10-15-2010

Characterizing InternetWorm Spatial-Temporal Infection Structures

Qian Wang
Florida International University, qian.wang@fiu.edu

DOI: 10.25148/etd.FI10120303

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

CHARACTERIZING INTERNET WORM SPATIAL-TEMPORAL INFECTION STRUCTURES

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL ENGINEERING

by

Qian Wang

2010

To: Dean Amir Mirmiran
College of Engineering and Computing

This dissertation, written by Qian Wang, and entitled Characterizing Internet Worm Spatial-Temporal Infection Structures, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

Zesheng Chen

Kia Makki

Jean Andrian

Jason Liu

Deng Pan

Niki Pissinou, Major Professor

Date of Defense: October 15, 2010

The dissertation of Qian Wang is approved.

Dean Amir Mirmiran
College of Engineering and Computing

Interim Dean Kevin O'Shea
University Graduate School

Florida International University, 2010

DEDICATION

To mentor, advisor, and friend,

Dr. Zesheng Chen,

and to my parents,

Mingchun Jiang and Yuanqin Wang.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest gratitude to Dr. Zesheng Chen, without whose incisive and enormous guidance this work would not have been possible. Dr. Chen's intuitive way of solving difficult problems with simple and elegant solutions will always be an immense source of inspiration and encouragement to me. Moreover, his simple, humble, and hard-working way of life sets me a Christian example to follow.

I would like to thank Dr. Niki Pissinou and Dr. Kia Makki for offering me the opportunity to pursue my doctoral degree and freedom to conduct research. I would also like to thank them for their encouraging advice. Many thanks to Dr. Chao Chen for her countless manuscript revisions and invaluable feedback. I also appreciate Dr. Jason Liu, Dr. Jean H. Andrian and Dr. Deng Pan for serving on my dissertation defense committee and giving enlightening comments.

I would like to thank all the members of the Telecommunications and Information Technology Institute for their help and friendship during my study. Special thanks go to Kai Chen, Dr. Xiaowen Zhang, Dr. Vinayak Ganapathy, Xinyu Jin, and Dr. Qutub Bakhtiar. I am also grateful to brothers and sisters in the Noah Chinese Christian Fellowship. Thanks to Florida International University for supporting my research, and this great country, the United States of America, for offering me a new way of life.

Finally, I thank my parents, Mingchun Jiang and Yuanqin Wang, for their unconditional love, and Jesus Christ, who loves me and saved me, for leading me both now and throughout eternity.

ABSTRACT OF THE DISSERTATION

CHARACTERIZING INTERNET WORM SPATIAL-TEMPORAL INFECTION STRUCTURES

by

Qian Wang

Florida International University, 2010

Miami, Florida

Professor Niki Pissinou, Major Professor

Since the Morris worm was released in 1988, Internet worms continue to be one of top security threats. For example, the Conficker worm infected 9 to 15 million machines in early 2009 and shut down the service of some critical government and medical networks. Moreover, it constructed a massive peer-to-peer (P2P) botnet. Botnets are zombie networks controlled by attackers setting out coordinated attacks. In recent years, botnets have become the number one threat to the Internet.

The objective of this research is to characterize spatial-temporal infection structures of Internet worms, and apply the observations to study P2P-based botnets formed by worm infection.

First, we infer temporal characteristics of the Internet worm infection structure, *i.e.,* the host infection time and the worm infection sequence, and thus pinpoint patient zero or initially infected hosts. Specifically, we apply statistical estimation techniques on Darknet observations. We show analytically and empirically that our proposed estimators can significantly improve the inference accuracy. Second, we reveal two key spatial characteristics of the Internet worm infection structure, *i.e.,* the number of children and the generation of the underlying tree topology formed by worm infection. Specifically, we apply probabilistic modeling methods and a sequential growth model. We show analytically and empirically that the number of children has asymptotically a geometric distribution with parameter 0.5,

and the generation follows closely a Poisson distribution. Finally, we evaluate bot detection strategies and effects of user defenses in P2P-based botnets formed by worm infection. Specifically, we apply the observations of the number of children and demonstrate analytically and empirically that targeted detection that focuses on the nodes with the largest number of children is an efficient way to expose bots. However, we also point out that future botnets may self-stop scanning to weaken targeted detection, without greatly slowing down the speed of worm infection. We then extend the worm spatial infection structure and show empirically that user defenses, *e.g.,* patching or cleaning, can significantly mitigate the robustness and the effectiveness of P2P-based botnets. To counterattack, we evaluate a simple measure by future botnets that enhances topology robustness through worm re-infection.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

Ω	Size of the scanning space ($\Omega = 2^{32}$)
ω	Size of the Darknet
s	Scanning rate (scans/time unit)
σ	Standard deviation of the scanning rate
p_a	Probability that an IP address with the same first l bits as the attacking host is chosen by LS
p	Probability that at least one scan from the same infected host hits the Darknet in a time unit
t_0	Host infection time
$\hat{t}_0$	Estimated host infection time
t_i	Discrete time tick when the infected host hits the Darknet for the i-th time $(i \geq 1)$
δ_i	Time interval between two consecutive hits of the Darknet ($\delta_i = t_{i+1} - t_i$, $i \geq 1$)
n	Number of hit events observed at the Darknet for an infected host (Chapter 3) / total number of vulnerable hosts (Chapter 4)
μ	Mean of δ
$\hat{\mu}$	Estimation of μ
D	Sequence distance
S_i	Worm infection sequence
$\hat{S}_i$	Estimated worm infection sequence
N	Length of the worm infection sequence considered for evaluation
$L_n(i, j)$	Number of nodes that have i children and belong to generation j
$C_n(i)$	Number of nodes that have i children
$G_n(j)$	Number of nodes in generation j
$p_n(i, j)$	Joint distribution of the number of children and the generation
$c_n(i)$	Distribution of the number of children
$g_n(j)$	Distribution of the generation

A	Accessed bot ratio
D_R	Average percentage of bots that can be exposed by random detection
D_T	Average percentage of bots that can be exposed by targeted detection
m	Number of hosts that an infected host has compromised to stop scanning
n_0	Total number of vulnerable hosts
r_p	Patching rate: the rate at which an infected or vulnerable machine becomes invulnerable
r_c	Cleaning rate: the rate at which the infection is cleaned on a machine without patching
n_d	Number of hosts that get patched or cleaned
n_r	Number of remaining infected hosts after n_d hosts get patched or cleaned
t_r	Number of trees in the worm forest
$B_{n_0}^{n_d}(i)$	Number of nodes that have i peers
$T_{n_0}^{n_d}(j)$	Number of trees that have j nodes
$b_{n_0}^{n_d}(i)$	Distribution of the number of peers
$t_{n_0}^{n_d}(j)$	Distribution of the botnet size

CHAPTER 1

INTRODUCTION

Internet worms are malicious software that can compromise vulnerable hosts and use them to attack other victims, and have been one of top security threats since the Morris worm in 1988. Botnets are zombie networks controlled by attackers through Internet relay chat (IRC) systems (*e.g.,* GT Bot) or peer-to-peer (P2P) systems (*e.g.,* Storm) to execute coordinated attacks and have become the number one threat to the Internet in recent years. The main difference between worms and botnets lies in that worms emphasize the procedures of infecting targets and propagating among vulnerable hosts, whereas botnets focus on the mechanisms of organizing the network of compromised computers and setting out coordinated attacks, such as sending denial-of-service attacks, producing spams, and stealing financial information. Most botnets, however, still apply worm-scanning methods to recruit new bots or collect network information [1, 2, 3, 4]. Moreover, although many P2P-based botnets use the existing P2P networks to build a bootstrap procedure, Conficker C forms a P2P botnet through scan-based peer discovery [5, 6]. Specifically, Conficker C searches for new peers by randomly scanning the entire Internet address space. As a result, the way that Conficker C constructs a P2P-based botnet is in principle the same as worm scanning/infection. Therefore, characterizing structures of worm infection is important and imperative for defending against current and future epidemics such as Internet worms and Conficker C like P2P-based botnets.

1.1 Internet Worm Spatial-Temporal Infection Structures

Since the Code Red worm in 2001, Internet worms have been an active research topic. Many research works have been developed to characterize the spread of worms, estimate worm behaviors, and contain worm propagation. Most previous works, however, have

focused on the *macro-level* characteristics of worm infection. For example, different analytical approaches have been applied to study the total number of infected hosts over time [7, 8, 9, 10, 11, 1, 12]. The *micro-level* information of worm infection that focuses on individual hosts, however, has been investigated little. In this thesis, we focus on individual infected hosts and study their infection relationships, *i.e.,* the Internet worm infection structure.

When a host infects another host, they form a "father-and-son" relationship, which is represented by a directed edge in a graph formed by worm infection, the worm infection family tree, called the "worm tree" in short (see Fig. 1.1). That is, the procedure of worm propagation constructs a directed tree where patient zero is the root and the infected hosts that do not compromise any vulnerable host are leaves. Based on the perspective from which the worm tree is investigated, we divide the Internet worm infection structure into two domains: the temporal and spatial infection structures. The worm temporal infection structure describes the temporal infection relationship between individual infected hosts in the worm tree by studying their infection times, and therefore sheds light on the information of "*who infects before whom*"; the worm spatial infection structure characterizes the spatial infection relationship between individual infected hosts by studying the topology of the worm tree, and therefore provides insights into the information of "*who infects whom*".

1.2 Research Objectives and Contributions

The objective of this thesis is to characterize the spatial-temporal infection structures of Internet worms, and apply the observations to study P2P-based botnets formed by worm infection. Specifically, we investigate the following three topics:

1. Characterizing Internet worm temporal infection structure: First, we infer the temporal infection relationship between individual infected hosts by answering the following two questions:

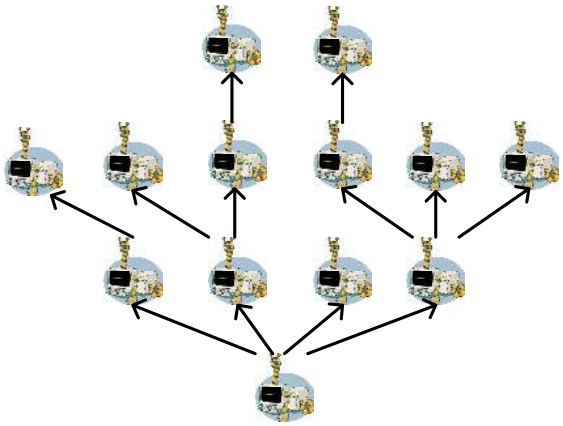


Figure 1.1: A worm tree.

- *Host infection time:* When exactly does a specific host get infected? This information is critical for the reconstruction of the worm infection sequence.
- *Worm infection sequence:* What is the order in which hosts are infected by worm propagation? Such an order can help identify patient zero or initially infected hosts.

The information of both the infection time and the infection sequence is important for defending against worms. First, the identification of patient zero or initially infected hosts and their infection times provide forensic clues for law enforcement against the attackers who wrote and spread the worm. Second, the knowledge of the infection sequence provides insights into how a worm spread across the Internet and how network defense systems were breached.

To answer these two questions analytically, we apply Internet worm tomography, which refers to inferring the characteristics of Internet worms from the observations of Darknet that monitor a routable but unused IP address space. Specifically, we introduce statistical estimation techniques and propose method of moments, maximum likelihood, and linear regression estimators. We then apply Monte Carlo simulation to verify our analytical results. Our research work makes several contributions:

- We propose method of moments, maximum likelihood, and linear regression statistical estimators to infer the host infection time. We show analytically and empirically that the mean squared error of our proposed estimators can be almost half of that of the naive estimator used in the previous work [13] in inferring the host infection time.
- We extend our proposed estimators to infer the worm infection sequence. Specifically, we formulate the problem of estimating the worm infection sequence as a detection problem and derive the probability of error detection for different estimators. We demonstrate analytically and empirically that our method performs much better than the algorithm proposed in the previous work [13].
- We show empirically that our estimators have a better performance in identifying patient zero or initially infected hosts of the smart worm than the naive estimator. We also demonstrate that our estimators can be applied to worms using different scanning strategies such as random scanning and localized scanning.

2. Characterizing Internet worm spatial infection structure: Next, we investigate the spatial infection relationship between individual infected hosts by studying the following two metrics of the worm tree:

- *Number of children:* For a randomly selected node in the tree, how many children does it have? This metric represents the infection ability of individual hosts.
- *Generation:* For a randomly selected node in the tree, which generation (or level) does it belong to? This metric indicates the average path length of the graph formed by worm infection.

These two metrics have important implications and applications for security analysis. For example, the distribution of the number of children can be used to answer questions such as what is the probability that an infected host compromises more than 10 vulnerable hosts.

Moreover, some schemes have been proposed to trace worms back to their origins through the cooperation between infected hosts [14], and the distribution of generations can provide the information on the number of hosts required to cooperate.

To study these two metrics analytically, we apply probabilistic modeling methods and a sequential growth model. Specifically, we start from a worm tree with only patient zero and add new nodes into the worm tree sequentially. We then investigate the relationship between the two worm trees before and after a new node is added and verify our analysis through simulation. Our research work makes several contributions:

- We show both analytically and empirically that if a worm uses a scanning method for which a new victim is compromised by each existing infected host with equal probability, the number of children has asymptotically a geometric distribution with parameter 0.5. This means that on average half of infected hosts never compromise any target and over 98% of infected hosts have no more than five children. On the other hand, this also indicates that a small portion of hosts infect a large number of vulnerable hosts.
- We demonstrate analytically and empirically that the generation closely follows a Poisson distribution with parameter $H_n - 1$, where n is the number of nodes and H_n is the n-th harmonic number [15]. This means that the average path length of the worm tree increases approximately logarithmically with the number of nodes.
- We show empirically that if a worm uses localized scanning, the number of children still has approximately a geometric distribution with parameter 0.5. Moreover, the generation still follows a Poisson distribution, but with the parameter depending on the probability of local scanning. Therefore, most previous observations also apply to localized-scanning worms.

3. Evaluating P2P-based botnets formed by worm infection: Finally, we study P2P-based botnets formed by worm infection and answer the following two questions:

- *Bot detection:* What is the most effective method to detect bots? This information is critical for defenders to combat against botnets.
- *User defenses:* How do user defenses (*e.g.,* host patching or cleaning) affect P2P-based botnets formed by worm infection? This information helps defenders evaluate the effectiveness of their defense systems.

The answer of the first question is directly related to the number of children of the worm spatial infection structure. For example, if a very small number of hosts infect a large number of machines and the majority of hosts have none or few children, such botnets are robust to random defenses, but are vulnerable to targeted defenses of a small portion of nodes with highest node degrees [16]. On the other hand, if each host has a similar node degree, then such botnets are robust to both defense schemes [16]. The answer of the second question reflects the robustness and the effectiveness of the botnet topology formed by worm infection under user defenses. For example, if user defenses disrupt the botnet into a collection of small isolated botnets, then the effectiveness is lower than the single connected botnet with the same total number of bots.

To answer these two questions, we first evaluate efficient bot detection methods both analytically and empirically by applying the results of the number of children of the worm spatial infection structure. We then empirically study effects of user defenses on the botnet topology formed by worm infection. Specifically, we study the number of peers (*i.e.,* the number of father and children for a randomly selected bot in the botnet topology), and the botnet size (*i.e.,* the number of bots for a randomly selected botnet in the topology). Our research work makes several contributions:

- We show both analytically and empirically that while randomly examining a small portion of nodes in a botnet (*i.e.,* random detection) can only expose a limited number of bots, examining the nodes with the largest number of children (*i.e.,* targeted detection) is much more efficient in detecting bots. For example, our simulation shows

that when 3.125% nodes are examined, random detection exposes totally 9.10% bots, whereas targeted detection reveals 22.36% bots.

- We find empirically that when user countermeasures are considered, the distribution of the number of peers has an exponential scaling with the decay constant increasing with the number of patched or cleaned hosts. This implies that a small percentage of bots have a large number of peers and the majority of bots have none or few peers. Moreover, the distribution of the disconnected botnet size has a power-law tail with the scaling exponent increasing with the number of patched or cleaned hosts. This reflects that patching or cleaning severely disrupts the single worm tree. We also find that the size of the largest isolated botnet is relatively small. Therefore, P2P-based botnets formed by worm infection are vulnerable to targeted defenses and ineffective due to patching or cleaning. However, we discover that botmasters may potentially enhance the robustness and the effectiveness of P2P-based botnets through worm re-infection.

1.3 Thesis Outline

The rest of this thesis is organized as follows. Chapter 2 surveys the related work. Chapter 3 infers temporal characteristics of the Internet worm infection structure, *i.e.,* the host infection time and the worm infection sequence. Chapter 4 characterizes the Internet worm spatial infection structure and reveals two key characteristics, *i.e.,* the number of children and the generation of the underlying tree topology formed by worm infection. Next, Chapter 5 evaluates bot detection strategies and effects of user defenses in P2P-based botnets formed by worm infection. Finally, Chapter 6 concludes the thesis and identifies future research directions.

CHAPTER 2

RELATED WORK

In the first chapter, we identified our research objectives and outlined our contributions. In this chapter, we answer the following questions: Why are existing techniques or models not sufficient for characterizing the worm infection structure? And how are they related to or different from our solution?

2.1 Internet Worm Temporal Infection Structure

Under the framework of Internet worm tomography, several works have applied Darknet observations to infer the characteristics of worms. For example, Chen *et al.* studied how the Darknet can be used to monitor, detect, and defend against Internet worms [9]. Moore *et al.* applied network telescope observations and least squares fitting methods to infer the number of infected hosts and scanning rates of infected hosts [17]. Some works have researched on how to use Darknet observations to detect the appearance of worms [18, 12, 19, 20]. For instance, Zou *et al.* used a Kalman filter to infer the infection rate of a worm and then detect the worm [12]. Moreover, the Darknet observations have been used to study the feature of a specific worm, such as Code Red [21], Slammer [22], and Witty [23].

Internet worm tomography has been applied to infer worm temporal characteristics. For example, Kumar *et al.* used network telescope data and analyzed the pseudo-random number generator to reconstruct the "who infected whom" infection tree of the Witty worm [24]. Hamadeh *et al.* further described a general framework to recover the infection sequence for both TCP and UDP scanning worms from network telescope data [25]. Rajab *et al.* applied the same data and studied the "infection and detection times" to infer the worm infection sequence [13]. Different from the above works, in Chapter 3, we employ advanced statistical estimation techniques to Internet worm tomography.

2.2 Internet Worm Spatial Infection Structure

Some efforts have been focused on individual infected hosts and studied the worm infection sequence [24, 13, 14, 26]. The prior work investigates the details of the random number generator of worm propagation [24] or infers the worm infection sequence through the observations of network telescopes [13, 26]. In Chapter 4, we apply probabilistic modeling methods and reveal key micro-level information of the worm spatial infection structure, such as the infection ability of individual hosts and the underlying tree topology formed by worm infection. Moreover, Sellke *et al.* applied a branching process to study the effectiveness of a containment strategy [27]. They assume that the total number of scans of an infected host is bounded. As a result, the worm tree studied in their work is fundamentally different from the one in our work.

Modeling the topology generation process has been an active research area. For example, Barabási *et al.* developed the well-known Barabási-Albert (BA) model and used a mean-field approach to characterize the growth of a topology with both preferential attachment and uniform attachment [28, 29]. Moreover, two exact mathematical models have been studied for the BA model [30, 31]. From the theoretical aspect, our proposed worm tree is similar to the random tree. For example, Devroye used the records theory to derive the distribution of the level of a random ordered tree in [32]. Compared with these theoretical efforts, our work studies a very different problem (*i.e.,* worm spatial infection structure) and uses a very different approach (*i.e.,* probabilistic modeling).

2.3 P2P-based Botnets Formed by Worm Infection

Botnets have become the top threat to the Internet in recent years [33, 34], and are rapidly transiting from IRC systems to P2P systems [35]. In [36], Wang *et al.* gave a systematic study on P2P-based botnets. Moreover, it has been shown that in current botnets, worm

infection is still a main tool for recruiting new bots or collecting network information, and random scanning has been widely used [2]. Several methods have been proposed to construct P2P-based botnets through worm infection and re-infection [3, 4]. Different from the above works, in our P2P-based botnets studied in Chapter 5, there is no grouping of bots or exchange of peers between bots. Infected hosts are only peers to their own infectors and infectees.

In [16], Dagon *et al.* surveyed different P2P-based botnet topologies, such as random graphs and power-law topologies. It has been shown that power-law topologies are robust to random node removal, but are vulnerable to the removal of a small portion of nodes with highest node degrees; random graphs are robust to both removal schemes [16, 37]. Our work takes one step further to quantitatively evaluate bot detection strategies and effects of user defenses by exploiting the P2P-based botnet topology formed by worm infection.

CHAPTER 3

CHARACTERIZING INTERNET WORM TEMPORAL INFECTION STRUCTURE

Since Code Red and Nimda worms were released in 2001, epidemic-style attacks have caused severe damages. Internet worms can spread so rapidly that existing defense systems cannot respond until most vulnerable hosts have been infected. For example, on January 25th, 2003, the Slammer worm reached its maximum scanning rate of more than 55 million scans per second in about 3 minutes, and infected more than 90% of vulnerable machines within 10 minutes [22]. It cost over one billion US dollars in cleanup and economic damages. Therefore, worm attacks pose significant threats to the Internet and meanwhile present tremendous challenges to the research community.

To counteract these notorious plague-tide attacks, various detection and defense strategies have been studied in recent years. According to where the detectors are located, these strategies can generally be classified into three categories: *source detection and defenses*, detecting infected hosts in the local networks [38, 39, 40, 41]; *middle detection and defenses*, revealing the appearance of worms by analyzing the traffic going through routers [14, 42, 43]; and *destination detection and defenses*, monitoring unwanted traffic arriving at *Darknet or network telescopes*, a globally routable address space where no active services or servers reside [44, 45, 46, 47, 48]. There are two types of Darknet: *active Darknet* that responds to malicious scans to elicit the payloads of the attacks [46, 47], and *passive Darknet* that observes unwanted traffic passively [45, 48].

Different from source and middle detection and defenses, destination detection and defenses offer unique advantages in observing large-scale network explosive events such as distributed denial-of-service (DDoS) attacks [49] and Internet worms [21, 22, 23]. There is no legitimate reason for packets destined to Darknet. Hence, most of the traffic arriving at Darknet is malicious or unintended, including hostile reconnaissance scans, probe activities

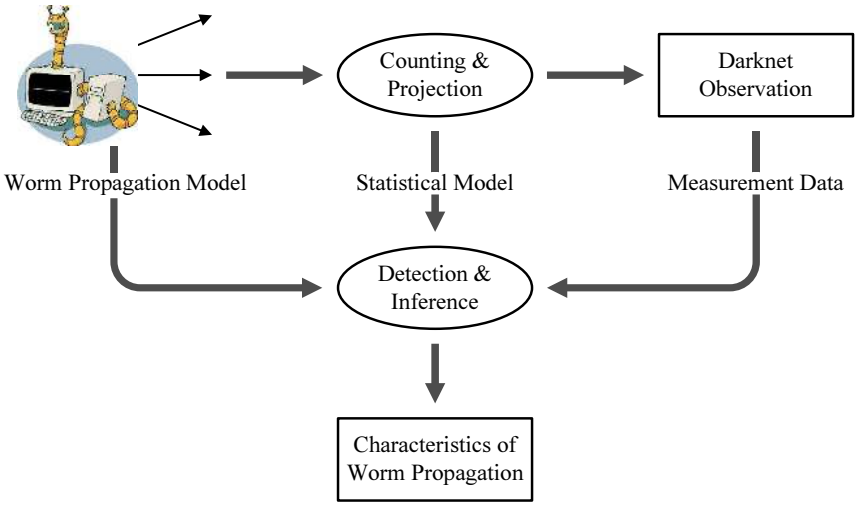


Figure 3.1: Internet worm tomography.

from active worms, DDoS backscatter, and packets from mis-configured hosts. Moreover, it has been shown that for a large-scale worm event, most of infected hosts, if not all, can be observed by the Darknet with a sufficiently large size [9].

In this chapter, we focus on the destination detection and defenses. Specifically, we study the problem of inferring the characteristics of Internet worms from Darknet observations. We refer to such a problem as *Internet worm tomography*, as illustrated in Fig.3.1. Most worms use scan-based methods to find vulnerable hosts and randomly generate target IP addresses. Thus, Darknet can observe partial scans from infected hosts. Together with the worm propagation model and the statistical model, Darknet observations can be used to detect worm appearance [18, 50, 19, 20] and infer worm characteristics (*e.g.*, infection rate [12], number of infected hosts [9, 17], and worm infection sequence [24, 13, 26]). Internet worm tomography is named after *network tomography*, which infers the characteristics of the internal network (*e.g.*, link loss rate, link delay, and topology) through the observations from end systems [51, 52]. Network tomography can be formulated as a linear inverse problem. Internet worm tomography, however, cannot be translated into the linear inverse problem due to the specific properties of worm propagation, and thus presents new challenges.

Under the framework of Internet worm tomography, researchers have studied worm temporal characteristics of the worm infection structure (*i.e.,* the host infection time and the worm infection sequence) [13, 24]. For example, a simple estimator has been proposed in [13] to infer worm temporal behaviors. The estimator uses the observation time when an infected host scans the Darknet for the first time as the approximation of the host infection time to infer the worm infection sequence. Such a naive estimator, however, does not fully exploit all information obtained by the Darknet. Moreover, an attacker can design a smart worm that uses lower scanning rates for patient zero or initially infected hosts and higher scanning rates for other infected hosts. In this way, the smart worm would weaken the performance of the naive estimator.

The goal of this chapter is to infer the Internet worm temporal characteristics accurately by exploiting Darknet observations and applying statistical estimation techniques. Specifically, we introduce statistical estimation techniques and propose method of moments, maximum likelihood, and linear regression estimators. We then apply Monte Carlo simulation to verify our analytical results.

The remainder of this chapter is organized as follows. Section 3.1 introduces estimators for inferring the host infection time. Section 3.2 presents our algorithms in estimating the worm infection sequence. Section 3.3 gives simulation results. Finally, Section 3.4 discusses the assumptions, the limitations, and the extensions of our estimators.

3.1 Estimating the Host Infection Time

We use Darknet observations to estimate when a host gets infected and use *hit* to denote the event that a worm scan hits the Darknet. As shown in Fig. 3.2, suppose that a certain host is infected at time t_0. The Darknet monitors a portion of the IPv4 address space and can observe some scans from this host and record hit times $t_1, t_2, \cdots, t_n$, where n is the number of hit events from this host. The problem of estimating the host infection time can

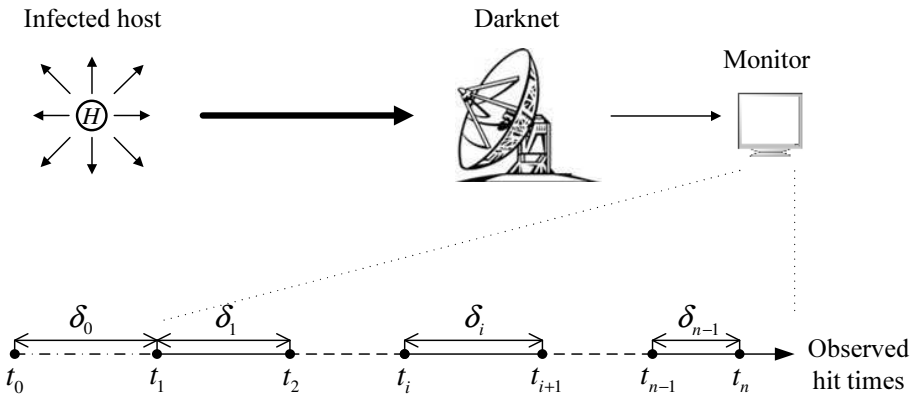


Figure 3.2: An illustration of Darknet observations.

then be stated as follows: Given the Darknet observations $t_1, t_2, \cdots, t_n$, what is the best estimate of t_0?

To study this problem analytically, we make the following assumptions: 1) There is no packet loss in the Internet. In Section 3.4, however, we relax this assumption and use simulations to study the effect of packet losses on different estimators. 2) An infected host uses its actual source IP address and does not apply IP spoofing, which is the case for TCP worms. 3) The scanning rate s (*i.e.*, the number of scans sent by an infected host per time unit) is time-invariant for an infected host, whereas the scanning rates of infected hosts can be different from each other. The last assumption comes from the observation that famous worms, such as Code Red, Nimda, Slammer, and Witty, do not apply any scanning rate variation mechanisms. An infected host always scans for vulnerable hosts at the maximum speed allowed by its computing resources and network conditions [53].

Obviously, inferring t_0 from Darknet observations is affected by the Internet-worm scanning methods. In this work, we focus on random scanning and localized scanning. However, our estimation techniques can be applied to other worm-scanning methods, such as importance scanning [54], for which a scan from an infected host hits Darknet with a time-invariant probability. To analytically estimate the host infection time, we consider a discrete-time system. For random scanning (RS), a worm selects targets randomly and

scans the entire IPv4 address space with Ω addresses (*i.e.,* $\Omega = 2^{32}$). We assume that Darknet monitors ω addresses. Thus, the probability for a scan to hit the Darknet is ω/Ω; and the probability of a hit event in the discrete-time system (*i.e.*, the probability that Darknet observes at least one scan from the same infected host in a time unit) is

$$\Pr_{\text{RS}}(\text{hit event}) = 1 - \left(1 - \frac{\omega}{\Omega}\right)^s. \tag{3.1}$$

Since s is time-invariant for a given infected host, $\Pr_{\text{RS}}(\text{hit event})$ is also time-invariant.

Localized scanning (LS) preferentially searches for vulnerable hosts in the "local" address space [55]. For simplicity, in this work we only consider the $/l$ LS: $p_a (0 \leq p_a < 1)$ of the time, a "local" IP address with the same first l bits as the attacking host is chosen as the target; $1 - p_a$ of the time, a random address is chosen. We consider a centralized Darknet that occupies a continuous address space and monitors ω addresses. Moreover, we assume that the Darknet is contained in a $/l$ prefix with no vulnerable hosts. For example, network telescopes used by CAIDA are such a centralized Darknet and contain a $/8$ subnet. Since no infected hosts exist in the $/l$ subnet where the Darknet resides, the probability for a worm scan to hit the Darknet is $(1 - p_a) \cdot \omega/\Omega$. Therefore, the probability of a hit event in the discrete-time system is

$$\Pr_{\text{LS}}(\text{hit event}) = 1 - \left(1 - (1 - p_a) \cdot \frac{\omega}{\Omega}\right)^s, \tag{3.2}$$

which is time-invariant. Since $\Pr_{\text{RS}}(\text{hit event})$ has a similar form as $\Pr_{\text{LS}}(\text{hit event})$ and is the special case of $\Pr_{\text{LS}}(\text{hit event})$ when p_a = 0, we use p $(0 < p < 1)$ to denote the hit probability in general for both cases to simplify our discussion.

Denote δ_0 as the time interval between when a host gets infected and when Darknet observes the first scan from this host, *i.e.,* $\delta_0 = t_1 - t_0$, as shown in Fig. 3.2. Denote δ_i as the time interval between i-th hit and $(i + 1)$-th hit on Darknet, *i.e.,* $\delta_i = t_{i+1} - t_i$, $i \geq 1$. Thus, $\delta_0, \delta_1, \cdots, \delta_{n-1}$ are independent and identically distributed (i.i.d.) and follow

a geometric distribution with parameter p, *i.e.,*

$$\Pr(\delta = k) = p \cdot (1-p)^{k-1},\ k = 1, 2, 3, \cdots, \tag{3.3}$$

$$\mathrm{E}(\delta) = \frac{1}{p} = \mu, \quad \mathrm{Var}(\delta) = \frac{1-p}{p^2}. \tag{3.4}$$

Denote μ as the mean value of δ and $\hat{\mu}$ as the estimate of μ. We then estimate t_0 by subtracting $\hat{\mu}$ from t_1, *i.e.,*

$$\hat{t_0} = t_1 - \hat{\mu}. \tag{3.5}$$

Therefore, our problem is reduced to estimating μ.

3.1.1 Naive Estimator

Since δ follows the geometric distribution as described by Equation (5.6), $\Pr(\delta)$ is maximized when $\delta = 1$. Then, a *naive estimator* (NE) of μ is

$$\hat{\mu}_{\text{NE}} = 1. \tag{3.6}$$

Thus, the NE of t_0 is

$$\hat{t_{0_{\text{NE}}}} = t_1 - \hat{\mu}_{\text{NE}} = t_1 - 1. \tag{3.7}$$

Note that $\hat{t_{0_{\text{NE}}}}$ depends only on t_1, but not on $t_2, t_3, \cdots, t_n$. This estimator has been used in [13] to infer the host infection time and the worm infection sequence. In this work, however, we consider more advanced estimation methods.

3.1.2 Method of Moments Estimator

Since $\mathrm{E}(\delta) = \mu$, we design a *method of moments estimator* (MME), *i.e.,*

$$\hat{\mu}_{\text{MME}} = \overline{\delta} = \frac{1}{n-1}\sum_{i=1}^{n-1} \delta_i = \frac{t_n - t_1}{n-1}. \tag{3.8}$$

Thus, the MME of t_0 is

$$\hat{t_0}_{\text{MME}} = t_1 - \hat{\mu}_{\text{MME}} = t_1 - \frac{t_n - t_1}{n-1}. \tag{3.9}$$

Note that $\hat{t_0}_{\text{MME}}$ is not only related to t_1, but also to n and t_n.

3.1.3 Maximum Likelihood Estimator

Rewrite the probability mass function of δ in Equation (5.6) with respect to μ,

$$\Pr(\delta;\mu) = \tfrac{1}{\mu}\Big(1 - \tfrac{1}{\mu}\Big)^{\delta-1}, \delta = 1, 2, 3, \cdots. \tag{3.10}$$

Since $\delta_1, \delta_2, \cdots, \delta_{n-1}$ are i.i.d., the likelihood function is given by the following product

$$\begin{aligned} \mathrm{L}(\mu) &= \prod_{i=1}^{n-1} \Pr(\delta_i;\mu) \\ &= \Big(\frac{1}{\mu}\Big)^{n-1}\Big(1-\frac{1}{\mu}\Big)^{(\sum_{i=1}^{n-1}\delta_i)-(n-1)}. \end{aligned} \tag{3.11}$$

We then design a *maximum likelihood estimator* (MLE), *i.e.,*

$$\hat{\mu}_{\text{MLE}} = \arg\max_{\mu} \mathrm{L}(\mu). \tag{3.12}$$

Rather than maximizing $\mathrm{L}(\mu)$, we choose to maximize its logarithm $ln\mathrm{L}(\mu)$. That is,

$$\frac{d}{d\mu} ln\mathrm{L}(\mu) = 0 \tag{3.13}$$

$$\Longrightarrow \hat{\mu}_{\text{MLE}} = \frac{1}{n-1}\sum_{i=1}^{n-1}\delta_i = \frac{t_n - t_1}{n-1}, \tag{3.14}$$

which has the same expression as the MME. Thus,

$$\hat{t_0}_{\text{MLE}} = t_1 - \hat{\mu}_{\text{MLE}} = t_1 - \frac{t_n - t_1}{n-1}. \tag{3.15}$$

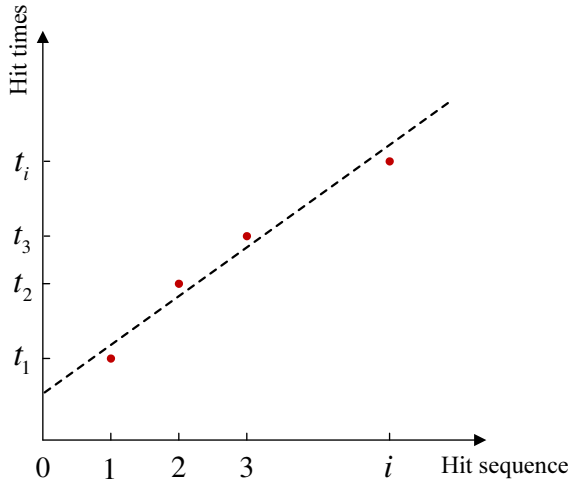


Figure 3.3: Linear regression model.

3.1.4 Linear Regression Estimator

Under the assumption that the scanning rate of an individual infected host is time-invariant, the relationship between t_i and i can be described by a linear regression model as illustrated in Fig. 3.3, *i.e.,*

$$t_i = \alpha + \beta \cdot i + \varepsilon_i, \tag{3.16}$$

where α and β are coefficients, and ε_i is the error term. To fit the observation data, we apply the least squares method to adjust the parameters of the model. That is, we choose the coefficients that minimize the residual sum of squares (RSS)

$$\text{RSS} = \sum_{i=1}^{n} [t_i - (\alpha + \beta \cdot i)]^2. \tag{3.17}$$

The minimum RSS occurs when the partial derivatives with respect to the coefficients are zero

$$\begin{cases} \dfrac{\partial \text{RSS}}{\partial \alpha} = -2 \displaystyle\sum_{i=1}^{n} (t_i - \alpha - \beta \cdot i) = 0 \\ \dfrac{\partial \text{RSS}}{\partial \beta} = -2 \displaystyle\sum_{i=1}^{n} i \cdot (t_i - \alpha - \beta \cdot i) = 0, \end{cases} \tag{3.18}$$

which leads to

$$\begin{cases} \hat{\alpha} = \overline{t} - \hat{\beta} \cdot \overline{i} \\ \hat{\beta} = \dfrac{\overline{i \cdot t} - \overline{i} \cdot \overline{t}}{\overline{i^2} - (\overline{i})^2}, \end{cases} \tag{3.19}$$

where the bar symbols denote the average values

$$\begin{cases} \overline{i} = \dfrac{1}{n} \sum_{i=1}^{n} i, \quad \overline{i^2} = \dfrac{1}{n} \sum_{i=1}^{n} i^2 \\ \overline{t} = \dfrac{1}{n} \sum_{i=1}^{n} t_i, \quad \overline{i \cdot t} = \dfrac{1}{n} \sum_{i=1}^{n} i \cdot t_i. \end{cases} \tag{3.20}$$

We then design a *linear regression estimator* (LRE), *i.e.,*

$$\hat{\mu}_{\text{LRE}} = \hat{\beta} = \hat{t_1} - \hat{t_0}. \tag{3.21}$$

Thus, the LRE of t_0 is

$$\hat{t_{0_{\text{LRE}}}} = t_1 - \hat{\mu}_{\text{LRE}} = t_1 - \frac{\overline{i \cdot t} - \overline{i} \cdot \overline{t}}{\overline{i^2} - (\overline{i})^2}. \tag{3.22}$$

There is another way to estimate t_0, which uses the point of interception shown in Fig. 3.3 as the estimation of t_0, *i.e.,*

$$\hat{t_{0_{\text{LRE}}}}' = \hat{\alpha} = \overline{t} - \hat{\mu}_{\text{LRE}} \cdot \overline{i}. \tag{3.23}$$

However, we find that the mean squared error of $\hat{t_{0_{\text{LRE}}}}'$ increases when n increases. That is, the performance of the estimator worsens with the increasing number of hits, which makes this estimator undesirable.

3.1.5 Comparison of Estimators

To compare the performance of the naive estimator and our proposed estimators, we compute the bias, the variance, and the mean squared error (MSE). For estimating μ,

$$\begin{cases} \text{Bias}(\hat{\mu}) &= \text{E}(\hat{\mu}) - \mu \\ \text{Var}(\hat{\mu}) &= \text{E}\left[(\hat{\mu} - \text{E}(\hat{\mu}))^2\right] \\ \text{MSE}(\hat{\mu}) &= \text{E}\left[(\hat{\mu} - \mu)^2\right] = \text{Bias}^2(\hat{\mu}) + \text{Var}(\hat{\mu}). \end{cases} \tag{3.24}$$

Table 3.1: Comparison of estimator properties ($\hat{\mu}$).

$\hat{\mu}$	Bias($\hat{\mu}$)	Var($\hat{\mu}$)	MSE($\hat{\mu}$)
$\hat{\mu}_{\text{NE}} = 1$	$1 - \frac{1}{p}$	0	$\frac{(1-p)^2}{p^2}$
$\hat{\mu}_{\text{MME}} = \hat{\mu}_{\text{MLE}} = \frac{t_n - t_1}{n-1}$	0	$\frac{1-p}{p^2(n-1)}$	$\frac{1-p}{p^2(n-1)}$
$\hat{\mu}_{\text{LRE}} = \frac{\overline{i \cdot t} - \bar{i} \cdot \bar{t}}{\overline{i^2} - (\bar{i})^2}$	0	$\frac{6(n^2+1)(1-p)}{5n(n^2-1)p^2}$	$\frac{6(n^2+1)(1-p)}{5n(n^2-1)p^2}$

Table 3.2: Comparison of estimator properties ($\hat{t_0}$).

$\hat{t_0} = t_1 - \hat{\mu}$	Bias($\hat{t_0}$)	Var($\hat{t_0}$)	MSE($\hat{t_0}$)	
$\hat{t_0}_{\text{NE}}$	$\frac{1-p}{p}$	$\frac{1-p}{p^2}$	$\frac{(1-p)(2-p)}{p^2}$	$(\approx \frac{2(1-p)}{p^2}, \text{ when } p \ll 1)$
$\hat{t_0}_{\text{MME}} = \hat{t_0}_{\text{MLE}}$	0	$\frac{1-p}{p^2} \cdot \frac{n}{n-1}$	$\frac{1-p}{p^2} \cdot \frac{n}{n-1}$	$(\approx \frac{1-p}{p^2}, \text{ when } n \gg 1)$
$\hat{t_0}_{\text{LRE}}$	0	$\frac{1-p}{p^2} \cdot \frac{5n^3+6n^2-5n+6}{5n(n^2-1)}$	$\frac{1-p}{p^2} \cdot \frac{5n^3+6n^2-5n+6}{5n(n^2-1)}$	$(\approx \frac{1-p}{p^2}, \text{ when } n \gg 1)$

Here, the *bias* denotes the average deviation of the estimator from the true value; the *variance* indicates the distance between the estimator and its mean; and the *MSE* characterizes the closeness of the estimated value to the true value. A smaller MSE indicates a better estimator. Table 3.1 summarizes the results of NE, MME (or MLE), and LRE for estimating μ. The details of the derivations of Table 3.1 are given in Appendix A. It is noted that MME and LRE are unbiased, while NE is biased. Moreover, MME and LRE have a smaller MSE than NE if $n > 2$ and $p < 0.5$, a condition that is usually satisfied. Specifically, when $n \to \infty$, MSE($\hat{\mu}_{\text{MME}}$) $\to 0$ and MSE($\hat{\mu}_{\text{LRE}}$) $\to 0$, but MSE($\hat{\mu}_{\text{NE}}$) $\to (1-p)^2/p^2$. It is also observed that MME is slightly better than LRE in terms of MSE when $n > 2$.

Similarly, we compute the bias, the variance, and the MSE of the estimators for estimating t_0 in Table 3.2. The details of the derivations of Table 3.2 are given in Appendix B. We also observe that MME (or MLE) and LRE are unbiased, whereas NE is biased. Moreover, MSE($\hat{t_0}_{\text{MME}}$) and MSE($\hat{t_0}_{\text{LRE}}$) are smaller than MSE($\hat{t_0}_{\text{NE}}$), and MSE($\hat{t_0}_{\text{MME}}$) is the smallest when $n > 3$ and $p < 0.5$. Specifically, in practice, Darknet only covers a relatively small portion of the IPv4 address space (*i.e.,* $\omega \ll \Omega$), which leads to $p \ll 1$. Thus, we have the following theorem:

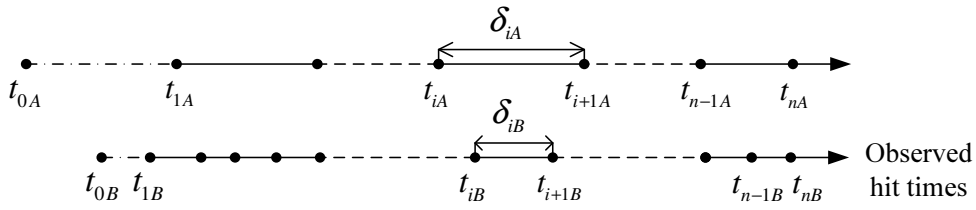


Figure 3.4: A scenario of the worm infection sequence.

Theorem 3.1.1 *When the Darknet observes a sufficient number of hits (*i.e., $n \gg 1$*) and* $p \ll 1$*,*

$$MSE(\hat{t_{0_{MME}}}) \approx MSE(\hat{t_{0_{LRE}}}) \approx \frac{1}{2} MSE(\hat{t_{0_{NE}}}). \tag{3.25}$$

That is, the MSE of our proposed estimators is almost half of that of the naive estimator. That is, our proposed estimators are nearly twice as accurate as the naive estimator in estimating the host infection time.

3.2 Estimating the Worm Infection Sequence

In this section, we extend our proposed estimators for inferring the worm infection sequence.

3.2.1 Algorithm

Our algorithm is that we first estimate the infection time of each infected host. Then, we reconstruct the infection sequence based on these infection times. That is, if $\hat{t}_{0\mathrm{A}} < \hat{t}_{0\mathrm{B}}$, we infer that host A is infected before host B. It is noted that the algorithm used in [13] to infer the worm infection sequence can be regarded as using this approach with the naive estimator.

The naive estimator, however, can potentially fail to infer the worm infection sequence in some cases. Fig. 3.4 shows an example, where hosts A and B get infected at $t_{0\mathrm{A}}$ and $t_{0\mathrm{B}}$,

respectively, and $t_{0A} < t_{0B}$. Moreover, these two infected hosts have scanning rates $s_A < s_B$ such that Darknet observes $t_{1A} > t_{1B}$. If the naive estimator is used, $\hat{t}_{0A} > \hat{t}_{0B}$, which means that host A is incorrectly inferred to be infected after host B. Intuitively, if our proposed estimators are applied, it is possible to obtain $\hat{t}_{0A} < \hat{t}_{0B}$ and thus recover the real infection sequence.

3.2.2 Performance Analysis

To analytically show that our estimators are more accurate than the naive estimator in estimating the worm infection sequence, we formulate the problem as a detection problem. Specifically, in Fig. 3.4, suppose that host B is infected after host A (*i.e.,* $t_{0A} < t_{0B}$). If $\hat{t}_{0A} < \hat{t}_{0B}$, we call it "success" detection; otherwise, if $\hat{t}_{0A} > \hat{t}_{0B}$, we call it "error" detection[1]. We intend to calculate the probability of error detection for different estimators.

Note that $\delta_{0A} = t_{1A} - t_{0A}$ and $\delta_{0B} = t_{1B} - t_{0B}$ follow the geometric distribution (*i.e.,* Equation (5.6)) with parameter p_A and p_B, respectively. Here, p_A (or p_B) is the probability that at least one scan from host A (or B) hits the Darknet in a time unit and follows Equation (3.1) for random scanning and Equation (3.2) for localized scanning. Moreover, p_A (or p_B) depends on s_A (or s_B) so that if $s_A < s_B$, then $p_A < p_B$. Since $\omega \ll \Omega$, we have $p_A \ll 1$ and $p_B \ll 1$. Hence, for simplicity we use the continuous-time analysis and apply the exponential distribution to approximate the geometric distribution for δ_{0A} and δ_{0B} [56], *i.e.,*

$$f(x;\lambda) = \begin{cases} \lambda e^{-\lambda x}, & x \geq 0 \\ 0, & x < 0, \end{cases} \tag{3.26}$$

where $\lambda = p_A$ or p_B.

To calculate the probability of error detection for different estimators, we first define a new random variable

$$Z = \delta_{0A} - \delta_{0B}, \tag{3.27}$$

[1]We ignore the case $\hat{t}_{0A} = \hat{t}_{0B}$ here.

and calculate its probability density function (pdf) $f_{\text{Z}}(z)$. From Equation (3.26), we can obtain the pdf of $\delta'_{0\text{B}} = -\delta_{0\text{B}}$, which is

$$f_{\delta'_{0\text{B}}}(x) = \begin{cases} p_{\text{B}}\, e^{p_{\text{B}} x}, & x \leq 0 \\ 0, & x > 0. \end{cases} \tag{3.28}$$

Since $\delta_{0\text{A}}$ and $\delta'_{0\text{B}}$ are independent, the pdf of $Z = \delta_{0\text{A}} + \delta'_{0\text{B}}$ is given by the convolution of $f_{\delta_{0\text{A}}}(x)$ and $f_{\delta'_{0\text{B}}}(x)$, *i.e.,*

$$f_{\text{Z}}(z) = \int_{-\infty}^{+\infty} f_{\delta_{0\text{A}}}(x) f_{\delta'_{0\text{B}}}(z - x)\, dx. \tag{3.29}$$

For $z \geq 0$, this yields

$$\begin{aligned} f_{\text{Z}}(z) &= \int_{z}^{+\infty} p_{\text{A}}\, e^{-p_{\text{A}} x} \cdot p_{\text{B}}\, e^{p_{\text{B}}(z-x)}\, dx \\ &= \tfrac{p_{\text{A}} p_{\text{B}}}{p_{\text{A}} + p_{\text{B}}}\, e^{-p_{\text{A}} z}. \end{aligned} \tag{3.30}$$

For $z < 0$, we obtain

$$\begin{aligned} f_{\text{Z}}(z) &= \int_{0}^{+\infty} p_{\text{A}}\, e^{-p_{\text{A}} x} \cdot p_{\text{B}}\, e^{p_{\text{B}}(z-x)}\, dx \\ &= \tfrac{p_{\text{A}} p_{\text{B}}}{p_{\text{A}} + p_{\text{B}}}\, e^{p_{\text{B}} z}. \end{aligned} \tag{3.31}$$

Hence,

$$f_{\text{Z}}(z) = \begin{cases} \tfrac{p_{\text{A}} p_{\text{B}}}{p_{\text{A}} + p_{\text{B}}}\, e^{-p_{\text{A}} z}, & z \geq 0 \\ \tfrac{p_{\text{A}} p_{\text{B}}}{p_{\text{A}} + p_{\text{B}}}\, e^{p_{\text{B}} z}, & z < 0. \end{cases} \tag{3.32}$$

Naive Estimator

The naive estimator uses $\hat{t_0} = t_{1} - 1$ to estimate t_0. Thus, the probability of error detection is

$$\text{Pr}_{\text{NE}}(\text{error}) = \text{Pr}(t_{1\text{A}} - 1 > t_{1\text{B}} - 1) = \text{Pr}(\delta_{0\text{A}} > \tau + \delta_{0\text{B}}), \tag{3.33}$$

where $\tau = t_{0\text{B}} - t_{0\text{A}}$, the time interval between the infection of host A and host B; and $\tau > 0$. We then have

$$\begin{aligned} \Pr_{\text{NE}}(\text{error}) &= \Pr(Z > \tau) \\ &= \frac{p_{\text{B}}}{p_{\text{A}}+p_{\text{B}}}\, e^{-p_{\text{A}}\tau}. \end{aligned} \tag{3.34}$$

Note that another way to derive $\Pr_{\text{NE}}(\text{error})$ is based on the memoryless property of the exponential distribution and $\Pr(\delta_{0\text{A}} > \delta_{0\text{B}}) = p_{\text{B}}/(p_{\text{A}} + p_{\text{B}})$, *i.e.,*

$$\Pr_{\text{NE}}(\text{error}) = \Pr(\delta_{0\text{A}} > \tau + \delta_{0\text{B}}) = \Pr(\delta_{0\text{A}} > \tau)\Pr(\delta_{0\text{A}} > \delta_{0\text{B}}), \tag{3.35}$$

which leads to the same result.

Proposed Estimators

We assume that Darknet observes a sufficient number of scans from hosts A and B so that our proposed estimators can estimate μ_{A} (*i.e.,* $\frac{1}{p_{\text{A}}}$) and μ_{B} (*i.e.,* $\frac{1}{p_{\text{B}}}$) accurately. Then, the probability of error detection of our proposed estimators is

$$\begin{aligned} \Pr_{\text{MME}}(\text{error}) &= \Pr_{\text{MLE}}(\text{error}) = \Pr_{\text{LRE}}(\text{error}) \\ &= \Pr(t_{1\text{A}} - \tfrac{1}{p_{\text{A}}} > t_{1\text{B}} - \tfrac{1}{p_{\text{B}}}) \\ &= \Pr(Z > \tau + \tfrac{p_{\text{B}}-p_{\text{A}}}{p_{\text{A}}p_{\text{B}}}) \\ &= \int_{\tau+\frac{p_{\text{B}}-p_{\text{A}}}{p_{\text{A}}p_{\text{B}}}}^{+\infty} f_{\text{Z}}(z)\, dz. \end{aligned} \tag{3.36}$$

When $\tau + \frac{p_{\text{B}}-p_{\text{A}}}{p_{\text{A}}p_{\text{B}}} \geq 0$,

$$\begin{aligned} \Pr_{\text{MME}}(\text{error}) &= \int_{\tau+\frac{p_{\text{B}}-p_{\text{A}}}{p_{\text{A}}p_{\text{B}}}}^{+\infty} \frac{p_{\text{A}}p_{\text{B}}}{p_{\text{A}}+p_{\text{B}}}\, e^{-p_{\text{A}}z}\, dz \\ &= \frac{p_{\text{B}}}{p_{\text{A}}+p_{\text{B}}}\, e^{-p_{\text{A}}\left(\tau+\frac{p_{\text{B}}-p_{\text{A}}}{p_{\text{A}}p_{\text{B}}}\right)}. \end{aligned} \tag{3.37}$$

When $\tau + \frac{p_B - p_A}{p_A p_B} < 0$,

$$\begin{aligned} \Pr_{\text{MME}}(\text{error}) &= \int_{\tau+\frac{p_B-p_A}{p_A p_B}}^{0} \frac{p_A p_B}{p_A+p_B} e^{p_B z}\, dz + \\ &\quad \int_{0}^{+\infty} \frac{p_A p_B}{p_A+p_B} e^{-p_A z}\, dz \\ &= \frac{1}{p_A+p_B}\Big(p_A + p_B - p_A\, e^{p_B\left(\tau+\frac{p_B-p_A}{p_A p_B}\right)}\Big). \end{aligned} \tag{3.38}$$

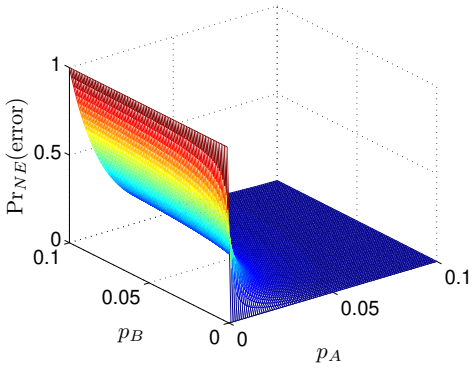
Performance Comparison

Since $\Pr_{\text{NE}}(\text{error}) = \Pr(Z > \tau)$ and $\Pr_{\text{MME}}(\text{error}) = \Pr\big(Z > \tau + \frac{p_B-p_A}{p_A p_B}\big)$, for a given τ ($\tau > 0$), comparing Equation (3.34) with Equations (3.37) and (3.38),

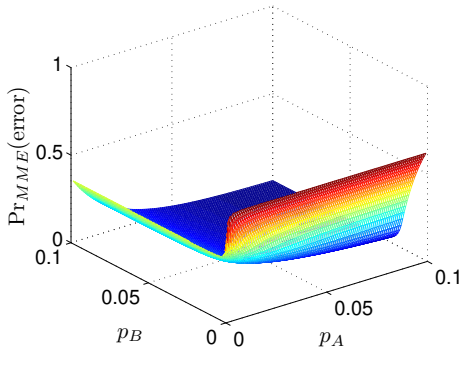
$$\begin{cases} \Pr_{\text{NE}}(\text{error}) > \Pr_{\text{MME}}(\text{error}), & p_A < p_B \\ \Pr_{\text{NE}}(\text{error}) < \Pr_{\text{MME}}(\text{error}), & p_A > p_B. \end{cases} \tag{3.39}$$

Hence, it is unclear which estimator is better based on the expressions of $\Pr_{\text{NE}}(\text{error})$ and $\Pr_{\text{MME}}(\text{error})$. However, we can compare the performance of our estimators with the naive estimator through numerical analysis. We first demonstrate the probabilities of error detection (*i.e.,* $\Pr_{\text{NE}}(\text{error})$ and $\Pr_{\text{MME}}(\text{error})$) as the functions of p_A and p_B in Figs. 3.5 (a) and (b), where τ = 50 time units. It can be seen that for the naive estimator, when host A hits the Darknet with a very low probability, $\Pr_{\text{NE}}(\text{error})$ is almost 1 regardless of p_B. However, the worst case of $\Pr_{\text{MME}}(\text{error})$ is slightly above 0.6 when p_B is small. Moreover, we show the probabilities of error detection as a function of τ with a given pair of p_A and p_B in Fig.s 3.5 (c) and (d). The performance of two estimators improves as τ increases. Furthermore, the sum of the integral $\int_0^{500} \Pr_{\text{NE}}(\text{error})\, d\tau$ of the two figures is 41.43, while the sum of the integral $\int_0^{500} \Pr_{\text{MME}}(\text{error})\, d\tau$ in these two cases is only 34.76. This shows that the improvement gain of our estimators over the naive estimator when $p_A < p_B$ outweighs the degradation suffered when $p_A > p_B$, indicating the benefits of applying our estimators.

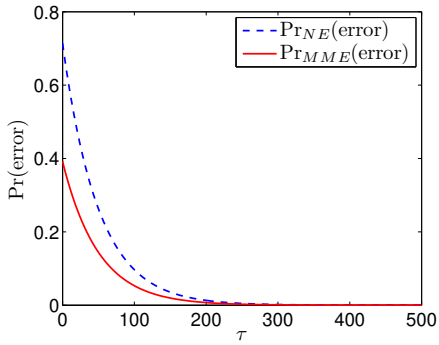
Note that p_A, p_B, and τ can be random variables. To evaluate the overall performance of each estimator, we consider the average probability of error detection over p_A, p_B, and τ,



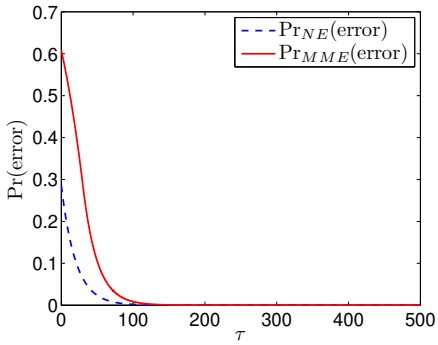
(a) $\Pr_{\text{NE}}(\text{error})$ (τ = 50 time units).



(b) $\Pr_{\text{MME}}(\text{error})$ (τ = 50 time units).



(c) $\Pr(\text{error})$ ($p_{\text{A}} = 0.02$, $p_{\text{B}} = 0.05$).



(d) $\Pr(\text{error})$ ($p_{\text{A}} = 0.05$, $p_{\text{B}} = 0.02$).

Figure 3.5: Numerical analysis of $\Pr(\text{error})$.

i.e.,

$$\mathrm{E}\left[\Pr(\text{error})\right] = \int_{\tau}\int_{p_{\text{A}}}\int_{p_{\text{B}}} \Pr(\text{error}) \cdot f(p_{\text{A}}, p_{\text{B}}, \tau)\; d\,p_{\text{B}}\, d\,p_{\text{A}}\, d\tau. \tag{3.40}$$

Since p_{A}, p_{B}, and τ are independent,

$$f(p_{\text{A}}, p_{\text{B}}, \tau) = f(p_{\text{A}}) \cdot f(p_{\text{B}}) \cdot f(\tau). \tag{3.41}$$

We then consider some cases in which we are interested and apply the numerical integration toolbox in Matlab [57] to calculate the triple integration. For example, we assume that s_{A} and s_{B} follow a normal distribution $N(u, \sigma^2)$ and τ is uniform over $(0, \tau_1]$. We find that when u, σ^2, and τ_1 are set to realistic values, we always have

$$\mathrm{E}\left[\Pr_{\text{NE}}(\text{error})\right] > \mathrm{E}\left[\Pr_{\text{MME}}(\text{error})\right]. \tag{3.42}$$

That is, our proposed estimators perform better than NE on average, which will further be verified in Section 3.3 through simulations.

Moreover, in Fig. 3.5(a), it can be seen that the majority of detection error for the naive estimator comes from the case that $p_{\rm A} < p_{\rm B}$. Specifically, it is obvious to derive the following theorem from Equations (3.34) and (3.37).

Theorem 3.2.1 *When $p_A < p_B$,*

$$\begin{aligned} Pr_{MME}(error) &= Pr_{MLE}(error) = Pr_{LRE}(error) \\ &= Pr_{NE}(error) \cdot e^{-\left(1-\frac{p_A}{p_B}\right)}. \end{aligned} \tag{3.43}$$

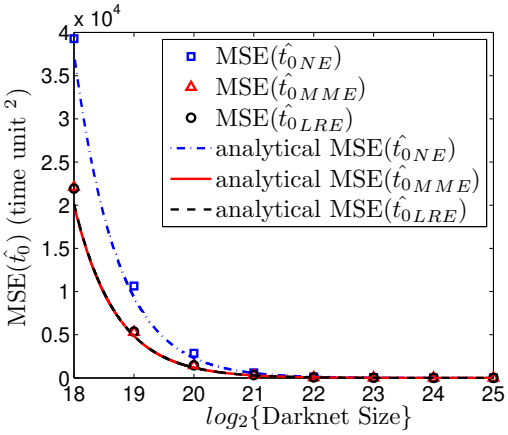
That is, the error probability is decreased by a factor of $e^{-\left(1-\frac{p_{\rm A}}{p_{\rm B}}\right)}$ by applying our estimators as compared with the naive estimator.

3.3 Simulations and Verification

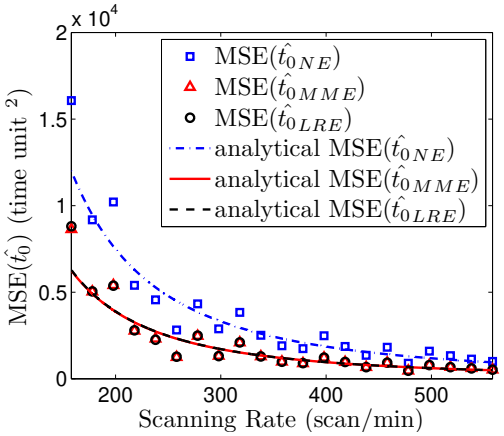
In this section, we use simulations to verify our analytical results and then apply estimators to identify the patient zero or the hitlist. As far as we know, there is no publicly available data to show the real worm infection sequence. That is, there is no dataset available with the real infection sequence to serve as the ground truth and a comparison basis for performance evaluation. Therefore, we apply empirical simulations to provide the simulated worm infection time and infection sequence.

3.3.1 Estimating the Host Infection Time

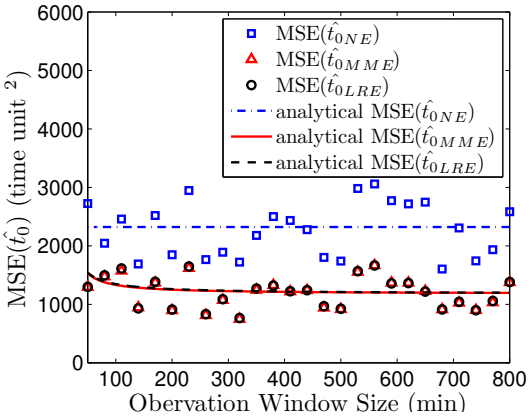
We evaluate the performance of estimators in estimating the host infection time. For the case of random-scanning worms, we simulate the behavior of a host infected by the Code Red v2 worm. The host is infected at time tick 0 and uses a constant scanning rate. The time unit is set to 20 seconds. The Darknet records hit times during an observation window. The



(a) Changing the Darknet size.



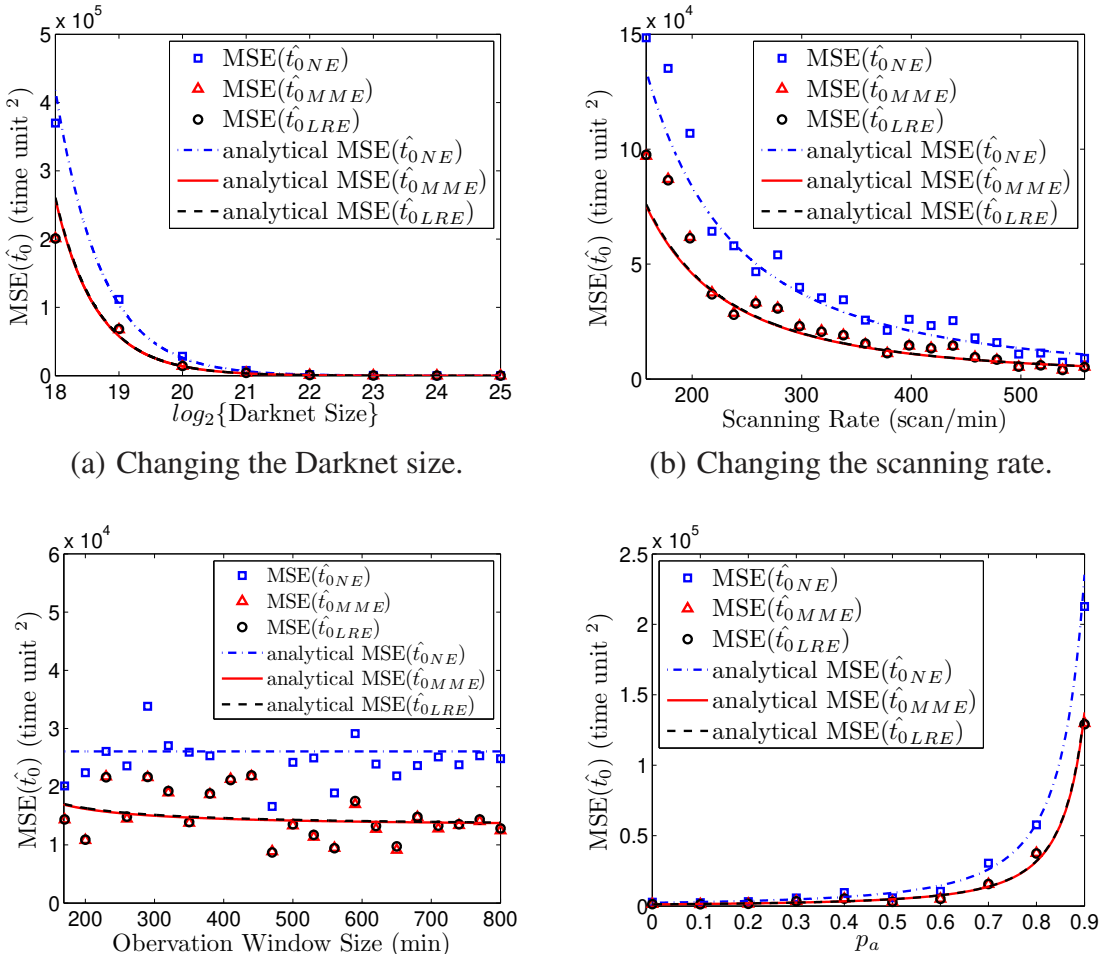
(b) Changing the scanning rate.



(c) Changing the observation window size.

Figure 3.6: Comparison of MSE($\hat{t_0}$) for random-scanning worms.

results are averaged over 100 independent runs. Fig. 3.6 compares the performance (*i.e.,* MSE of estimators for t_0) of NE, MME, and LRE. In our simulations, we use a Darknet size of 2^{20}, a scanning rate of 358 scans/min, and an observation window size of 800 mins as default values. Moreover, when a parameter is studied and varied, we keep other parameters unchanged. Specifically, we consider the effects of the Darknet size, the scanning rate, and the observation window size on the performance of the estimators. It is observed that for all cases, our proposed estimators have a better performance (smaller MSE) than the naive estimator in estimating the host infection time. Specifically, the simulation results verify



(a) Changing the Darknet size.

(b) Changing the scanning rate.

(c) Changing the observation window size.

(d) Changing the p_a.

Figure 3.7: Comparison of MSE($\hat{t_0}$) for localized-scanning worms.

Theorem 3.1.1, *i.e.,* that the MSE of our estimators is almost half of that of the naive estimator, when the observation window size is sufficiently large (*e.g.,* $>$ 200 mins).

Next, we study a host infected by localized-scanning worms and adopt the same simulation parameters and settings as the above. The main difference is that here the host preferentially searches for vulnerable hosts in the "local" address space with a probability p_a. In Fig. 3.7, we compare MSE($\hat{t_0}$) for different estimators. The default parameter values are a Darknet size of 2^{20}, a scanning rate of 358 scans/min, an observation window size of 800 mins, and a p_a value of 0.7. We find that the results are similar to those for the

random-scanning case shown in Fig. 3.6. That is, the MSE of our estimators is almost half of that of the naive estimator. On the other hand, it can be seen that the MSE($\hat{t_0}$) in Fig.s 3.7 (a)-(c) is larger for all cases than that in Fig. 3.6 since the localized-scanning worm hits the Darknet less frequently than the random-scanning worm.

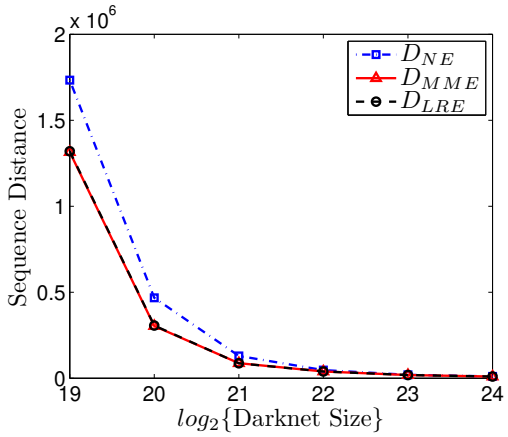
3.3.2 Estimating the Worm Infection Sequence

We evaluate the performance of our algorithms in estimating the worm infection sequence and simulate the propagation of the Code Red v2 worm. Specifically, the simulator considers a discrete-time system and mimics the random-scanning behavior of infected hosts during each discrete time interval. Moreover, the parameter setting is based on the Code Red v2 worm's characteristics. The Code Red worm has a vulnerable population of 360,000. Different infected hosts may have different scanning rates. Thus, we assign a scanning rate (scans/min) from a normal distribution $N(358, \sigma^2)$ to a newly infected host. Moreover, we start our simulation at time tick 0 from one infected host. The time unit is set to 20 seconds. Detailed information about how the parameters are chosen can be found in Section VII of [12]. Each point in Fig. 3.8 is averaged over 20 independent runs. Table 3.3 gives the results of a sample run with a Darknet size of 2^{20}, an observation window size of 1,600 mins, and σ = 110. In the table, S_i is the actual infection sequence (*i.e.,* $S_i = i$), whereas $\hat{S}_i$ is the estimated sequence. In this example, we find that MME and LRE can pinpoint the patient zero successfully, while NE fails.

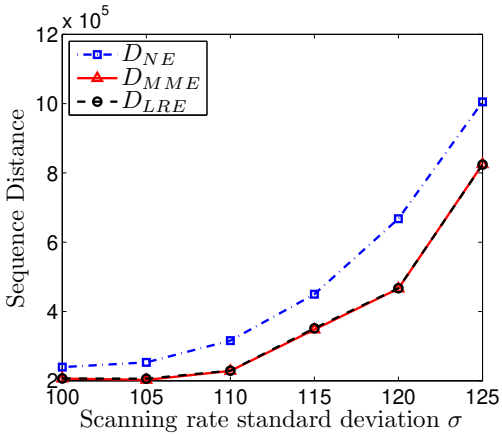
To compare the performance of estimators quantitatively, we consider a simple l_1 *sequence distance*, *i.e.,*

$$D = \sum_{i=1}^{N} \left| S_i - \hat{S}_i \right|, \tag{3.44}$$

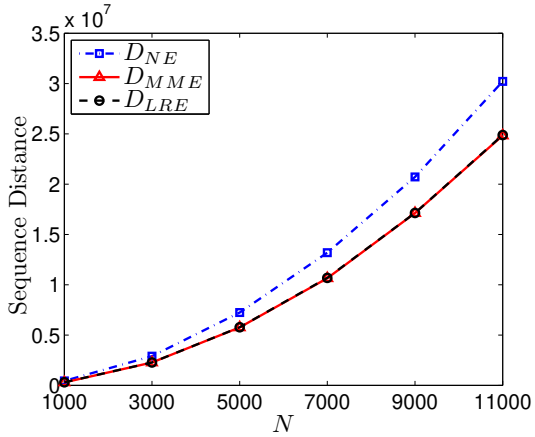
where N is the length of the infection sequence considered, S_i is the actual infection sequence (*i.e.,* $S_i = i$), and $\hat{S}_i$ is the estimated sequence. Note that the smaller the sequence distance is, the better the estimator performance will be. Fig. 3.8 compares the perfor-



(a) Changing the Darknet size.



(b) Changing the scanning rate standard deviation.



(c) Changing the infection sequence length.

Figure 3.8: Comparison of the sequence distance for random-scanning worms.

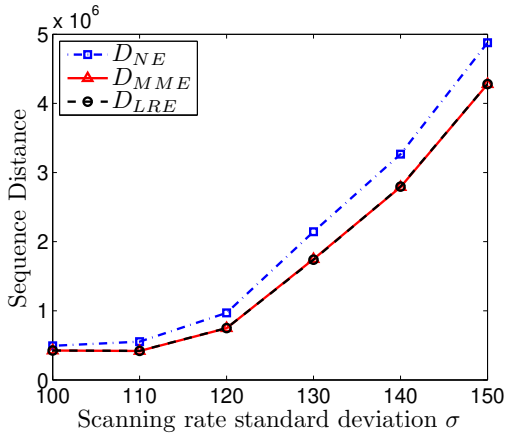
mance of different estimators for random-scanning worms, where the default parameter values are a Darknet size of 2^{20}, a scanning rate standard deviation of 115, an observation window size of 1,600 mins, and a length of the infection sequence considered of 1,000. Specifically, Fig. 3.8(a) shows the sequence distances of NE, MME, and LRE with varying Darknet sizes from 2^{19} to 2^{24}. It is observed that when the Darknet size increases, the performance of all estimators improves dramatically. Moreover, the performance of MME and LRE is always better than that of NE. For example, when the Darknet size equals 2^{19}, MME and LRE improve the inference accuracy by 24%, compared with NE. Fig. 3.8(b)

Table 3.3: A sample run of simulations for random scanning.

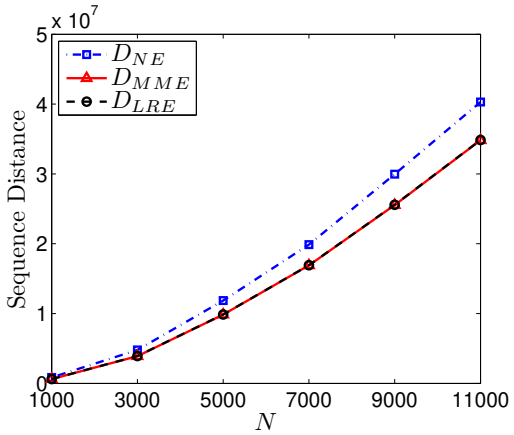
S_i	$\hat{S}_{i_{\text{NE}}}$	$\hat{S}_{i_{\text{MME}}}$	$\hat{S}_{i_{\text{LRE}}}$	t_0	$\hat{t_0}_{\text{NE}}$	$\hat{t_0}_{\text{MME}}$	$\hat{t_0}_{\text{LRE}}$
1	2	1	1	0	114	20	20
2	1	2	2	85	98	74	73
3	3	3	3	105	165	116	116
:	:	:	:	:	:	:	:
520	498	533	534	593	622	589	589
521	433	488	477	594	611	581	580
:	:	:	:	:	:	:	:

demonstrates the sequence distances of these three estimators by changing the standard deviation of the scanning rate (*i.e.,* σ) from 100 to 125. It is noted that when σ increases, the performance of all estimators deteriorates. The performance of MME and LRE, however, is always better than that of NE. For example, when σ = 120, MME and LRE reduce the sequence distance by 30%, compared with NE. In Fig. 3.8(c), we increase the length of the infection sequence considered, N, from 1,000 to 11,000. It is intuitive that the sequence distances of all estimators become larger as N increases. However, MME and LRE are always better than NE.

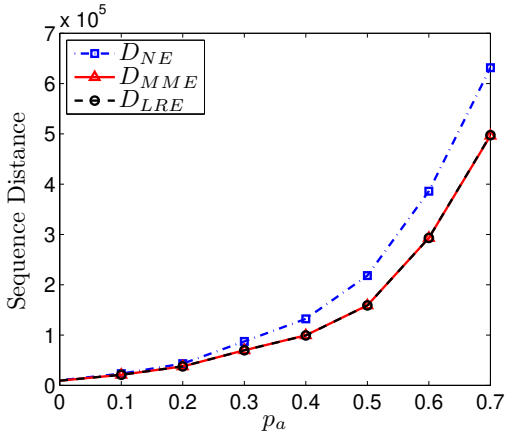
Next, we extend our simulator to imitate the spread of localized-scanning worms. Specifically, we consider /8 localized-scanning worms and a centralized /8 Darknet with 2^{24} IP addresses. We still use the Code Red v2 worm parameters and the same setting as random scanning, except that the observation window size is 1,000 mins (this is because localized-scanning worms spread faster). The distribution of vulnerable hosts is extracted from the dataset provided by DShield [58]. DShield obtains the information of vulnerable hosts by aggregating logs from more than 1,600 intrusion detection systems distributed throughout the Internet. Specifically, we use the dataset with port 80 (HTTP) that is exploited by the Code Red v2 worm to generate the vulnerable-hosts distribution. Each point in Fig. 3.9 is averaged over 20 independent runs. Fig. 3.9 compares the sequence distances of different estimators for localized scanning. Specifically, the results in Fig. 3.9(a) and (b)



(a) Changing the scanning rate standard deviation.



(b) Changing the infection sequence length.



(c) Changing the p_a.

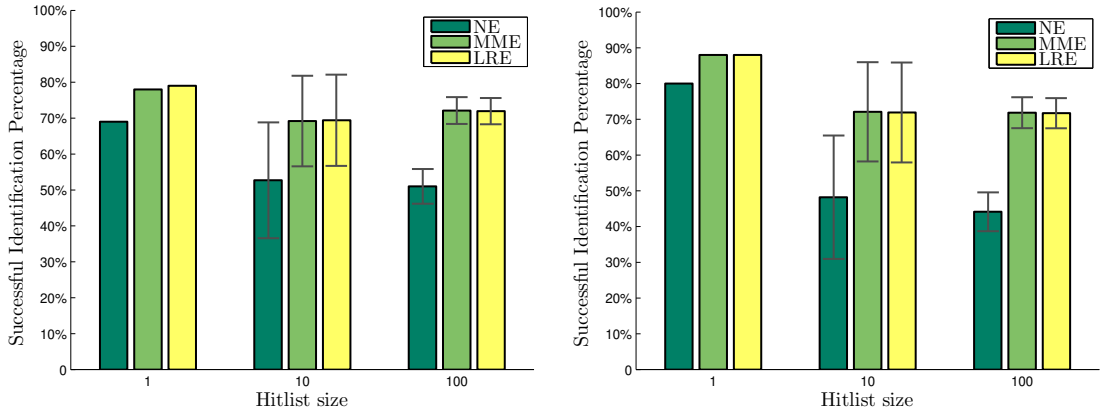
Figure 3.9: Comparison of the sequence distance for localized-scanning worms.

are similar to those in Fig. 3.8(b) and (c). In Fig. 3.9(c), we compare the performance of the estimators by increasing p_a from 0 to 0.7. Here, N = 1,000, and σ = 115. It is observed that the sequence distances of all estimators increase as p_a becomes larger. However, our estimators are always better than NE. For example, when p_a = 0.5, MME and LRE increase the inference accuracy by 27%, compared with NE.

Therefore, our proposed estimators perform much better than the naive estimator for both random-scanning and localized-scanning worms in estimating the worm infection sequence.

3.3.3 Identifying the Patient Zero or the Hitlist

A smart worm can assign lower scanning rates to the initially infected host(s) and higher scanning rates to other infected hosts. In this way, the Darknet might observe later infected hosts first, and therefore the smart worm would weaken the performance of the naive estimator. In Fig. 3.10, we compare the performance of estimators in identifying the hitlist of such a smart worm. Specifically, the worm assigns scanning rates from $N(50, 20^2)$ to the host(s) on the hitlist and scanning rates from $N(358, 110^2)$ to other infected hosts. Then, we calculate the percentage of the host(s) on the hitlist that are successfully identified by an estimator. For example, if the size of the hitlist is 100 and 50 hosts that belong to the hitlist are identified among the first 100 hosts of the estimated infection sequence, the successful identification percentage of the estimator is 50%. The results are averaged over 100 independent runs. Fig. 3.10(a) shows the case of random scanning, where the Darkent size is 2^{20} and the observation window size is 1,000 mins. It is seen that our estimators have a higher successful identification percentage and a smaller variance than the naive estimator. For instance, when the size of the hitlist is 1 (*i.e.,* the worm starts from the patient zero), MME and LRE can pinpoint the patient zero around 80% of the time, while NE can detect it only 70% of the time. When the size of the hitlist is 10 or 100, compared with NE, our proposed estimators increase the number of successfully identified hosts from 5 to 7 or 51 to 72, and reduce the variance from 2.6 to 1.6 or 23 to 13, respectively. Fig. 3.10(b) shows the results of localized scanning, where the Darkent size is 2^{24}, p_a = 0.7, and all other parameters are the same as the case of random scanning. The results are similar to those in Fig. 3.10(a). Therefore, the simulation results demonstrate that our proposed estimators are much more effective in identifying the histlist of the smart worm than the naive estimator.



(a) Random scanning (the Darknet size is 2^{20} IP addresses).

(b) Localized scanning (the Darknet size is 2^{24} IP addresses, and the value of p_a is 0.7).

Figure 3.10: Comparison of estimators in identifying the patient zero or the hitlist.

3.4 Discussions

In this section, we first analyze the chance that Darknet misses an infected host and then discuss the limitations and the extensions of our proposed estimators.

3.4.1 Host Missing Probability

By applying Darknet observations, we have made an assumption: The infected host will hit the Darknet. Then, an intuitive question would be: What is the probability that the Darknet misses an infected host within a given observation window?

We consider the case of localized scanning and regard random scanning as a special case of localized scanning when $p_a = 0$. The probability for a scan from an infected host to hit the Darknet is $(1-p_a)\cdot\omega/\Omega$; and then the probability that the Darknet misses observing the host in a time unit is $(1-(1-p_a)\cdot\omega/\Omega)^s$. Thus, the host missing probability (*i.e.*, the probability that the Darknet misses the infected host in a k time units observation window) is

$$\Pr_{\text{LS}}(\text{missing}) = \left(1-(1-p_a)\cdot\frac{\omega}{\Omega}\right)^{s\cdot k}. \tag{3.45}$$

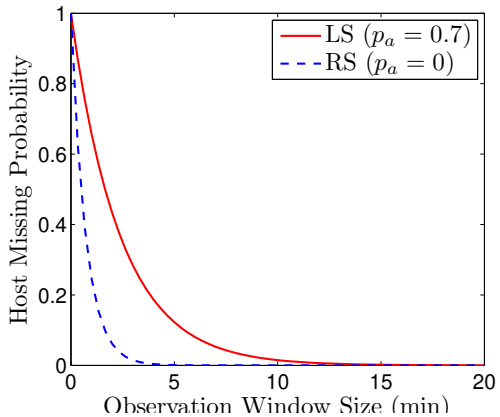


Figure 3.11: Host missing probability.

In Fig. 3.11, we show the host missing probability as the observation window size changes. In this example, we set $\omega = 2^{24}$, time unit = 20 seconds, and s = 358 scans/min. We find that if p_a = 0.7, the infected host will almost hit the Darknet for sure when the observation window size is larger than 20 mins. If p_a = 0, which is the case of random scanning, a 5-min observation window is sufficient to guarantee the capture of the infected host. Therefore, in our previous analysis and simulation, the assumption that the Darknet can observe scans from the infected host, especially at the early stage, is reasonable. Moreover, our estimator can still work even for self-stopping worms [59].

3.4.2 Estimator Limitations and Extensions

Our proposed estimators are built based on some assumptions listed in Section 3.1. Attackers that design future worms may exploit these assumptions to weaken the accuracy of our estimators. In the following, we discuss some limitations of our estimators and the potential extensions.

Darknet Avoidance

The majority of active worms up to date do not attempt to avoid the detection of Darknet. As a result, CAIDA's network telescopes have been observing many active Internet worms such as Code Red, Slammer, Witty, and even recently the Conficker worm (also known as the April Fool's worm). Most worms apply random scanning and localized scanning, and Darknet can observe the traffic from such worms.

Recent work, however, has shown that attackers can potentially detect the locations of Darknet or network sensors [60]. Thus, a future worm can be specially designed to avoid scanning the address space of the Darknet. The countermeasure against such an intelligent worm is to apply the distributed Darknet instead of the centralized Darknet [17]. That is, unused IP addresses in many subnets are used to observe worm traffic, which is then reported to a collection center for further processing. A prototype of distributed Darknet has been designed and evaluated in [61].

Scanning Rate Variation

Although there have been no observations of worms that use scanning rate variation mechanisms (*i.e.,* the scanning rate of an individual infected host is time-variant) [53], future worms may employ such schemes to invalidate our basic assumption and thus weaken the performance of our estimators. Changing the scanning rate, however, introduces additional complexity to worm design and can slow down worm spreading. Moreover, if the change of scanning rates is relatively slow, our estimators can be enhanced with the change-point detection [62] to detect and track when the scanning rate has a significant change and then apply the early observations to derive the infection time of an infected host.

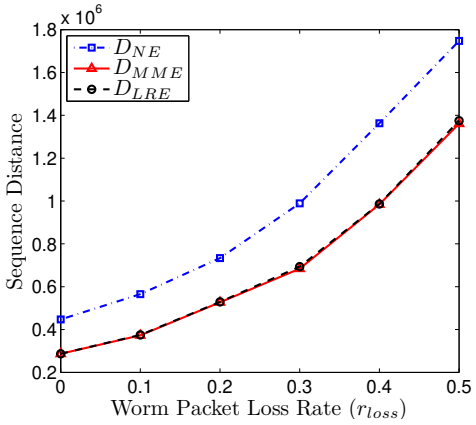


Figure 3.12: Comparison of the sequence distance varying with the worm packet loss rate.

Measurement Errors

The measurement errors can affect the performance of estimators. There are two types of measurement errors. The false positive denotes that Darknet incorrectly classifies the traffic from a benign host as worm traffic, whereas the false negative is that Darknet incorrectly classifies worm traffic as benign traffic or misses worm traffic due to congestion or device malfunction.

For the false positives, most of time we can distinguish worm traffic from other traffic. First, our estimation techniques are used as a form of post-mortem analysis on worm records logged by Darknet. As a result, we can limit our analysis to the records logged during the outbreak of the worm when it is most rampant. More importantly, worm packages always contain information about infection vectors that distinguish worm traffic from other traffic. For example, the Witty worm uses a source port of 4,000 to attack Internet Security Systems firewall products [16]. It is very unlikely that a benign host uses a source port of 4,000. By filtering the records based on infection vectors specific to the worm under investigation, we can eliminate most of the effects of false positives on Darknet observations.

False negatives are much harder to eliminate. A packet towards Darknet may be lost due to congestion caused by the worm (such as the Slammer worm [22]) or the malfunc-

tion of Darknet monitoring devices. To study the effects of false negatives, we modify our simulator to mimic the packet loss and evaluate the performance of our estimators under false negatives. Here we assume that the loss rate of the worm packets towards Darknet (denoted as r_{loss}) is the same for each infected host. Fig. 3.12 shows how the sequence distances of different estimators vary with the worm packet loss rate. The results are averaged over 20 independent runs. It is intuitive that when the packet loss rate becomes larger, the performance of all estimators worsens. Our proposed estimators, however, always perform much better than NE. For example, compared with NE, our estimators (*i.e.,* MME and LRE) improve the inference accuracy by 28% when $r_{\text{loss}} = 0.4$. A mechanism to recover from worm-induced congestion has been proposed in [53], which estimates the packet loss rates of infected hosts based on Darknet observations and BGP atoms. This method can be incorporated into our estimators to enhance their robustness against worm-induced congestion.

CHAPTER 4

CHARACTERIZING INTERNET WORM SPATIAL INFECTION STRUCTURE

Modeling Internet worm infection has been focused on the macro level. Most, if not all, mathematical models study the total number of infected hosts over time [7, 8, 9, 11, 1]. For example, Staniford *et al.* used a simple differential equation to estimate the global propagation speed of the Code Red v2 worm [7], whereas Rohloff *et al.* applied a stochastic model to reflect the variation of the number of infected hosts at the early stage of worm infection [10]. The models of some key micro-level information of worm infection, such as the infection ability of individual hosts and the underlying topology formed by worm infection, has been investigated little.

The goal of this chapter is to bridge the gap by characterizing the spatial infection relationship between individual infected hosts, *i.e.,* the worm spatial infection structure. Specifically, we reveal the key characteristics of the underlying topology formed by worm infection, *i.e.,* the number of children and the generation of the worm tree. To study these two metrics analytically, we apply probabilistic modeling methods and derive the probability distributions of the number of children and the generation through a sequential growth model. Different from other models that characterize the dynamics of worm propagation (*e.g.,* the total number of infected hosts over time), our sequential growth model aims at capturing the main features of the topology formed by worm infection (*e.g.,* the number of children and the generation). To the best of our knowledge, there is yet no mathematical model for characterizing the structure of the worm tree. We then verify the analytical results through simulations.

The remainder of this chapter is structured as follows. Section 4.1 presents our sequential growth model and assumptions used in analyzing the worm tree. Section 4.2 gives our analysis on the worm tree. Section 4.3 then uses simulations to verify the analytical results and provide observations on the worm tree using the localized-scanning method.

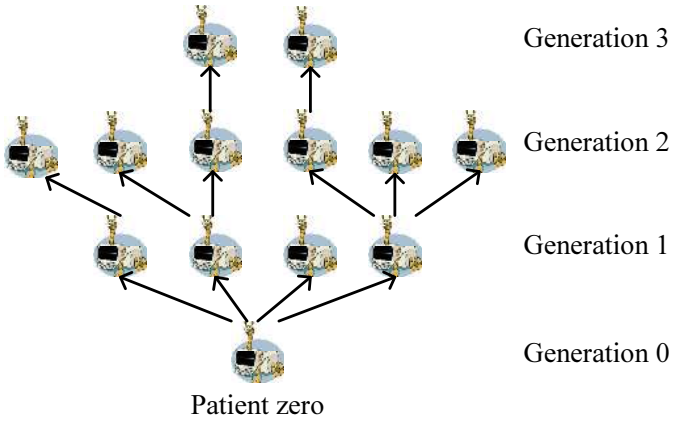
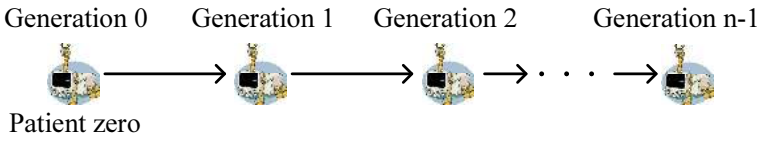


Figure 4.1: An example of the worm tree.

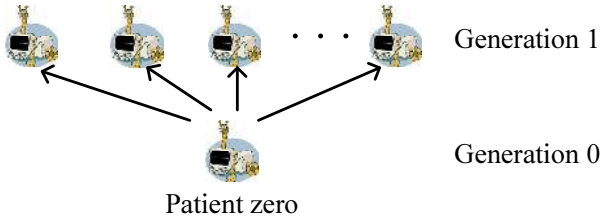
4.1 Worm Tree and Sequential Growth Model

In this section, we provide the background on the worm tree, and present the assumptions and the growth model.

An example of a worm tree is given in Fig. 4.1. Here, patient zero is the root and belongs to generation 0. The tail of an arrow is from the "father" or the infector, whereas the head of an arrow points to the "son" or the infectee. If a father belongs to generation i, then its children lie in generation $i+1$. In a worm tree with n nodes, we use $L_n(i,j)$ $(0 \leq i,j \leq n-1)$ to denote the number of nodes that have i children and belong to generation j. Note that $\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} L_n(i,j) = n$. We also use $C_n(i)$ $(i = 0,1,2,\cdots,n-1)$ to denote the number of nodes that have i children and $G_n(j)$ $(j = 0,1,2,\cdots,n-1)$ to denote the number of nodes in generation j. Moreover, $L_n(i,j)$, $C_n(i)$, and $G_n(j)$ are random variables. Thus, we define $p_n(i,j) = \frac{\mathrm{E}[L_n(i,j)]}{n}$, representing the joint distribution of the number of children and the generation. Similarly, we define $c_n(i) = \frac{\mathrm{E}[C_n(i)]}{n}$ to represent the marginal distribution of the number of children and $g_n(j) = \frac{\mathrm{E}[G_n(j)]}{n}$ to represent the marginal distribution of the generation. Note that $c_n(i) = \sum_{j=0}^{n-1} p_n(i,j)$ and $g_n(j) = \sum_{i=0}^{n-1} p_n(i,j)$.



(a) Extreme case 1.



(b) Extreme case 2.

Figure 4.2: Two extreme cases of worm trees.

Although we model worm infection as a tree, different worm trees can show very different structures. Fig. 4.2 demonstrates two extreme cases of worm trees. Specifically, in Fig. 4.2 (a), each infected host compromises one and only one host except the last infected host. In this case, if the total number of nodes is n, $C_n(0) = 1$, and $C_n(1) = n - 1$, which lead to $c_n(0) = \frac{1}{n}$ and $c_n(1) = \frac{n-1}{n} \approx 1$ when n is large. That is, almost each node has one and only one child. Moreover, $G_n(j) = 1$, $j = 0, 1, 2, \cdots, n-1$, which means that $g_n(j) = \frac{1}{n}$. Thus, the average path length is $\sum_{j=0}^{n-1} j \cdot g_n(j) = \frac{n-1}{2} \sim O(n)$. That is, the average path length increases linearly with the number of nodes. Comparatively, Fig. 4.2 (b) shows another case where all hosts (except patient zero) are infected by patient zero. For the distribution of the number of children, $c_n(n-1) = \frac{1}{n}$, and $c_n(0) = \frac{n-1}{n} \approx 1$ when n is large, indicating that almost every node has no child. For the distribution of the generation, $g_n(0) = \frac{1}{n}$, and $g_n(1) = \frac{n-1}{n}$, which leads to that the average path length is $\frac{n-1}{n} \approx 1$ when n is large. Thus, the path length is close to a constant of 1. In this chapter, we attempt to identify the structure of the worm tree formed by Internet worm infection.

To study the worm tree analytically, in this work we make several assumptions and considerations. First, to simplify the model, we assume that infected hosts have the same

scanning rate. This assumption is removed in Section 4.3.2, where we use simulations to study the effect of the variation of scanning rates on the worm tree. Second, we consider a wide class of worms for which a new victim is compromised by each existing infected host with equal probability. Such worms include random-scanning worms, routable-scanning worms, importance-scanning worms, OPT-STATIC worms, and SUBOPT-STATIC worms. Random scanning selects targets in the IPv4 address space randomly and has been the main scanning method for both worms and botnets [7, 2]; routable scanning finds victims in the routable IPv4 address space [63, 8]; and importance scanning probes subnets according to the vulnerable-host distribution [54]. OPT-STATIC and SUBOPT-STATIC are optimal and suboptimal scanning methods that are proposed in [64] to minimize the number of worm scans required to reach a predetermined fraction of vulnerable hosts. In Section 4.3.3, we extend our study to localized scanning, which preferentially searches for targets in the local subnet and has also been used by real worms [65, 55]. Third, we consider the classic susceptible $\rightarrow$ infected (SI) model, ignoring the cases that an infected host can be cleaned and becomes vulnerable again, or can be patched and becomes invulnerable. The SI model assumes that once infected, a host remains infected. Such a simple model has been widely applied in studying worm infection [7, 8, 64], and presents the worst case scenario. Fourth, we assume that there is no re-infection. That is, if an infected host is hit by a worm scan, this host will not be further re-infected. As a result, every infected host has one and only one father except for patient zero, and the resulting graph formed by worm infection is a tree. Fifth, we assume that the worm starts from one infected host, *i.e.,* patient zero or a hitlist size of 1. When the hitlist size is larger than 1, the underlying infection topology is a worm forest, instead of a worm tree. Our analysis, however, can easily be extended to model the worm forest. Finally, to simplify the analysis, we assume that no two nodes are added to the worm tree at the same time. That is, no two vulnerable hosts are infected simultaneously. We relax this assumption in Section 4.3 where simulations are performed.

Based on these considerations and assumptions, the sequential growth model of a worm tree works as follows: We consider a fixed sequence of infected hosts (*i.e.,* nodes) $v_1, v_2, \cdots$ and inductively construct a random worm tree $(T_n)_{n\geq 1}$, where n is the number of nodes and T_1 has only patient zero. Infecting a new host is equivalent to adding a new node into the existing worm tree. Hence, given T_{n-1}, T_n is formed by adding node v_n together with an edge directed from an existing node v_f to v_n. According to the assumption, v_f is randomly chosen among the $n-1$ nodes in the tree, *i.e.*, $\Pr(f = k) = \frac{1}{n-1}$, $k = 1, 2, \cdots, n-1$. Note that such a sequential growth model and its variations have been widely used in studying topology generators [28]. In this chapter, we apply this model to characterize worm spatial infection strucutre.

4.2 Characterizing Internet Worm Spatial Infection Structure

In this section, we characterize the topology of the worm tree through mathematical analysis. Specifically, we first derive the joint distribution of the number of children and the generation, *i.e.,* $p_n(i, j)$, by applying probabilistic methods. We then use $p_n(i, j)$ to analyze two marginal distributions, *i.e.,* $c_n(i)$ and $g_n(j)$, and obtain their closed-form approximations. Finally, we find a closed-form approximation to $p_n(i, j)$.

4.2.1 Joint Distribution

For a worm tree with only patient zero (*i.e.,* $n = 1$), since $L_1(0, 0) = 1$ with probability 1, $p_1(0, 0) = 1$. Similarly, for a worm tree with $n = 2$, it is evident that $L_2(1, 0) = L_2(0, 1) = 1$. Thus, $p_2(1, 0) = p_2(0, 1) = \frac{1}{2}$. We now consider $p_n(i, j)$ $(0 \leq i, j \leq n-1)$ when $n \geq 3$. Specifically, we study two cases:

(1) $p_n(0, j)$, *i.e.,* the proportion of the number of leaves in generation j in T_n. Assume that T_{n-1} is given, and there are $L_{n-1}(0, j)$ leaves in generation j and totally $G_{n-1}(j-1) =$

$\sum_{i=0}^{n-2} L_{n-1}(i, j-1)$ nodes in generation $j-1$. Note that we have extended the notation so that $G_{n-1}(-1) = L_{n-1}(i,-1) = 0$, $0 \le i \le n-2$. When a new node v_n is added, v_n becomes a leaf of T_n. If v_n is connected to one of existing nodes in generation $j-1$, v_n belongs to generation j; and the probability of such an event is $\frac{G_{n-1}(j-1)}{n-1}$. Moreover, if a leaf in generation j in T_{n-1} connects to v_n, this node is no longer a leaf and now has one child; and the probability of this event is $\frac{L_{n-1}(0,j)}{n-1}$. Therefore, we can obtain the stochastic recurrence of $L_n(0,j)$:

$$L_n(0,j) = \begin{cases} L_{n-1}(0,j)+1, & \text{w.p. } \frac{G_{n-1}(j-1)}{n-1} \\ L_{n-1}(0,j)-1, & \text{w.p. } \frac{L_{n-1}(0,j)}{n-1} \\ L_{n-1}(0,j), & \text{otherwise.} \end{cases} \tag{4.1}$$

Given T_{n-1} (*i.e.,* $L_{n-1}(0,j)$ and $G_{n-1}(j-1)$), the conditional expected value of $L_n(0,j)$ is $[L_{n-1}(0,j)+1]\cdot\frac{G_{n-1}(j-1)}{n-1} + [L_{n-1}(0,j)-1]\cdot\frac{L_{n-1}(0,j)}{n-1} + L_{n-1}(0,j)\cdot\left[1 - \frac{G_{n-1}(j-1)+L_{n-1}(0,j)}{n-1}\right]$, *i.e.,*

$$\mathrm{E}[L_n(0,j)|T_{n-1}] = \tfrac{n-2}{n-1} L_{n-1}(0,j) + \tfrac{1}{n-1} G_{n-1}(j-1). \tag{4.2}$$

Applying $\mathrm{E}[L_n(0,j)] = \mathrm{E}[\mathrm{E}[L_n(0,j)|T_{n-1}]]$ (*i.e.,* the law of total expectation), we obtain

$$\mathrm{E}[L_n(0,j)] = \tfrac{n-2}{n-1}\mathrm{E}[L_{n-1}(0,j)] + \tfrac{1}{n-1}\mathrm{E}[G_{n-1}(j-1)]. \tag{4.3}$$

Using the definitions $p_n(0,j) = \frac{\mathrm{E}[L_n(0,j)]}{n}$ and $g_{n-1}(j-1) = \frac{\mathrm{E}[G_{n-1}(j-1)]}{n-1} = \sum_{i=0}^{n-2} p_{n-1}(i, j-1)$, the above equation leads to

$$\begin{aligned} p_n(0,j) &= \tfrac{n-2}{n} p_{n-1}(0,j) + \tfrac{1}{n} g_{n-1}(j-1) & (4.4) \\ &= \tfrac{n-2}{n} p_{n-1}(0,j) + \tfrac{1}{n}\textstyle\sum_{i=0}^{n-2} p_{n-1}(i,j-1). & (4.5) \end{aligned}$$

(2) $p_n(i,j)$, $1 \le i \le n-1$. Given $L_{n-1}(i,j)$ and $L_{n-1}(i-1,j)$ in T_{n-1}, we study $L_n(i,j)$ in T_n. When the new node v_n is added into T_{n-1}, v_n is connected to a node with $i-1$ children and in generation j with probability $\frac{L_{n-1}(i-1,j)}{n-1}$, or is connected to a node

with i children and in generation j with probability $\frac{L_{n-1}(i,j)}{n-1}$. Thus, in T_n,

$$L_n(i,j) = \begin{cases} L_{n-1}(i,j)+1, & \text{w.p. } \frac{L_{n-1}(i-1,j)}{n-1} \\ L_{n-1}(i,j)-1, & \text{w.p. } \frac{L_{n-1}(i,j)}{n-1} \\ L_{n-1}(i,j), & \text{otherwise.} \end{cases} \tag{4.6}$$

This relationship leads to

$$\mathrm{E}[L_n(i,j)|T_{n-1}] = \tfrac{n-2}{n-1}L_{n-1}(i,j) + \tfrac{1}{n-1}L_{n-1}(i-1,j). \tag{4.7}$$

Therefore,

$$\mathrm{E}[L_n(i,j)] = \tfrac{n-2}{n-1}\mathrm{E}[L_{n-1}(i,j)] + \tfrac{1}{n-1}\mathrm{E}[L_{n-1}(i-1,j)]. \tag{4.8}$$

That is,

$$p_n(i,j) = \tfrac{n-2}{n}p_{n-1}(i,j) + \tfrac{1}{n}p_{n-1}(i-1,j). \tag{4.9}$$

Summarizing the above two cases, we have the following theorem:

Theorem 4.2.1 *When $n \geq 3$, the joint distribution of the number of children and the generation in a worm tree T_n follows*

$$p_n(i,j) = \begin{cases} \frac{n-2}{n}p_{n-1}(0,j) + \frac{1}{n}g_{n-1}(j-1), & i = 0 \\ \frac{n-2}{n}p_{n-1}(i,j) + \frac{1}{n}p_{n-1}(i-1,j), & \textit{otherwise}, \end{cases} \tag{4.10}$$

where $0 \leq i, j \leq n-1$.

Theorem 4.2.1 provides a way to calculate $p_n(i,j)$ recursively from $p_2(i,j)$. Fig. 4.3 shows a snapshot of $p_n(i,j)$ when $n = 2000$. It can be seen that when the generation is specified (*i.e.,* j is fixed), $p_n(i,j)$ is a monotonous function and decreases quickly as i increases. On the other hand, when the number of children is given (*i.e.,* i is fixed), $p_n(i,j)$ has a bell shape. Moreover, since $\sum_{i=0}^{10}\sum_{j=0}^{15} p_n(i,j) = 0.9976$, most nodes do not have a large number of children, and the worm tree does not have a large average path length.

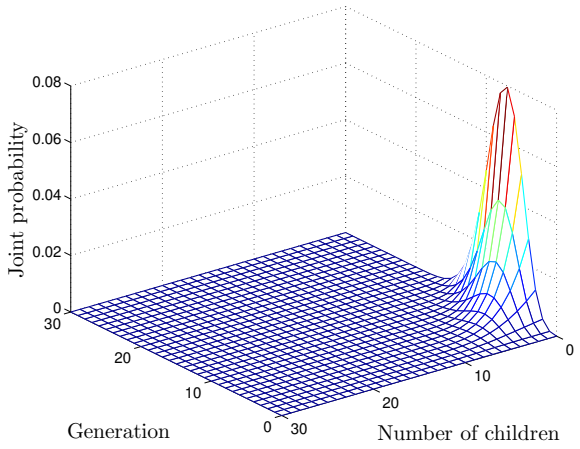


Figure 4.3: Joint distribution of the number of children and the generation.

4.2.2 Number of Children

We use $p_n(i,j)$ to derive the marginal distribution of the number of children, *i.e.,* $c_n(i)$. Similarly, we study two cases:

(1) $c_n(0)$, *i.e.,* the proportion of the number of leaves in T_n. Since $c_n(0) = \sum_{j=0}^{n-1} p_n(0,j)$ and $\sum_{j=0}^{n-1} g_{n-1}(j-1) = 1$, we obtain the recursive relationship of $c_n(0)$ from Equation (4.4):

$$c_n(0) = \tfrac{n-2}{n} c_{n-1}(0) + \tfrac{1}{n}. \tag{4.11}$$

Moreover, note that $c_2(0) = \frac{1}{2}$. If we assume that $c_{n-1}(0) = \frac{1}{2}$, we can obtain by induction that

$$c_n(0) = \tfrac{1}{2}. \tag{4.12}$$

This indicates that no matter how many nodes are in the worm tree, on average half of nodes are leaves, *i.e.,* on average 50% of infected hosts never compromise any target.

(2) $c_n(i)$, $1 \le i \le n-1$. From Equation (4.9) and $c_n(i) = \sum_{j=0}^{n-1} p_n(i,j)$, we find the recurrence of $c_n(i)$ as follows

$$c_n(i) = \tfrac{n-2}{n} c_{n-1}(i) + \tfrac{1}{n} c_{n-1}(i-1). \tag{4.13}$$

Summarizing the above two cases, we have the following theorem on the distribution of the number of children:

Theorem 4.2.2 *When $n \geq 3$, the distribution of the number of children in a worm tree T_n follows*

$$c_n(i) = \begin{cases} \frac{1}{2}, & i = 0 \\ \frac{n-2}{n} c_{n-1}(i) + \frac{1}{n} c_{n-1}(i-1), & 1 \leq i \leq n-1. \end{cases} \tag{4.14}$$

From Theorem 4.2.2, we can derive the statistical properties of the number of children as follows.

Corollary 4.2.3 *When $n \geq 1$, the expectation and the variance of the number of children are*

$$E_n[C] = \sum_{i=0}^{n-1} i \cdot c_n(i) = \frac{n-1}{n} \tag{4.15}$$

$$Var_n[C] = \sum_{i=0}^{n-1} \left(i - E_n[C]\right)^2 \cdot c_n(i) = 2 - \frac{n-1}{n^2} - \frac{2H_n}{n}, \tag{4.16}$$

where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ is the n-th harmonic number [15].

The proof of Corollary 4.2.3 is given in Appendix B.1. One intuitive way to derive $\mathrm{E}_n[C]$ is that in worm tree T_n, there are $n-1$ directed edges and n nodes. Thus, the average number of edges (*i.e,* the average number of children) of a node is $\frac{n-1}{n}$. Moreover, since H_n is $O(1 + \ln n)$, $\lim_{n\to\infty} \mathrm{E}_n[C] = 1$, and $\lim_{n\to\infty} \mathrm{Var}_n[C] = 2$.

Theorem 4.2.2 also leads to a simple closed-form expression of the distribution of the number of children when n is very large, as shown in the following corollary.

Corollary 4.2.4 *When $n \to \infty$, the number of children has a geometric distribution with parameter $\frac{1}{2}$,* i.e.,

$$c(i) = \lim_{n\to\infty} c_n(i) = \left(\frac{1}{2}\right)^{i+1}, \quad i = 0, 1, 2, \cdots. \tag{4.17}$$

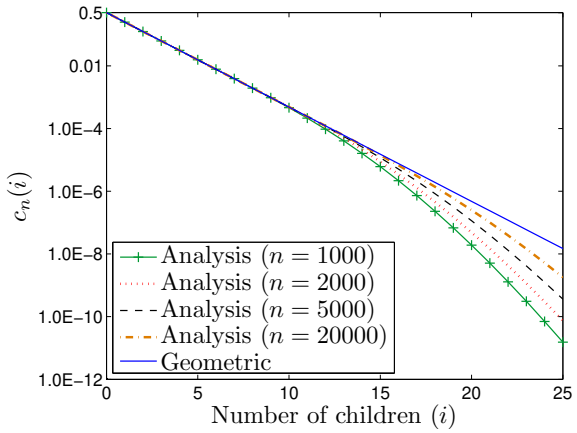


Figure 4.4: Number of children.

Proof. It is already known that $c(0) = \frac{1}{2}$. When $i \geq 1$, this corollary follows readily from Equation (4.13). Since $n \to \infty$, $c_{n-1}(i) = c_n(i) = c(i)$, which yields

$$c(i) = \tfrac{n-2}{n} c(i) + \tfrac{1}{n} c(i-1). \tag{4.18}$$

That is,

$$c(i) = \tfrac{1}{2} c(i-1), \;\; i \geq 1. \tag{4.19}$$

Hence, from $c(0) = \frac{1}{2}$, we can recursively obtain Equation (5.6). □

Corollary 4.2.4 indicates that when n is very large, $c_n(i)$ decreases approximately exponentially with a decay constant of $\ln 2$ as the number of children increases. We further study when both n and i are finite and large, how $c_n(i)$ varies with n, *i.e.,* how the tail of the distribution of the number of children changes with n. First, note that $c_3(0) = \frac{1}{2}$, $c_3(1) = \frac{1}{3}$, and $c_3(2) = \frac{1}{6}$. Thus, from Equation (4.13), we can prove by induction that $c_n(i)$ $(n \geq 3)$ is a decreasing function of i, *i.e.,* $c_n(i) < c_n(i-1)$, for $1 \leq i \leq n-1$. Next, putting this inequality into Equation (4.13), we have $c_n(i) > \frac{n-1}{n} c_{n-1}(i)$. Hence, when n is very large, $\frac{n-1}{n} \approx 1$, and $c_n(i) > c_{n-1}(i)$, which indicates that the tail of $c_n(i)$ increases with n. Fig. 4.4 verifies this result, showing $c_n(i)$ obtained from Theorem 4.2.2 when $n = 1000$, 2000, 5000, and 20000, as well as the geometric distribution with parameter 0.5 obtained from

Corollary 4.2.4. Note that the y-axis uses log-scale. It can be seen that when n increases from 1000 to 20000, the tail of $c_n(i)$ also increases to approach the tail of the geometric distribution. Moreover, it is shown that the geometric distribution well approximates the distribution of the number of children when n is large.

4.2.3 Generation

Next, we derive the generation distribution (*i.e.,* $g_n(j)$) in a similar manner to the case of $c_n(i)$. Using Theorem 4.2.1 and $g_n(j) = \sum_{i=0}^{n-1} p_n(i,j)$, we obtain the following theorem:

Theorem 4.2.5 *When $n \geq 3$, the distribution of the generation in a worm tree T_n follows*

$$g_n(j) = \tfrac{n-1}{n} g_{n-1}(j) + \tfrac{1}{n} g_{n-1}(j-1), 0 \leq j \leq n-1, \tag{4.20}$$

where $g_{n-1}(-1) = 0$.

Theorem 4.2.5 gives a method to calculate the distribution of the generation recursively. Moreover, from Theorem 4.2.5, we can derive the statistical properties of the generation distribution in the following corollary.

Corollary 4.2.6 *When $n \geq 1$, the expectation and the variance of the generation are*

$$E_n[G] = \textstyle\sum_{j=0}^{n-1} j \cdot g_n(j) = H_n - 1. \tag{4.21}$$

$$Var_n[G] = \textstyle\sum_{j=0}^{n-1} \left(j - E_n[G]\right)^2 \cdot g_n(j) = H_n - H_{n,2}, \tag{4.22}$$

where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ and $H_{n,2} = \sum_{i=1}^{n} \frac{1}{i^2}$.

The proof of Corollary 4.2.6 is given in Appendix B.2. From Corollary 4.2.6, we have some interesting observations. Since H_n is $O(1 + \ln n)$ and $H_{\infty,2} = \zeta(2) = \frac{\pi^2}{6} \approx 1.645$ is the Riemann zeta function of 2 [66], both $\mathrm{E}_n[G]$ and $\mathrm{Var}_n[G]$ are $O(1 + \ln n)$. This indicates that the average path length of the worm tree (*i.e.*, $\mathrm{E}_n[G]$) increases approximately logarithmically with n. Moreover, when $n \to \infty$, $\lim_{n\to\infty} \mathrm{E}_n[G] - \ln n = \gamma - 1$, and

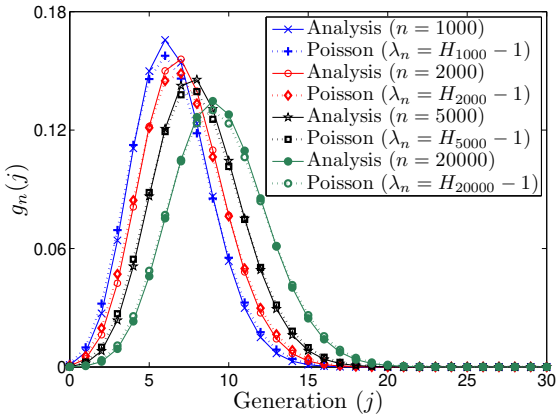


Figure 4.5: Generation.

$\lim_{n\to\infty} \text{Var}_n[G] - \ln n = \gamma - \zeta(2)$, where $\gamma \approx 0.577$ is the Euler-Mascheroni constant [67]. Therefore, when n is large, $\text{E}_n[G] \approx \text{Var}_n[G]$. Furthermore, we can use Theorem 4.2.5 to obtain a closed-form approximation to $g_n(j)$ as follows.

Corollary 4.2.7 *When n is very large, the generation distribution $g_n(j)$ can be approximated by a Poisson distribution with parameter $\lambda_n = E_n[G] = H_n - 1$. That is,*

$$g_n(j) \approx \frac{\lambda_n^j}{j!} e^{-\lambda_n}, \quad 0 \le j \le n-1. \tag{4.23}$$

Proof. We prove this corollary by applying z-transform. If a random variable X follows a Poisson distribution with parameter λ,

$$\Pr(X = k) = \frac{\lambda^k}{k!} e^{-\lambda}, \quad k = 0, 1, 2, \cdots. \tag{4.24}$$

Using z-transform, we have

$$X(z) = \sum_{k=0}^{\infty} \Pr(X = k) z^{-k} = e^{\lambda\left(z^{-1}-1\right)}. \tag{4.25}$$

Meanwhile, using Equation (4.20) in Theorem 4.2.5, we find the z-transform of $g_n(j)$

$$Y_n(z) = \textstyle\sum_{j=0}^{n-1} g_n(j) z^{-j} = \left(1 + \frac{z^{-1}-1}{n}\right) Y_{n-1}(z). \tag{4.26}$$

Note that when $x \to 0$, $e^x \approx 1 + x$. Thus, when n is very large, $1 + \frac{z^{-1}-1}{n} \approx \exp((z^{-1} - 1)/n)$. That is,

$$Y_n(z) \approx e^{\frac{z^{-1}-1}{n}} Y_{n-1}(z). \tag{4.27}$$

Using $Y_1(z) = 1$, we can recursively obtain

$$Y_n(z) \approx e^{(z^{-1}-1)\sum_{i=2}^{n} \frac{1}{i}} = e^{(H_n-1)(z^{-1}-1)}. \tag{4.28}$$

Therefore, comparing Equations (4.25) and (4.28), we find that $g_n(j)$ can be approximated by the Poisson distribution with parameter $H_n - 1$ as in Equation (4.23). □

Fig. 4.5 verifies Corollary 4.2.7, showing $g_n(j)$ obtained from Theorem 4.2.5 when $n = 1000, 2000, 5000$, and 20000, as well as the Poisson distribution with parameter $\mathrm{E}_n[G]$. It can be seen that when n is large, the Poisson distribution fits the generation distribution closely.

4.2.4 Approximation to the Joint Distribution

Finally, we derive a closed-form approximation to the joint distribution $p_n(i, j)$. From Equation (4.9), we can see that when $n \to \infty$, $p_n(i, j) = p_{n-1}(i, j)$, which yields

$$p_n(i, j) = \tfrac{1}{2} p_n(i - 1, j). \tag{4.29}$$

Hence, we can obtain

$$p_n(i, j) = \left(\tfrac{1}{2}\right)^i p_n(0, j) \approx \left(\tfrac{1}{2}\right)^{i+1} g_n(j). \tag{4.30}$$

Since when n is very large, $g_n(j)$ follows closely the Poisson distribution as in Corollary 4.2.7,

$$p_n(i, j) \approx \left(\tfrac{1}{2}\right)^{i+1} \cdot \tfrac{\lambda_n^j}{j!} e^{-\lambda_n}, \;\; 0 \le i, j \le n - 1, \tag{4.31}$$

where $\lambda_n = H_n - 1$. The above derivation also shows that when n is very large, the number of children and the generation are almost independent random variables.

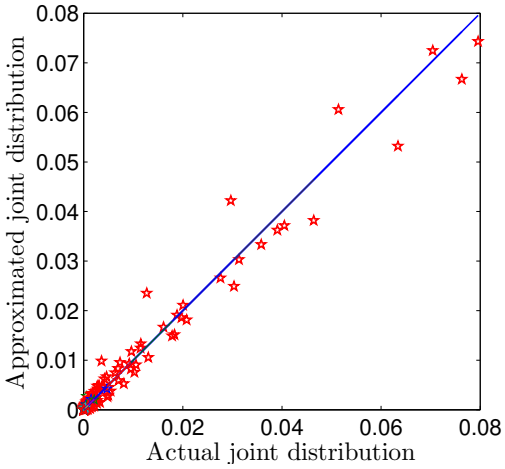
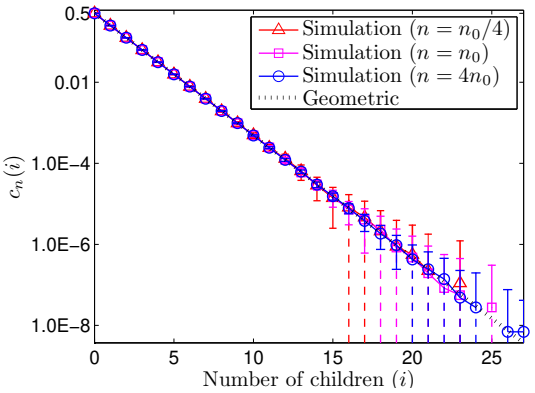


Figure 4.6: Joint distribution.

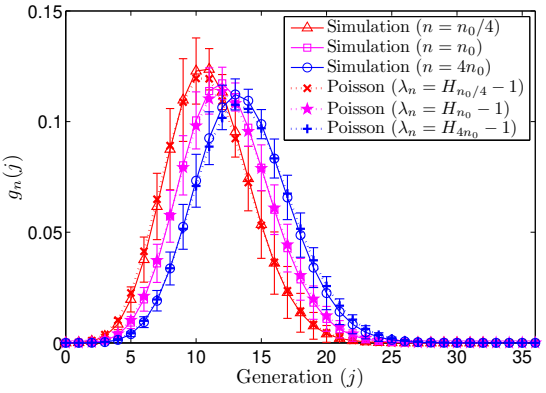
Fig. 4.6 shows the parity plot of the approximation to the joint distribution when $n = 2000$. In the figure, the x-axis is the actual $p_n(i, j)$ obtained from Theorem 4.2.1, and the y-axis is the approximated $p_n(i, j)$ from Equation (4.31), where $0 \leq i, j \leq 30$. It can be seen that most points are on or near the diagonal line, indicating that the approximation to the joint distribution is reasonable.

4.3 Simulations and Verification

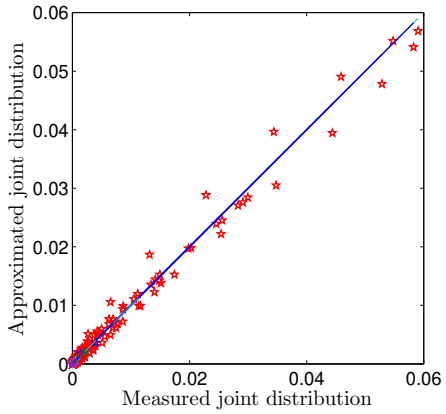
In this section, we study the worm spatial infection structure through simulations. As far as we know, there is no publicly available data to show the real worm tree and verify our analytical results. Moreover, real experiments in a controlled environment are impractical for this study since the closed-form approximations are derived based on the assumption that the number of nodes is very large. Therefore, we apply empirical simulations. Specifically, we first simulate the spatial infection structure of the Code Red v2 worm and then study the effects of important parameters on the worm tree. Finally, we extend our simulation to localized-scanning worms.



(a) Number of children.



(b) Generation.



(c) Joint distribution ($n = n_0$).

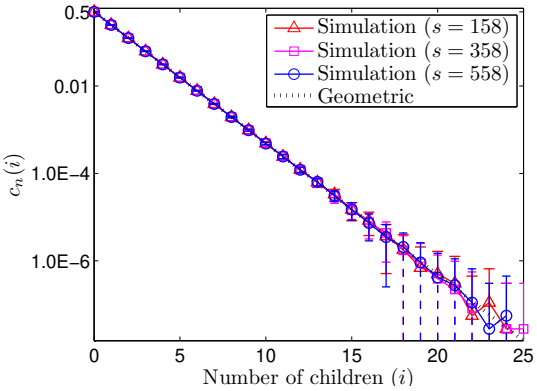
Figure 4.7: Simulating the spatial infection structure of the Code Red v2 worm.

4.3.1 Code Red v2 Worm Verification

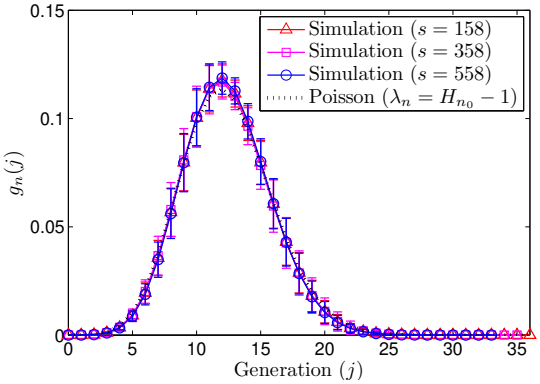
Similarly to Chapter 3, we simulate the propagation of the Code Red v2 worm by using and extending the simulator in [68]. Here, the vulnerable population is $n_0 = 360,000$, and a newly infected host is assigned with a scanning rate of 358 scans/min. We then extend the simulator to track the worm spatial infection structure by adding the information of the number of children and the generation to each infected host. Moreover, we set the time unit to 20 seconds and start our simulation at time tick 0 with patient zero. Note that we remove the assumption used in the sequential growth model that no two hosts are compromised at

the same time. That is, multiple hosts can be compromised at one time tick. Moreover, all new victims of the current time tick start scanning at the next time tick. The simulation results (mean $\pm$ standard deviation) are obtained from 100 independent runs with different seeds and are presented in Fig. 4.7.

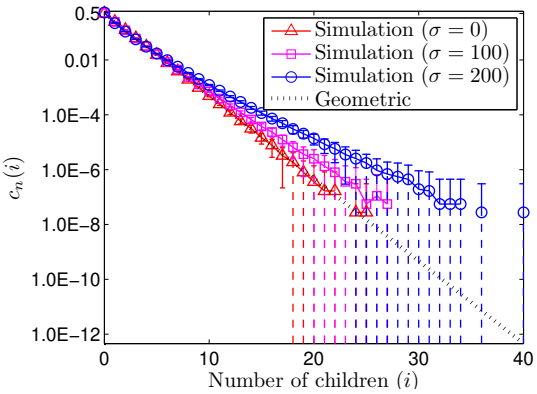
Fig. 4.7(a) shows the distribution of the number of children, comparing the simulation results of $c_n(i)$ for $n = n_0/4$, n_0, and $4n_0$ with the geometric distribution obtained from Corollary 4.2.4. Note that the y-axis uses the log-scale. The vertical dotted line represents the standard deviation that goes into the negative territory. It can be seen that the distribution of the number of children can be well approximated by the geometric distribution with parameter 0.5. This implies that $c_n(i)$ decreases approximately exponentially with a decay constant of $\ln 2$. Specifically, in all three cases, on average 50.0% of the infected hosts do not have children, about 98.4% of them have no more than five children, and 0.1% of them have no less than ten children. We also calculate the expectation and the variance of the number of children from the simulation and find that they are identical to the analytical results obtained from Corollary 4.2.3. Fig. 4.7(b) demonstrates the generation distribution, comparing the simulation results of $g_n(j)$ for $n = n_0/4$, n_0, and $4n_0$ with the Poisson distributions with parameter $\mathrm{E}_n[G] = H_n - 1$ obtained from Corollary 4.2.7. It can be seen that the simulation results of $g_n(j)$ closely follow the Poisson distributions for all three cases. Hence, simulation results verify that the average path length of the worm tree increases approximately logarithmically with the total number of infected hosts. Moreover, we also compute the expectation and the variance of the generation in simulations and verify the analytical results in Corollary 4.2.6. Fig. 4.7(c) compares the measured joint distribution from simulations with the approximated joint distribution from Equation (4.31) by using the parity plot. It can be seen that most points are on or near the diagonal line, indicating that the approximation works well.



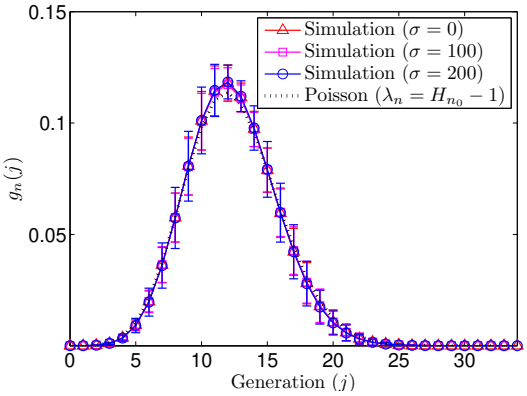
(a) Effect of s on $c_n(i)$.



(b) Effect of s on $g_n(j)$.



(c) Effect of σ on $c_n(i)$.

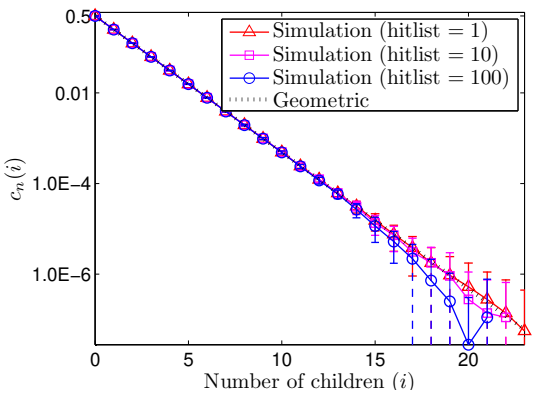


(d) Effect of σ on $g_n(j)$.

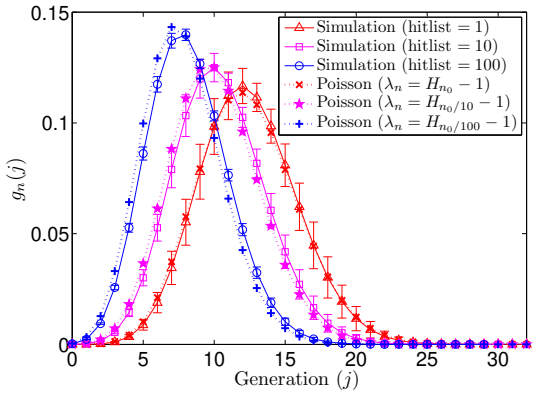
4.3.2 Effects of Worm Parameters

Next, we extend our simulator to examine the effects of three important parameters of worm propagation on the worm tree: the scanning rate, the scanning rate standard deviation, and the hitlist size. When a parameter is studied and varied, we set other parameters to the parameters of the Code Red v2 worm as used in Section 4.3.1. The simulation results are obtained from 100 independent simulation runs and are shown in Fig. 4.8.

Fig.s 4.8(a) and (b) show the effect of varying the scanning rate s (scans/min) from 158 to 558 on the distributions of the number of children and the generation. Here, the scanning rate is set to a fixed value for every infected host, *i.e.,* the scanning rate standard deviation



(e) Effect of hitlist sizes on $c_n(i)$.



(f) Effect of hitlist sizes on $g_n(j)$.

Figure 4.8: Effects of s, σ, and the hitlist size on $c_n(i)$ and $g_n(j)$.

is 0. The figures also plot the geometric distribution with parameter 0.5 and the Poisson distribution with parameter $H_{n_0} - 1$ for reference. It can be seen that the scanning rate does not affect the worm tree structure.

Fig.s 4.8(c) and (d) demonstrate the effect of the variation of the scanning rates among different hosts (*i.e.*, σ). In our simulation, a newly infected host is assigned with a scanning rate (scans/min) from a normal distribution $N(358, \sigma^2)$. The figures show the simulation results when $\sigma = 0$, 100, and 200. It can be seen that while the scanning rate standard derivation σ has no effect on the generation distribution, it does affect the distribution of the number of children. Specifically, when σ increases, the tail of $c_n(i)$ moves upward from the geometric distribution with parameter 0.5. This is because when σ becomes larger, the variation of the scanning rate among infected hosts is greater. That is, there are more hosts with high scanning rates and also more hosts with low scanning rates. As a result, those hosts with high scanning rates tend to infect a large number of hosts, making the tail of $c_n(i)$ move upward. However, it is also observed that when σ is not very large (the case for real worms), the geometric distribution with parameter 0.5 is still a good approximation.

In Fig.s 4.8(e) and (f), we show the effect of the hitlist size on the worm tree. As pointed

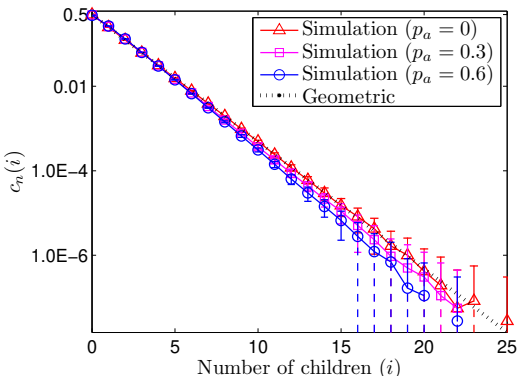
out in Section 4.1, when the hitlist size is greater than 1, the underlying infection topology is a worm forest with the number of trees equal to the hitlist size. Moreover, in a worm forest, it is intuitive that each tree is a smaller version of the single worm tree of hitlist size 1 and has fewer nodes. Hence, it is not surprising to see that in Fig. 4.8(f), the generation distribution moves leftward when the hitlist size increases. However, the generation distribution can still be well approximated by the Poisson distribution with parameter $H_{n_h} - 1$, where n_h is the average number of nodes in a tree. Moreover, since in each tree the distribution of the number of children can be approximated by the geometric distribution with parameter 0.5, in the worm forest $c_n(i)$ still follows closely the same distribution.

4.3.3 Localized Scanning

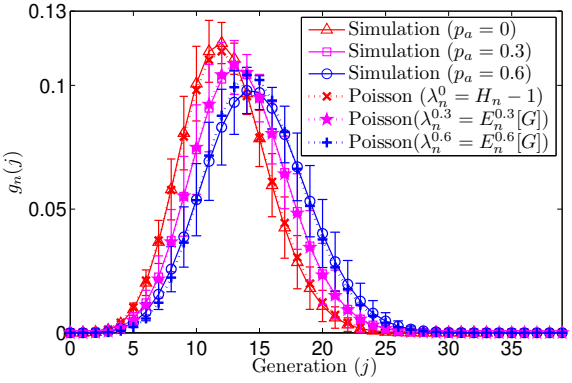
Finally, we extend our simulation study to the infection tree of localized-scanning worms. Different from random scanning, localized scanning preferentially searches for targets in the "local" address space [7]. As a result, when a new node is added to the worm tree, it connects to one of the existing nodes that are in the same "local" address space with a higher probability. That is, the growth model is no longer uniform attachment as studied in Section 4.2. For simplicity, in this work we only consider the $/l$ localized scanning [55]:

- *Local scanning*: $p_a (0 \leq p_a < 1)$ of the time, a "local" IP address with the same first l $(0 \leq l \leq 32)$ bits as the attacking host is chosen as the target.
- *Global scanning*: $1 - p_a$ of the time, a random address is chosen.

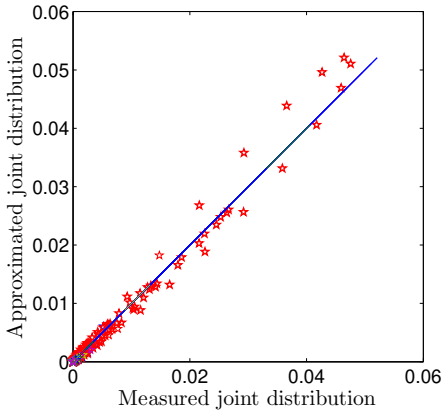
Note that random scanning can be regarded as a special case of localized scanning when $p_a = 0$. Moreover, if local scanning is selected, it can be regarded as random scanning in a local $/l$ subnet. It has been shown that since the vulnerable-hosts distribution is highly uneven, localized scanning can spread a worm much faster than random scanning [65].



(a) Number of children.



(b) Generation.



(c) Joint distribution ($p_a = 0.6$).

Figure 4.9: Simulating the spatial infection structure of the localized-scanning worm.

We extend our simulator to imitate the spread of localized-scanning worms. We extract the distribution of vulnerable hosts in $/l$ subnets from the dataset provided by DShield [58, 69]. Specifically, we use the dataset in [69] with port 80 (HTTP) that is exploited by the Code Red worm to generate the vulnerable-host distribution. Moreover, we use similar parameters as in Section 4.3.1 (*e.g.*, $n = 360,000$, $s = 358$ scans/min, $\sigma = 0$, and $hitlist = 1$) and set the subnet level to 8 (*i.e.*, $l = 8$). The results are obtained from 100 independent simulation runs and are shown in Fig. 4.9. For each run, patient zero is randomly chosen from vulnerable hosts.

Fig. 4.9(a) compares the simulation results of the distributions of the number of children (*i.e.,* $c_n(i)$) when $p_a = 0$, 0.3, and 0.6 with the geometric distribution with parameter 0.5. It is surprising that $c_n(i)$ of localized-scanning worms can still be well approximated by the geometric distribution. That is, the majority of nodes have few children, whereas a small portion of compromised hosts infect a large number of hosts. An intuitive explanation is given as follows. From Fig. 4.7(a), it can be seen that the total number of nodes has a minor effect on $c_n(i)$. Hence, if in a /8 subnet the majority of vulnerable hosts are infected through local scanning, it is expected that $c_n(i)$ of these hosts still closely follows the geometric distribution since the local scanning can be regarded as random scanning inside a /8 subnet. Therefore, both local infection and global infection lead $c_n(i)$ towards the geometric distribution with parameter 0.5. On the other hand, it can also be seen that when p_a increases, the tail of $c_n(i)$ moves slightly downward. This is because as p_a increases, more vulnerable hosts are infected through local scanning. Hence, it is more difficult for an infected host to find targets after vulnerable hosts in this host's local subnet have been exhausted. As a result, when p_a increases, fewer nodes can have a large number of children.

Fig. 4.9(b) demonstrates that the generation distribution of localized-scanning worms (*i.e.,* $g_n(j)$) can be well approximated by the Poisson distribution for the cases of $p_a = 0$, 0.3, and 0.6. The Poisson parameter, however, depends not only on n, but also on p_a. We further define $\lambda_n^{p_a} = \mathrm{E}_n^{p_a}[G]$ as the expectation of the generation for a localized-scanning worm with parameter p_a. Here, $\mathrm{E}_n^{p_a}[G]$ can be easily estimated from the simulation results of $g_n(j)$. Fig. 4.9(c) further shows the parity plot of the simulated joint distribution and the approximated joint distribution from Equation (4.31) when $p_a = 0.6$, indicating that the approximation is reasonable.

Moreover, Fig. 4.10 shows the effect of the subnet level (*i.e.,* l) on the distribution of the number of children (*i.e.,* $c_n(i)$). It can be seen that when l increases, the tail of $c_n(i)$ moves downward. The reason is similar to the argument used in Fig. 4.9(a), *i.e.,* as l increases,

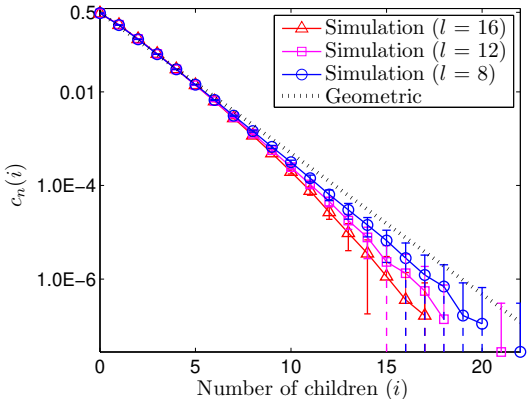


Figure 4.10: Effect of the subnet level ($p_a = 0.6$).

fewer nodes can infect a large number of children. However, the figure also demonstrates that the geometric distribution with parameter 0.5 is still a good approximation to $c_n(i)$, especially when the number of children is not large.

CHAPTER 5

EVALUATING P2P-BASED BOTNETS FORMED BY WORM INFECTION

A botnet is a zombie network controlled by a malicious attacker called the *botmaster* and is capable of sending denial-of-service attacks, producing spams, and stealing financial information. For example, the Storm botnet affected tens of millions of hosts and was used for spam emails and distributed DoS attacks in 2007 [35]. Therefore, botnets have become one of top threats to the Internet.

There are two major types of botnets: IRC-based botnets and P2P-based botnets. While IRC-based botnets require central servers for command delivery, P2P-based botnets make use of peer-to-peer systems and can form different command communication networks such as random graphs or power-law topologies [16]. As a result, P2P-based botnets are more resilient to defenses and have plagued the Internet [36]. In this chapter, we consider a P2P-based botnet formed by worm scanning/infection. That is, once a host infects another host, they become peers in the resulting P2P-based botnet. Note that P2P-based botnets formed by worm infection are a real threat. For example, Conficker C uses random scanning to locate peers and forms a P2P botnet through scan-based peer discovery [5, 6]. Thus, the way that Conficker C builds the botnet is in principle the same as worm infection.

Our observations on the worm spatial infection structure in Chapter 4 have important applications on Conficker C like P2P-based botnets. For example, we have found that the generation distribution closely follows the Poisson distribution and the average path length increases approximately logarithmically with the number of nodes. This average path length reflects the delay or the effort for a botmaster to deliver a command to all bots in a P2P-based botnet like Conficker C, and our results show that the botnet is scalable and can efficiently forward commands to a large number of bots. In this chapter, we further study other aspects of a Conficker C like P2P-based botnet for both defenders and attackers.

The goal of this chapter is to evaluate bot detection strategies and effects of user defenses in P2P-based botnets formed by worm infection. Specifically, we first apply the observations of the number of children in Chapter 4 on a Conficker C like P2P-based botnet to study efficient bot detection strategies. We then further extend the worm spatial infection structure to investigate the P2P-based botnet topologies under user patching and cleaning through simulations.

The remainder of this chapter is structured as follows. Section 5.1 evaluates bot detection methods and studies potential countermeasures by future botnets. Section 5.2 evaluates the effect of user defenses on the P2P-based botnet structure, and further studies effects of worm re-infection against user countermeasures.

5.1 Evaluating Bot Detection Strategies

In this section, we evaluate efficient bot detection methods by applying the observations of the number of children in Chapter 4 and then study a potential countermeasure by future botnets.

5.1.1 Bot Detection Strategies

In a P2P-based botnet formed by worm scanning/infection (*e.g.*, Fig. 5.1), when a defender captures an infected host x in a botnet, the defender can process the historic records inside the host or monitor the traffic going into or out of the host, and will potentially detect other infected hosts such as the father (host y) and the children (host z) of the infected host x. Then, our question is that if a defender can only access a small portion of nodes in a botnet, how many bots will be detected by the defender. Moreover, inspired by the random removal and targeted removal methods used in analyzing the robustness of a topology [37], here we study two bot detection strategies:

- Random detection: Access bots randomly.

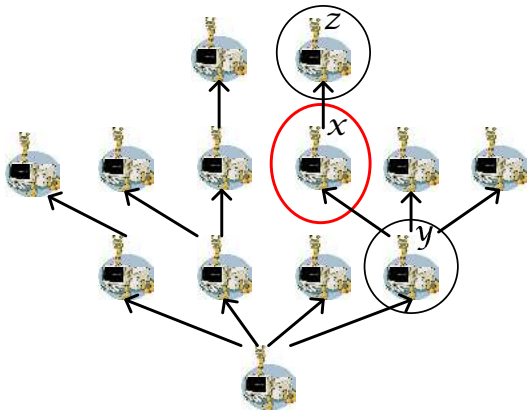


Figure 5.1: Bot detection in P2P-base botnet formed by worm infection.

- Targeted detection: Access bots that have the largest number of children.

Analytically, we suppose that a defender can access a small ratio of bots in a botnet. We assume that an accessed bot exposes itself, its father, and its children to the defender. To simplify the analysis, we also assume that the accessed bot ratio, A, is a power of 0.5 and all exposed nodes are different nodes. We then calculate the average percentages of exposed bots by random detection and targeted detection.

Since from Corollary 4.2.3 a randomly selected node has approximately one child, the average percentage of bots that can be exposed by random detection is then

$$D_R = 3A. \tag{5.1}$$

For targeted detection, since the nodes with the largest number of children are chosen and the number of children follows asymptotically a geometric distribution with parameter 0.5 as shown in Corollary 4.2.4,

$$A = \sum_{i \geq d} c_n(i) = \sum_{i=d}^{\infty} \left(\tfrac{1}{2}\right)^{i+1} = \left(\tfrac{1}{2}\right)^{d}, \tag{5.2}$$

where d is the smallest number of children of accessed nodes. That is, $d = -\log_2 A$. Therefore, the average percentage of exposed nodes by targeted detection is

$$D_T = \sum_{i=d}^{\infty} (2+i) \cdot c_n(i) = (d+3)\left(\tfrac{1}{2}\right)^{d} = A(3 - \log_2 A). \tag{5.3}$$

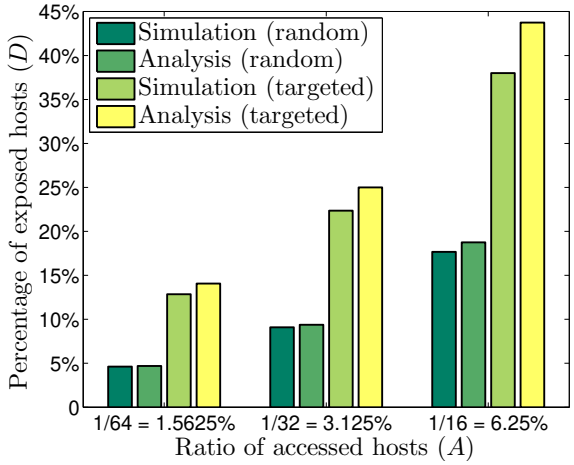


Figure 5.2: Random and targeted detection.

Compared with random detection, targeted detection can expose $(-A \log_2 A) \times n$ more nodes. For example, if $A = \frac{1}{64}$, on average random detection can detect 4.69% of nodes, whereas targeted detection can expose 14.06% of bots.

We simulate a P2P-based botnet formed through worm infection by using our simulator in Section 4.3.1. We then extend the simulator to study the effectiveness of random and targeted detection strategies. Fig. 5.2 shows the simulation results over 100 independent runs for both strategies, as well as the analytical results from Equations (5.1) and (5.3), when $A = \frac{1}{64}$, $\frac{1}{32}$, and $\frac{1}{16}$. It can be seen that the analytical results slightly overestimate the exposed host percentage. This is because in our analysis we ignore the case that two exposed nodes can be duplicate. Fig. 5.2 also demonstrates that targeted detection performs much better than random detection. For example, in our simulation, when $A = 3.125\%$, 9.10% of the bots are exposed under random detection, whereas 22.36% of the bots are detected under targeted detection. Therefore, when a small portion of bots are examined, the botnets formed by worm infection are robust to random detection, but are relatively vulnerable to targeted detection.

5.1.2 A Countermeasure by Future Botnets

To counteract the targeted detection method, an intuitive way for botnets is to limit the maximum number of children for each node. That is, set a small number m. Once an infected host has compromised m other hosts, this host stops scanning. In this way, there is no node with a large number of children. Moreover, the infected hosts can self-stop scanning, potentially reducing the worm traffic [59].

To analyze the robustness of such botnets against targeted detection, we extend Corollary 4.2.4 to obtain an approximated distribution of the number of children in a botnet with the countermeasure:

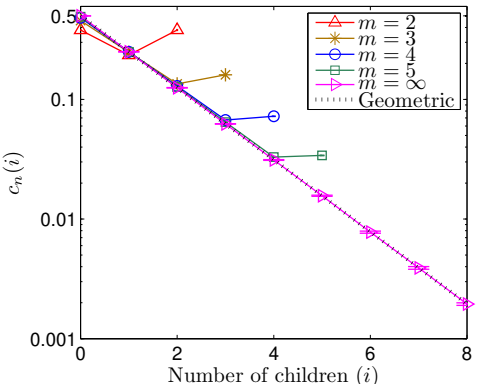
$$c_n(i) = \begin{cases} \left(\frac{1}{2}\right)^{i+1}, & i = 0, 1, 2, \cdots, m-1 \\ \left(\frac{1}{2}\right)^{m}, & i = m. \end{cases} \tag{5.4}$$

The distribution is based on the observation that those nodes having more than m children in a botnet without the countermeasure can now have only m children. Hence, the expected percentage of exposed nodes under targeted detection can be calculated:

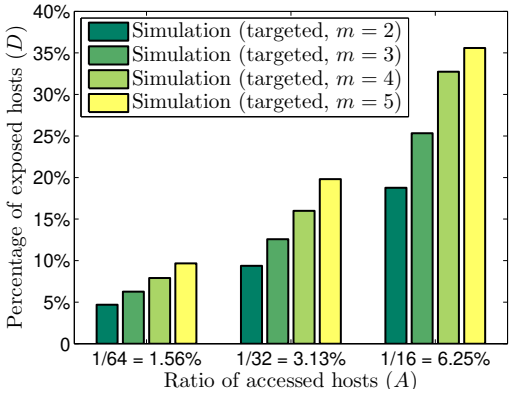
$$D'_T = \begin{cases} (m+2) \cdot A, & A \leq \left(\frac{1}{2}\right)^{m} \\ A(3 - \log_2 A) - \left(\frac{1}{2}\right)^{m}, & A > \left(\frac{1}{2}\right)^{m}. \end{cases} \tag{5.5}$$

Compared with D_T in Equation (5.3), D'_T is smaller. This means that under the countermeasure the number of exposed nodes can be reduced significantly. For example, when $m = 3$ and $A = \frac{1}{64}$, $D_T = \frac{9}{64}$, and $D'_T = \frac{5}{64}$.

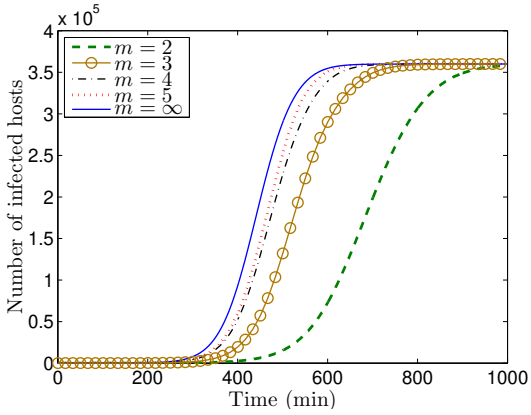
We then extend our simulation in Section 5.1.1 to simulate the worm tree generated using the above countermeasure and evaluate its performance against targeted detection. Fig. 5.3(a) shows the distribution of the number of children when $m = 2$, 3, 4, and 5. It can be seen that except for $m = 2$, $c_n(i)$ is well approximated by Equation (5.4). For $m = 2$, since an infected host stops scanning when it has hit two vulnerable hosts, leaves in the worm tree have more chances to recruit a child. Fig. 5.3(b) demonstrates the expected



(a) Distribution of the number of children.



(b) Targeted detection.



(c) Worm propagation speed.

Figure 5.3: A worm countermeasure via limiting the maximum number of children.

percentage of exposed nodes (*i.e.,* D'_T), when $A = \frac{1}{64}$, $\frac{1}{32}$, and $\frac{1}{16}$, and $m = 2$, 3, 4, and 5. It can be seen that D'_T follows approximately the analytical results in Equation (5.5). Moreover, the expected percentage of exposed nodes under the countermeasure is reduced significantly. For example, when $A = \frac{1}{32}$, the percentage is reduced from 22.36% without the countermeasure to 19.80%, 15.99%, 12.58%, and 9.38% when $m = 5$, 4, 3, and 2, respectively.

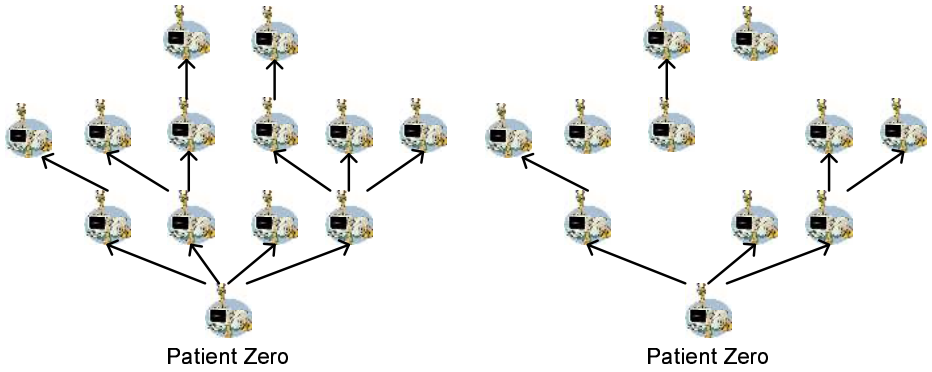
On the other hand, since not every infected host keeps scanning the targets, the countermeasure can potentially slow down the speed of worm infection. Thus, we also simulate

the propagation speed of worms that limit the maximum number of children and plot the results in Fig. 5.3(c) for $m = 2$, 3, 4, and 5, as well as the original worm without the countermeasure. It can be seen that except for $m = 2$, the worm does not slow down much. But even when $m = 2$, the worm can infect most vulnerable hosts within 17 hours. Moreover, Fig.s 5.3(b) and (c) demonstrate the tradeoff between the efficiency of worm infection and the robustness of the formed botnet topology. That is, a worm with the countermeasure spreads slower, but the resulting botnet is more robust against targeted detection.

5.2 Evaluating Effects of User Defenses

In Chapter 4, we studied the worm tree, *i.e.,* the network structure of P2P-based botnets formed by Internet worm infection. Specifically, we considered that once an infected host compromises another host, they form the "father" and "child" relationship, as shown in Fig. 5.4(a). In Chapter 4, we found through theoretical analysis that the number of children has asymptotically a geometric distribution with parameter 0.5 and the generation follows closely a Poisson distribution. In our prior work, however, we focused on the process of worm infection and the formation of P2P-based botnets, and did not consider the potential countermeasures from users.

Users can respond to worm outbreaks by patching or cleaning discovered infected hosts. For example, to counterattack the Conficker worm, Microsoft released a removal guide to clean and patch the Conficker compromised machines after the outbreak of the worm [70]. When an infected host is patched, it becomes invulnerable; and when it is cleaned, it is no longer infectious, but is still vulnerable to worm infection. It is obvious that a patched or cleaned infected host can break its relationships with its father and children in the worm tree. Specifically, when an infected host is patched or cleaned, the corresponding node along with its associated links are removed from the worm tree. As a result, the infection topology is no longer a tree, but a forest, as shown in Figure 5.4(b). When user countermea-



(a) A worm tree. (b) A worm forest.

Figure 5.4: User defenses in P2P-based botnets formed by worm infection.

sures are considered, therefore, two interesting questions arise: Are patching and cleaning methods effective against P2P-based botnets, and how do user countermeasures affect the botnets formed by worm infection?

To answer these questions, in this section we extend our previous work to investigate the structure of P2P-based botnets under user countermeasures. Specifically, we consider that a vulnerable host has three states: susceptible, infected, and removed. A susceptible host can become infected through worm infection. An infected host can either become removed by user patching or become susceptible again by user cleaning. Note that user cleaning is a real method against some worms. For example, a Code-Red infected host becomes susceptible once rebooted [21]. The effectiveness of patching and cleaning against worms has been studied in terms of the total number of infected hosts over time [9, 71]. In this work we focus on the effect of user countermeasures on the P2P-based botnet structure. To characterize the key features of botnet topologies under both worm infection and user countermeasures, we study two important metrics in particular:

- *Number of peers:* For a randomly selected node in the botnet topology, how many peers (*i.e.*, an infected host's father and children) does it have? This metric represents the node degree of individual hosts.

- *Botnet size:* For a randomly selected tree in the forest, how many nodes does it have? This metric represents the size of disconnected botnets after node removal.

These two metrics shed light on the robustness and the effectiveness of formed P2P-based botnets. For example, if a very small number of hosts have a large number of peers and the majority of hosts have none or few peers, such botnets are robust to random defenses, but are vulnerable to targeted defenses (*i.e.*, quarantining the hosts with the largest node degree) [72, 16]. On the other hand, if each host has a similar node degree, then such botnets are robust to both defense schemes [72, 16]. Moreover, the bigger a botnet is, the more effective and dangerous it is [16]. For example, if the forest consists of a collection of small isolated botnets, then its effectiveness is significantly lower than the single connected botnet with the same total number of nodes.

5.2.1 Worm Forest and Simulation Settings

In this section, we first provide the background of the worm forest and then introduce our simulation settings.

In Chapter 4, we studied the topology of P2P-based botnets formed by Internet worm infection without considering user defenses. Specifically, we analyzed the tree structure of P2P-based botnets formed by a wide class of worms starting from patient zero, for which a new victim is compromised by each existing infected host with equal probability. Such worms include well known random-scanning worms, routable-scanning worms, importance-scanning worms, OPT-STATIC worms, and SUBOPT-STATIC worms. Here, we assume that all vulnerable hosts are globally reachable and do not consider the effect of network address translation [73]. In this section, we construct the worm forest by randomly patching or cleaning hosts in the worm tree studied in [72]. Since most Internet worms spread so fast that existing defense systems cannot respond until they have infected most vulnerable hosts [22, 23], we assume that user patching or cleaning starts when the

entire vulnerable population (denoted as n_0) gets infected. We use r_p to denote the patching rate at which a machine is patched and becomes invulnerable, and r_c to denote the cleaning rate at which the infection is cleaned on a machine without patching. Once patched or cleaned, the node and its associated links are then removed from the botnet topology. Suppose that n_d hosts get patched or cleaned, and the number of remaining infected hosts and trees are denoted as n_r and t_r, respectively. We use $B_{n_0}^{n_d}(i)$ ($i = 0, 1, 2, \cdots, n_r - 1$) to denote the number of nodes that have i peers and $T_{n_0}^{n_d}(j)$ ($j = 1, 2, 3, \cdots, n_r$) to denote the number of trees that have j nodes. Note that $\sum_{i=0}^{n_r-1} B_{n_0}^{n_d}(i) = n_r$, and $\sum_{j=1}^{n_r} T_{n_0}^{n_d}(j) = t_r$. Moreover, $B_{n_0}^{n_d}(i)$ and $T_{n_0}^{n_d}(j)$ are random variables. Thus, we define $b_{n_0}^{n_d}(i) = \frac{\mathrm{E}[B_{n_0}^{n_d}(i)]}{n_r}$ to represent the distribution of the number of peers and $t_{n_0}^{n_d}(j) = \frac{\mathrm{E}[T_{n_0}^{n_d}(j)]}{t_r}$ to represent the distribution of the botnet size. Note that the worm tree is a special case of the worm forest when $n_d = 0$ (*i.e.,* without user defenses). For such a tree, we have

$$\lim_{n_0 \to \infty} b_{n_0}^{0}(i) = (\tfrac{1}{2})^i, \;\; i = 1, 2, 3, \cdots \tag{5.6}$$

by extending the result in Chapter 4. While our previous work only considers the number of children, this section studies the number of peers including both the father and children. Therefore, in P2P-based botnets formed by worm infection without user countermeasures, the distribution of the number of peers has asymptotically a geometric distribution with parameter 0.5, and decreases exponentially with a decay constant of $\ln 2$. Moreover, Since there is only one botnet, we then have the distribution of the botnet size $t_{n_0}^{0}(n_0) = 1$.

To investigate the P2P-based botnet topology under user patching and cleaning, in this work we study $b_{n_0}^{n_d}(i)$ and $t_{n_0}^{n_d}(j)$ through simulations. As far as we know, there is no publicly available data to show the real botnet topologies. Moreover, the complex dynamics of patching and cleaning make the botnet structure difficult to be characterized analytically. Therefore, we apply Monte Carlo simulation. Monte Carlo simulation is widely applied in probability modeling and is the only viable method for the modeling of many complex stochastic systems [74]. Specifically, we simulate a P2P-based botnet formed through

worm infection by using our simulator in Section 4.3.1. We then extend the simulator to mimic the dynamics of user countermeasures and capture the resulting botnet structure. Specifically, after all vulnerable machines get compromised, we randomly patch or clean hosts with $r_p = 2 \times 10^{-5}$/sec or $r_c = 2 \times 10^{-5}$/sec. We also record the information of the number of peers and the botnet size to track the botnet structure. Moreover, we set the time unit to 20 seconds and start our simulation at time tick 0 with patient zero. The simulation results are obtained from 100 independent runs with different seeds.

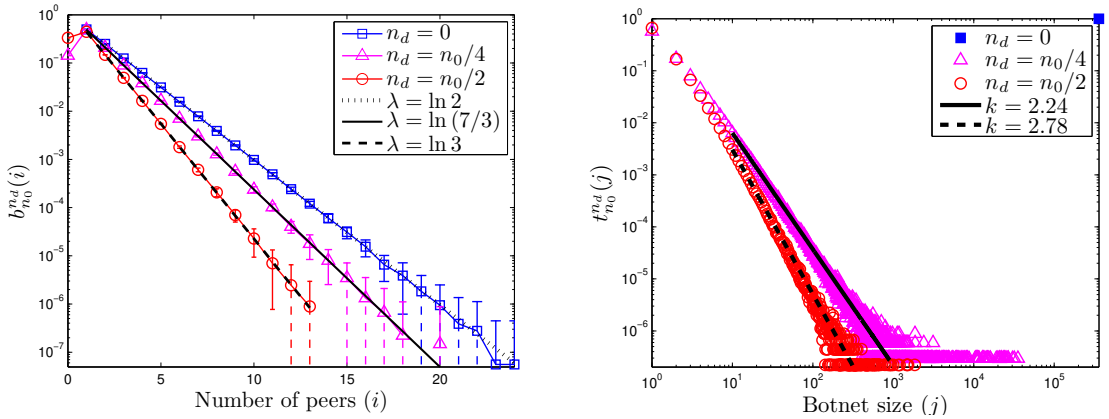
5.2.2 P2P-based Botnet Structure under User Countermeasures

In this section, we present the P2P-based botnet structure under user countermeasures. Specifically, we examine the distributions of the number of peers and the botnet size under three different defense schemes: host patching only, host cleaning only, and host patching/cleaning schemes. The results are shown in Fig.s 5.5-5.7. Scaling parameters λ and k are estimated through regression analysis on empirical data by using the Matlab curve fitting toolbox [75], and the coefficient of determination R^2 is very close to 1 for all estimates.

Host Patching Only Scheme

Under this defense scheme, we begin to randomly patch infected hosts with $r_p = 2 \times 10^{-5}$/sec after all vulnerable machines get infected. Once patched, an infected host becomes invulnerable, and the node and its associated links are removed from the worm forest. We then examine the P2P-based botnet structure when n_d hosts get patched. The results are shown in Fig. 5.5.

Fig. 5.5(a) shows the distribution of the number of peers, comparing the simulation results of $b_{n_0}^{n_d}(i)$ for $n_d = 0$, $n_0/4$, and $n_0/2$ with the exponential scaling obtained through regression. Note that the y-axis uses the log-scale and the error bar represents the standard



(a) Number of peers. (b) Botnet size.

Figure 5.5: Host patching only scheme.

deviation over 100 runs. The dotted line represents the standard deviation that goes into the negative territory. It can be seen that the distribution of the number of peers has an exponential tail. Specifically, without user defenses (*i.e.,* when $n_d = 0$), $b_{n_0}^0(i)$ can be well approximated by the geometric distribution with parameter 0.5 shown in Equation (5.6), and therefore decreases exponentially with the decay constant $\lambda = \ln 2$. However, as infected hosts get patched, nodes that do not have any peer emerge in the forest. Moreover, when n_d increases, $b_{n_0}^{n_d}(i)$ still has an exponential tail, but decays faster. This is because when more infected hosts get removed, there are fewer hosts with a large node degree and more hosts becoming isolated nodes without any peer. On one hand, the exponential scaling of $b_{n_0}^{n_d}(i)$ implies that after random patching, a small portion of bots still have a large number of peers and the majority of bots have none or few peers. For example, when $n_d = n_0/2$, on average over 99.7% of bots have no more than five peers. On the other hand, an increasing decay constant indicates that the node degree of a bot decreases due to patching. For example, the average node degree decreases from 2 when $n_d = 0$ to 1 when $n_d = n_0/2$. Moreover, through extensive regression analysis, we find that after user patching, in the resulting P2P-based botnet topology, the decay constant $\lambda \approx \ln((n_0 + n_r)/n_r)$, where

$n_r = n_0 - n_d$. For example, when half of infected hosts are patched, $b_{n_0}^{n_d}(i)$ decreases exponentially with a decay constant approximately of $\ln 3$.

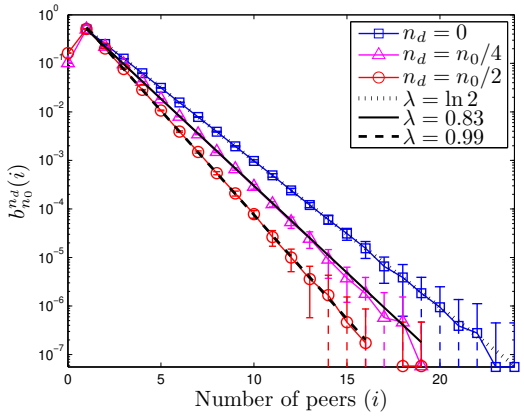
Fig. 5.5(b) demonstrates the distribution of the botnet size, comparing the simulation results of $t_{n_0}^{n_d}(j)$ for $n_d = 0$, $n_0/4$, and $n_0/2$ with the power-law tails obtained through regression. Note that the x- and y-axes use the log-scale. It can be seen that when $n_d = 0$, $t_{n_0}^{0}(n_0) = 1$. That is, without patching, worm infection forms a single botnet with n_0 nodes. However, with infected hosts being patched, the distribution of the botnet size has a power-law tail. Moreover, when n_d increases, the scaling exponent k becomes larger. This is because as we patch more infected hosts, the number of trees in the forest increases, whereas the maximum size of trees decreases. For example, when $n_d = n_0/2$, on average there are $90,011$ trees[1] in the forest with an average size of 2 nodes. The average maximum tree size is 622 nodes, comprising less than 0.04% of infected hosts in the forest. Therefore, the size of the largest botnet is relatively small, indicating that patching infected hosts severely disrupts the single botnet formed by worm infection.

After performing sensitivity analysis on the parameter r_p when n_d is fixed, we find that the patching rate does not affect the botnet structure.

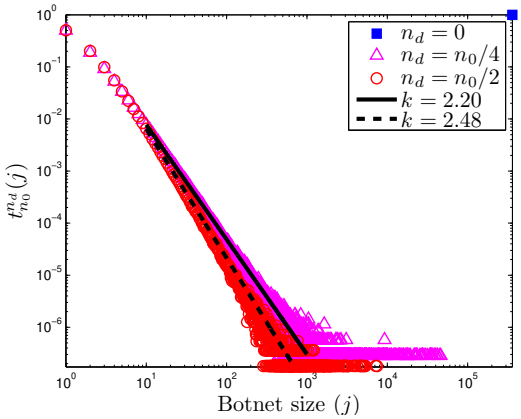
Host Cleaning Only Scheme

Under this defense scheme, we begin to randomly clean infected hosts with $r_c = 2 \times 10^{-5}$/sec after all vulnerable machines get compromised. Once cleaned, an infected host becomes susceptible, and the host and its associated links are removed from the forest. Note that different from patching, cleaned infected hosts can be compromised again and rejoin the forest. We then examine the P2P-based botnet structure when n_d hosts get cleaned. The results are shown in Fig. 5.6.

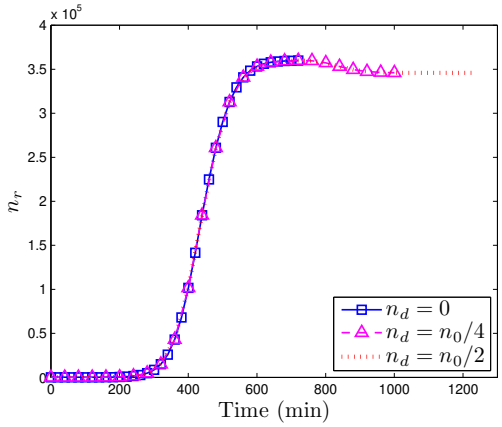
[1]We consider that isolated nodes without any peer are a special tree of size one.



(a) Number of peers.



(b) Botnet size.

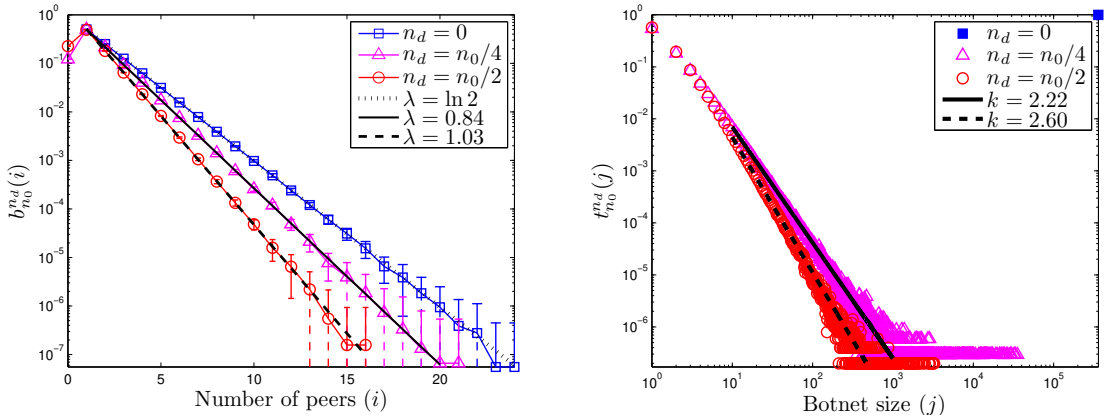


(c) Number of infected hosts.

Figure 5.6: Host cleaning only scheme.

Fig.s 5.6(a) and (b) show the results of the distributions of the number of peers and the botnet size. It can be seen that $b_{n_0}^{n_d}(i)$ still has an exponential decay and $t_{n_0}^{n_d}(j)$ has a power-law tail. As a result, after user cleaning, a small portion of bots still have a large number of peers, and the majority of bots have none or few peers. For example, when $n_d = n_0/2$, the average node degree of bots is 1.36, and on average about 99.3% of them have a node degree of no more than five. Moreover, cleaning infected hosts severely disrupts the single botnet formed by worm infection. For example, when $n_d = n_0/2$, on average there are $110,740$ disconnected botnets in the forest with an average size of 3 nodes. The average

maximum size of the disconnected botnets is $2,954$ nodes, comprising about 0.85% of the remaining infected hosts in the forest. However, compared with the patching only scheme, the exponential and power-law scaling parameters under the host cleaning only scheme are smaller. This is due to the different nature of patching and cleaning. Under the host cleaning only scheme, when n_d hosts are cleaned, some of them get compromised again and rejoin the worm forest. As a result, the number of remaining infected hosts in the forest $n_r > (n_0 - n_d)$. Comparatively, under the host patching only scheme, when n_d nodes are patched, $n_r = n_0 - n_d$. Therefore, as expected, the host cleaning only scheme less disrupts the botnet structure than the host patching only scheme. Moreover, as shown in Fig. 5.6(c), we find that under the host cleaning only scheme, on average n_r stabilizes at around $345,950$. This happens when the number of nodes being cleaned, $n_r \cdot r_c$, is about the same with the number of susceptible hosts getting infected again, $(n_0 - n_r) \cdot p_i$, where $p_i = n_r \cdot s \cdot \frac{1}{2^{32}}$ is the probability of a susceptible host being compromised. Setting $n_r \cdot r_c = (n_0 - n_r) \cdot n_r \cdot s \cdot \frac{1}{2^{32}}$, we then obtain that the number of nodes in the worm forest will stabilize at $n_r = n_0 - \frac{r_c}{s} \cdot 2^{32}$. For example, with $r_c = 2 \times 10^{-5}$/sec and $s = 358$ scans/min, $n_r = 345,603$, which is very close to our simulation result. In the figure, we also find that n_r is about the same for the cases of $n_d = n_0/4$ and $n_0/2$. However, $b_{n_0}^{n_d}(i)$ and $t_{n_0}^{n_d}(j)$ of the case $n_d = n_0/2$ has larger scaling parameters. This is due to the fact that hosts with a large number of peers might get cleaned, whereas susceptible hosts rejoin the forest as leaves with a node degree of one. As a result, although the number of infected hosts stabilizes at the same level, the host cleaning process decreases the node degree of infected hosts over time and further disrupts the worm forest. Furthermore, we find that the cleaning rate r_c has little effect on the botnet structure when n_d is fixed. On one hand, a smaller cleaning rate corresponds to a larger stabilized botnets population n_r. One the other hand, it takes more time to clean n_d nodes with a smaller cleaning rate.



(a) Number of peers. (b) Botnet size.

Figure 5.7: Host patching/cleaning scheme.

Host Patching/Cleaning Scheme

Under this defense scheme, we consider both user patching and cleaning, which is the case in real world scenarios. Specifically, we begin to randomly clean infected hosts with $r_c = 2 \times 10^{-5}$/sec after all vulnerable hosts get compromised. Meanwhile, susceptible and infected hosts are randomly patched with $r_p = 2 \times 10^{-5}$/sec. We then examine the P2P-based botnet structure when n_d hosts get patched or cleaned. The results are shown in Fig. 5.7. It is intuitive that the distributions of the number of peers and the botnet size exhibit the combined effects of the host patching only and the host cleaning only schemes. Specifically, the exponential decay constant λ and the power-law scaling exponent k are smaller than those under the host patching only scheme but greater than those under the host cleaning only scheme. For example, when $n_d = n_0/2$, the average node degree of bots is 1.21, and on average about 99.5% of them have no more than five peers. Moreover, on average there are $100,535$ disconnected botnets in the forest with an average size of 2.5 nodes. The average maximum size of the disconnected botnets is $1,636$ nodes, comprising about 0.64% of the remaining infected hosts in the forest.

The simulation results of all three defense schemes show that when users patch or clean

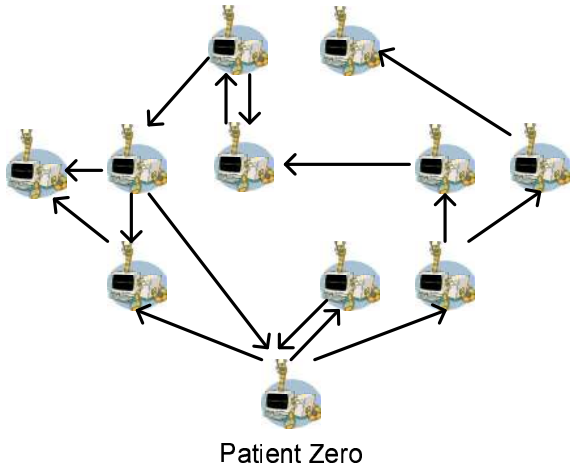


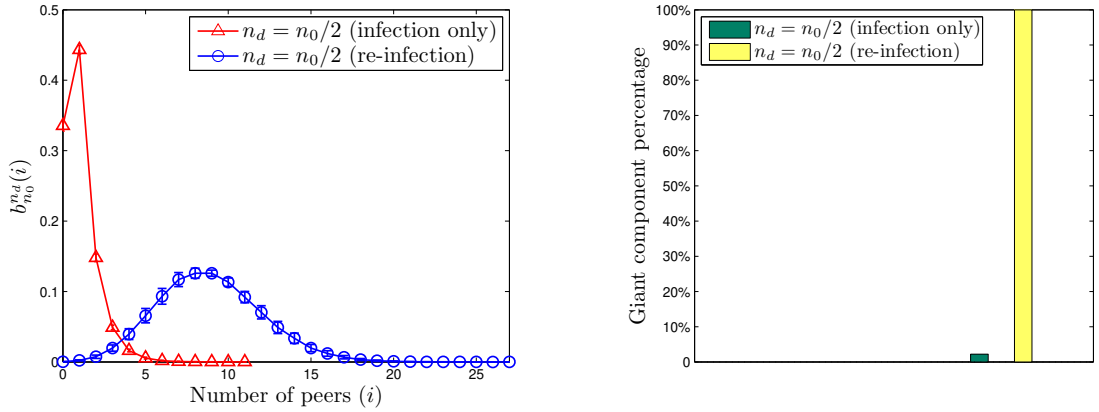
Figure 5.8: Worm re-infection topology.

part of infected hosts, P2P-based botnets formed by worm infection suffer two weaknesses. First, the botnets are highly centralized to a small percentage of the "hub" bots that have a large node degree, and thus vulnerable to targeted defenses [72, 16]. Second, the single botnet formed by worm infection is severly disrupted into a collection of small isolated low-effective botnets.

5.2.3 P2P-based Botnets Formed by Worm Re-infection

In this section, we study a potential countermeasure by future botnets to combat against user patching or cleaning.

A simple potential countermeasure for botmasters to construct more robust and effective P2P-based botnets is through worm re-infection. That is, if an infected host is hit by a worm scan, this host will be further re-infected and become a peer of the infector. As a result, the remaining bots may have a balanced node degree and stay well connected even when some infected hosts get patched or cleaned (see Fig. 5.8). Note that different from the botnet formed by re-infection discussed in [4], in our P2P-based botnet, there is no exchange of peers between bots. Infected hosts are only peers to their own infectors and infectees.



(a) Number of peers (host patching only). (b) Giant component percentage (host patching only).

Figure 5.9: P2P-based botnets formed by worm re-infection.

To show the effectiveness of worm re-infection on P2P-based botnets against user patching or cleaning, we consider the host patching only scheme, which is the worst case scenario. As shown in Section 5.2.2, under the host patching only scheme, $b_{n_0}^{n_d}(i)$ and $t_{n_0}^{n_d}(j)$ have the largest scaling parameters among the three schemes, and therefore the resulting P2P-based botnets are least robust and effective. In Fig. 5.9, we compare the network structure of botnets formed by worm infection only and by worm re-infection when n_d hosts get patched. Here, the vulnerable population n_0 is set to $10,000$. All other parameters ramain the same as the ones used in Section 5.2.2. Moreover, for worm re-infection, once a vulnerable host gets compromised, it is open for re-infection from the next time tick. We begin to randomly patch infected hosts with $r_p = 2 \times 10^{-5}$/sec when all vulnerable machines get compromised. Once patched, the infected host becomes invulnerable, and the host and its associated links are then removed from the botnet topology. Fig. 5.9(a) shows the distribution of the number of peers. It can be seen that in the P2P-based botnet formed by worm re-infection, when half of infected hosts get patched, $b_{n_0}^{n_d}(i)$ has a bell shape and therefore the node degree of a bot is more evenly distributed. For example, on average 92.56% of bots have a node degree between 5 and 15, and the average node degree of bots is 9. On

one hand, such a botnet is resilient to both random and targeted defenses [72, 16]. On the other hand, the P2P-based botnet formed by worm re-infection may have an average node degree similar to other P2P networks [76]. As a result, it may appear to have normal P2P traffic and can potentially avoid detection [16]. Moreover, by further connecting to other bots, the P2P-based botnet formed by worm re-infection stays well connected. In [16], Dagon *et al.* used the giant component or the size of the largest connected botnet to measure the effectiveness. In Fig. 5.9(b), we show the percentage of the giant component to the available bots. It can be seen that for the botnets formed by worm re-infection, almost all of the remaining bots are connected, whereas the giant component of the botnets formed by worm infection comprises only 2.2% of the remaining infected hosts. Therefore, P2P-based botnets formed by worm re-infection are much more robust and effective than those formed by worm infection only.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

6.1 Characterizing Internet Worm Temporal Infection Structure

In Chapter 3, we have attempted to estimate the temporal characteristics of Internet worms through both analysis and simulation under the framework of Internet worm tomography. Specifically, we have proposed method of moments, maximum likelihood, and linear regression estimators to infer the host infection time and reconstruct the worm infection sequence. We have shown analytically and empirically that the mean squared error of our proposed estimators can be almost half of that of the naive estimator in estimating the host infection time. Moreover, we have formulated the problem of estimating the worm infection sequence as a detection problem and have calculated the probability of error detection for different estimators. We have demonstrated empirically that our estimation techniques perform much better than the algorithm used in [13] in estimating the worm infection sequence and in identifying the hitlist for both random-scanning and localized-scanning worms.

6.2 Characterizing Internet Worm Spatial Infection Structure

In Chapter 4, we have attempted to capture the key characteristics of the tree topology formed by worm infection. We have shown analytically and empirically that for the infection tree formed by a wide class of worms, the number of children asymptotically has a geometric distribution with parameter 0.5; and the generation closely follows a Poisson distribution with parameter $\mathrm{E}_n[G]$ (*i.e.,* $H_n - 1$). As a result, on average half of infected hosts never compromise any target, over 98% of nodes have no more than five children, and a small portion of hosts have a large number of children. Moreover, the average path

length of the worm tree increases approximately logarithmically with the number of nodes. We have also demonstrated empirically that similar observations can be found in localized-scanning worms.

6.3 Evaluating P2P-based Botnets Formed by Worm Infection

In Chapter 5, we have attempted to evaluate bot detection strategies and effects of user defenses in P2P-based botnets formed by worm infection. Specifically, we have applied the observations of the number of children to bot detection and found analytically and empirically that targeted detection is an efficient way to expose bots in a Conficker C like botnet. However, we have also pointed out that a simple countermeasure by future botnets can weaken the performance of targeted detection, without greatly slowing down the speed of worm infection. Moreover, we have characterized the network structure of P2P-based botnets formed by worm infection under user countermeasures. We have shown that when part of infected hosts are randomly patched or cleaned after all vulnerable hosts get compromised, the distribution of the number of peers of a bot has an exponential scaling and the distribution of the size of disconnected botnets has a power-law tail. As a result, P2P-based botnets formed by worm infection are vulnerable to targeted defenses and ineffective due to patching or cleaning. We have then applied the observations to design future botnets and found that botmasters can significantly enhance the robustness and the effectiveness of P2P-based botnets through worm re-infection.

6.4 Future Work

6.4.1 Real-World Data Verification

One limitation of this work is that our analytical results are verified through simulations rather than real-world data. As far as we know, there is no direct dataset of the worm spatial-temporal infection structures publicly available. However, we may extract some indirect knowledge from worm traces to serve as an approximation of the ground truth. For example, for the worm temporal infection structure, we may use first hits observed at a large Darknet (*e.g.*, a /8 network telescope) to serve as a comparison basis, and then apply estimators to observations of a much smaller Darknet (*e.g.*, a /24 network telescope) for performance evaluation. Moreover, some works have inferred the information of "who infected whom" [24, 14], which may be used as an approximation of the real worm tree to verify our analytical results of the worm spatial infection structure.

6.4.2 Fractal Analysis

A fractal is a rough or fragmented geometric shape that can be split into parts, each of which is a reduced-size copy of the whole [77]. The defining characteristic of a fractal is self-similarity. Fractals have broad applications in ecology, biology and the Earth sciences [78]. One of the most familiar examples of self-similarity is a tree. The pattern of branching is very similar and repeated throughout the tree. If we capture a small group of infected hosts that are connected as a branch in the worm tree, one interesting question is that, by analyzing the fractal patterns of the captured branch, can we predict characteristics of worm propagation or P2P-based botnets formed by worm infection as a whole? This enables us to understand and defend against worms or botnets with significantly reduced efforts and costs.

BIBLIOGRAPHY

[1] D. Dagon, C. C. Zou, and W. Lee, "Modeling Botnet Propagation Using Time Zones," in *Proc. NDSS*, Feb. 2006.

[2] Z. Li, A. Goyal, Y. Chen, and V. Paxson, "Automating Analysis of Large-Scale Botnet Probing Events," in *Proc. ACM Symposium on Information, Computer and Communication Security (ASIACCS'09)*, Mar. 2009.

[3] R. Vogt, J. Aycock, and M. Jacobson, Jr., "Army of Botnets," in *Proc. NDSS*, Feb. 2007.

[4] P. Wang, S. Sparks, and C. C. Zou, "An Advanced Hybrid Peer-to-Peer Botnet," *IEEE Transactions on Dependable and Secure Computing*, vol. 7, no. 2, pp. 113–127, Apr.-Jun. 2010.

[5] P. Porras, H. Saidi, and V. Yegneswaran, "Conficker C P2P Protocol and Implementation," *SRI International Technical Report*, Sept. 2009.

[6] CAIDA. Conficker/Conflicker/Downadup as seen from the UCSD Network Telescope. [Online]. Available: http://www.caida.org/research/security/ms08-067/conficker.xml.

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

[8] C. C. Zou, D. Towsley, and W. Gong, "On the Performance of Internet Worm Scanning Strategies," *Elsevier Journal of Performance Evaluation*, vol. 63, no. 7, pp. 700–723, Jul. 2006.

[9] Z. Chen, L. Gao, and K. Kwiat, "Modeling the Spread of Active Worms," in *Proc. IEEE INFOCOM*, Apr. 2003.

[10] K. Rohloff and T. Basar, "Stochastic Behavior of Random Constant Scanning Worms," in *Proc. 14th ICCCN*, Oct. 2005.

[11] M. Vojnovic and A. J. Ganesh, "On the Race of Worms, Alerts and Patches," *IEEE/ACM Transactions on Networking*, vol. 16, no. 5, pp. 1066–1079, Oct. 2008.

[12] C. C. Zou, W. Gong, D. Towsley, and L. Gao, "The Monitoring and Early Detection of Internet Worms," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 967–974, Oct. 2005.

[13] M. A. Rajab, F. Monrose, and A. Terzis, "Worm Evolution Tracking via Timing Analysis," in *Proc. Workshop on Rapid Malcode (WORM)*, Nov. 2005.

[14] Y. Xie, V. Sekar, D. A. Maltz, M. K. Reiter, and H. Zhang, "Worm Origin Identification Using Random Walks," in *Proc. IEEE Symposium on Security and Privacy*, May 2005.

[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms (Second Edition)," *The MIT Press and McGraw-Hill*, 2002.

[16] D. Dagon, G. Gu, C. Lee, and W. Lee, "A Taxonomy of Botnet Structures," in *Proc. 23 Annual Computer Security Applications Conference (ACSAC'07)*, Dec. 2007.

[17] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, "Network Telescopes: Technical Report," *Technical Report*, Jul. 2004.

[18] J. Wu, S. Vangala, L. Gao, and K. Kwiat, "An Effective Architecture and Algorithm for Detecting Worms with Various Scan Techniques," *NDSS*, 2004.

[19] T. Bu, A. Chen, S. Wiel, and T. Woo, "Design and Evaluation of a Fast and Robust Worm Detection Algorithm," in *Proc. IEEE INFOCOM*, Apr. 2006.

[20] S. Soltani, S. A. Khayam, and H. Radha, "Detecting Malware Outbreaks Using a Statistical Model of Blackhole Traffic," in *Proc. IEEE International Conference on Communications*, May 2008.

[21] D. Moore, C. Shannon, and J. Brown, "Code-Red: a Case Study on the Spread and Victims of an Internet Worm," in *Proc. ACM SICGOMM Internet Measurement Workshop*, Nov. 2002.

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

[23] C. Shannon and D. Moore, "The Spread of the Witty Worm," in *IEEE Security and Privacy*, vol. 2, no. 4, Jul-Aug 2004, pp. 46–50.

[24] A. Kumar, V. Paxson, and N. Weaver, "Exploiting Underlying Structure for Detailed Reconstruction of an Internet-scale Event," in *Proc. Internet Measurement Conference*, 2005.

[25] I. Hamadeh and G. Kesidis, "Toward a Framework for Forensic Analysis of Scanning Worms," in *Proc. International Conference on Emerging Trends in Information and Communication Security*, Jun. 2006.

[26] Q. Wang, Z. Chen, K. Makki, N. Pissinou, and C. Chen, "Inferring Internet Worm Temporal Characteristics," in *Proc. IEEE GLOBECOM*, Dec. 2008.

[27] S. Sellke, N. B. Shroff, and S. Bagchi, "Modeling and Automated Containment of Worms," *IEEE Transactions on Dependable and Secure Computing*, vol. 5, no. 2, pp. 71–86, Apr.-Jun. 2008.

[28] A.-L. Barabási and R. Albert, "Emergence of Scaling in Random Networks," *Science*, vol. 286, pp. 509–512, Oct. 1999.

[29] A.-L. Barabási, R. Albert, and H. Jeong, "Mean-field Theory for Scale-free Random Networks," *Physica A 272*, 1999.

[30] B. Bollobás, O. Riordan, J. Spencer, and G. Tusnady, "The Degree Sequence of a Scale-free Random Graph Process," *Random Structures Algorithms*, vol. 18, no. 3, pp. 279–290, Apr. 2001.

[31] S. N. Dorogovtsev, J. F. F. Mendes, and A. N. Samukhin, "Structure of Growing Networks with Preferential Linking," *Phys. Rev. Lett.*, vol. 85, pp. 4633–4636, Nov. 2000.

[32] L. Devroye, "Applications of the Theory of Records in the Study of Random Trees," *Acta Inf.*, vol. 26, no. 1-2, pp. 123–130, 1988.

[33] E. Cooke, F. Jahanian, and D. McPherson, "The Zombie Roundup:Understanding, Detecting, and Disrupting Botnets," in *Proc. USENIX SRUTI Workshop: Steps to Reducing Unwanted Traffic on the Internet*, Jul. 2005.

[34] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "A Multifaceted Approach to Understanding the Botnet Phenomenon," in *Proc. ACMSIGCOMM/USENIX Internet Measurement Conference (IMC'06)*, Oct. 2006.

[35] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling, "Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm," in *Proc. 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, 2008.

[36] P. Wang, L. Wu, B. Aslam, and C. C. Zou, "A Systematic Study on Peer-to-Peer Botnets," in *Proc. International Conference on Computer Communications and Networks (ICCCN)*, Aug. 2009.

[37] R. Albert and A.-L. Barabási, "Statistical Mechanics of Complex Networks," *Review of Modern Physics*, vol. 74, pp. 47–97, 2002.

[38] N. Weaver, S. Staniford, and V. Paxson, "Very Fast Containment of Scanning Worms," in *Proc. 13th Usenix Security Conference*, Aug. 2004.

[39] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan, "Fast Portscan Detection Using Sequential Hypothesis Testing," in *Proc. IEEE Symposium on Security and Privacy*, May 2004.

[40] M. M. Williamson, "Throttling viruses: Restricting propagation to defeat mobile malicious code," in *Proc. 18th Annual Computer Security Applications Conference*, Dec. 2002.

[41] S. A. Khayam, H. Radha, and D. Loguinov, "Worm Detection at Network Endpoints Using Information-Theoretic Traffic Perturbations," in *Proc. IEEE International Conference on Communications*, May 2008.

[42] X. Chen and J. Heidemann, "Detecting Early Worm Propagation through Packet Matching," *Technical Report ISI-TR-2004-585*, Feb. 2004.

[43] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies Using Traffic Feature Distributions," in *Proc. ACM SIGCOMM*, Aug. 2005.

[44] Darknet. [Online]. Available: http://www.cymru.com/Darknet/.

[45] Network Telescope. [Online]. Available: http://www.caida.org/research/security/telescope/.

[46] Honeypots: Tracking Hackers. [Online]. Available: http://www.tracking-hackers.com/.

[47] Internet Motion Sensor. [Online]. Available: http://ims.eecs.umich.edu/.

[48] Internet Sink. [Online]. Available: http://wail.cs.wisc.edu/anomaly.html.

[49] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," in *ACM Transactions on Computer Systems (TOCS)*, vol. 24, no. 2, May 2006, pp. 115–139.

[50] D. W. Richardson, S. D. Gribble, and E. D. Lazowska, "The Limits of Global Scanning Worm Detectors in the Presence of Background Noise," in *Proc. ACM workshop on Rapid malcode*, Nov. 2005.

[51] R. Caceres, N. G. Duffield, J. Horowitz, and D. Towsley, "Multicast-based Inference of Network-internal Loss Characteristics," *IEEE Transactions on Information Theory*, vol. 45, no. 7, pp. 2462–2480, Nov. 1999.

[52] M. Coates, A. Hero, R. Nowak, and B. Yu, "Internet Tomography," *IEEE Signal Processing Magazine*, pp. 47–65, May 2002.

[53] S. Wei and J. Mirkovic, "Correcting Congestion-Based Error in Network Telescope's Observations of Worm Dynamics," in *Proc. Internet Measurement Conference*, Oct. 2008.

[54] Z. Chen and C. Ji, "Optimal Worm-Scanning Method Using Vulnerable-Host Distributions," *International Journal of Security and Networks (IJSN): Special Issue on Computer and Network Security*, vol. 2, no. 1/2, pp. 71 – 80, 2007.

[55] Z. Chen, C. Chen, and C. Ji, "Understanding Localized-Scanning Worms," in *Proc. IEEE IPCCC*, Apr. 2007.

[56] R. Jain, *The Art of Computer Systems Performance Analysis*. John Willy & Sons, Inc., 1991.

[57] Numerical Integration Toolbox. [Online]. Available: http://www.math.umd.edu/users/jmr/241/mfiles/nit/.

[58] Distributed Intrusion Detection System (DShield). [Online]. Available: http://www.dshield.org/.

[59] J. Ma, G. M. Voelker, and S. Savage, "Self-stopping Worms," in *Proc. ACM Workshop on Rapid Malcode*, Nov. 2005.

[60] J. Bethencourt, J. Franklin, and M. Vernon, "Mapping Internet Sensors with Probe Response Attacks," in *Proc. 14th USENIX Security Symposium*, Aug. 2005.

[61] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson, "The Internet Motion Sensor: A Distributed Blackhole Monitoring System," in *Proc. of Network and Distributed System Security Symposium (NDSS'05)*, Feb. 2005.

[62] M. Basseville and I. V. Nikiforov, *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall, Inc., 1993.

[63] J. Xia, S. Vangala, J. Wu, L. Gao, and K. Kwiat, "Effective Worm Detection for Various Scan Techniques," *Journal of Computer Security*, vol. 14, no. 4, pp. 359 – 387, 2006.

[64] M. Vojnovic, V. Gupta, T. Karagiannis, and C. Gkantsidis, "Sampling Strategies for Epidemic-Style Information Dissemination," *to appear in IEEE/ACM Transactions on Networking*.

[65] M. A. Rajab, F. Monrose, and A. Terzis, "On the Effectiveness of Distributed Worm Monitoring," in *Proc. 14th USENIX Security Symposium (Security'05)*, Aug. 2005.

[66] E. C. Titchmarsh, "The Theory of the Riemann Zeta Function," *Oxford University Press*, 1986.

[67] J. Havil, "Gamma: Exploring Euler's Constant," *Princeton University Press*, 2003.

[68] C. C. Zou. Internet Worm Propagation Simulator. [Online]. Available: http://www.cs.ucf.edu/~czou/research/wormSimulation/simulator-codered-100run.cpp.

[69] P. Barford, R. Nowak, R. Willett, and V. Yegneswaran, "Toward a Model for Sources of Internet Background Radiation," in *Proc. Passive and Active Measurement Conference (PAM'06)*, Mar. 2006.

[70] Wikipedia. Conficker. [Online]. Available: http://en.wikipedia.org/wiki/Conficker.

[71] C. C. Zou, W. Gong, and D. Towsley, "Code Red Worm Propagation Modeling and Analysis," in *Proc. 9th ACM Conference on Computer and Communication Security (CCS'02)*, Nov. 2002.

[72] Q. Wang, Z. Chen, and C. Chen, "Charaterizing Internet Worm Infection Structure," Preprint. [Online]. Available: http://arxiv.org/abs/1001.1195.

[73] M. A. Rajab, F. Monrose, and A. Terzis, "On the Impact of Dynamic Addressing on Malware Propagation," in *Proc. 4th ACM Workshop on Recurring Malcode*, Nov. 2006.

[74] M. M. Meerschaert, "Mathematical Modeling (Third Edition)," *Academic Press*, 2007.

[75] The MathWorks. Matlab Curve Fitting Toolbox. [Online]. Available: http://www.mathworks.com/products/curvefitting/.

[76] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," in *Proc. 16th International Conference on Supercomputing (ICS'02)*, Jun. 2002.

[77] Wikipedia. Fractal. [Online]. Available: http://en.wikipedia.org/wiki/Fractal.

[78] B. B. Mandlebrot, "The Fractal Geometry of Nature ," *New York: Freeman*, 1983.

APPENDIX A

INTERNET WORM TEMPORAL INFECTION STRUCTURE

A.1 Estimator Properties ($\hat{\mu}$)

We calculate the bias, the variance, and the MSE of different estimators for estimating μ.

A.1.1 Naive Estimator

Since $\hat{\mu}_{\text{NE}} = 1$, the bias of NE is

$$\text{Bias}(\hat{\mu}_{\text{NE}}) = \text{E}(\hat{\mu}_{\text{NE}}) - \mu = 1 - \tfrac{1}{p}. \tag{1}$$

Note that $\hat{\mu}_{\text{NE}}$ is constant. Thus, the variance of NE is

$$\text{Var}(\hat{\mu}_{\text{NE}}) = \text{E}\left[(\hat{\mu}_{\text{NE}} - \text{E}(\hat{\mu}_{\text{NE}}))^2\right] = 0. \tag{2}$$

Therefore,

$$\text{MSE}(\hat{\mu}_{\text{NE}}) = \text{Bias}^2(\hat{\mu}_{\text{NE}}) + \text{Var}(\hat{\mu}_{\text{NE}}) = \tfrac{(1-p)^2}{p^2}. \tag{3}$$

A.1.2 Method of Moments Estimator / Maximum Likelihood Estimator

Since $\text{E}(\delta_i) = \mu$ for $i = 1, 2, \cdots, n-1$ and Equations (3.8) and (3.14) hold, the bias of $\hat{\mu}_{\text{MME}}$ (or $\hat{\mu}_{\text{MLE}}$) is calculated as

$$\text{Bias}(\hat{\mu}_{\text{MME}}) = \text{E}\left(\tfrac{1}{n-1}\textstyle\sum_{i=1}^{n-1} \delta_i\right) - \mu = 0, \tag{4}$$

which is unbiased. Note that $\text{Var}(\delta_i) = \frac{1-p}{p^2}$ for $i = 1, 2, \cdots, n-1$ and δ_i's are independent. Thus, we have

$$\text{Var}(\hat{\mu}_{\text{MME}}) = \text{Var}\left(\tfrac{1}{n-1}\textstyle\sum_{i=1}^{n-1} \delta_i\right) = \tfrac{1-p}{p^2(n-1)}. \tag{5}$$

Therefore, the MSE of $\hat{\mu}_{\text{MME}}$ (or $\hat{\mu}_{\text{MLE}}$) is

$$\text{MSE}(\hat{\mu}_{\text{MME}}) = \text{Bias}^2(\hat{\mu}_{\text{MME}}) + \text{Var}(\hat{\mu}_{\text{MME}}) = \tfrac{1-p}{p^2(n-1)}. \tag{6}$$

It is noted that for an unbiased estimator, the MSE is identical to its variance.

A.1.3 Linear Regression Estimator

Note that $\hat{\mu}_{\text{LRE}} = \frac{\overline{i\cdot t}-\overline{i}\cdot\overline{t}}{\overline{i^2}-(\overline{i})^2}$. From Equation (3.20) and $t_i = t_0 + \sum_{j=0}^{i-1}\delta_j$, $i = 1, 2, \cdots, n$, we have

$$\begin{aligned}
\overline{i\cdot t} &= \tfrac{1}{n}\textstyle\sum_{i=1}^{n} i\cdot t_i \\
&= \tfrac{n+1}{2}t_0 + \tfrac{1}{n}\textstyle\sum_{i=0}^{n-1}\sum_{j=i+1}^{n} j\cdot\delta_i \\
&= \tfrac{n+1}{2}t_0 + \textstyle\sum_{i=0}^{n-1}\tfrac{(n-i)(n+i+1)}{2n}\delta_i
\end{aligned} \tag{7}$$

and

$$\overline{i}\cdot\overline{t} = \overline{i}\cdot\tfrac{1}{n}\textstyle\sum_{i=1}^{n} t_i = \overline{i}\cdot t_0 + \overline{i}\cdot\sum_{i=0}^{n-1}\tfrac{n-i}{n}\delta_i. \tag{8}$$

Since $\overline{i} = \frac{n+1}{2}$ and $\overline{i^2} = \frac{(n+1)(2n+1)}{6}$,

$$\overline{i\cdot t} - \overline{i}\cdot\overline{t} = \textstyle\sum_{i=1}^{n-1}\tfrac{i(n-i)}{2n}\delta_i \tag{9}$$

and

$$\overline{i^2} - \left(\overline{i}\right)^2 = \tfrac{n^2-1}{12}. \tag{10}$$

Note that $\text{E}(\delta_i) = \mu$ and $\text{Var}(\delta_i) = \frac{1-p}{p^2}$, $i = 0, 1, \cdots, n-1$, and δ_i's are independent. Moreover, $\sum_{i=1}^{n} i^3 = \left(\frac{n(n+1)}{2}\right)^2$ and $\sum_{i=1}^{n} i^4 = \frac{1}{30}(6n^5 + 15n^4 + 10n^3 - n)$. Then, we have

$$\text{E}(\overline{i\cdot t} - \overline{i}\cdot\overline{t}) = \textstyle\sum_{i=1}^{n-1}\tfrac{i(n-i)}{2n}\mu = \tfrac{n^2-1}{12}\mu \tag{11}$$

and

$$\begin{aligned}\text{Var}(\overline{i \cdot t} - \overline{i} \cdot \overline{t}) &= \sum_{i=1}^{n-1} \left(\tfrac{i(n-i)}{2n}\right)^2 \cdot \tfrac{1-p}{p^2} \\ &= \tfrac{1-p}{4n^2p^2}(n^2 \sum_{i=1}^{n-1} i^2 - 2n \sum_{i=1}^{n-1} i^3 + \sum_{i=1}^{n-1} i^4) \\ &= \tfrac{1-p}{p^2} \cdot \tfrac{n^4-1}{120n}. \end{aligned} \tag{12}$$

Therefore, the bias of $\hat{\mu}_{\text{LRE}}$ can be calculated as

$$\text{Bias}(\hat{\mu}_{\text{LRE}}) = \text{E}\left(\tfrac{\overline{i \cdot t} - \overline{i} \cdot \overline{t}}{\overline{i^2} - (\overline{i})^2}\right) - \mu = 0, \tag{13}$$

which is unbiased. Moreover, the variance and the MSE of $\hat{\mu}_{\text{LRE}}$ are

$$\begin{aligned}\text{MSE}(\hat{\mu}_{\text{LRE}}) &= \text{Var}(\hat{\mu}_{\text{LRE}}) \\ &= \text{Var}\left(\tfrac{\overline{i \cdot t} - \overline{i} \cdot \overline{t}}{\overline{i^2} - (\overline{i})^2}\right) \\ &= \tfrac{6(n^2+1)(1-p)}{5n(n^2-1)p^2}. \end{aligned} \tag{14}$$

A.2 Estimator Properties ($\hat{t_0}$)

We calculate the bias, the variance, and the MSE of different estimators for estimating t_0.

A.2.1 Naive Estimator

Since $\hat{t_{0\text{NE}}} = t_1 - \hat{\mu}_{\text{NE}} = t_0 + \delta_0 - 1$, $\text{E}(\delta_0) = \frac{1}{p}$, and $\text{Var}(\delta_0) = \frac{1-p}{p^2}$,

$$\text{Bias}(\hat{t_{0\text{NE}}}) = t_0 + \text{E}(\delta_0) - 1 - t_0 = \tfrac{1-p}{p} \tag{15}$$

$$\text{Var}(\hat{t_{0\text{NE}}}) = \text{Var}(t_0 + \delta_0 - 1) = \tfrac{1-p}{p^2} \tag{16}$$

$$\begin{aligned}\text{MSE}(\hat{t_{0\text{NE}}}) &= \text{Bias}^2(\hat{t_{0\text{NE}}}) + \text{Var}(\hat{t_{0\text{NE}}}) \\ &= \tfrac{(1-p)(2-p)}{p^2}. \end{aligned} \tag{17}$$

Note that when $p \ll 1$, $\text{MSE}(\hat{t_{0\text{NE}}}) \approx \frac{2(1-p)}{p^2}$.

A.2.2 Method of Moments Estimator / Maximum Likelihood Estimator

Note that $\hat{t_{0_{\text{MME}}}} = \hat{t_{0_{\text{MLE}}}} = t_0 + \delta_0 - \hat{\mu}_{\text{MME}}$ and $\text{E}(\delta_0) = \text{E}(\hat{\mu}_{\text{MME}}) = \mu$. Thus,

$$\text{Bias}(\hat{t_{0_{\text{MME}}}}) = t_0 + \text{E}(\delta_0) - \text{E}(\hat{\mu}_{\text{MME}}) - t_0 = 0 \tag{18}$$

$$\text{MSE}(\hat{t_{0_{\text{MME}}}}) = \text{Var}(\hat{t_{0_{\text{MME}}}}) = \text{Var}(\delta_0 - \hat{\mu}_{\text{MME}}). \tag{19}$$

Since $\hat{\mu}_{\text{MME}} = \frac{1}{n-1}\sum_{i=1}^{n-1} \delta_i$ that is independent of δ_0,

$$\begin{aligned} \text{MSE}(\hat{t_{0_{\text{MME}}}}) &= \text{Var}(\hat{t_{0_{\text{MME}}}}) \\ &= \text{Var}(\delta_0) + \text{Var}(\hat{\mu}_{\text{MME}}) \\ &= \tfrac{1-p}{p^2} \cdot \tfrac{n}{n-1}, \end{aligned} \tag{20}$$

based on Equation (5) and $\text{Var}(\delta_0) = \frac{1-p}{p^2}$. Note that when $n \gg 1$, $\text{MSE}(\hat{t_{0_{\text{MME}}}}) \approx \frac{1-p}{p^2}$.

A.2.3 Linear Regression Estimator

Since $\hat{t_{0_{\text{LRE}}}} = t_0 + \delta_0 - \hat{\mu}_{\text{LRE}}$ and $\text{E}(\delta_0) = \text{E}(\hat{\mu}_{\text{LRE}}) = \mu$,

$$\text{Bias}(\hat{t_{0_{\text{LRE}}}}) = t_0 + \text{E}(\delta_0) - \text{E}(\hat{\mu}_{\text{LRE}}) - t_0 = 0 \tag{21}$$

$$\text{MSE}(\hat{t_{0_{\text{LRE}}}}) = \text{Var}(\hat{t_{0_{\text{LRE}}}}) = \text{Var}(\delta_0 - \hat{\mu}_{\text{LRE}}). \tag{22}$$

Note that from Equations (9) and (10), $\hat{\mu}_{\text{LRE}} = \frac{12}{n^2-1}\sum_{i=1}^{n-1} \frac{i(n-i)}{2n}\delta_i$ that is independent of δ_0. Hence,

$$\begin{aligned} \text{MSE}(\hat{t_{0_{\text{LRE}}}}) &= \text{Var}(\hat{t_{0_{\text{LRE}}}}) \\ &= \text{Var}(\delta_0) + \text{Var}(\hat{\mu}_{\text{LRE}}) \\ &= \tfrac{1-p}{p^2} \cdot \tfrac{5n^3+6n^2-5n+6}{5n(n^2-1)}, \end{aligned} \tag{23}$$

based on Equation (14) and $\text{Var}(\delta_0) = \frac{1-p}{p^2}$. Note that when $n \gg 1$, $\text{MSE}(\hat{t_{0_{\text{LRE}}}}) \approx \frac{1-p}{p^2}$.

APPENDIX B

INTERNET WORM SPATIAL INFECTION STRUCTURE

B.1 Statistical Properties of the Number of Children

We apply z-transform to derive the expectation and the variance of the number of children. First, note that Corollary 4.2.3 holds for $n = 1$ and 2. Next, when $n \geq 3$, we define z-transform

$$X_n(z) = \sum_{i=0}^{n-1} c_n(i) z^{-i}. \tag{24}$$

Setting $c_{n-1}(-1) = 1$, we can transform Theorem 4.2.2 to

$$c_n(i) = \tfrac{n-2}{n} c_{n-1}(i) + \tfrac{1}{n} c_{n-1}(i-1),\ 0 \leq i \leq n-1, \tag{25}$$

when $n \geq 3$. Then, putting Equation (25) into Equation (24), we can obtain the difference equation of z-transform

$$X_n(z) = \left(\tfrac{1}{n} z^{-1} + \tfrac{n-2}{n}\right) X_{n-1}(z) + \tfrac{1}{n}. \tag{26}$$

Note that $\mathrm{E}_n[C] = -\frac{dX_n(z)}{dz} \mid_{z=1}$ and $X_{n-1}(1) = 1$, which leads to

$$\mathrm{E}_n[C] = \tfrac{n-1}{n} \mathrm{E}_{n-1}[C] + \tfrac{1}{n}. \tag{27}$$

Since $\mathrm{E}_2[C] = \frac{1}{2}$, we can show by induction that

$$\mathrm{E}_n[C] = \tfrac{n-1}{n}. \tag{28}$$

Moreover, $\mathrm{E}_n[C^2] = \frac{d}{dz}\left[z \frac{dX_n(z)}{dz}\right] \mid_{z=1}$ yields

$$\mathrm{E}_n[C^2] = \tfrac{n-1}{n} \mathrm{E}_{n-1}[C^2] + \tfrac{2}{n} \mathrm{E}_{n-1}[C] + \tfrac{1}{n} \tag{29}$$

$$= \tfrac{n-1}{n} \mathrm{E}_{n-1}[C^2] + \tfrac{3n-5}{n^2}. \tag{30}$$

Thus, we can use $\mathrm{E}_2[C^2] = \frac{1}{2}$ to prove by induction that

$$\mathrm{E}_n[C^2] = 2 + \tfrac{(n-1)(n-2)}{n^2} - \tfrac{2H_n}{n}, \tag{31}$$

where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ is the n-th harmonic number [15]. Therefore,

$$\begin{aligned} \mathrm{Var}_n[C] &= \mathrm{E}_n[C^2] - \mathrm{E}_n^2[C] && (32) \\ &= 2 - \tfrac{n-1}{n^2} - \tfrac{2H_n}{n}. && (33) \end{aligned}$$

B.2 Statistical Properties of the Generation

Similar to the proof of Corollary 4.2.3, we apply z-transform to derive the expectation and the variance of the generation. First, note that Corollary 4.2.6 holds for $n = 1$ and 2. Next, when $n \geq 3$, we define z-transform

$$Y_n(z) = \textstyle\sum_{j=0}^{n-1} g_n(j) z^{-j}. \quad (34)$$

Putting Equation (4.20) into Equation (34), we can obtain the difference equation of z-transform

$$Y_n(z) = \left(\tfrac{1}{n} z^{-1} + \tfrac{n-1}{n}\right) Y_{n-1}(z). \quad (35)$$

Note that $\mathrm{E}_n[G] = -\frac{dY_n(z)}{dz} \mid_{z=1}$ and $Y_{n-1}(1) = 1$, which leads to

$$\mathrm{E}_n[G] = \mathrm{E}_{n-1}[G] + \tfrac{1}{n}. \quad (36)$$

Since $\mathrm{E}_2[G] = \frac{1}{2}$, we can show by induction that

$$\mathrm{E}_n[G] = H_n - 1. \quad (37)$$

Moreover, $\mathrm{E}_n[G^2] = \frac{d}{dz}\left[z \frac{dY_n(z)}{dz}\right] \mid_{z=1}$ yields

$$\mathrm{E}_n[G^2] = \mathrm{E}_{n-1}[G^2] + \tfrac{2}{n}\mathrm{E}_{n-1}[G] + \tfrac{1}{n}. \quad (38)$$

Therefore, combining Equations (36) and (38) gives

$$\begin{aligned} \mathrm{Var}_n[G] &= \mathrm{E}_n[G^2] - \mathrm{E}_n^2[G] \\ &= \mathrm{E}_{n-1}[G^2] + \tfrac{1}{n}(2\mathrm{E}_{n-1}[G] + 1) \\ &\quad -(\mathrm{E}_{n-1}[G] + \tfrac{1}{n})^2 \\ &= \mathrm{Var}_{n-1}[G] + \tfrac{1}{n} - \tfrac{1}{n^2}. && (39) \end{aligned}$$

Thus, we can use $\text{Var}_2[G] = \frac{1}{4}$ to prove by induction that

$$\text{Var}_n[G] = H_n - H_{n,2}, \tag{40}$$

where $H_n = \sum_{i=1}^{n} \frac{1}{i}$ and $H_{n,2} = \sum_{i=1}^{n} \frac{1}{i^2}$.

VITA

QIAN WANG

1983 Born, Dalian, China

2005 B.E., Electronic and Information Engineering
Dalian University of Technology
Dalian, China

2008 M.S., Telecommunications and Networking
Florida International University
Miami, Florida

2010 Ph.D. Candidate, Electrical Engineering
Florida International University
Miami, Florida

PUBLICATIONS AND PRESENTATIONS

Q. Wang, Z. Chen, and C. Chen, "Characterizing Internet Worm Infection Structure," submitted to *IEEE INFOCOM 2011*.

Q. Wang, Z. Chen, and C. Chen, "Darknet-Based Inference of Internet Worm Temporal Characteristics," submitted to *IEEE Transactions on Information Forensics and Security*.

Q. Wang, Z. Chen, C. Chen, and N. Pissinou, "On the Robustness of the Botnet Topology Formed by Worm Infection," in *Proc. IEEE GLOBECOM 2010*, Dec. 2010.

Z. Chen, C. Chen, and Q. Wang, "On the Scalability of Delay-Tolerant Botnets," *International Journal of Security and Networks (IJSN)*, vol. 5, no. 4, 2010.

Z. Chen, C. Chen, and Q. Wang, "Delay-Tolerant Botnets," in *Proc. SecureCPN 2009*, in conjunction with IEEE ICCCN 2009, Aug. 2009.

Q. Wang, Z. Chen, K. Makki, N. Pissinou, and C. Chen, "Inferring Internet Worm Temporal Characteristics," in *Proc. IEEE GLOBECOM 2008*, Dec. 2008.