Mr. Uriti Suresh, Mr. M.V.A. Naidu, Prof. D.S. Sharma / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 4, July-August 2012, pp.478-485

Spectral Based Detection of Smart Worms

Mr. Uriti Suresh[1] Mr. M.V.A. Naidu[2] Prof. D.S. Sharma[3]

[1]Student, Department of CSE, Sri Sivani College of Engineering, Chilakapalem-532001
[2] Asst.Professor, Department of CSE, Sri Sivani College of Engineering, Chilakapalem-532001
[3]Assoc.Professor & HOD, Department of CSE, Sri Sivani College of Engineering, Chilakapalem-532001

ABSTRACT—

The easy access and wide usage of the Internet makes it a prime target for malicious activity. In particular, the Internet has become a powerful mechanism for propagating malicious software programs designed to annoy (e.g., deface web pages), spread misinformation (e.g., false news reports or stock quotes), deny service (e.g., corrupt hard disks), steal financial information (e.g. credit card numbers), enable remote login (e.g., Trojan horses), etc. Smart worms cause most important security threats to the Internet. This is due to the ability of Smart worms spread in an automated fashion and can flood the Internet in a very short time. Smart worms develop during their propagation and thus create great challenges to defend against them. In this paper, we look into "Detection of Smart Worms". The Smart Worms are different from traditional worms because of its ability to intelligently manipulate its scan traffic volume over time. Thereby, we analyze characteristics of the Smart Worms and conduct a comprehensive comparison between its traffic and non-worm traffic (background traffic). Motivated by our observations, we design a novel spectrum-based scheme to detect the Smart Worm. Our scheme uses the Power Spectral Density (PSD) distribution of the scan traffic volume and its corresponding Spectral Flatness Measure (SFM) to distinguish the Smart Worm traffic from background traffic. Using a comprehensive set of detection metrics and real-world traces as background traffic, we conduct extensive performance evaluations on our proposed spectrum-based detection scheme. The performance data clearly demonstrates that our scheme can effectively detect the Smart Worm propagation. Furthermore, we show the generality of our spectrum-based scheme in effectively detecting not only the Smart Worm, but traditional worms as well.

Index Terms—Worm, Camouflage, Spectrum Based Detection, Smart Worm.

1 INTRODUCTION

The easy access and wide usage of the Internet makes it a prime target for malicious activity. In particular, the Internet has become a powerful mechanism for propagating malicious software programs designed to annoy (e.g., deface web pages), spread misinformation (e.g., false news reports or stock quotes), deny service (e.g., corrupt hard disks), steal financial information (e.g. credit card numbers), enable remote login (e.g., Trojan horses), etc. The two most popular ways to spread such malicious software are commonly referred to as worms (like the Code Red) and email viruses (like the infamous Melissa and Love Bug). However it is increasingly difficult to distinguish malicious software programs using these terms. For example, the recent Nimda attack was especially vicious because it combined both attack methods. Active worms have been a persistent security threat on the Internet since the Morris worm arose in 1988. The Code Red and Nimda worms infected hundreds of thousands of systems, and cost both the public and private sectors millions of dollars [1], [2]. Active worms propagate by infecting computer systems and by using infected computers to spread the worms in an automated fashion. Active worms can potentially spread across the Internet within seconds. It is therefore of great importance to characterize and monitor the spread of active worms, and be able to derive methods to effectively defending our systems against them. An active worm refers to a malicious software program that propagates itself on the Internet to infect other computers. Many real-world worms have caused important damage on the Internet. These worms include "Code-Red" worm in 2001 [1], "Slammer" worm in 2003 [2], and "Witty"/"Sasser" worms in 2004 [3]. Many active worms are used to infect a large number of computers and recruit them as bots or zombies, which are networked together to form botnets [4]. These botnets can be used to: (a) launch massive Distributed Denial-of-Service (*DDoS*) attacks that interrupt the Internet utilities [5], (b) access confidential information that can be misused [6] through large scale traffic sniffing, key logging, identity theft etc, (c) destroy data that has a high monetary value [7], and (d) distribute large-scale unwanted advertisement emails (as spam) or software

(as malware). Due to the substantial damage caused by worms in the past years, there have been significant efforts on developing detection and defense mechanisms against worms. A network based worm detection system plays a major role by monitoring, collecting, and analyzing the scan traffic (messages to identify vulnerable computers) generated during worm attacks. In this system, the detection is commonly based on the self propagating behavior of worms that can be described as follows: after a worm-infected computer identifies and infects a vulnerable computer on the Internet, this newly infected computer will automatically and continuously scan several IP addresses to identify and infect other vulnerable computers. As such, numerous existing detection schemes are based on a unstated assumption that each worm-infected computer keeps scanning the Internet and propagates itself at the highest possible speed. Furthermore, it has been shown that the worm scan traffic volume and the number of worm-infected computers exhibit exponentially increasing patterns [2], [11], [12], [13], [14]. Nevertheless, the attackers are crafting attack strategies that intend to defeat existing worm detection systems. In particular, 'stealth' is one attack strategy used by a recently-discovered active worm called "Atak" worm [15] and the "self-stopping" worm [16] avoid detection by hibernating (i.e., stop propagating) with a pre-determined period. Worm might also use the ambiguous scan [17] and traffic morphing technique to hide the detection [18]. This worm attempts to remain hidden by sleeping (suspending scans) when it suspects it is under detection. Worms that adopt such smart attack strategies could exhibit overall scan traffic patterns different from those of traditional worms. Since the existing worm detection schemes will not be able to detect such scan traffic patterns, it is very important to understand such smart-worms and develop new countermeasures to defend against them. In this paper, we conduct a systematic study on smart-worms. The Smart Worms have a self-propagating behavior similar to traditional worms, i.e., they intend to rapidly infect as many vulnerable computers as possible. However, the Smart Worms are quite different from traditional worms in which it camouflages any noticeable trends in the number of infected computers over time. The camouflage is achieved by manipulating the scan traffic volume of worm-infected computers.

Such a manipulation of the scan traffic volume prevents exhibition of any exponentially increasing trends or even crossing of thresholds that are tracked by existing detection schemes [19], [20], [21]. We note that the propagation controlling nature of the Smart Worm cause a slow down in the propagation speed. However, by carefully controlling its scan rate, the Smart Worms can: (a) still achieve their ultimate goal of infecting as many computers as possible before being detected, and (b) position them self to launch subsequent attacks [4], [5], [6], [7]. Based on the observations, we adopt frequency domain analysis techniques and develop a detection scheme against wide-spreading of the Smart Worms. Particularly, we develop a novel spectrum-based detection scheme that uses the *Power Spectral Density* (*PSD*) distribution of scan traffic volume in the frequency domain and its corresponding *Spectral Flatness Measure* (*SFM*) to distinguish the Smart Worm traffic from nonworm traffic (background traffic). Our frequency domain analysis studies use the real-world Internet traffic traces (Shield logs dataset) provided by SANs Internet Storm Center (*ISC*) [22], [23]2. Our results reveal that non-worm traffic (e.g., port-scan traffic for port 80, 135 and 8080) has relatively larger *SFM* values for their *PSD* distributions. Whereas, the Smart Worm traffic shows comparatively smaller *SFM* value for its respective *PSD* distribution.

Furthermore, We define several new metrics. *Maximal Infection Ratio* (*MIR*) is the one to quantify the infection damage caused by a worm before being detected. Other metrics include *Detection Time* (*DT*) and *Detection Rate* (*DR*). Our evaluation data clearly demonstrate that our spectrum-based detection scheme achieves much better detection performance against the Smart Worm propagation compared with existing detection schemes. Our evaluation also shows that our spectrum-based detection scheme is general enough to be used for effective detection of traditional worms as well. The remainder of the paper is organized as follows. In Section 2, we introduce the background and review the related work. In Section 3, we introduce the propagation model of the Smart Worm. We present our spectrum-based detection scheme against the Smart Worm in Section 4. We conclude this paper in Section 5.

2 BACKGROUND AND RELATED WORK

2.1 Active Worms

Active worms are similar to biological viruses in terms of their infectious and self-propagating nature. They identify vulnerable computers, infect them and the worm-infected computers propagate the infection further to other vulnerable computers. In order to understand worm behavior, we first need to model it. With this understanding, effective detection and defense schemes could be developed to mitigate the impact of the worms. For this reason, tremendous research effort has focused on this area [12], [24], [14], [25], [16]. Active worms use various scan mechanisms to propagate themselves efficiently. The basic form of active worms can be categorized as having the Pure Random Scan (PRS) nature. In the PRS form, a worm-infected computer continuously scans a set of random Internet IP addresses to find new vulnerable computers. Other worms propagate themselves more

effectively than PRS worms using various methods, e.g., network port scanning, email, file sharing, Peer-to-Peer (P2P) networks, and Instant Messaging (IM) [26], [27]. In addition, worms use different scan strategies during different stages of propagation. In order to increase propagation efficiency, they use a local network or hit list to infect previously identified vulnerable computers at the initial stage of propagation [12], [28]. They may also use DNS, network topology and routing information to identify active computers instead of randomly scanning IP addresses [11], [21], [27], [29]. They split the target IP address space during propagation in order to avoid duplicate scans [21]. Li *et al.* [30] studied a divide-conquer scanning technique that could potentially spread faster and stealthier than a traditional random-scanning worm. Ha *et al.* [31] formulated the problem of finding a fast and resilient propagation topology and propagation schedule for Flash worms. Yang *et al.* [32] studied the worm propagation over the sensor networks. Different from the above worms, which attempt to accelerate the propagation with new scan schemes, the Smart worms studied in this paper aims to avoid the detection by the worm defense system during worm propagation. Active worms that are polymorphic [33], [34] in nature. Polymorphic worms are able to change their binary representation or signature as part of their propagation process. This can be achieved with self-encryption mechanisms or semantics preserving code manipulation techniques. The Smart Worm also shares some similarity with stealthy port-scan attacks. Such attacks try to find out available services in a target system, while avoiding detection [35], [36]. It is accomplished by decreasing the port scan rate, hiding the origin of attackers, etc. Due to the nature of self-propagation, the C-Worm must use more complex mechanisms to manipulate the scan traffic volume over time in order to avoid detection.

2.2 Worm Detection

Worm detection has been intensively studied in the past and can be generally classified into two categories: "host-based" detection and "network-based" detection. Host-based detection systems detect worms by monitoring, collecting, and analyzing worm behaviors on end-hosts. Since worms are malicious programs that execute on these computers, analyzing the behavior of worm executables plays an important role in host based detection systems. Many detection schemes fall under this category [37], [38]. In contrast, network-based detection systems detect worms primarily by monitoring, collecting, and analyzing the scan traffic (messages to identify vulnerable computers) generated by worm attacks. Many detection schemes fall under this category [19], [20], [21], [39], [40]. Ideally, security vulnerabilities must be prevented to begin with, a problem which must addressed by the programming language community. However, while vulnerabilities exist and pose threats of large-scale damage, it is critical to also focus on network-based detection, as this paper does, to detect wide spreading worms. In order to rapidly and accurately detect Internet-wide large scale propagation of active worms, it is imperative to monitor and analyze the traffic in multiple locations over the Internet to detect suspicious traffic generated by worms. The widely adopted worm detection framework consists of multiple distributed monitors and a worm detection center that controls the former [23], [41]. This framework is well adopted and similar to other existing worm detection systems, such as the Cyber center for disease controller [11], Internet motion sensor [42], SANS ISC (Internet Storm Center) [23], Internet sink [41], and network telescope [43]. The monitors are distributed across the Internet and can be deployed at end hosts, router, or firewalls etc. Each monitor passively records irregular port-scan traffic, such as connection attempts to a range of void IP addresses (IP addresses not being used) and restricted service ports. Periodically, the monitors send traffic logs to the detection center. The detection center analyzes the traffic logs and determines whether or not there are suspicious scans to restricted ports or to invalid IP addresses. Network-based detection schemes commonly analyze the collected scanning traffic data by applying certain decision rules for detecting the worm propagation. For example, Venkataraman *et al.* and Wu *et al.* in [20], [21] proposed schemes to examine statistics of scan traffic volume, Zou *et al.* presented a trend-based detection scheme to examine the exponential increase pattern of scan traffic [19], Lakhina *et al.* in [40] proposed schemes to examine other features of scan traffic, such as the distribution of destination addresses. Other works study worms that attempt to take on new patterns to avoid detection [39]. Besides the above detection schemes that are based on the global scan traffic monitor by detecting traffic anomalous behavior, there are other worm detection and defense schemes such as sequential hypothesis testing for detecting worm-infected computers [44], payload-based worm signature detection [34], [45]. In addition, Cai *et al.* in [46] presented both theoretical modeling and experimental results on a collaborative worm signature generation system that employs distributed fingerprint filtering and aggregation and multiple edge networks. Dantu *et al.* in [47] presented a state-space feedback control model that detects and control the spread of these viruses or worms by measuring the velocity of the number of new connections an infected computer makes. Despite the different approaches described above, we believe that detecting widely scanning anomaly behavior continues to be a useful weapon against worms, and that in practice multifaceted defence has advantages.

3 PROPOSED MODEL OF THE SMART WORM

3.1 Smart Worm

When an active worm is fired into the Internet, it simultaneously scans many machines in an attempt to find a vulnerable machine to infect. When it finally finds its Victim, it sends out a probe to infect the target. If successful, a copy of this worm is transferred to this new host. This new host then begins running the worm and tries to infect other machines. When an invulnerable machine or an unused IP address is reached, the worm poses no threat. During the worm's spreading process, some machines might stop functioning properly, forcing the users to reboot these computers or at least kill some of the processes that may have been exploited by the worm. Then these infected machines become vulnerable machines again, and are still inclined to further infection. When the worm is detected, people will try to slow it down or stop it. A patch, which repairs the security hole of the machines, is used to defend against worms. When an infected or vulnerable machine is patched, it becomes an invulnerable machine. To speed up the spread of active worms, Weaver presented the "hitlist" idea [10]. Long before an attacker releases the worm, he/she gathers a list of potentially vulnerable machines with good network connections. After the worm has been fired onto an initial machine on this list, it begins scanning down the list. Hence, the worm will first start infecting the machines on this list. Once this list has been exhausted, the worm will then start infecting other vulnerable machines. The machines on this list are referred to as the "hitlist". After the worm infects the hitlist rapidly, it uses these infected machines as "stepping stones" to search for other vulnerable machines. In this paper we do not consider the amount of time it takes a worm to infect the hitlist since the hitlist can be acquired well before a worm is released and be infected in a very short period of time. There are several different scanning mechanisms that active worms employ, such as random, local subnet, permutation and topological scanning [5]. In this paper we focus on two mechanisms, random scanning and local subnet scanning. In random scanning, it is assumed that every computer in the Internet is just as likely to infect or be infected by other computers. Such a network can be pictured as a fully-connected graph in which the nodes represent computers and the arcs represent connections (neighboring-relationships) between pairs of nodes. This topology is called "homogeneous mixing" in the theoretical epidemiology [7]. In local subnet scanning, computers also connect to each other directly, forming "homogeneous mixing". However, instead of selecting targets randomly, the worms preferentially scan for hosts on the "local" address space. For example, the Nimda worm selects target IP addresses as follows:

- 50% of the time, an address with the same first two octets will be chosen.
- 25% of the time, an address with the same first octet will be chosen.
- 25% of the time, a random address will be chosen.

The Smart Worm camouflages its propagation by controlling scan traffic volume during its propagation. The simplest way to manipulate scan traffic volume is to randomly change the number of worm instances conducting port-scans. In order to effectively avoid detection, the overall scan traffic for the Smart Worm should be comparatively slow and variant enough to not show any notable increasing trends over time. On the other hand, a very slow propagation of the Smart Worm is also not desirable, since it delays rapid infection damage to the Internet. Hence, the Smart Worm needs to adjust its propagation so that it is neither too fast to be easily detected, nor too slow to delay rapid damage on the Internet. To regulate the Smart Worm scan traffic volume, we introduce a control parameter called *attack probability* P(t) for each worm-infected computer. P(t) is the probability that a Smart Worm instance participates in the worm propagation (i.e., scans and infects other computers) at time t. Our Smart Worm model with the control parameter P(t) is general. P(t) = 1 represents the cases for traditional worms, where all worm instances actively participate in the propagation. For the Smart Worm, P(t) needs not be a constant value and can be set as a time varying function. In order to achieve its camouflaging behavior, the C-Worm needs to obtain an appropriate P(t) to manipulate its scan traffic. Specifically, the Smart Worm will regulate its overall scan traffic volume such that: (a) it is similar to non-worm scan traffic in terms of the scan traffic volume over time, (b) it does not exhibit any notable trends, such as an exponentially increasing pattern or any mono-increasing pattern even when the number of infected hosts increases (exponentially) over time, and (c) the average value of the overall scan traffic volume is sufficient to make the Smart Worm propagate fast enough to cause rapid damage on the Internet. We assume that a worm attacker intends to manipulate scan traffic volume so that the number of worm instances participating in the worm propagation follow a random distribution with mean $\overline{Mc}$ This $\overline{Mc}$ can be regulated in a random fashion during worm propagation in order to camouflage the propagation of Smart Worm. Correspondingly, the worm instances need to adjust their attack probability P(t) in order to ensure that the total number of worm instances launching the scans is approximately $\overline{Mc}$. To regulate $\overline{Mc}$, it is obvious that P(t) must be decreased over time since M(t) keeps increasing during the worm

propagation. We can express P(t) using a simple function as follows: $\boldsymbol{P(t) = min\left(\frac{\overline{Mc}}{\overline{M}(t)}, 1\right)}$, where $\overline{M}$(t) represents the estimation of M(t) at time t. From the above expression, we know that the Smart Worm needs to obtain the value of $\overline{M}$(t) (as close to M(t) as possible) in order to generate an effective P(t). Here, we discuss one approach for the Smart Worm to estimate M(t). The basic idea is as follows: A Smart Worm could estimate the percentage of computers that have already been infected over the total number of IP addresses as well as M(t), through checking a scan attempt as a *new hit* (i.e., hitting an uninfected vulnerable computer) or a *duplicate hit* (i.e., hitting an already infected vulnerable computer). This method requires each worm instance (i.e., infected computer) to be marked indicating that this computer has been infected. Thus, when a worm instance (for example, computer *A*) scans one infected computer (for example, computer *B*), then computer *A* will detect such a mark, thereby becoming aware that computer *B* has been infected. Through validating such marks during the propagation, a Smart Worm infected computer can estimate M(t).

3.2 Propagation Model of the SMART Worm

To analyze the Smart Worm, we adopt the epidemic dynamic model for disease propagation, which has been extensively used for worm propagation modeling [2], [12]. Based on existing results [2], [12], this model matches the dynamics of real worm propagation over the Internet quite well. For this reason, similar to other publications, we adopt this model in our paper as well. Since our investigated Smart Worm is a novel attack, we modified the original Epidemic dynamic formula to model the propagation of the Smart Worm by introducing the P(t) - the attack probability that a worm-infected computer participates in worm propagation at time t. We note that there is a wide scope to notably improve our modified model in the future to reflect several characteristics that are relevant in real-world practice. Particularly, the epidemic dynamic model assumes that any given computer is in one of the following states: immune, vulnerable, or infected. An immune computer is one that cannot be infected by a worm; a vulnerable computer is one that has the potential of being infected by a worm; an infected computer is one that has been infected by a worm. The simple epidemic model for a finite population of traditional PRS worms can be expressed as,

$$\frac{\boldsymbol{dM(t)}}{\boldsymbol{dt}} = \boldsymbol{\beta.M(t).[N - M(t)]} \quad \text{(Formula1)}$$

where M(t) is the number of infected computers at time t; N(= T · P1 · P2) is the number of vulnerable computers on the Internet; T is the total number of IP addresses on the Internet; P1 is the ratio of the total number of computers on the Internet over T ; P2 is the ratio of total number of *vulnerable* computers on the Internet over the total number of computers on the Internet; **β = S/V** is called the pair wise infection rate; S is the scan rate defined as the number of scans that an infected computer can launch in a given time interval. We assume that at t = 0, there are M(0) computers being initially infected and N−M(0) computers being susceptible to further worm infection. The Smart Worm has a different propagation model compared to traditional PRS worms because of its P(t) parameter. Consequently, Formula (1) needs to be rewritten as,

$$\frac{\boldsymbol{dM(t)}}{\boldsymbol{dt}} = \boldsymbol{\beta.M(t).P(t).[N - M(t)]}$$

Assuming that $\overline{M(t)}$ = (1+ε) ·M(t), where ε is the estimation error, the Formula (2) can be rewritten as,

$$\frac{\boldsymbol{dM(t)}}{\boldsymbol{dt}} = \boldsymbol{\beta.}\frac{\overline{\boldsymbol{Mc}}}{\boldsymbol{1+\varepsilon(t)}}\boldsymbol{.[N - M(t)]}$$

With Formula (3), we can derive the propagation model
for the Smart Worm as

$$\boldsymbol{M(t) = N - e^{\beta.\frac{\overline{Mc}}{1+\varepsilon(t)}.t\,(N-M(0))}}$$

where M(0) is the number of infected computers at time
0.

4. DETECTING THE SMART WORM

In this section, we develop a novel *spectrum-based detection scheme*. Our detection scheme captures the distinct pattern of the Smart Worm in the frequency domain, and thereby has the potential of effectively detecting the Smart Worm propagation. In order to identify the Smart Worm propagation in the frequency domain, we use the distribution of *Power Spectral Density* (*PSD*) and its corresponding *Spectral Flatness Measure* (*SFM*) of the scan traffic. Particularly, *PSD* describes how the power of a time series is distributed in the frequency domain. Mathematically, it is defined as the *Fourier* transform of the auto-correlation of a time series. In our case, the time series corresponds to the changes in the number of worm instances that actively conduct scans over time. The *SFM* of *PSD* is defined as the ratio of *geometric mean* to *arithmetic mean* of the coefficients of *PSD*. The range of *SFM* values is [0, 1] and a larger *SFM* value implies flatter *PSD* distribution and vice versa.

To illustrate *SFM* values of both the Smart Worm and normal non-worm scan traffic, we plot the *Probability Density Function* (PDF) of *SFM* for both C-Worm and normal non-worm scan traffic as shown in Fig. 3 and Fig. 4, respectively. The normal non-worm scan traffic data shown in Fig. 1 is based on real-world traces collected by the *ISC*. Note that we only show the data for port 8080 as an example, and other ports show

similar observations. From this figure, we know that the *SFM* value for normal non-worm traffic is very small (e.g., *SFM* ∈ (0.02, 0.04) has much higher density compared with other magnitudes). The Smart Worm data shown in Fig. 2 is based on 800 C-Worms attacks generated by varying attack parameters defined in Section 3 such as P(t) and Mc(t). From this figure, we know that the *SFM* value of the Smart Worm attacks is high (e.g., *SFM* ∈ 0.5, 0.6 has high density). From the above two figures, we can observe that there is a clear separation range of *SFM* ∈ (0.3, 0.38) between the Smart Worm and normal non-worm scan traffic. As such, the *SFM* can be used to sensitively detect the Smart Worm scan traffic. The large *SFM* values of normal non-worm scan traffic can be explained as follows. The normal non-worm scan traffic does not tend to concentrate at any particular frequency since its random dynamics is not caused by any recurring phenomenon. The small value of *SFM* can be reasoned by the fact that the power of Smart Worm scan traffic is within a narrow-band frequency range. Such concentration within a *narrow* range of frequencies is unavoidable since the Smart Worm adapts to the dynamics of the Internet in a recurring manner for manipulating the overall scan traffic volume. In reality,
the above recurring manipulations involve steady increase followed by a decrease in the scan traffic volume. Notice that the frequency domain analysis will require more samples in comparison with the time domain analysis, since the frequency domain analysis technique such as the Fourier transform, needs to derive power spectrum amplitude for different frequencies. In order to generate the accurate spectrum amplitude for relatively high frequencies, a high granularity of data sampling will be required. In our case, we rely on Internet threat monitoring (ITM) systems to collect traffic traces from monitors (motion sensors) in a timely manner. As a matter of fact, other existing detection schemes based on the scan traffic rate [20], variance [21] or trend [19] will also demand a high sampling frequency for ITM systems in order to accurately detect worm attacks. Enabling the ITM system with timely data collection will benefit worm detection in real-time.

4.1 Spectrum-based Detection Scheme

We now present the details of our spectrum-based detection scheme. Similar to other detection schemes [19], [21], we use a "destination count" as the number of the unique destination IP addresses targeted by launched scans during worm propagation. To understand how the destination count data is obtained, we recall that an ITM system collects logs from distributed monitors across the Internet. On a side note, Internet Threat Monitoring (ITM) systems are a widely deployed facility to detect, analyze, and characterize dangerous Internet threats such as worms. In general, an ITM system consists of one centralized data center and a number of monitors distributed across the Internet. Each monitor records traffic that addressed to a range of IP addresses (which are not commonly used IP address also called the dark IP addresses) and periodically sends the traffic logs to the data center. The data center then analyzes the collected traffic LOGS and publishes reports (e.g., statistics of monitored traffic) to ITM system users. Therefore the baseline traffic in our study is scan traffic. With reports in a sampling window Ws, the source count X(t) is obtained by counting the unique source IP addresses in received logs. To conduct spectrum analysis, we consider a detection sliding window Wd in the worm detection system. Wd consists of q (> 1) continuous detection sampling windows and each sampling window lasts Ws. The detection sampling window is the unit time interval to sample the detection data (e.g., the destination count). Hence, at time i, within a sliding window Wd, there are q samples denoted by $(X(i-q-1), X(i-q-2), \ldots, X(i))$, where $X(i-j-1)$ $(j \in (1, q))$ is the j-th destination count from time $i-j-1$ to $i-j$.

4.1.1 Detection Decision Rule

We now describe the method of applying an appropriate
detection rule to detect Smart Worm propagation. As the *SFM* value can be used to sensitively distinguish the Smart Worm and normal non-worm scan traffic, the worm detection is performed by comparing the *SFM* with a predefined threshold Tr. If the *SFM* value is smaller than a predefined threshold Tr, then a C-Worm propagation alert is generated. The value of the threshold Tr used by the Smart Worm detection can be fittingly set based on the knowledge of statistical distribution (e.g., *PDF*) of *SFM* values that correspond to the non-worm scan traffic. Notice that the Tr value for the non-worm traffic can be derived by analyzing the historical data provided by SANs Internet Storm Center (ISC). In the worm detection systems, monitors collect port-scan traffic to certain area of dark IP addresses and periodically reports scan traffic log to the data center. Then the data center aggregates the data from different monitors on the same port and publishes the data. Based on the historical data for different ports, we can build the statistical profiles of port-scan traffic on different ports and then derive the Tr value for the non-worm traffic. Based on the continuous reported data, the value of Tr will be tuned and adaptively used to carry out worm detection. If we can obtain the *PDF* of *SFM* values for the Smart Worm through comprehensive simulations and even real-world profiled data in the future, the optimal threshold can be obtained by applying the Bayes classification [65]. If the *PDF* of *SFM* values for the Smart Worm is not available, based on the *PDF* of *SFM* values of the normal non-worm scan traffic, we

can set an appropriate Tr value. For example, the Tr value can be determined by the Chebyshev inequality [65] in order to obtain a reasonable false positive rate for worm detection. Hence in Section 5, we evaluate our spectrum-based detection scheme against the C-Worm on two cases: (a) the PDF of *SFM* values are known for both the normal non-worm scan traffic and the C-Worm scan traffic, (b) the PDF of *SFM* values is only known for the normal non-worm scan traffic. Notice that even if the Smart Worm monitors the port-scan traffic report, it will be hard for the Smart Worm to make the SFM similar to the background traffic. This can be reasoned by two factors. First, the low value of SFM is mainly caused by the closed-loop control nature of Smart worm. The concentration within a narrow range of frequencies is unavoidable since the Smart Worm adapts to the dynamics of the Internet in a recurring manner for manipulating the overall scan traffic volume. Based on our analysis, the non-worm traffic on a port is rather random and its SFM has a flat pattern. That means that the non-worm traffic on the port distributes similar power across different frequencies. Second, as we indicated in other responses, without introducing the closed-loop control, it will be difficult for the attacker to hide the irregularity of worm propagation traffic in the time domain. When the worm attacks incorporate the closed-loop control mechanism to camouflage their traffic, it will expose a relative small value of SFM. Hence, integrating our spectrum-based detection with existing traffic rate-based anomaly detection in the time domain, we can force the worm attacker into a dilemma: if the worm attacker does not use the closed-loop control, the existing traffic rate-based detection scheme will be able to detect the worm; if the worm attacker adopt the closed-loop control, it will cause the relatively small SFM due to the process of closed-loop control. This makes the worm attack to be detected by our spectrum-based scheme along with other existing traffic-rate based detection schemes.

5 FINAL REMARKS

In this paper, we studied a new class of worms called Smart Worm, which has the capability to camouflage its propagation and further avoid the detection. Our investigation showed that, by deploying network monitor system in the entire network the Smart worm can be defended. Based on observation, we developed a novel spectrum-based detection scheme to detect the Smart Worm. This paper lays the foundation for ongoing studies of "smart" worms that intelligently adapt their propagation patterns to reduce the effectiveness of countermeasures.

6 REFERENCES

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

2. D. Moore, V. Paxson, and S. Savage, "Inside the slammer worm," in *IEEE Magazine of Security and Privacy*, July 2003.

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

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

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

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

7. Charles Wright, Scott Coull, and Fabian Monrose, "Traffic morphing: An efficient defense against statistical traffic analysis," in *Proceedings of the 15th IEEE Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Febrary 2008.

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

9. Y. Shinoda, K. Ikai, and M. Itoh, "Vulnerabilities of passive internet threat monitors," in *Proceedings of the 14-th USNIX Security Symposium*, Baltimore, MD, July-August 2005.

10. X. Wang, W. Yu, X. Fu, D. Xuan, and W. Zhao, "iloc: An invisible localization attack to internet threat monitoring systems," in *Proceedings of the 27th IEEE International Conference on Computer Communications*

(INFOCOM) Mini-conference, Phoenix, AZ, April 2008.

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

12. John Bethencourt, Dawn Song, and Brent Waters, "Analysis-resistant malware," in *Proceedings of the 15th IEEE Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Febrary 2008.

13. Monirul Sharif, Jonathon Giffin, Wenke Lee, and Andrea Lanzi, "Impeding malware analysis using conditional code obfuscation," in *Proceedings of the 15th IEEE Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, Febrary 2008.

14. Yubin Li, Zesheng Chen, and Chao Chen, "Understanding divideconquer-scanning worms," in *Proceedings of International Performance Computing and Communications Conference (IPCCC)*, Austin, TX, December 2008.

15. C. C. Zou, D. Towsley, and W. Gong, "Modeling and simulation study of the propagation and defense of internet e-mail worm," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 2, pp.105–118, 2007.

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

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

18. M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson, "The internet motion sensor: A distributed blackhole monitoring system," in *Proceedings of the 12-th IEEE Network and Distributed Systems Security Symposium (NDSS)*, San Diego, CA, February 2005.

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

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

21. X. Wang, W. Yu, A. Champion, X. Fu, and D. Xuan, "Detecting worms via mining dynamic program execution," in *Proceedings of IEEE International Conference on Security and Privacy in Communication Networks (SECURECOMM)*, Nice, France, September 2007.