, we investigate the relationship between the maximum infection rate and the maximum tolerable false positive rate $\delta$ when the trace-back interval $t_B$ is not enough to eliminate worm propagation. As we can see from the figure, for the defense against both the static self-disciplinary worm and dynamic self-disciplinary worm, the more false alarms the system can tolerate, the less computers a dynamic self-disciplinary worm can infect. In particular, for the defense against the dynamic self-disciplinary worm, when the maximum tolerable false positive rate increases from $1\%$ to $8\%$, the maximum information rate decreases from $36\%$ to $12\%$ correspondingly.

In Figure 5, we also demonstrate the relationship between the minimum required trace-back interval $t_B$ to eliminate worm propagation and the maximum number of computers for manual check in trace-back in the cases of defending against the static self-disciplinary worm and defending against the dynamic self-disciplinary worm, respectively. As we can see from the figure, the larger the number of computers the system allows to carry out a manual check for trace back, the less trace-back interval $t_B$ can be achieved for the system to eliminate worm propagation. For example, for the static self-disciplinary worm, when the number of suspect computers for manual check increases from 1000 to 2000, the minimum required trace-back interval $t_B$ decreases from 1.50 days to 1.32 days correspondingly. From this figure, we know that minimum trace-back interval for the dynamic self-disciplinary worm is comparatively shorter than that for the static self-disciplinary worm. This is because the spectrum-based detection scheme can effectively suppress the worm propagation

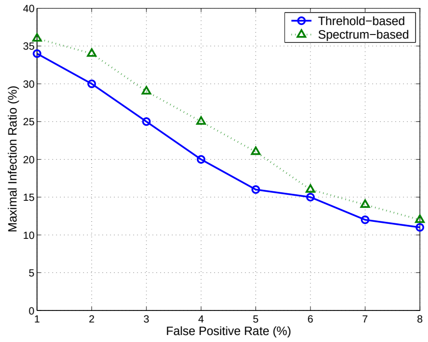

Fig. 4. Relationship between maximum infection rate and maximum false positive rate

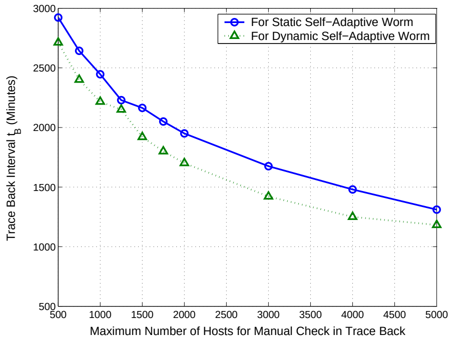

Fig. 5. Maximum infection rate for dynamic self-disciplinary worms

growth rate in the early stage of the worm propagation.

## VIII. Discussion

### A. Generalized Utility Function

We now discuss how to generalize the utility function of the worm propagator which we proposed in Section IV-C. Note that in Section IV-C, we assumed that the worm propagator either receives infinite penalty from trace-back (i.e., $u_{\mathrm{A}} = -\infty$ when $f(t_{\mathrm{D}} - t_{\mathrm{B}}) > m$), or none at all (when $f(t_{\mathrm{D}} - t_{\mathrm{B}}) > m$). In practice, however, different worm propagators may evaluate the risk of being traced back. Some risk-averse worm propagators may stop propagating the worm when the probability of being traced back is 10%, while others may choose to propagate regardless of whether or not they will be traced back. Thus, we generalize the utility function of a worm propagator to a continuous function, in order to model the threats from worm propagators with different risk aversion levels.

In particular, let $h(x)$ be the loss of the worm propagator if the defender can trace-back to $x$ infected computers at the earliest trace-back time $\max(0, t_{\mathrm{D}} - t_{\mathrm{B}})$. Apparently, $h(x)$ should be monotonically decreasing with $x$, as a larger $x$ makes it more difficult to identify the worm propagator. Let $\alpha > 0$ be a preferential parameter pre-determined by the worm propagator. The generalized objective of a worm propagator is to maximize

$$U_{\mathrm{A}} = f(t_{\mathrm{D}}) - \alpha \cdot h(f(\max(0, t_{\mathrm{D}} - t_{\mathrm{B}}))). \tag{9}$$

As we can see, our utility function defined in Section IV-C is a special case of this generalized version when $h$ is defined as follows:

$$h(x) = \begin{cases} \infty, & \text{if } x \le m; \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Given the generalized utility function, Theorem 2 and Theorem 4 can be restated as follows:

*Theorem 5:* When the worm propagator propagates a static self-disciplinary worm in the system and the defender uses an integration of the threshold-based and trace-back schemes, the Nash equilibrium of the game is as follows:

- When $\alpha \cdot h\left(\frac{N(e^{\beta \cdot p_{\mathrm{E}} \cdot N t_{\mathrm{E}}})^{1-q}}{(e^{\beta \cdot p_{\mathrm{E}} \cdot N t_{\mathrm{E}}})^{1-q} + N}\right) > p_{\mathrm{E}} T_{\mathrm{R}}^{0}$, the worm propagator chooses not to propagate the worm (i.e., $p = 0$). The defender chooses $T_{\mathrm{R}} = T_{\mathrm{R}}^{0}$.
- Otherwise, the worm propagator chooses $p = p_{\mathrm{E}}$. The defender chooses $T_{\mathrm{R}} = T_{\mathrm{R}}^{0}$.

*Theorem 6:* When the worm propagator propagates a dynamic self-disciplinary worm in the system and the defender uses an integration of the threshold-based, trace-back, and spectrum-based schemes, the Nash equilibrium of the game is as follows:

- If there exists $T_{\mathrm{M}}$ and $T_{\mathrm{R}}$ such that 1) $\alpha \cdot h(f(t_{\mathrm{E}} - t_{\mathrm{B}})) > p_{\mathrm{E}} T_{\mathrm{R}}$, and 2) the false positive rate $\Lambda < \delta$, then the worm propagator chooses not to propagate the worm (i.e., $p(t) \equiv 0$). The defender chooses the corresponding $T_{\mathrm{R}}$ and $T_{\mathrm{M}}$.
- Otherwise, the worm propagator chooses

  $$p(t) = \begin{cases} \min(1, T_{\mathrm{R}}/f(t)), & \text{with Prob.} \gamma_{\mathrm{M}}; \\ \min(p_{\mathrm{M}}, T_{\mathrm{R}}/f(t)), & \text{with Prob.} 1 - \gamma_{\mathrm{M}}. \end{cases} \tag{11}$$

  The defender chooses the integration of $T_{\mathrm{R}}$ and $T_{\mathrm{M}}$ that 1) minimizes $f(t_{\mathrm{D}})$ when the worm uses the above strategy, and 2) satisfies $\Lambda \le \delta$.

The basic idea for proving the above two theorems is similar to the proof of Theorem 2 and Theorem 4. The optimal strategy for the worm propagator is to select the maximum propagation growth rate $p$ or $p(t)$ that delays the detection time to $\mathrm{t_E}$. The condition for a static self-disciplinary worm to stop propagation is to make the utility function, defined in (11) less than 0.

### B. Self-Adaptation as a General Principle

In this paper, we focus on a new class of worms, referred to as *self-disciplinary worms*. These worms adapt their propagation traffic patterns in order to reduce the probability of detection, and to eventually infect more computers. The self-disciplinary worm is different from the polymorphic worms that deliberately change their *payload signatures* during the propagation [38], [39]. For example, MetaPHOR [40] and Zmist [41]) worms intensively metamorphose their payload signature to hide themselves from detection schemes that rely on expensive packet payload analysis. Bethencourt *et al.* studied worms which employed private information retrieval techniques to find and retrieve specific pieces of sensitive information from compromised computers while hiding their search criteria [42]. Sharif *et al.* [43] presented an obfuscation technique that automatically conceals specific condition dependent malicious behavior from worm detector that have no prior knowledge of program inputs. Popov *et al.* [44] investigated a technique that allows the worm programs to be obfuscated by changing many control transfers into signals (traps) and inserting dummy control transfers and "junk" instructions after the signals. The resulted code can significantly reduce the chance to be detected. Worm might also other evasive scan [45] and traffic morphing techniques to hide the detection [46].

## IX. Related Work

In this section, we first introduce the modeling of worm propagation. Then we explore work related to worm detection and network forensics, followed by the work related to game-theoretic studies for network security.

Due to substantial damage caused by worms in past years, there have been significant efforts on modeling and detecting worms as well as other defenses. With the help of worm modeling, effective detection and defense schemes could be further developed to mitigate worms' impact; hence, tremendous research effort has focused on this area [2], [47]–[50]. extensive work on the propagation of specific worms. For example, All these studies focus on traditional worms discussed in Section II-A.1, while our study focuses on modeling the propagation of self-disciplinary worms that can dynamically manipulate the propagation traffic pattern to reduce the probability of being detected. The "self-stopping" worms investigated in [10], [12] are special cases of self-disciplinary worms studied in this paper.

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 [18], [19], which mainly focus on detecting worms via software anomalies. For example, Wang *et al.* [18] 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.* [19] 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. 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, Jung *et al.* [51] developed a threshold-based detection algorithm to identify anomalous scan traffic generated by a computer. Venkataraman *et al.* and Wu *et al.* [13], [14] proposed schemes to examine statistics of scan traffic volume. Zou *et al.* presented a trend-based detection scheme to examine the exponentially-increasing scan-traffic pattern [52]. Lakhina *et al.* [53] proposed a scheme to examine other features of scan traffic, such as the distribution of destination addresses. Perdisci *et al.* [38] studied worms that attempt to "take on" new payload patterns to avoid detection. port [54]. Besides the network-based detection, there is also extensive research on post-detection defense, including containment, throttling, and filtering [17], [55].

Forensic analysis is another critical countermeasure to defend against worm attacks by tracing network logs to identify attack origins. Many forensic analysis mechanisms have been proposed [15], [26], [27]. For example, in order to render Internet forensic analysis feasible, Xie *et al.* [31] proposed an infrastructure in which distributed collection points record log traffic and store them in repositories for querying. Carrier *et al.* [56] presented a protocol that can assist in the forensic analysis of a computer involved in malicious network activity. Their system was designed to help automate the process of tracing attackers who log on to a series of hosts to hide their identity. Xie *et al.* [15] studied a random-walk algorithm to determine the computer responsible for worm propagation, using an insight that the propagation traffic forms a causal tree with root at the source of worm attacks.

The application of game theory has also been extensively studied in distributed systems and network security research [57]–[59]. For example, Moscibroda *et al.* [60] applied the game-theory to study how much the presence of selfish/Byzantine players can deteriorate or improve the virus propagation over a specific grid topology. In contrast, our work investigates a general form of worms that can dynamically manipulate the overall propagation patterns and analyze the performance of integrated countermeasures against such worms over the Internet.

## X. Final Remarks

In this paper, we modeled a new class of worms called self-disciplinary worms, which adapt their propagation pattern in order to avoid detection. Based on the degree of control of the propagation growth rate, we classified self-disciplinary worms into two categories: static self-disciplinary worms and dynamic self-disciplinary worms. We demonstrated that existing worm detection schemes based on traffic volume and variance are insufficient to detect self-disciplinary worms. Based on a game-theoretic formulation of the interaction between the worm propagator and the defender, we showed that an effective integration of multiple defensive schemes is critical for defending against self-disciplinary worms. In particular, we considered three schemes: threshold-based scheme, trace-back scheme, and spectrum-based scheme. We showed that the combination of the first two schemes can effectively defend against static self-disciplinary worms, while the combination of all three schemes can effectively defend against dynamic self-disciplinary worms. This paper lays the foundation for ongoing studies of "smart" worms that intelligently adapt their propagation patterns to countermeasures.

## Acknowledgments

We thank the anonymous reviewers for their invaluable feedback. This work was supported in part by the National Science Foundation under grants 0808419, 0324988, 0721571, 0329181, 0721783, 0747150, 0721766 and 0722856. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors would like to acknowledge Ms. Larisa Archer for her dedicated help to improve the paper.

# References

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

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

[3] M. Casado, T. Garfinkel, W. Cui, V. Paxson, and S. Savage, "Opportunistic measurement: Extracting insight from spurious traffic," in *Proceedings of the 4th ACM SIGCOMM HotNets Workshop (HotNets)*, College Park, MD, November 2005.

[4] J. Mirkovic, G. Prier, and P. Reiher, "Attacking ddos at source," in *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP)*, Paris, France, November 2002.

[5] Y. Pan and X. Ding, "Anomaly based web phishing page detection," in *Proceedings of the 22th Annual Computer Security Applications Conference (ACSAC)*, Miami Beach, FL, November 2006.

[6] B. Leiba and N. Borenstein, "A multifaceted approach to spam reduction," in *Proceedings of the 1st Conference on Email and Anti-Spam*, Mountain View, CA, July 2004.

[7] J. Binkley and S. Singh, "An algorithm for anamoly-based botnet detection," in *Proceedings of the 2nd Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, San Jose, CA, July 2006.

[8] E. Cooke and F. Jahanian, "The zombie roundup: Understanding, detecting, and disrupting botnets," in *Proceedings of the 1st Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, Cambridge, MA, July 2005.

[9] F. C. Freiling, T. Holz, and G. Wicherski, "Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks," in *Proceedings of the 10th European Symposium on Research in Computer Security (ESORICS)*, Milan, Italy, September 2005.

[10] R. Vogt, J. Aycock, and M. Jacobson, "Quorum sensing and self-stopping worms," in *Proceedings of 5th ACM Workshop on Recurring Malcode (WORM)*, Alexandria VA, October 2007.

[11] Zdnet, *Smart worm lies low to evade detection*, `http://news.zdnet.co.uk/internet/security/0,39020375,39160285,00.htm`, 2005.

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

[13] J. Wu, S. Vangala, and L. X. Gao, "An effective architecture and algorithm for detecting worms with various scan techniques," in *Proceedings of the 11th IEEE Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Febrary 2004.

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

[15] Y. Xie, V. Sekar, D. A. Maltz, M. K. Reiter, and H. Zhang, "Worm origin identification using random moonwalks," in *Proceeding of the IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 2005.

[16] A. Ahmad and A. B. Ruighaver, "Design of a network-access audit log for security monitoring and forensic investigation," in *Proceedings of the 1st Australian Computer Network, Information & Forensics Conference*, Western Australia, Australia, November 2003.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 2001.

[22] M. Christodorescu, S. Jha, S. A. Seshia, D. Song, and R. E. Bryant, "Semantics-aware malware detection," in *Proceedings of IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 2005.

[23] SANS, *Internet Storm Center*, `http://isc.sans.org/`, 2004.

[24] V. Yegneswaran, P. Barford, and D. Plonka, "On the design and utility of internet sinks for network abuse monitoring," in *Proceeding of Symposium on Recent Advances in Intrusion Detection (RAID)*, Pittsburgh, PA, September 2003.

[25] D. Moore, "Network telescopes: Observing small or distant security events," in *Invited Presentation at the 11th USENIX Security Symposium (SEC)*, San Francisco, CA, August 2002.

[26] X. Wang and D. S. Reeves, "Robust correlation of encrypted attack traffic through stepping stones by manipulation of inter-packet delays," in *Proceedings of the 2003 ACM Conference on Computer and Communications Security (CCS)*, Washingtion, DC, November 2003.

[27] W. Yu, X. Fu, S. Graham, D. Xuan, and W. Zhao, "Dsss-based flow marking technique for invisible traceback," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 2007.

[28] D. J. Daley and J. Gani, *Epidemic Modeling: An Introduction*, Cambridge Univ. Press, 1999.

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

[30] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[31] V. Sekar, Y. Xie, D. Maltz, M. Reiter, and H. Zhang, "Toward a framework for internet forensic analysis," in *Proceeding of the 3rd Workshop on Hot Topics in Networks (HotNets)*, San Diego, CA, November 2004.

[32] R. L. Allen and D. W. Mills, *Signal Analysis: Time, Frequency, Scale, and Structure*, Wiley and Sons, 2004.

[33] M. S. Kim, T. Kim, Y. J. Shin, S. S. Lam, and E. J. Powers, "A wavelet-based approach to detect shared congestion," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 4, pp. 293–306, 2004.

[34] Y. Zhao, Y. Chen, and D. Bindel, "Towards unbiased end-to-end network diagnosis," in *Proceeding of ACM SIGCOMM*, Pisa, Italy, Septermber 2006.

[35] H. Balakrishnan, S. Seshan, and H. Rahul, "An integrated congestion management architecture for internet hosts," in *Proceeding of ACM SIGCOMM*, Cambridge, MA, Septermber 1999.

[36] R. E. Yantorno, K. R. Krishnamachari, J. M. Lovekin, D. S. Benincasa, and S. J. Wenndt, "The spectral autocorrelation peak valley ratio (sapvr) - a usable speech measure employed as a co-channel detection system," in *Proceedings of IEEE International Workshop on Intelligent Signal Processing (WISP)*, Budapest, Hungary, May 2001.

[37] DShield.org, *Distributed Intrusion Detection System*, `http://www.dshield.org/`, 2004.

[38] 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.

[39] D. Bruschi, L. Martignoni, and M. Monga, "Detecting self-mutating malware using control flow graph matching," in *Proceedings of the Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, Berlin, Germany, July 2006.

[40] MetaPHOR, `http://securityresponse.symantec.com/avcenter/venc/data/w32.simile.html`.

[41] P. Ferrie and P. Ször. Zmist, *Zmist opportunities*, Virus Bullettin, `http://www.virusbtn.com`.

[42] J. Bethencourt, D. Song, and B. Waters, "Analysis-resistant malware," in *Proceedings of the 15th IEEE Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Febrary 2008.

[43] M. Sharif, J. Giffin, W. Lee, and A. Lanzi, "Impeding malware analysis using conditional code obfuscation," in *Proceedings of the 15th IEEE Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Febrary 2008.

[44] I. V. Popov, S. K. Debray, and G. R. Andrews, "Binary obfuscation using signals," in *Proceedings of the 17th USENIX Security Symposium (SECURITY)*, San Jose, CA, July 2008.

[45] M. G. Kang, J. Caballero, and D. Song, "Distributed evasive scan techniques and countermeasuress," in *Proceedings of International Conference on Detection of Intrusions & Malware, and Vulnerability Assessment (DIMVA)*, Lucerne, Switzerland, July 2007.

[46] C. Wright, S. Coull, and F. Monrose, "Traffic morphing: An efficient defense against statistical traffic analysis," in *Proceedings of the 15th*

*IEEE Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Febrary 2008.

[47] 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.

[48] Y. Li, Z. Chen, and C. Chen, "Understanding divide-conquer-scanning worms," in *Proceedings of International Performance Computing and Communications Conference (IPCCC)*, Austin, TX, December 2008.

[49] D. Ha and H. Ngo, "On the trade-off between speed and resiliency of flash worms and similar malcodes," in *Proceedings of 5th ACM Workshop on Recurring Malcode (WORM)*, Alexandria VA, October 2007.

[50] Y. Yang, S. Zhu, and G. Cao, "Improving sensor network immunity under worm attacks: A software diversity approach," in *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, Hong Kong, May 2008.

[51] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan, "Fast portscan detection using sequential hypothesis testing," in *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, Oakland, CA, May 2004.

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

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

[54] G. F. Gu, D. Dagon, M. I. Sharif X. Z. Qin, W. Lee, and G. F. Riley, "Worm detection, early warning, and response based on local victim information," in *Proceedings of Proceedings of the 20th Annual Computer Security Applications Conference (ACSAC)*, Tucson, Arizona, December 2004.

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

[56] B. Carrier and C. Shields, "The session token protocol for forensics and traceback," *ACM Transactions on Information and System Security (TISSEC)*, vol. 7, no. 3, pp. 332–362, 2004.

[57] P. Liu, W. Y., and M. Yu, "Incentive-based modeling and inference of attacker intent, objectives, and strategies," *ACM Transaction on Information System and Security*, vol. 8, no. 1, pp. 78–118, 2005.

[58] W. Yu and K. J. R. Liu, "Game theoretic analysis of cooperation stimulation and security in autonomous mobile ad hoc networks," *IEEE Transaction on Mobile Computing*, vol. 6, no. 5, pp. 459–473, 2007.

[59] Y. Liu, C. Comaniciu, and H. Man, "A bayesian game approach for intrusion detection in wireless ad hoc networks," in *Proceedings of the 2006 Workshop on Game Theory for Communications and Networks*, Pisa, Italy, 2006.

[60] T. Moscibroda, S. Schmid, and R. Wattenhofer, "When selfish meets evil: Byzantine players in a virus inoculation game," in *Proceeding of the 25th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, Denver, CO, July 2006.

[61] J. Farlow, J. E. Hall, J. M. McDill, and B. H. West, *Differential Equations and Linear Algebra*, Prentice-Hall, Inc, 2002.

## APPENDIX

### A. Derivation of $f(t)$

To derive the relationship between $f(t)$ and other system parameters, we take an approach similar to the epidemic model used for analyzing traditional worms [28]. Note that

$$f(t+\Delta t) = f(t) + X(t,\Delta t), \tag{12}$$

where $X(t,\Delta t)$ is the number of computers infected during time internal $(t, t+\Delta t]$. $X(t,\Delta t)$ can be estimated as:

$$\begin{aligned} X(t,\Delta t) &= (\text{Number of worm scans in } (t,t+\Delta t]) && (13)\\ &\cdot(\text{Success rate of each scan})\cdot\Delta t. && (14)\end{aligned}$$

When $\Delta t \to 0$, the number of scans made during $(t, t+\Delta t]$ tends to be $S \cdot p \cdot f(t) \cdot \Delta t$. Let $V$ and $N$ be the total number of IP addresses and vulnerable computers on the Internet, respectively. At time $t$, the number of computers that are vulnerable to infection is $N - f(t)$. Then, the success rate of a scan is $(N - f(t))/V$. Due to (15), we have

$$X(t,\Delta t) = S \cdot p \cdot f(t) \cdot \frac{N - f(t)}{V}\Delta t. \tag{15}$$

Let $\beta = S/V$. $\beta$ is commonly referred to as the *pairwise propagation rate* in the epidemic research literature [28]. Substituting (17) into (14), we have $\frac{df(t)}{dt} = \lim_{\Delta t\to 0}\frac{f(t+\Delta t)-f(t)}{\Delta t} = \beta\cdot p\cdot f(t)\cdot(N - f(t))$. As we can see, this is a differential equation of $f(t)$ in terms of system parameters $\beta$, $p$, and $N$. With the initial condition in (2), the equation 3 can be solved using Laplace transform [61].

### B. Proof of Theorem 3

*Proof:* We first consider the case where $t_{\rm B} \geq t_{\rm E} - \log m/(N\cdot\beta)$. In this case, the proof of Nash equilibrium is similar to that of Theorem 2. Thus, we only demonstrate why the lower bound on $t_{\rm B}$ changes to $t_{\rm E} - \log m/(N\cdot\beta) \approx t_{\rm E}$. Consider the case where the worm propagator adopts a strategy as follows:

1) First, the worm propagator uses $p(t) = \min(1, T_{\rm R}^0/f(t))$ to infect $m$ computers as soon as possible, say at time $t_{\rm A}$ (i.e., $f(t_{\rm A}) = m$).
2) After that, the worm propagator chooses $p(t) = 0$.

As we can see, since $m \ll N$, the worm will not be detected before $t_{\rm A}$. Thus, the worm propagator cannot be traced back as long as $t_{\rm A} < t_{\rm E} - t_{\rm B}$. As such, in order to force the worm propagator not to propagate the worm, there must be $f(t_{\rm E} - t_{\rm B}) \leq m$ for the above strategy. That is, $\frac{N\cdot e^{\beta\cdot N(t_{\rm E}-t_{\rm B})}}{e^{\beta\cdot N(t_{\rm E}-t_{\rm B})}+N} \leq m$. With some mathematical manipulation, we have $t_{\rm B} \geq t_{\rm E} - \frac{1}{\beta\cdot N}\log\frac{N\cdot m}{N-m} \approx t_{\rm E} - \frac{\log m}{\beta\cdot N} \approx t_{\rm E}$. Thus, a necessary condition to force the worm propagator not to propagate the worm is $t_{\rm B} \geq t_{\rm E} - (\log m)/(N\cdot\beta) \approx t_{\rm E}$.

We now consider the case where $t_{\rm B} < t_{\rm E} - \log m/(N\cdot\beta)$. In particular, we prove the correctness of the Nash equilibrium specified in the theorem by showing that no player can benefit by unilaterally changing its strategy. As we have shown in Theorem 2, the defender cannot benefit by deviating from $T_{\rm R} = T_{\rm R}^0$. For the worm propagator, suppose that it uses a different propagation growth rate function $p_1(t)$. In order for the worm propagator to benefit from the strategy change, there must exist $t_1 \in [0, t_{\rm E}]$ such that $p_1(t_1) > p(t_1) = \min(1, T_{\rm R}^0/f(t))$. Nevertheless, the worm will then be detected at time $t_1$ due to the threshold-based scheme, resulting in a reduced $u_{\rm A}$. Thus, no player can benefit by changing its strategy unilaterally from the equilibrium specified in the theorem. ■

### C. Proof of Theorem 4

*Proof:* We first consider the case where $f_{\rm M}^0(t_{\rm E} - t_{\rm B}) \leq m$. Apparently, the defender already reaches the maximum possible $u_{\rm D} = 0$ and cannot benefit by changing its strategy. For the worm propagator, suppose that it changes the propagation growth rate function to $p_1(t)$. Let the changed function of the number of infected computers be $f_1(t)$. Due to the definition of spectrum-based scheme and $f_{\mathrm{M}}^{0}(t)$, there must be $f_1(t) \leq f_{\mathrm{M}}^{0}(t)$ for all $t \in [0, t_{\mathrm{E}}]$. Thus, $f_1(t_{\mathrm{E}} - t_{\mathrm{B}}) \leq f_{\mathrm{M}}^{0}(t_{\mathrm{E}} - t_{\mathrm{B}}) \leq m$. That is, the worm propagator will be traced back with probability of at least $50\%$, resulting in $u_{\mathrm{A}} = -\infty$. As such, the worm propagator cannot benefit by changing its strategy unilaterally.

We now consider the case where $f_{\mathrm{M}}^{0}(t_{\mathrm{E}} - t_{\mathrm{B}}) > m$. Note that in order to avoid being detected by the threshold-based scheme, the worm propagator must maintain $p(t) \leq T_{\mathrm{R}}/f(t)$. Based on our discussion above, it is easy to verify that the worm propagator cannot benefit by changing its strategy unilaterally. For the defender, if it changes either $T_{\mathrm{R}}$ or $T_{\mathrm{M}}$, there will be only two possible outcomes: 1) an increased $f(t_{\mathrm{D}})$, and/or 2) $\Lambda > \delta$. Either way, the defender will have a decreased utility function $u_{\mathrm{D}}$. Thus, the defender cannot benefit by changing its strategy unilaterally. ■

**Wei Yu** Dr. Wei Yu is an Assistant Professor in the Department of Computer and Information Sciences, Towson University. He received the BS degree in Electrical Engineering from Nanjing University of Technology in 1992, the MS degree in Electrical Engineering from Tongji University in 1995, and the PhD degree in computer engineering from Texas A&M University in 2008. He worked for Cisco Systems Inc. for nine years. His research interests include cyber space security, computer network, and distributed systems.

**Nan Zhang** Dr. Nan Zhang is an Assistant Professor of Computer Science at the George Washington University. He received the BS degree from Peking University in 2001 and the PhD degree from Texas A&M University in 2006, both in computer science. His current research interests include security and privacy issues in databases, data mining, and computer networks, in particular privacy and anonymity in data collection, publishing, and sharing, privacy-preserving data mining, and wireless network security and privacy.

**Xinwen Fu** Dr. Xinwen Fu is an assistant professor in the Department of Computer Science, University of Massachusetts Lowell. He received his BS (1995) and MS (1998) in Electrical Engineering from Xi'an Jiaotong University, China and University of Science and Technology of China respectively. He obtained his PhD (2005) in Computer Engineering from Texas A&M University. From 2005 to 2008, he was an assistant professor with the College of Business and Information Systems at Dakota State University. In summer 2008, he joined University of Massachusetts Lowell as a faculty member. Dr. Fu has been publishing papers in prestigious conferences such as S&P, INFOCOM and ICDCS, journals such as TPDS, and book chapters. His group won the best paper award at International Conference on Communications (ICC) 2008. His current research interests are in network security and privacy.

**Wei Zhao** Dr. Wei Zhao is currently the Rector of the University of Macau. Before joining the University of Macau, he served as the Dean of the School of Science at Rensselaer Polytechnic Institute. Between 2005 and 2006, he served as the director for the Division of Computer and Network Systems in the US National Science Foundation when he was on leave from Texas A&M University, where he served as Senior Associate Vice President for Research and Professor of Computer Science. Dr. Zhao completed his undergraduate program in physics at Shaanxi Normal University, Xian, China, in 1977. He received the MS and PhD degrees in Computer and Information Sciences at the University of Massachusetts at Amherst in 1983 and 1986, respectively. Since then, he has served as a faculty member at Amherst College, the University of Adelaide, and Texas A&M University. As an elected IEEE fellow, Wei Zhao has made significant contributions in distributed computing, real-time systems, computer networks, and cyber space security.