Creation of the importance scanning worm using information collected by Botnets

Yoon-Ho Choi\textsuperscript{a}, Peng Liu\textsuperscript{a,*,} Seung-Woo Seo\textsuperscript{b,*}

\textsuperscript{a}College of Information Sciences and Technology, Pennsylvania State University, University Park, PA 16802, USA
\textsuperscript{b}School of Electrical Engineering and Computer Sciences, Seoul National University, Seoul 151-744, Republic of Korea

\textbf{A R T I C L E I N F O}

Article history:
Received 2 June 2009
Received in revised form 26 October 2009
Accepted 12 November 2009
Available online 3 December 2009

Keywords:
Importance scanning worms
Worm virulence estimation
Importance scan
Botnet

\textbf{A B S T R A C T}

Importance scanning worm exploits a non-uniform distribution of vulnerable hosts on the Internet. To realize an importance scanning worm, the attacker needs to obtain or estimate the distribution of vulnerable hosts. Zesheng Chen and Chuanyi Ji claimed that a worm can infer the distribution of vulnerable hosts on the Internet by either using public information (e.g., empirical distribution of web servers) or using the distribution of worm-infected hosts during worm propagation. However, the first method may often fail and the second method may not be fast as expected. In this paper, we answer the question, “How do we determine which part on the Internet is more vulnerable, while maintaining a simple worm propagation mechanism?”. To learn the distribution of vulnerable hosts on the Internet, the proposed estimation method applies statistical sampling and estimation theory while using a Botnet, which is a distributed network of Bots. From analytical models and their validation results, we show the proposed estimation method can get sufficiently accurate estimations; in many cases, the good-enough sampling ratio is as small as 0.6%. Also, it is shown that the estimated distribution is unbiased toward the actual distribution of vulnerable hosts on the Internet. Thus, we believe that the estimated distribution table of vulnerable hosts on the Internet will help the worm identify target systems more effectively.

\textcopyright 2009 Elsevier B.V. All rights reserved.

1. Introduction

Self-propagating Internet worms can propagate rapidly over the Internet by exploiting security flaws in widely deployed services on the Internet. Internet worms are a major concern, because computer systems which do not have the latest security updates can be infected just by being connected to the Internet.

Internet worms probe the entire IPv4 address space or the distinct address space, such as the local address space, to find target systems. For example, the Code Red [1] and Witty [2] worms use ‘random scanning’ to find target IP addresses at random. Also, the Code Red II [3] and Nimda [4] use ‘localized scanning’ to preferentially search for vulnerable hosts in the local address space. In the near future, using a certain amount of knowledge, more worms are likely to employ more effective scanning strategies for identifying vulnerable target systems. For example, routable scanning worms [6,7] can select targets only in the routable address space by using the BGP routing table and hitlist worms [8,9] can do using pre-computed hitlists of vulnerable hosts on the Internet. Also, importance scanning worms aim to target the more vulnerable portions on the Internet (address space) by exploiting a non-uniform distribution of vulnerable hosts on the Internet [10–13]; this scanning method is called importance scan. By using such vulnerability information, future worms can determine who to be scanned and who not to be scanned so that they can avoid scanning invulnerable target systems. Thus, it is important to characterize and understand the various types of worms that employ invulnerability-avoiding scanning strategies. In this work, we focus on importance scanning worms which rely on a certain statistic of an underlying vulnerable-host distribution.

The worm infection rate, which means how fast worms can spread, mainly depends on the distinct scanning address space used to find target systems. As shown in many papers [5–13], the worm infection rate is mainly determined by the scanning target space and its vulnerability prevalence (proportion). Let us assume that a certain number of worm scan packets are used by a worm. In the early propagation stage of the worm, the infection rate of the importance scanning worm is much faster than that of the random scanning worm. This is because by using the vulnerable-host distribution on the Internet, the probability that a target host is identified by an importance scan is much higher than that by a random scan. However, implementing such worm is more destructing because the attacker needs to have more knowledge. Here, the attacker is the worm programmer. Also, as different worms may use different vulnerabilities and thus, some real vulnerable-host distributions based on the previous worm propagation, more specifically worm-infected host distribution, may not be used by future worms. In fact, Internet-scale vulnerability...
distribution estimation is the “soul” (or crux) of the importance scanning problem.

Chen et al. [10–13] propose three types of importance scanning worms: dynamic optimal scanning worm; static optimal scanning worm; and self-learning worm using importance scanning. For the first two worms, the authors believe that the attacker can estimate the vulnerable-host distribution by using public information, such as an empirical distribution of web servers provided by a random URL generator and statistical analysis of network telescope observations. However, as we will show shortly in Section 2, such public information is insufficient in many if not most cases. Also, self-learning worms are not as fast as expected (and contradict the design philosophy of worms) because of the reasons we will mention shortly in Section 2.

As a practical approach to infer Internet-wide distribution of vulnerable hosts, Collins et al. [15] focus on predicting how likely a network is to contain compromised hosts from the defender’s viewpoint. For this purpose, they used reports of network activity and traffic logs of a large network. However, this information is not obtainable by attackers. To our best knowledge, there has ever been no practical approach for attackers to infer Internet-wide distribution of vulnerable hosts. The goal of this work is to fill this “hole”.

Specifically, we intend to answer the following questions:

1. Which information should be collected?
2. How to collect the side information used as the input for estimating the vulnerable-host distribution?
3. How to estimate the vulnerable-host distribution from the input as output?

In this work, we first focus on determining the scanning target space. The target space consists of active IP addresses, which are routable, have OSes, are running applications vulnerable to worms, and have open or listening ports exploitable by worms.

Next, we focus to how to collect the side information from the scanning target space. Note that since the side information is collected before worm propagation using a Botnet, this learning process does not limit the speed of worm propagation. A ‘Botnet’ is a distributed network of ‘Bots’, which have been set up to operate as malicious programs, such as Trojan horses (Trojans), on the Internet. For example, there exist real-world Botnets such as IRC-based [16], HTTP-based [17–20], and peer-to-peer (P2P) [21–23] Botnets. All the Bots once connected to the communication channel, such as Internet Relay Chat (IRC) channel, form network of bots while awaiting the attacker command [16]. For the following reasons, we believe that the Botnet is useful for collecting the side information used as the input for estimating the vulnerable-host distribution in a gradual, very low-speed, scan-activity-concealing manner [16–26]. First, a large Botnet consists of about 1.5 million compromised computers [24]. They are ideal for targeted attacks aimed exclusively at a specific organization or network. Second, Bot codes can be customized for specific tasks by the attacker. Third, Bots can allow a system to be remotely controlled without the owner’s knowledge and via private secure channels such as IRC channels. Fourth, some Bots in a Botnet have a high Internet bandwidth capacity. This allows a Bot to act as a remote command and control (C&C) server. Thus, Bot can function as: a Trojan which performs a scan to search for multiple active hosts with a specific open or listening port, where a specific service runs; a Trojan which operates as a proxy server; or as a Trojan which sends information to a C&C server.

Based on these characteristics of the Botnet, each Bot collects information about the proportion of vulnerable hosts in a target group, i.e. categorical data with two possible values, vulnerable or invulnerable to a worm. Here, a target group (shortly group) means any space that consists of either a large subnet or a combination of small subnets. The group is determined by the attacker and will be probed by Bots in the learning stage and by worms in the worm scanning stage.

Finally, we find how to estimate the group distribution of the vulnerable hosts, called the vulnerable-host group distribution, from the collected side information. The side information collected by the Bots is sent to the C&C server. To determine the vulnerable-host group distribution, the C&C server uses simple inferential statistics [30,31]. In this inferential process, the C&C server attempts to minimize the difference between the actual vulnerable-host group distribution and the estimated vulnerable-host group distribution by using the methodology for maximum likelihood estimation (MLE) [30,31]. From a statistical point of view, although the methodology for MLE is simple, it yields estimators with good statistical properties. In addition, MLE provides efficient methods for quantifying uncertainty through confidence bounds. By using the method for MLE based on the collected information, the C&C server estimates the probable vulnerable-host proportion in each group with a given routable IP address space, called the vulnerable-host group proportion. Also, by combining the vulnerable-host group proportions from every group, the C&C server estimates the vulnerable-host group distribution on the Internet. Finally, the attacker designs worms using the estimated vulnerable-host group distribution. Here, we call this type of worm into the ‘Botnet-nurtured worm using the vulnerable-host group distribution’ or the ‘Botnet-nurtured worm’.

The main merits and significance of this work are as follows: (a) To our best knowledge, the proposed approach is the first practical approach for attackers to infer Internet-scale distribution of vulnerable hosts; (b) this work is the first quantitative approach to estimate the vulnerable-host group distribution using statistical sampling and estimation theory while using a Botnet; (c) our simulation results show that by probing (or sampling) a very small portion of the Internet (e.g., the good-enough sampling ratio is as small as 0.6%), very accurate Internet-scale vulnerability distribution estimations can typically be inferred; (d) this study shows that Botnet-nurtured worms are no longer a concept. These worms are a real threat; and (e) this study shows that Botnets introduce a serious new Internet-scale surveillance threat.

The remaining of this paper is organized as follows. In Section 2, we analyze the assumptions behind the importance scanning worms. In Section 3, we define the problem. Section 4 shows how to collect the side information used as the input for estimating the vulnerable-host group distribution. After showing how to estimate the vulnerable-host group distribution in Section 5, we validate the estimation error of our proposed estimation method in Section 6. Finally, after discussing about possible countermeasures in Section 7, we conclude the work in Section 8.

2. Characteristics of importance scanning worms

In this section, we analyze importance scanning worms in detail [10–13]. The importance scanning worm is inspired by the observation that if we divide the Internet into several groups, the vulnerable-host distribution in these groups is highly non-uniform. From this observation, Chen et al. propose three types of importance scanning worms: dynamic optimal scanning worm; static optimal scanning worm; and self-learning worm using importance scanning. Note that it is clear that to realize an importance scanning worm, it is very important to estimate a vulnerable-host distribution in these groups is highly non-uniform. From this observation, Chen et al. propose three types of importance scanning worms: dynamic optimal scanning worm; static optimal scanning worm; and self-learning worm using importance scanning. Note that it is clear that to realize an importance scanning worm, it is very important to estimate a vulnerable-host group distribution. In this paper, we investigate the importance scanning worms by focusing on the estimation process of the vulnerable-host group distribution and answer the question, “Are such importance scanning worms practical?”.
Dynamic and static optimal scanning worms, which are shown in [10,11], assume that the vulnerable-host group distribution can be made available or obtainable by attackers before the worms propagate. The authors believe that public information, such as an empirical distribution of web servers provided by a random URL generator and statistical analysis of network telescope observations, can be used while estimating the vulnerable-host group distribution. Our question is, "Is public information sufficient to estimate a vulnerable-host group distribution?"

First, the distribution of many vulnerabilities may not be available or obtainable, because there are many types of worms; These worms use various security flaws in different services. For example, contrast to Code Red [1], which affects systems with a vulnerability in Windows web server, Witty [2] worm affects systems with a vulnerability in ICQ parsing by Internet Security Systems (ISS) products. Second, it is difficult to guarantee how accurate this public information is. Thus, we believe that public information is not sufficient to estimate the distribution required to design a practical worm. Also, these worms need a lot of information exchange among worm-infected hosts; the information exchange results in bandwidth overhead. For information exchange to take place, there needs to be pre-installed executable programs allowing the worms to send information to each other. However, how this operation works is not clear in the previous papers [10,11] on this subject. For the above reasons, we believe that dynamic and static optimal scanning worms are not practical. This indicates that we need to consider the question as to how to learn the vulnerable-host group distribution in more details.

As one candidate for practicality, Chen et al. propose a self-learning worm using importance scanning [11]. By assuming that it may not be easy for attackers to get the vulnerable-host group distribution before a worm is released, they propose a method of learning the vulnerable-host group distribution while the worm is propagating. Self-learning worms have two propagation stages: the learning stage and importance-scanning stage. In the learning stage, each worm-infected host, called a worm client, infects new hosts by performing random scanning or routable scanning, and then the record of the worm clients are collected by a host with a high Internet bandwidth capacity, called a worm server. When the worm server records a sufficient number of the worm clients, it estimates the vulnerable-host group distribution and sends the corresponding group distribution to all worm clients in the list. In importance-scanning stage, the initially worm-infected hosts and the newly worm-infected hosts during the learning operation perform importance scanning using the group scanning distribution. Our question is "Is self-learning worm really fast as expected?"

For the following reasons, we believe that even a self-learning worm using importance scanning is still not fast as expected. First, it is clear that a simple self-propagating worm can quickly spread across the Internet [32]. However, the self-learning worm learns the vulnerable-host group distribution while propagating. Learning while propagating cannot benefit from importance scanning; this learning process may result in decreasing the propagation speed at least in the early propagation stage. For example, in addition to collecting samples to estimate the vulnerable-host group distribution, a self-learning worm could spend substantial time on the learning stage to observe almost 500–10,000 worm-infected hosts before launching the importance scanning stage, as indicated in [11,12]. Second, the worm clients during the learning operation based on random scanning or routable scanning can be detected by the defender. Also, since the learning phase is pretty long, if some worm clients are detected during this phase, the defenders could be able to patch the vulnerable services before the importance-scanning phase starts. On the other hand, in the proposed worm, since Bots were typically pre-farmed using a vulnerability different from the one used by the Botnet-nurtured worm, detection of some Bots may not be a big concern (or threat) to the worm attacker. Third, we need to use the different number of observations as the input for estimating the vulnerable-host group distribution depending on the size of the target group. However, the self-learning worm simply assumes that if a sufficient number of worm-infected hosts are harvested, the attackers have the ability to estimate the underlying vulnerable-host group distribution. To satisfy a desired accuracy of estimation, while the authors in [11] roughly derive the total number of samples based on MSE, we derive the required sample size in a target group. Moreover, the effect of estimation error on the worm propagation with a small or medium sample size is not studied in [11].

To realize a practical importance scanning worm, we propose a new importance scanning worm called Botnet-nurtured worm. Botnet-nurtured worm estimates the vulnerable-host group distribution using a Botnet and then, such distribution is used in worm propagation, while an importance scanning worm proposed by Zesheng Chen and Chuan Yi Ji form a Botnet by using worm-infected hosts while worm propagating.

3. Botnet-nurtured worm: problem definition

In this section, we introduce the 'Botnet-nurtured worm'. After describing a Botnet architecture for the collection of side information, we describe the objective of this work.

3.1. What is a 'Botnet-nurtured worm'? 

To answer this question, we consider how to obtain the Botnet and describe how a Botnet-nurtured worm is designed using the Botnet.

An attacker may compromise hosts using a known vulnerability in order to construct a Botnet. However, the attacker's behavior can be detected by the defender while he or she is in the process of compromising hosts. Note that the goal of the attacker is not to construct Botnets, but to use them as a controlled network to collect the input required for estimating vulnerable-host group distribution. Thus, the attacker may obtain a Botnet by simply cooperating with a 'Botnet Farmer', which is an individual or group that manages one or more Botnets to generate revenue by selling systems [14], or by simply buying a Botnet from a 'Botnet Farmer'. Once the Botnet is obtained, the Botnet-nurtured worm can be designed in the following ways.

Botnet-nurtured worms have two propagation stages: the learning stage by the Botnet and the scanning stage by the worm. In the learning stage, Bots collect the information from active IP addresses and send the collected information to the C&C server. By combining the information collected by the Bots, the C&C server determines the proportion of vulnerable hosts in each group, and then estimates the vulnerable-host group distribution in the scanning target space. In the scanning stage, the Botnet-nurtured worm scans the target groups based on the group scanning distribution, i.e. based on the estimated vulnerable-host group distribution.

The propagation process of the Botnet-nurtured worm has the following characteristics. In the learning stage, the Botnet is used to collect the information used as the input for estimating the vulnerable-host group distribution. While collecting the information, Bots check that the targets are vulnerable to the worm via very low-speed scanning manner and thus, hide themselves from the defender. Next, to satisfy the expected estimation error between the actual distribution and the estimated distribution, the C&C server collects the required samples depending on the size of the groups. In contrast to a self-learning worm using importance scanning, the vulnerable-host group distribution is estimated before the worm propagates. Thus, in the scanning stage, the Botnet-nurtured worm...
simply fast propagates over the Internet using the estimated group scanning distribution.

3.2. Network set-up

As shown by the Zou et al. in [7], we can divide the target space into several groups based on the routable IP address prefixes in the Border Gateway Protocol (BGP) routing tables such as the tables from Route Views [27] and RIPE NCC [28]. Also, as shown by the Rajab et al. in [29], we may employ lightweight sampling of the IP space to identify prefixes that contain routable IP addresses. Through this process called local scanning, we may scan the target space and collect information from the target space, but not include Bots or subnets that do include Bots. If Bots are not included in a target group or its subnet, Bots outside the target group should collect side information in the following ways. Also, the attacker combines continuous small subnets together when their prefix lengths are longer than /8. For example, if 128.119.0.0/16 and 128.119.85.0/24 appear in the BGP routing tables, only 128.119.0.0/16 that has a longer prefix length may be a target group. Also, if 128.120.84.0/24 and 128.120.85.0/24 appear in BGP routing tables, the attacker combines these two continuous networks into [128.120.84.0, 128.120.85.255], which may be a target group. Thus, a target group consists of either one subnet or several subnets. Note that a target group can consist of subnets that do not include Bots or subnets that do include Bots. If Bots are not included in a target group or its subnet, Bots outside the target group scan the target space and collect information from the target space, through a process called remote scanning. Otherwise, Bots in each target space are used to collect information from the target space, through a process called local scanning. Fig. 1 shows the network set-up established by the attacker.

We suppose that the attacker divides the worm's target space into \( N_g \) target group(s). A target group \( i \) consists of \( N(i) \) routable IP addresses \( \sum_{i=1}^{N(i)} N(i) = N_g \) and may have Firewalls (F/Ws) and honeypots against sampling. \( N_v \) vulnerable hosts are non-uniformly deployed over \( N_g \) routable IP addresses in \( N_v \) target group(s). Also, a Botnet consists of one C&C server and \( N_B \) Bots. \( m(i) \) Bots among \( N_B \) Bots scan a target group \( i \). These variables are controlled by a C&C server. In Table 1, we summarize the parameters, which are used in this work.

3.3. Objectives and preliminaries

The objective of this work is to find a practical way of estimating the vulnerable-host group distribution while maintaining a simple worm propagation mechanism. Specifically, an attacker tries to look for the estimated proportion of vulnerable hosts in a group \( i \) over the Internet, \( p(i) \), with maximum likelihood to be the actual proportion of vulnerable hosts in a group \( i \) over the Internet, \( \hat{p}(i) \). The estimation error between \( \hat{p}(i) \) and \( p(i) \) is defined as the following mean squared error (MSE):

\[
MSE(\hat{p}(i)) = E\left( (\hat{p}(i) - p(i))^2 \right).
\]

(1)

where \( E(\cdot) \) is the arithmetic mean. Thus, the attacker's objective function is given as

\[
\min MSE(\hat{p}(i)).
\]

(2)

To minimize the estimation error, the information collected for a target group needs to accurately represent the entire address space in the group. To do so, we focus to the small difference between the actual vulnerable-host proportion in a target group \( i \) with \( N(i) \) routable IP addresses, \( p(i,N(i)) \), and the estimated vulnerable-host proportion in a target group \( i \) with \( N(i) \) routable IP addresses, \( p(i,\hat{N}(i)) \). We suppose that Bots collect information in the following ways.

- **Determine the required scanning space in a large group:** If the size of a target group is large, it is not usually practical to scan the entire address space. It is costly to obtain side information from a large space. For example, in the worst case that a single IPv4 address space itself is a group, Bots should collect side information from 2^32 addresses. On the other hand, it is easier to collect and to summarize the observations with a sample than with a complete count. Also, a large number of scans can be easily detected by a defender. Thus, to make predictions or inferences concerning the vulnerable-host proportion in a large group, the C&C server determines the required scanning space, i.e. the required sample size.

- **Use simple random sampling while collecting side information:** The Bots scan the sampling units based on simple random sampling, which causes each sampling element to have an equal probability of selection.

- **Bots close to a target group scan the target group:** Bots in target group \( i \) scan the group or Bots in a subnet of the group \( i \) scan the subnet. If there is no Bot in a target group, Bots close to the target group scan the target group. This increases the probability that the scans succeed.

Now, we go into details about how to collect the side information used as the input and how to estimate the output, i.e. the estimated vulnerable-host group distribution.

4. Botnet-nurtured worm: sample collection

Now, we consider the question, “How do the Bots scan the target groups to collect the samples, which are used to estimate the proportion of vulnerable hosts in a group?” To answer this question, we propose two methods of collecting samples from a target group. Here, the Bots attempt to collect categorical data with two possible values, i.e. vulnerable or invulnerable to a worm, which are summarized with proportions.

4.1. Information collection for categorical data analysis

If the target space is set-up, the C&C server controls and commands \( m(i) \) Bots to collect the categorical data with two possible values from target group \( i \). After collecting the side information from a target group \( i \), the \( m(i) \) Bots send it back to the C&C server. To keep their information exchange stealthy, the Bots and C&C server use their own private secure channel while exchanging information. Also, to keep the Bots' traffic load, some Bots with a high Internet bandwidth capacity are used as proxy servers while scanning each target group. Proxy servers with random ports or...
random addresses can avoid possible countermeasures, which are discussed in Section 7.2. Even though proxy servers are detected by the defender, Bots may remain undetected. However, the actual scanning is carried out by some fixed proxies and may cause the single-point-of-failure problem [29]. That is, if you use a fixed proxy to scan a network, it is so easy to detect and block such an IP sweep scanning from a single source scanner. Also, if this proxy is taken down, then most scans are not able to carry out. A distributed and coordinated scan is much more stealthy in this sense. Thus, the difference between the two methods is their scanning methods. Fig. 2 shows the difference between the two sample collection methods when scanning a target group. The details of the data collection in target group \( i \) are as follows.

1. **C&C server \( \rightarrow \) Bots:** After the C&C server determines the required sample size corresponding to a target group \( i \), i.e. \( n(i) \),
   - In method 1: It commands each Bot to scan the IP addresses in a target group \( i \) uniformly at random and collects \( n(i) \) elements.
   - In method 2: It logically divides the entire set of IP addresses in a target group \( i \) into \( m(i) \) sub-groups, whose IP addresses are randomly selected and whose sizes are the same as each other. It commands each Bot to scan the IP addresses in the corresponding sub-group uniformly at random and collects \( n(i)/m(i) \) samples.

2. **Bots \( \rightarrow \) Target group:**
   - In method 1: Using a proxy server, each Bot scans its corresponding IP address space in a group \( i \) and collects the scanning results against the IP addresses in target group \( i \).
   - In method 2: Each Bot scans its corresponding IP address sub-group in a group \( i \) and collects the scanning results against the IP addresses in target group \( i \).

3. **Bots \( \rightarrow \) C&C server:** If the size of the samples collected by Bots is equal to the required group sample size, the Bots send the samples to the C&C server.

4. **C&C server:**
   - In method 1: C&C server eliminates overlapped samples, whose IP addresses are the same as each other. If the number of samples is equal to the required sample size in target group \( i \), the C&C server estimates the vulnerable-host parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( N_g )</td>
<td>The number of target groups in the target space</td>
</tr>
<tr>
<td>( N_{k(i)} )</td>
<td>Routable IP address population in the target space</td>
</tr>
<tr>
<td>( N_V )</td>
<td>Vulnerable IP address population over the routeable IP address space</td>
</tr>
<tr>
<td>( N_{v(i)} )</td>
<td>Actual vulnerable-host population in a target group ( i )</td>
</tr>
<tr>
<td>( \hat{N}_{v(i)} )</td>
<td>Estimated vulnerable-host population in a target group ( i )</td>
</tr>
<tr>
<td>( N(m, \sigma) )</td>
<td>Normal distribution, whose mean is ( m ) and standard deviation is ( \sigma )</td>
</tr>
<tr>
<td>( N_k )</td>
<td>Bot population in the Botnet</td>
</tr>
<tr>
<td>( m(i) )</td>
<td>The number of Bots that scan a target group ( i )</td>
</tr>
<tr>
<td>( s_0(i) )</td>
<td>Scan rate of each Bot that probes a target group ( i )</td>
</tr>
<tr>
<td>( p_{r(i)} )</td>
<td>The proportion of IP addresses in ( N_k ) that are selected, called sampling ratio, i.e. ( \sum_{i=1}^{N_k} m(i)/N_k )</td>
</tr>
<tr>
<td>( p(i, N(i)) )</td>
<td>Actual vulnerable-host proportion in a target group ( i ) with ( N(i) ) routeable IP addresses, i.e. ( N_{v(i)}/N(i) )</td>
</tr>
<tr>
<td>( p(i, \hat{N}_{v(i)}) )</td>
<td>Estimated vulnerable-host proportion in a target group ( i ) for ( p(i, N(i)) ), i.e. ( \hat{N}<em>{v(i)}/\hat{N}</em>{v(i)} )</td>
</tr>
<tr>
<td>( p(i) )</td>
<td>Actual proportion of vulnerable hosts in a target group ( i ) over the Internet, called actual vulnerable-host group distribution, i.e. ( N_{v(i)}/N_{v} )</td>
</tr>
<tr>
<td>( p(\hat{i}) )</td>
<td>Estimated proportion of vulnerable hosts in a target group ( i ) over the Internet, called estimated vulnerable-host group distribution, i.e. ( \hat{N}<em>{v(i)}/\hat{N}</em>{v} )</td>
</tr>
<tr>
<td>( e_{p(i)} )</td>
<td>The error in estimating ( p(i) ) by ( \hat{p}(i) ), which is denoted into (</td>
</tr>
<tr>
<td>( e_{p(\hat{i})} )</td>
<td>The relative error in estimating ( p(i) ) by ( \hat{p}(i) ), which is denoted into (</td>
</tr>
</tbody>
</table>

**Fig. 2.** Scanning paradigm for collecting categorical data with two possible values, i.e. vulnerable (gray circle in targeted group) or invulnerable (white circle in targeted group) to a worm: (a) in method 1; (b) in method 2. Here, dotted circles in targeted group mean the logical sub-groups targeted by Bots.
proportion in target group \( i \). And then, by combining the vulnerable-host proportions in every target group, the C&C server estimates the vulnerable-host group distribution, where \( N_y(i) \) is the number of samples collected in group \( i \). If the number of samples is less than the required sample size, the C&C server will send a command to collect the remaining samples.

- **In method 2**: If the number of samples from \( m(i) \) Bots is equal to the required sample size in target group \( i \), the C&C server estimates the vulnerable-host proportion in target group \( i \). And then, by combining the vulnerable-host proportions in every target group, the C&C server estimates the vulnerable-host group distribution. Otherwise, the C&C server commands some of the \( m(i) \) Bots to collect the remaining samples.

Method 1 is simpler than method 2, because each Bot randomly scans a target group without determining the corresponding target systems. However, it takes much more time to collect the samples. If several Bots scan the same host, the C&C server should discard the duplicate sampling elements and thus may make the Bots collect more samples several times. Also, the scanning activity of each Bot may easily be detected by the defender, due to the frequent access to the target group. In addition, the scanning activity of each Bot can be easily detected and blocked due to the single-point-of-failure problem. In contrast to method 1, method 2 can reduce the time and effort required to collect the samples. In method 2, a group is divided into \( m(i) \) sub-groups and each Bot scans its own IP address space in group \( i \). Such a distributed and coordinated scan is much more stealthy compared to an IP sweep scanning from a single source scanner. This results in each Bot scanning different target space and collecting non-overlapping samples. Also, by scanning its own IP address space in group \( i \), each Bot avoids to scan the large IP address space on Internet and to expose its existence in scanning. Thus, in the following section, we go into details based on estimation method 2.

### 4.2. Example

**Fig. 3** shows the attacker’s network set-up that is considered in this example. We assume that all of the remote side information belongs to the five groups from group \( A \) to group \( E \). The Bots in groups \( A, B, C \) and \( E \) collect information from a target group \( D \); the collected information is sent to the C&C server. In the target group, no Bots exist. Also, we assume that a worm propagates through a host whose Operating System is Windows XP and through port 80. 240 IP addresses among 1000 routable IP addresses in the target group \( D \) (\( N(D) = 1000 \)) are using Windows XP. For 236 of these 240 IP addresses, port 80 is in the ‘open’ or ‘listening’ status while the remaining ones are in the ‘closed’ or ‘established’ status, i.e. \( p(D, N(D)) = 0.236 \). Suppose that the Botnet consists of 10 Bots, which are randomly deployed in four groups, i.e. group \( A \) to group \( C \), and group \( E \). Bot \( A-1 \) acts as a C&C server. A Trojan Notifier is installed on each Bot. Also, group \( B \) and group \( C \) are located close to the target group \( D \), i.e. \( m(D) = 6 \).

First, as six Bots are located close to the target group \( D \), Bot \( A-1 \) divides the 1000 IP addresses in target group \( D \) into six sub-groups by randomly selecting IP addresses in group \( D \). The information about each sub-group is sent to each Bot (Step 1). After receiving this information, three Bots in group \( B \) scan group \( D \) and three Bots in group \( C \) scan group \( D \) in a distributed manner (Step 2). Here, let us assume that \( n(D) = 300 \) and the required 50 samples are collected by each Bot. Now, the six Bots send their collected samples to Bot \( A-1 \) (Step 3). By analyzing the samples obtained from the six Bots, Bot \( A-1 \) estimates the proportion of vulnerable hosts in group \( D \). Let us assume that \( p(D, N(D)) = 0.220 \) and \( p(i, N(i)) \) for the other groups are given as: \( p(A, N(A)) = 0.015 \), \( p(B, N(B)) = 0.112 \), \( p(C, N(C)) = 0.142 \), and \( p(E, N(E)) = 0.274 \) (Step 4).

### 5. Botnet-nurtured worm: estimation of the vulnerable-host group distribution

Now, we answer the question, “How does the C&C server estimate the vulnerable-host group distribution?”. More specifically, using the method of \( \text{MLE} \), we maximize the probability of occurrence of the required sample values, which results in minimizing the difference between the actual vulnerable-host group distribution and the estimated vulnerable-host group distribution.

#### 5.1. Point estimator of proportion of vulnerable hosts in a group

Let \( X \) be a discrete random variable with probability distribution \( f(x; p(i, N(i))) \). Since, using simple random scanning, the Bots collect a number of samples corresponding to group sample size \( n(i) \), let \( x_{nk}(i) \) denote the event that the \( j \)-th host in a target group \( i \), which is scanned by \( k \)-th Bot, is vulnerable, i.e.

\[
 x_{nk}(i) = \begin{cases} 1, & \text{if vulnerable;} \\ 0, & \text{otherwise.} \end{cases}
\] (3)

**Fig. 3.** Example of an attacker’s network set-up, where Bot-A-1 operates as a C&C server.
That is, a discrete random variable \(X\) follows a Bernoulli random variable with the probability of finding a vulnerable host by one Bot. Here, the probability of finding a vulnerable host for the scans sent out by one Bot equals the actual vulnerable-host proportion in a target group \(i\) whose population size is \(N(i)\), i.e. \(p(i, N(i))\). Thus, the probability mass function (pmf) is

\[
f(x; p(i, N(i))) = \begin{cases} p(i, N(i))^x (1 - p(i, N(i)))^{1-x}, & \text{if } x = 0, 1; \\ 0, & \text{otherwise.} \end{cases} \tag{4}
\]

where \(E[X] = p(i, N(i))\). Also, the likelihood function \((L(p(i, N(i))))\) is defined as

\[
L(p(i, N(i))) = p(i, N(i))^{x_{i1}} (1 - p(i, N(i)))^{1-x_{i1}} p(i, N(i))^{x_{i2}} (1 - p(i, N(i)))^{1-x_{i2}} \ldots p(i, N(i))^{x_{iN}} (1 - p(i, N(i)))^{1-x_{iN}}.
\]

Based on the fact that \(p(i, \hat{N}(i))\) which maximizes \(L(p(i, N(i)))\) also maximizes \(\ln L(p(i, N(i)))\), we find the MLE of \(p(i, N(i))\) as

\[
p(i, \hat{N}(i)) = \arg \max_{p(i, N(i))} \ln L(p(i, N(i))) = \arg \max_{p(i, N(i))} \left[ \sum_{j=1}^{m(i)} \sum_{k=1}^{n(i)} x_{jk}(i) \ln p(i, N(i)) + (n(i)m(i) - \sum_{j=1}^{m(i)} \sum_{k=1}^{n(i)} x_{jk}(i)) \ln (1 - p(i, N(i))) \right].
\tag{6}
\]

By estimating \(\ln \left( \frac{\ln(p(i, \hat{N}(i)))}{\ln(p(i, N(i)))} \right)\) to be zero and solving for \(p(i, N(i))\), the MLE of \(p(i, \hat{N}(i))\) is given as

\[
p(i, \hat{N}(i)) = \frac{1}{n(i) m(i)} \left[ \sum_{j=1}^{m(i)} \sum_{k=1}^{n(i)} x_{jk}(i) \right].
\tag{7}
\]

Here, Eq. (7) implies that if \(N(i), n(i), \) and \(m(i)\) are identical for all groups, the MLE is indeed the simple proportional estimator in [11].

Finally, the expected value of \(p(i, \hat{N}(i))\) is given as

\[
E[p(i, \hat{N}(i))] = \frac{1}{n(i) m(i)} \left[ \sum_{j=1}^{m(i)} \sum_{k=1}^{n(i)} x_{jk}(i) \right] = p(i, N(i)).
\tag{8}
\]

This indicates that the estimator is unbiased, i.e. the expected value of the difference between the estimated vulnerable-host proportion and the actual vulnerable-host proportion in a target group is almost zero. Also, this indicates that the estimated population of vulnerable hosts in a target group \(i\) represents the population of vulnerable hosts in the group.

### 5.2. Required group sample size

Suppose that there is an infinitely large number of IP addresses in a target group, i.e. \(N(i) = \infty\), and \(n_0(i)\) samples are collected. Given \(p(i, \hat{N}(i))\), we denote the error in estimating \(p(i, N(i))\) by \(e(p(i, \hat{N}(i))) = |p(i, N(i)) - p(i, \hat{N}(i))|\). Note that when a discrete random variable \(X\) follows a Bernoulli random variable, if group sample size \(n_0(i)\) is small relative to the group space, the number of successes (vulnerable hosts) in the samples has an approximately binomial distribution. If \(n_0(i)\) is large, the sample proportion is approximately normally distributed. Also, we note that we are approximately 100(1 - \(\alpha\)) percent confident that \(e(p(i, \hat{N}(i)))\) is less than the sampling error \(e_\alpha = \frac{Z_{\alpha/2}}{\sqrt{\frac{n_0(i)}{1 - n_0(i)}}}\).

Here, \(e_\alpha\) is the sampling error of the proportion for the infinite group size \(N(i)\), \(\alpha\) is the significance level used to compute the confidence level and \(SE_{p(i, \hat{N}(i))} = \sqrt{p(i, \hat{N}(i))(1 - p(i, \hat{N}(i)))/n_0(i)}\) is the standard error of the estimated proportion in a target group \([i]\). Thus, when we select the group sample size, we may choose it to be 100(1 - \(\alpha\)) percent confident that the error is less than some specified value \(e(p(i, \hat{N}(i)))\). By solving \(e(p(i, \hat{N}(i)))\) for \(n_0(i)\), the appropriate group sample size is

\[
n_0(i) = \frac{Z_{\alpha/2}^2 e^2(p(i, \hat{N}(i)))}{p(i, \hat{N}(i))(1 - p(i, \hat{N}(i)))}. \tag{9}
\]

However, to use Eq. (9), an estimate of \(p(i, \hat{N}(i))\) is required. Since the group sample size obtained from Eq. (9) will always be a maximum for \(p(i, \hat{N}(i)) = 0.5\) (maximum variability), we obtain an upper bound on \(n_0(i)\).

\[
n_0(i) \leq \left( \frac{Z_{\alpha/2}^2}{e^2(p(i, \hat{N}(i)))} \right)^2 (0.25). \tag{10}
\]

In other words, we are at least 100(1 - \(\alpha\)) percent confident that the error in estimating \(p(i, N(i))\) by \(p(i, \hat{N}(i))\) is less than \(e(p(i, \hat{N}(i)))\) if the group sample size is

\[
n_0(i) = \left( \frac{Z_{\alpha/2}^2}{e^2(p(i, \hat{N}(i)))} \right)^2 (0.25), \text{ where } p(i, \hat{N}(i)) = 0.5. \tag{11}
\]

From the premise that the samples are selected from an infinitely large population, Eq. (11) does not depend on the size of the population in a group \(i\). If the group sample size is small in comparison with \(N(i)\), this assumption is reasonable according to the sampling theory. However, when the group size \(N(i)\) is finite and the group sample size \(n(i)\) in the finite population is not small, we need to adjust the standard error \(SE_{p(i, \hat{N}(i))}\) using a finite population correction (fpc); this parameter accounts for the added precision gained by sampling close to a larger percentage of the population. Here, \(fpc = \sqrt{\frac{N(i) - n(i)}{N(i) - 1}}\) and \(N(i) - n(i) < N(i) - 1\) since \(n(i)\) is greater than 1 for all practical cases. Thus, the standard error of the proportion for the finite group size \(N(i)\) is always smaller than that before it is corrected. The corrected standard error is given as

\[
SE_{p(i, \hat{N}(i))} = fpc \cdot SE_{p(i, \hat{N}(i))}', \tag{12}
\]

and the sampling error of the proportion for the finite group size \(N(i)\) is given as

\[
e = \frac{Z_{\alpha/2}^2 SE_{p(i, \hat{N}(i))}'}{n_0(i) - 1}. \tag{13}
\]

From Eqs. (9) and (13), given \(e(p(i, \hat{N}(i)))\), \(\alpha\) and \(N(i)\), the required sample size in a target group \(i\) is defined as

\[
n_0(i) = \frac{n_0(i)N(i)}{n_0(i) + (N(i) - 1)}. \tag{14}
\]

### 5.3. Estimating vulnerable-host group distribution

Now, based on the required sample size \(n_0(i)\) obtained from Eq. (14) and the estimated proportion of vulnerable hosts in a group
iE[p(i, NV(i))] obtained from Eq. (8), we estimate the vulnerable-host group distribution, i.e. p(i). In Eq. (8), we show that E[p(i, NV(i))] is unbiased. That is, using the proposed estimation method, the estimated proportion in a target group has a maximum likelihood to be the actual proportion in the target group. Thus, by combining the estimated proportions from every group, we can minimize the estimation error shown in Eq. (1).

When NVv vulnerable hosts exist in every target group, the actual vulnerable-host group distribution is defined as

\[ p(i) = \frac{NV(i)}{NV}. \] (15)

Also, the estimated population of vulnerable hosts in every target group is defined as \( NV = \sum_{j=1}^{NV} NV(j) = \sum_{j=1}^{NV} p(j, NV(j)) \times NV(j). \) Thus, the estimated vulnerable-host group distribution is defined as

\[ \hat{p}(i) = \frac{NV(i)}{NV} = \frac{p(i, NV(i)) \times NV(i)}{\sum_{j=1}^{NV} p(j, NV(j)) \times NV(j)}. \] (16)

5.4. Example

From the example shown in Section 4.2, let us assume that N(A) = 1000, N(B) = 1500, N(C) = 500, and N(E) = 1000. As \( NV(A) = p(A, NV(A)) \times N(A) = 15 \), \( NV(B) = p(B, NV(B)) \times N(B) = 168 \),
\[ NV(C) = p(C, NV(C)) \times N(C) = 71 \], \( NV(D) = p(D, NV(D)) \times N(D) = 220 \), \( NV(E) = p(E, NV(E)) \times N(E) = 274 \), and \( \sum_{j=1}^{NV} p(j, NV(j)) \times NV(j) = 748 \), the vulnerable-host group distribution is given as: \( p(A) = 0.020 \), \( p(B) = 0.223 \), \( p(C) = 0.094 \), \( p(D) = 0.300 \), and \( p(E) = 0.363 \).

6. Botnet-nurtured worm: validation

We compare the actual vulnerable-host group distribution on the target space and the estimated vulnerable-host group distribution based on MSE under the following conditions: VC++ 6.0 is used for embodying simulator. We note that the estimation error varies across different setups. We measured the estimation error under various characteristics of the sample: under different sample sizes; and under different vulnerable-host populations. Also, we measured the estimation error under various characteristics of a Botnet: under different numbers of Bots; and under different distributions of Bots over the target groups. To investigate the influence of a target space on the estimation error, we measure the estimation error under different probabilities that a target system replies for a scan sent out by one Bot.

A set of estimation results are shown in Table 2 in Appendix A, where the sampling ratio is 2.0898% (x = 0.01 and z_0.05 = 2.58). To make the vulnerable-host group distribution table interpretable, we compare the distributions based on the error in estimating \( p(i) \) by \( p(i) \), i.e. \( e(p(i)) \), and the relative error in estimating \( p(i) \) by \( p(i) \), i.e. \( e(p(i)) \). In Table 2, it is shown that the absolute estimation errors are consistently less than 0.01%, and the most relative estimation errors are less than 2%.

Based on the following observations, we believe that the proposed statistical sampling and estimation method using the Botnet estimates the actual vulnerable-host group distribution efficiently.

6.1. Description of network set-up

We suppose that there are \( 2^{24} \) IP addresses in the attacker’s target space and are NVv vulnerable hosts on the target space. We assume that the attacker divides the worm’s target space into \( 2^v \) target groups, each of which consists of \( 2^{16} \) routable IP addresses and has a F/W. We do experiments on two types of non-uniform distributions of vulnerable hosts on the target groups: in Type 1, NVv vulnerable hosts are normally distributed on the groups, i.e. \( N(NV/256, NV \times (2/3)) \); and in Type 2, a certain group has more vulnerable-host proportion than other groups. Here, \( N(\mu, \sigma^2) \) is the normal distribution with mean \( \mu \) and variance \( \sigma^2 \). That is, in Type 1 experiment, all the groups have the same percentage of vulnerable hosts, and this percentage ranges from 0.01% to 4.00%. In contrast, in Type 2 experiment, different groups can have different percentages of vulnerable hosts, and these percentages range from 0.5% to 45%. In two types of distribution of vulnerable hosts, we set up the “ground truth” for group i as follows. First, we set up the percentage of vulnerable hosts in this group, which is denoted \( p(i, NV(i)) = (p(i) \times NV) / NV(i) \). Second, we determine the set of vulnerable hosts in this group by assigning the same “being-vulnerable” probability to every host in this group. That is, we do not differentiate hosts in determining which hosts are vulnerable. In fact, the value of the probability is equivalent to \( p(i, NV(i)) \).

To measure the estimation error under various characteristics of the sample, we assume that the sufficient number of Bots are deployed over every target group. Also, we assume that every target system replies for each scan sent out by the Bots with the probability 1.0. Contrast to these assumptions, to measure the estimation error under various characteristics of the Botnet, we assume that m(i) Bots among NVi Bots scan a group i. Each of m(i) Bots scans a target group i with scan rate s(i) and collects n(i) samples from each target group. Also, to measure the estimation error under the characteristic of the target space, we assume that a target system replies for a scan sent out by one Bot with the probability, p(i).

6.2. Experiments under different number of independent trials

We measure the estimation errors by increasing the number of independent trials (simulations) from 1 to 300 in a step-wise manner with size 50 in average. Here, the estimation errors are averaged over NVi independent trials. The simulation results estimated on Type 1 distribution are shown in Fig. 4, where x-axis is the number of independent trials (NVi) and y-axis is the expected estimation error, in terms of MSE. The results in Fig. 4 show that the estimation errors are bounded as NVi approaches 100. Also, it is shown that when the routable address population (NV = 2^{24} or 2^{25}) is given, the estimation errors are bounded regardless of the vulnerable-host population (NVi) as NVi approaches 100. Based on these observations, the estimation errors in the other experiments are averaged over 100 independent trials of group sample size n(i).

6.3. Experiments under different sample sizes

By increasing group sample size (n(i)) from 33 to 1369 in a step-wise manner with size 200 on the average, we measure the estimation errors. Here, n(i) = 33 and n(i) = 1369 are the required group sample sizes obtained from Eq. (14) by assuming that the attacker determines the best estimate of the proportion of vulnerable hosts in each group population as \( (NV_i/NV)/% \) and is willing to accept the error rate of 5% \( e(p(i, NV(i))) = 0.05 \) and 1% \( e(p(i, NV(i))) = 0.01 \) for each experiment. Also, it is assumed that the attacker wants \( p(i, NV(i)) \) to be 95% and 99% certain that his or her finding does not differ from the true proportion by more than 5% (x = 0.05, i.e. \( z_{0.05} = 1.96 \)) and 1% (x = 0.01, i.e. \( z_{0.01} = 2.58 \)) for each experiment. We note that the sample size on the target space is \( \sum_{i=1}^{NV} n(i) \). By assuming that each group has the same routable IP addresses, the simulation results are shown based on different group sample sizes, i.e. the sample size on the target space ranges from 8,448 to 350,464.
as that the estimation error decreases as
expected estimation error, in terms of
A vulnerable-host group distribution table, where parameters are predefined in Table 1. Here,
Table 2

![Table 2](image)

Fig. 5 shows the simulation results estimated on Type 1 distribution. Here, x-axis is group sample size \( n(i) \) and y-axis is the expected estimation error, in terms of \( ESE \). The results in Fig. 5 show that the estimation error decreases as \( n(i) \) increases and is bounded by \( ESE \) and \( \alpha \). In Fig. 5, it is shown that the absolute errors from the sampling ratio 2.089% to 0.05% range from 0.01% to 0.05%. Especially, when the sampling ratio approaches 0.6%, the good-enough sampling ratio is shown, i.e. 0.01%. These observations show that under the small sample size, the proposed estimation method can show the good estimation for the actual vulnerable-host group distribution. Also, based on Type 2 distribution, \( ESE \) ranges from 4.8E–9 to 6E–10 for each \( N_v \) from 180,000 to 720,000. The expected absolute error ranges from 2.55E–3 to 8.98E–4 for the different vulnerable-host populations from 180,000 to 720,000, and the most relative estimation errors \( \alpha \) are less than 2%. Here, the sampling ratio \( r_s \) is 2.089% (although not small compared to 0.05% or 0.6%). These experiment results show that the proposed estimation method is not sensitive to the number (or percentage) of vulnerable hosts in each group.
6.4. Experiments under different numbers of Bots

We note that the larger $N_B$ is needed, the more cost the attacker may spend. For example, since Bots use random scanning, as the number of Bots increases, the probability of non-overlapping samples also increases. To measure the estimation errors under different number of Bots, we vary the Bot population in the Botnet from 500 to 4000 in a step-wise manner with size 500. We assume that 360,000 vulnerable hosts are in the target space, each Bot scans a target group with scan rate, 50 scans/s, and a target system replies for a scan sent out by one Bot with the probability, 0.3. The simulation results in Fig. 7, where x-axis is the number of Bots in the Botnet ($N_B$) and y-axis is the expected estimation error, in terms of MSE, show that the estimation error decreases as $N_B$ increases. Also, it is shown that when the Bots are uniformly deployed over every target group, the estimation errors are smaller than when the Bots are non-uniformly deployed on the groups, i.e. $N(N_B/256, N_B \times (2/3))$. This observation shows that by uniformly deploying $N_B$ Bots on every target groups and by increasing $N_B$, the estimation error will decrease.

6.5. Experiments under different scan rates

We measure the estimation errors by varying scan rates of the Bots. We vary the scan rate from 5 to 50 in a step-wise manner
with size 5. We assume that 360,000 vulnerable hosts are in the target space, 2000 Bots are deployed on the target space, and a target system replies for a scan sent out by one Bot with the probability 0.3. The simulation results are shown in Fig. 8, where x-axis is the scan rate of each Bot which scans a target group \( i(s_b(i)) \) and y-axis is the expected estimation error, in terms of \( \text{MSE} \). The results in Fig. 8 show that as the scan rate increases, the estimation error decreases. Also, compared to the non-uniformly (normally) distributed Bots over the target groups, the uniformly distributed Bots obtain the smaller estimation error. This observation shows that by increasing scan rate and by uniformly deploying Bots over every target group, we can decrease the estimation error effectively.

6.6. Experiments under different probabilities that a target system replies for a scan

We need to consider how the proposed estimation method would be affected by current countermeasures such as a F/W and a honeypot against sampling. Specifically, we note that scans of Bots can be stealthy or not depending on a certain threshold of possible countermeasure. Thus, we measure the estimation errors by varying the probabilities that a target system replies for a scan sent out by a Bot. We vary the probability from 0.1 to 1.0 in a stepwise manner with size 0.1. We assume that 360,000 vulnerable hosts are in the target space, 2000 Bots are deployed on the target space, and each Bot scans a target group \( i \) with the low scan rate, 10 scans/s. The simulation results are shown in Fig. 9, where x-axis is the probability \( p_s \) and y-axis is the expected estimation error, in terms of \( \text{MSE} \). The results in Fig. 9 show that the higher \( p_s \) is, the smaller estimation error is.

7. Discussion

From analytical models and their validation results, it is shown that the proposed estimation method can get sufficiently accurate estimations with the small sampling ratio and the low scan rate. Now, we discuss about limitation of the study on Botnet-nurtured worms and the possible countermeasures.

7.1. Influence of IP address dynamics

Our study on Botnet-nurtured worms may be limited by the influence of IP address dynamics such as those of Dynamic Host Configuration Protocol (DHCP) and Network Address Translation (NAT). That is, by the influence of IP address dynamics, the estimated distribution may be biased toward the actual distribution of vulnerable hosts on the Internet. In this paper, we do not deal with the influence of such IP dynamics on the estimated distribution specifically. However, since such IP dynamics may affect to scans of Bots which are not included in a target group, the attacker can reduce the influence of IP address dynamics by increasing scans of Bots in the target group.

7.2. Possible countermeasures against Botnet-nurtured worms

As Botnet-nurtured worms estimate the distribution of vulnerable hosts using Botnet, their countermeasures can be analyzed based on the characteristics of Botnet.

We focus on the fact that this type of importance-scanning worms use the centralized C&C server to collect the side information used as the input for estimating the vulnerable-host distribution. Even though many Botnets are still designed using this way since centralized C&C structures are effective in process of designing many Botnets, this architecture suffers from the single-point-of-failure problem. That is, if the malicious behavior of C&C servers such as the Web server and the IRC chat server are detected and prevented before or while they command and control the Bots, the Botnet loses its C&C structure, which cannot be used to collect the side information. Many approaches are designed for detecting Botnets that use IRC-based or HTTP-based C&C servers [36–40].

However, as Bots can allow a system to be remotely controlled via private secure channels such as IRC encrypted communication channels, it is still difficult to detect and prevent the malicious behavior of C&C servers. For example, in case of IRC-based Botnet, Bot can connect to the IRC server via a secure SSL connection and then, the aforementioned detection approaches will not identify what is passing between Bot and the IRC server with a low false positive rate. Also, since the Botnet can change its C&C server address, e.g., using fast-flux service networks [41], such detection approaches may become ineffective.

On the other hand, to prevent Botnet-nurtured worms, the defender may hide the hosts of a live network segment within a population of seemingly live phantom addresses, which are called white holes [42]. By increasing the difficulty that legitimate hosts
can be targeted, the successful hiding of live hosts prevents the Botnet from accurately measuring the address distribution of the network segment. However, C&C servers may fingerprint the existence of white holes by observing that almost all IP/ports in its network segment are responsive to connect attempts, which can be a direct indicator of a potential probe monitoring system.

These observations indicate that C&C servers are important for both attackers and defenders. Thus, to detect and prevent Botnet-nurtured worms effectively, it is important to analyze their behavior in the current and new Botnets, whose analysis is outside the scope of this paper.

8. Conclusion

It is well-known that importance scanning worms can increase their propagation speed by exploiting a non-uniform distribution of vulnerable hosts. However, before worms propagating, no simple methodology that the attacker can estimate the non-uniform distribution of vulnerable hosts is known yet. In this work, by analyzing importance scanning worms proposed by Zesheng Chen and Chuanyi Ji [10–13], we show that even a self-learning worm using importance scanning is still not fast as expected. As a method for estimating the non-uniform distribution of vulnerable hosts to quickly spread across the Internet, we propose a new type of importance scanning worm called the Botnet-nurtured. By using the Botnet, the proposed Botnet-nurtured worm collects the required information for estimating the non-uniform distribution of vulnerable hosts in a gradual, very low-speed, scan-activity-concealing manner. As this learning stage is performed before the worm propagates, the proposed Botnet-nurtured worm can use the estimated non-uniform distribution of vulnerable hosts while maintaining a simple worm propagation mechanism. From analytical model and the validation experiments, it is shown that the estimated vulnerable-host group distribution is unbiased towards the actual vulnerable-host group distribution with a small sample size. In many cases, it is shown that the good-enough sampling ratio is as small as 0.6%. Thus, we believe that the proposed method based on statistical sampling and estimation theory is practical.

Acknowledgments

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government [NRF-2009-352-D00284], the IT R&D program of MKE/IITA [2008-F-034-02, Development of Security-Quality Guarantee Technology in Resilient Networks], NSF CNS-0716479, AFOSR MURI: Autonomic Recovery of Enterprise-wide Systems after Attack or Failure with Forward Correction, AFRL award FA8750-08-C-0137, ARO MURI: Computer-aided Human Centric Cyber Situation Awareness, and NSF CNS-0905131.

Appendix A. A vulnerable-host group distribution table

Table 2 shows a vulnerable-host group distribution table, each line of which consists of five values: group index \( i \); the actual vulnerable-host group distribution on 256 groups \( p(i) \); the absolute error in estimating \( p(i) \) by \( \hat{p}(i)(e(i)) \); and the relative error in estimating \( p(i) \) by \( \hat{p}(i)(e(i)) \). Here, the digits in Table 2 are shown up to the 3 digits to the right of the point. Among the parameters in each line, the attacker estimates \( p(i) \) by \( \hat{p}(i) \).

Appendix B. A real vulnerable-host group distribution: DShield data

By aggregating firewall and intrusion detection system logs from networks throughout the global Internet, DShield.org analyzed source distribution of activity observed at each site over all ports during the seven days. To evaluate our proposed MLE estimator, we use a real distribution of vulnerable hosts in /8 subnets provided by DShield.org [35] (file name: p0.7days.dshield slash8). In Fig. 10, the real distribution of vulnerable hosts is based on the web-server (port 80) distribution and the proportion of vulnerable hosts in a target group \( i \) over the Internet is formed as

\[
p(i) = \frac{\text{number of addresses with the first byte equal to } i}{\text{total number of collected addresses}},
\]

where \( i = 1, 2, \ldots, 256 \).

References


