ISSN: 0975-5662, Dec, 2013 www.ijrct.org

Detection of Camouflaging Worm(C-worm)

#B.Supriya[1]M.Tech CSE (Software engineering),e-mail:supriya.barigela@gmail.com
#M.S.B.Prudhviraj[2]**,** Asst.Professor, e -mail: phridviraj@gmail.com
#Kakatiya Institute of Technology & Science, Warangal, A.P, INDIA

Abstract

Active worms pose major security threats to the Internet. This is due to the ability of active worms to propagate in an automated fashion as they continuously compromise computers on the Internet. Active worms evolve during their propagation, and thus, pose great challenges to defend against them. In this paper, we investigate a new class of active worms, referred to as Camouflaging Worm (C-Worm in short). The C-Worm is different from traditional worms because of its ability to intelligently manipulate its scan traffic volume over time. Thereby, the C-Worm camouflages its propagation from existing worm detection systems based on analyzing the propagation traffic generated by worms. We analyze characteristics of the C-Worm and conduct a comprehensive comparison between its traffic and nonworm traffic (background traffic). We observe that these two types of traffic are barely distinguishable in the time domain. However, their distinction is clear in the frequency domain, due to the recurring manipulative nature of the C-Worm. Motivated by our observations, we design a novel spectrum-based scheme to detect the C-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 C-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 C- Worm propagation. Furthermore, we show the generality of our spectrum-based scheme in effectively detecting not only the C-Worm, but traditional worms as well.

Index Terms—Worm, camouflage, anomaly detection.

1 INTRODUCTION

An active worm refers to a malicious software program that propagates itself on the Internet to infect other computers. The propagation of the worm is based on exploiting vulnerabilities of computers on the Internet. Many real-world worms have caused notable 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: 1. launch massive Distributed Denial-of-Service (DDoS) attacks that disrupt the Internet utilities [5],

2. access confidential information that can be misused [6] through large-scale traffic sniffing, key logging, identity theft, etc., 3. destroy data that has a high monetary value [7], and 4. distribute large-scale unsolicited advertisement emails (as spam) or software (as malware). There is evidence showing that infected computers are being rented out as "Botnets" for creating an entire black-market industry for renting, trading, and managing "owned" computers, leading to economic incentives for attackers [4], [8], [9]. Researchers also showed possibility of "superbotnets," networks of independent botnets that can be coordinated for attacks of unprecedented scale [10]. For an adversary, superbotnets would also be extremely versatile and resistant to countermeasures. 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 computer1 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 tacit 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] circumvent detection by hibernating (i.e., stop propagating) with a predetermined period. Worm might also use the evasive 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 a new class of such smart-worms denoted as Camouflaging Worm (C-Worm in short). The C-Worm has a self-propagating behavior similar to traditional worms, i.e., it intends to rapidly infect as many vulnerable computers as possible. However, the C-Worm is 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 worminfected 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 C-Worm (and similar smart-worms, such as "Atak") cause a slow down in the propagation speed. However, by carefully controlling its scan rate, the C-Worm can: 1) still achieve its ultimate goal

of infecting as many computers as possible before being detected, and 2) position itself to launch subsequent attacks [4], [5], [6], [7].We comprehensively analyze the propagation model of the C-Worm and corresponding scan traffic in both time and frequency domains. We observe that although the C-Worm scan traffic shows no noticeable trends in the time domain, it demonstrates a distinct pattern in the frequency domain. Specifically, there is an obvious concentration within a narrow range of frequencies. This concentration within a narrow range of frequencies is inevitable, since the C-Worm adapts to the dynamics of the Internet in a recurring manner for manipulating and controlling its overall scan traffic volume. The above recurring manipulations involve steady increase, followed by a decrease in the scan traffic volume, such that the changes do not manifest as any trends in the time domain or such that the scan traffic volume does not cross thresholds that could reveal the C-Worm propagation. Based on the above observation, we adopt frequency domain analysis techniques and develop a detection scheme against wide spreading of the C-Worm. 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 C-Worm traffic from non worm traffic (background traffic). Our frequency-domain analysis studies use the real-world Internet traffic traces (Shield logs data set) provided by SANs Internet Storm Center (ISC) [22], [23].2 Our results reveal that nonworm traffic (e.g., port-scan traffic for port 80, 135, and 8080) has relatively larger SFM values for their PSD distributions. Whereas, the C-Worm traffic shows comparatively smaller SFM value for its respective PSD distribution. Furthermore, we demonstrate the effectiveness of our spectrum-based detection scheme in comparison with existing worm-detection schemes. 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 C-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 C-Worm. We present our spectrum-based detection scheme against the C-Worm in Section 4. The performance evaluation results of our spectrum-based detection scheme is provided in Section 5. We conclude this paper in Section 6.

2 BACKGROUND AND RELATED WORK

2.1 Active Worms

Active worms are similar to biological viruses in terms of their infectous 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 hitlist 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 and Ngo [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 C-Worm studied in this paper aims to elude the detection by the worm defense system during worm propagation. Closely related, but orthogonal to our work, are the evolved 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 C-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 selfpropagation, 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: “hostbased” detection and “network-based” detection. Hostbased 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 [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. [20] andWuet al. [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. [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] and payload-based worm signature detection [34], [45]. In addition, Cai et al. [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. [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 defense has advantages.

3 MODELING OF THE C-WORM

3.1 C-Worm

The C-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. As other alternatives, a worm attacker may use an open loop control (no feedback) mechanism by choosing a randomized and time-related pattern for the scanning and infection in order to avoid being detected. Nevertheless, the open-loop control approach raises some issues of the invisibility of the attack. First, as we know, worm propagation over the Internet can be considered a dynamic system. When an attacker launches worm propagation, it is very challenging for the attacker to know the accurate parameters for worm propagation dynamics over the Internet. Given the inaccurate knowledge of worm propagation over the Internet, the open-loop control system will not be able to stabilize the scan traffic. This is a known

result from control system theory [48]. Consequently, the overall worm scan traffic volume in the open-loop control system will expose a much higher probability to show an increasing trend with the progress of worm propagation. As more and more computers get infected, they, in turn, take part in scanning other computers. Hence, we consider the C-worm as a worst case attacking scenario that uses a closed-loop control for regulating the propagation speed based on the feedback propagation status. In order to effectively evade detection, the overall scan traffic for the C-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 C-Worm is also not desirable, since it delays rapid infection damage to the Internet. Hence, the C-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 C-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 C-Worm instance participates in the worm propagation (i.e., scans and infects other computers) at time t. Our C-Worm model with the control parameter P(t)is generic . P(t)=1 represents the cases for traditional worms, where all worm instances actively participate in the propagation. For the C-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 C-Worm will regulate its overall scan traffic volume such that: 1) it is similar to nonworm scan traffic in terms of the scan traffic volume over time, 2) 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 3) the average value of the overall scan traffic volume is sufficient to make the C-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 $\bar{M}_C$.This $\bar{M}_C$ can be regulated in a random fashion during worm propagation in order to camouflage the propagation of C-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 $\bar{M}_C$. To regulate $\bar{M}_C$, it is obvious that P(t) must be decreased over time, since M(t k)eeps increasing during the worm propagation. We can express P(t) using a simple function as P(t)= $\min(\frac{\bar{M}_C}{\bar{M}(t)}, 1)$,where $\bar{M}(t)$ represents the estimation of M(t) at time t.
From the above expression, we know that the C-Worm needs to obtain the value of $\bar{M}(t)$ (as close to M(t) as possible) in order to generate an effective P(t). Here, we discuss one approach for the C-Worm to estimate M(t). The basic idea is as follows: A C-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 C-Worm infected computer can estimate M(t). Appendix A discusses one alternative how the C-Worm could estimate M(t) to obtain $\bar{M}(t)$ as the propagation proceeds. There are other approaches to achieve this goal, such as incorporating the Peer-to-Peer techniques to disseminate information through secured IRC channels [49], [50].

3.2 Propagation Model of the C-Worm

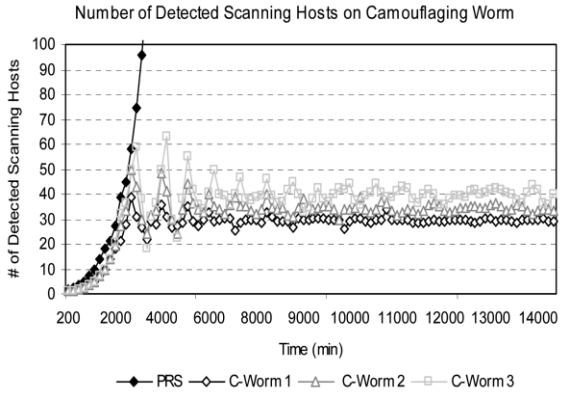
To analyze the C-Worm, we adopt the epidemic dynamic model for disease propagation, which has been extensively used for worm propagation modeling [2], [12]. Based on existing results [2], [12], this model matches the dynamics of real-worm propagation over the Internet quite well. For this reason, similar to other publications, we adopt this model in our paper as well. Since our investigated C-Worm is a novel attack, we modified the original epidemic dynamic formula to model the propagation of the C-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

$$dM(t)/dt=\beta.M(t).[N-M(t)], \quad (1)$$

where M(t) is the number of infected computers at time t; $N(=T.P1.P_2)$ is the number of vulnerable computers on the Internet; T is the total number of IP addresses on the Internet;

P_1 is the ratio of the total number of computers on the Internet over T; P_2 is the ratio of total number of vulnerable computers on the Internet over the total number of computers on the Internet; $\beta=S/V$ is called the pair wise infection rate [51]; and 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 C-Worm has a different propagation model compared to traditional PRS worms because of its P(t) parameter. Consequently, (1) needs to be rewritten as

$$dM(t)/dt=\beta.M(t).P(t).[N-M(t)], \quad 2$$



Observed infected instance number for the C-Worm and PRS worms

ISSN: 0975-5662, Dec, 2013 www.ijrct.org

Recall that P(t)= $\frac{\bar{M}_C}{\bar{M}(t)}$, $\bar{M}(t)$ is the estimation of M(t) at time t,and assuming the $\bar{M}(t)$=(1+ ϵ). M(t),where ϵ is estimation error, the (2) is rewritten as

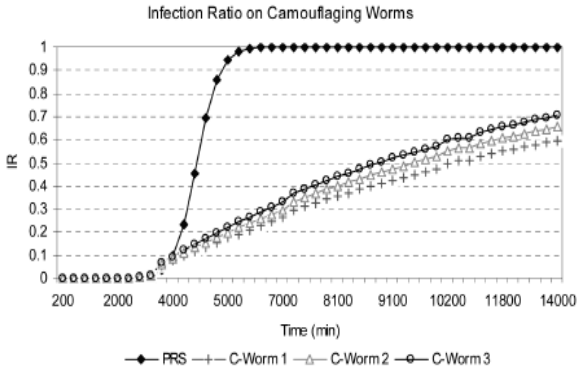
dM(t)/dt =β. $\bar{M}(t)$ /1+ $\epsilon(t)$.[N-M(t)]. (3)

With (3), we can derive the propagation model for the C-Worm as M(t)= N -e $^{\frac{\beta \cdot \bar{M}_C}{1+\epsilon}(t) \cdot t}$(N-M(0)),where M(0) is the number of infected computers at time 0. Assume that the worm detection system can monitor P_m($P_m \in$[0,1]) of the whole Internet IP address space. Without loss of generality, the probability that at least one scan from a worm-infected computer (it generates S scans in unit time on average) will be observed by the detection system is 1-(1-P_m)$^{P(t).S}$.We define that $M_A(t)$ is the number of worm instances that have been observed by the worm detection system at time t, then there are M(t) –M_A(t) unobserved infected instances at time t. At the worm propagation early stage, M(t)-M_A(t) $\simeq$ M(t). The expected number of newly observed infected instances at t + δ _ (where δ is the interval of monitoring) is (M(t)-M_A(t)).[1-(1-P_m)$^{P(t).S}$] $\simeq$ M(i) [1-(1-P_m)$^{P(t).S}$].Thus, we have M_A(t+ δ)= M_A(t)+ M(t) [1-(1-P_m)$^{P(t).S}$]. Using simple mathematical manipulations, the number of worm instances observed by the worm detection system at time t is

$M_A(t)=P(t).M(t).P_m=(Pm. \bar{M}_C)/(1+\epsilon(t))$ (4)

3.3 Effectiveness of the C-Worm

We now demonstrate the effectiveness of the C-Worm in evading worm detection through controlling P(t). Given random selection of $\bar{M}_c$, we generate three C-Worm attacks (viz., C-Worm 1, C-Worm 2, and C-Worm 3) that are characterized by different selections of mean and variance magnitudes for $\bar{M}_c$. In our simulations, we assume that the scan rate of the traditional PRS worm follow a normal distribution S_n =N(40,40) (note that if the scan rate generated by above distribution is less than 0, we set the scan rate as 0). We also set the total number of vulnerable computers on the Internet as 360,000, which is the total number of infected computers in "Code-Red" worm incident [1].

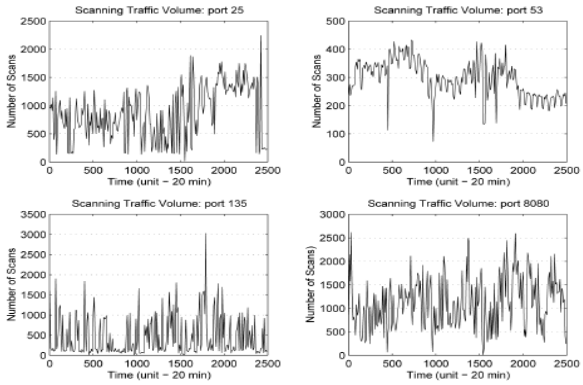


Infected ratio for the c-worm an PRS worm

Fig. 1 shows the observed number of worm-infected computers over time for the PRS worm and the above three C-Worm attacks. Fig. 2 shows the infection ratio (IR) for the PRS worm and the above three C-Worm attacks. These simulations are for a worm detection system discussed in Section 2.2 that covers a 2^{20} IPv4 address space on the Internet. The reason for choosing 220 IP addresses as the coverage space of the worm detection system is due to the fact that the SANs ISC, a representative Internet threat monitoring (ITM) system, has similar coverage space [23]. In the ITM systems, a large number of monitors are commonly deployed all over the Internet and each monitor collects the traffic directed to a small set of IP address spaces, which are not commonly used (also called dark IP addresses). Therefore, the address space of ITM system is

not a narrow range address space, rather a large number of small chunks of addresses randomly spread across the global IP address space.

For the C-Worm, the trend of observed number of worm instances over time (MAðtÞ) (defined in (4)) is much different from that of the traditional PRS worm, as shown in Fig. 2. This clearly demonstrates how the C-Worm successfully camouflages its increase in the number of worm instances (MAðtÞ) and avoids detection by worm detection systems that expect exponential increases in worm instance numbers during large-scale worm propagation. Fig. 3 shows the number of scanning computers from normal nonworm port-scanning traffic (background traffic) for several well-known ports, (i.e., 25, 53, 135, and 8080) obtained over several months by the ISC. Comparing Fig. 3 YU ET AL.: MODELING AND DETECTION OF CAMOUFLAGING WORM 5 Fig. 1. Observed infected instance number for the C-Worm and PRS worm.
Fig. 2. Infected ratio for the C-Worm and PRS worm. Fig. 3.



Infected ratio for the C-Worm and PRS worm.

with Fig. 1, we can observe that it is hard to distinguish the C-Worm port traffic from background port-scanning traffic in the time domain.From above Figs. 1 and 2,wealso observe that the C-Worm is still able to maintain a certain magnitude of scan traffic so as to cause significant infection on the Internet. As a note regarding the speed of C-Worm propagation, we can observe from Fig. 1 that the C-Worm takes approximately 10 days to infect 75 percent of total vulnerable hosts in comparison with the 3.3 days taken by a PRS worm.5 Hence, the C-Worm could potentially adjust its propagation speed such that it is still effective in causing wide-spreading propagation, while avoiding being detected by the worm detection schemes.

We discussed the "Atak" worm in Section 1 and mentioned that it is similar to the C-Worm, since it tries to avoid being detected, when it suspects that it is being detected by antiworm software. However, it differs from the C-Worm in its behavior. The "Atak" worm attempts to hide only during times it suspects its propagation will be detected by antiworm software. Whereas, the C-Worm proactively camouflages itself at all times. In addition, the "Self-stopping" worm attempts to hide by co-ordinating with its members to halt propagation activity only after the vulnerable population is subverted [16]. This behavior leaves enough evidence for worm detection systems to recognize its propagation. The C-Worm, on the other hand, hides itself even during its propagation, and thus, keeps the worm detection schemes completely unaware of its propagation. The C-Worm also has some similarity in spirit with polymorphic worms that manipulate the byte stream of worm payload in order to avoid the detection of signature (payload)-based detection scheme [33], [34]. The manipulation of worm payload can be achieved by various mechanisms:
1. inter leaving meaningful instructions with NOP (no operation),

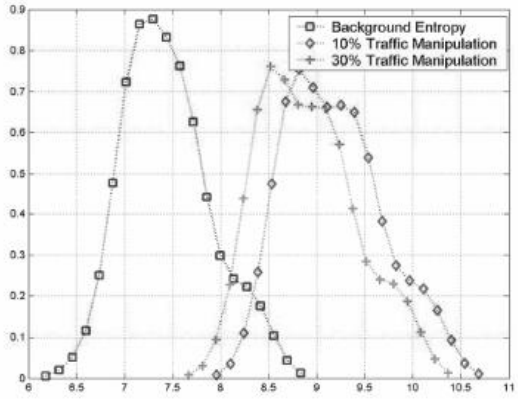
ISSN: 0975-5662, Dec, 2013 www.ijrct.org

2. using different instructions to achieve the same results,
3. shuffling the register set in each worm propagation program code copy, and
4. using cryptography mechanisms to change worm payload signature with every infection attempt [33], [34].

In contrast, the C-Worm tries to manipulate the scan traffic pattern to avoid detection.

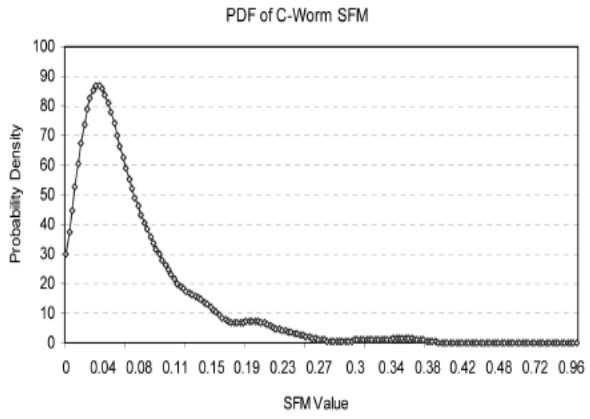
3.4 Discussion

In this paper, we focus on a new class of worms, referred to as the C-Worm. The C-Worm adapts their propagation traffic patterns in order to reduce the probability of detection, and to eventually infect more computers. The C-Worm is different from polymorphic worms that deliberately change their payload signatures during propagation [34], [52]. For example, MetaPHOR [53] and Zmist [54] worms intensively metamorphose their payload signature to hide themselves from detection schemes that rely on expensive packet payload analysis. Bethencourt et al. [55] studied the worm, which employs private information retrieval techniques to find and retrieve specific pieces of sensitive information from compromised computers while hiding its search criteria. Sharif et al. [56] presented an obfuscation-based technique that automatically conceals specific condition dependent malicious behavior from virus detectors that have no prior knowledge of program inputs. Popov et al. [57] investigated a technique that allows the worm programs to be obfuscated by changing many control transfers into signals (traps) and inserting dummy control transfers and "junk" instructions after the signals. The resulting code can significantly reduce the chance to be detected. Recent studies also showed that existing commercial antiworm detection systems fail to detect brand new worms and can also be easily circumvented by worms that use simple mutation techniques to manipulate their payload [58]. Although, in this paper, we only demonstrate effectiveness of the C-Worm against existing traffic volume-based detection schemes, the design principle of the C-Worm can be extended to defeat other newly developed detection schemes, such as destination distribution-based detection [39], [40]. In the following, we discuss this preliminary concept. Recall that the attack target distribution-based schemes analyze the distribution of attack targets (the scanned destination IP addresses) as basic detection data to capture the fundamental features of worm propagation, i.e., they continuously scan different targets, which is not the expected behavior of nonworm scan traffic. However, our initial investigation shows that the worm attacker is still able to defeat such a countermeasure via manipulating the attack target distribution. For example, the attacker may launch a portion of scan traffic bound for some IP addresses monitored by ITM system. Recall that those dedicated IP addresses monitored by ITM system can be obtained via probing attacks or other means



Manipulation of attack target distribution entropy.

[59], [60], [61]. Using port 135 reported by SANs ISC as an example, we analyze the traces and obtain the traffic target distribution in a window lasting 10 min. Following existing work [39], [40], we use entropy as the metric to measure the attack target distribution. Fig. 4 shows the Probability Density Function (PDF) of background traffic's entropy values. We also simulate the worm propagation traffic, which allocates a portion of scan traffic bound for IP addresses monitored by the ITM system. Following this, we obtain the PDF of the entropy value for combined traffic including both worm propagation and background traffic. From Fig. 4, we know that when the attacker uses a portion of attack traffic to manipulate the target distribution, the entropy-based



PDF of SFM on C-Worm traffic.

detection scheme can degrade significantly. For example, when the attacker uses 10 percent traffic to manipulate the traffic's entropy value, the false positive rate of entropybased detection scheme is 14 percent. When the attacker uses 30 percent traffic to manipulate the traffic's entropy value, the false positive rate becomes 40 percent. Hence, in order to preserve the performance, entropy-based detection scheme needs to evolve correspondingly and integrate with other detection schemes. We will perform a more detailed study of this aspect in our future work.

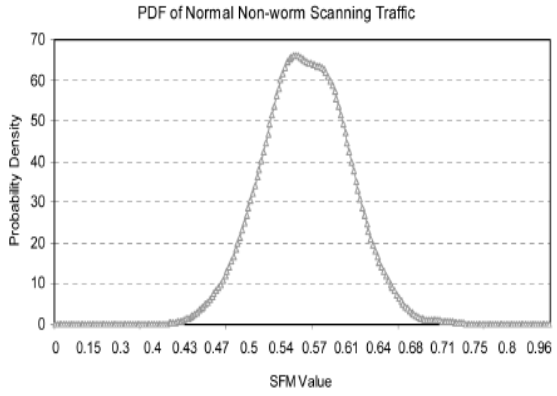
4 DETECTING THE C-WORM

4.1 Design Rationale

In this section, we develop a novel spectrum-based detection scheme. Recall that the C-Worm goes undetected by detection schemes that try to determine the worm propagation only in the time domain. Our detection scheme captures the distinct pattern of the C-Worm in the frequency domain, and thereby has the potential of effectively detecting the C-Worm propagation. In order to identify the C-Worm propagation in the frequency domain, we use the distribution of PSD and its corresponding 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 autocorrelation 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 C-Worm and normal nonworm scan traffic, we plot the PDF of SFM for both C-Worm and normal nonworm scan traffic, as shown in Figs. 5 and 6, respectively. The normal nonworm scan traffic data shown in Fig. 6 is based on real-world traces collected by the ISC[6] .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

nonworm traffic is very small (e.g., SFM ∈ (0.02, 0.04) has much higher density compared with other magnitudes). The C-Worm data shown in Fig. 5 is based on 800 C-Worms's attacks generated by varying attack parameters defined in



PDF of SFM on normal nonworm traffic.

Section 3, such as P(t) and $M_c(t)$. From this figure, we know that the SFM value of the C-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 demarcation range of SFM ∈0.3, 0.38 between the C-Worm and normal non worm scan traffic. As such, the SFM can be used to sensitively detect the C-Worm scan traffic.

The large SFM values of normal nonworm scan traffic can be explained as follows: The normal nonworm 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 C-Worm scan traffic is within a narrowband frequency range. Such concentration within a narrow range of frequencies is unavoidable, since the C-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 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.

4.2 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, 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),…..,X(i)),where X(i-j-1) (j ∈ (1,q)) is the jth destination count from the i-j-1 to i-j.
In our spectrum-based detection scheme, the distribution of PSD and its corresponding SFM are used to distinguish the C-Worm scan traffic from the nonworm scan traffic. Recall that the definition of PSD distribution and its corresponding SFM are introduced in Section 4.1. In our worm detection scheme, the detection data (e.g., destination counter), is further processed in order to obtain its PSD and SFM. In the following, we detail how the PSD and SFM are determined during the processing of the detection data.

4.2.1 Power Spectral Density

To obtain the PSD distribution for worm detection data, we need to transform data from the time domain into the frequency domain. To do so, we use a random process X(t),t ∈[0,n] to model the worm detection data. Assuming X(t) is the source count in time period [t-1.t] (t ∈[1,n],we define the autocorrelation of X(t) by

$$R_X(L)=E[X(t)X(t+L). \quad (5)$$

In (5), $R_X(L)$ is the correlation of worm detection data in an interval L. If a recurring behavior exists, a Fourier transform of the autocorrelation function of $R_X(L)$ can reveal such behavior. Thus, the PSD function (also represented by $S_{X(}f)$, where f refers to frequency) of the scan traffic data is determined using the Discrete Fourier Transform (DFT) of its autocorrelation function as follows:

$$\psi(R_X[L],K)=\sum_{n=0}^{N-1}(R_X[L])\cdot e^{-j2\pi Kn/N}, \quad (6)$$

where K =0,1,2,3. . .,N -1.As the PSD inherently captures any recurring pattern in the frequency domain, the PSD function shows a comparatively even distribution across a wide spectrum range for the normal nonworm scan traffic. The PSD of C-Worm scan traffic shows spikes or noticeably higher concentrations at a certain range of the spectrum.

4.2.2 Spectral Flatness Measure

We measure the flatness of PSD to distinguish the scan traffic of the C-Worm from the normal nonworm scan traffic. For this, we introduce the SFM, which can capture anomaly behavior in certain range of frequencies. The SFM is defined as the ratio of the geometric mean to the arithmetic mean of the PSD coefficients [62], [63]. It can be expressed as

$$SFM=\frac{[\prod_{k=1}^{n}S(f_k)]^{\frac{1}{n}}}{\frac{1}{n}\sum_{k=1}^{n}S(f_k)}, \quad (7)$$

where $S(f_k)$is an PSD coefficient for the PSD obtained from the results in (6). SFM is a widely existing measure for

discriminating frequencies in various applications, such as voiced frame detection in speech recognition [63], [64]. In general, small values of SFM imply the concentration of data at narrow frequency spectrum ranges. Note that the C-Worm has unpreventable recurring behavior in its scan traffic; consequently its SFM values are comparatively smaller than the SFM values of normal nonworm scan traffic. To be useful in detecting C-Worms, we introduce a sliding window to capture a noticeably higher concentrations at a small range of spectrum. When such noticeably concentration is recognized, we derive the SFM within a wider frequency range. From Fig. 5, we can observe that the SFM value for the C-Worm is very small (e.g., with a mean value of approximately 0.075). A formal analysis of SFM for the C-Worm is presented in the Appendix B.

4.2.3 Detection Decision Rule

We now describe the method of applying an appropriate detection rule to detect C-Worm propagation. As the SFM value can be used to sensitively distinguish the C-Worm and normal nonworm 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 C-Worm detectioncan be fittingly set based on the knowledge of statistical distribution (e.g., PDF) of SFM values that correspond to the nonworm scan traffic. Notice that the Tr value for the nonworm traffic can be derived by analyzing the historical data provided by SANs 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 nonworm 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 C-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 C-Worm is not available, based on the PDF of SFM values of the normal nonworm 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

TABLE 1
Evaluation Metrics

Notation	Definition
Infection Ratio (IR)	Ratio of worm-infection over time without the presence of detection/defense system
Maximal Infection Ratio (MIR)	Ratio of worm infection at the moment that worm is being detected
Detection time (DT)	Time taken to successfully detect a wide-spreading worm from its birth

two cases: 1) the PDF of SFM values are known for both the normal nonworm scan traffic and the C-Worm scan traffic, and 2) the PDF of SFM values is only known for the normal nonworm scan traffic.

In addition, our spectrum-based scheme is also generic for detecting the PRS worms. This is due to the fact that propagation traffic of PRS worms has an exponentially increasing pattern. Thus, in the propagation traffic of PRS worms, the PSD values in the low-frequency range are much higher compared with other frequency ranges. A

formal analysis of SFM for the PRS worm is presented in Appendix C.

Notice that even if the C-Worm monitors the port-scan traffic report, it will be hard for the C-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 C-worm. The concentration within a narrow range of frequencies is unavoidable, since the C-Worm adapts to the dynamics of the Internet in a recurring manner for manipulating the overall scan traffic volume. Based on our analysis, the nonworm traffic on a port is rather random and its SFM has a flat pattern. This means that thenonwormtraffic 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,wecan force the wormattacker 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 PERFORMANCE EVALUATION

In this section, we report our evaluation results that illustrate the effectiveness of our spectrum-based detection scheme against both the C-Worm and the PRS worm in comparison with existing representative detection schemes for detecting wide-spreading worms. In addition, we also take into consideration destination-distribution-based detection schemes and evaluate their performance against the C-Worm.

5.1 Evaluation Methodology

5.1.1 Evaluation Metrics

In order to evaluate the performance of any given detection scheme against the C-Worm, we use the following three metrics listed in Table 2. The first metric is the worm IR, which is defined as the ratio of the number of infected computers to the total number of vulnerable computers, assuming there is no worm detection/defense system in place. The other two metrics are the DT and the MIR. DT is defined as the time taken to successfully detect a wide spreading worm from the moment the worm propagation starts. It quantifies the detection speed of a detection scheme.

MIR defines the ratio of an infected computer number over the total number of vulnerable computers up to the moment when the worm spreading is detected. It quantifies the damage caused by a worm before being detected. The objective of any detection scheme is to minimize the damage caused by a rapid worm propagation. Hence, MIR and DT can be used to quantify the effectiveness of any worm detection scheme. The higher the values, the more effective the worm attack and the less effective the detection. In addition, we use two more metrics—Detection Rate (P_D) and False Positive Rate (P_F). The P_D is defined as the probability that a detection scheme can correctly identify a worm attack. The P_F is defined as the probability that a detection

scheme mistakenly identifies a nonexistent worm attack.

5.1.2 Simulation Setup

In our evaluation, we considered both experiments with real world "nonworm" traffic and simulated C-Worm traffic. To make our experiments reflect real-world practice, some key parameters that we used to generate C-worm traffic in our simulation were based on previous results from a real-worm incidence—"Code-Red" worm in 2001 [1]. Specifically, we set the total number of vulnerable computers on the Internet

TABLE 2
Detection Results for the C-Worm

Schemes	VAR	TREND	MEAN	SPEC(W)	SPEC
Detection Rate (DR)	48%	0%	14%	96.4%	99.3%
Maximal Infection Ratio (MIR)	14.4%	100%	7.5%	4.4%	2.8%
Detection Time (DT) in minutes	2367	∞	1838	1707	1460

as 360,000, which is the maximum number of computers, which could be infected by "Code-Red" worm. Additionally, we set the scan rate S (number of scans per minute) to be variable within a range, this allows us to emulate the infected computers in different network environments. In our evaluation, the scan rates are predetermined and follow a Gaussian distribution S=N(S_m , S_σ^2),where Sm and S_σ^2 are in [(20, 70)], similar to those used in [19]. In our evaluation, we merged the simulated C-worm attack traffic into replayed "nonworm" traffic traces and carried out evaluation study.

We simulate the C-Worm attacks by varying the attack parameters, such as attack probability P(t) and the number of worm instances participating in the scan $(\bar{M}_C)$ defined in Section 3. The $\bar{M}_C$ follows the Gaussian distribution N(m σ) and are changed dynamically by the C-Worm during its propagation. Particularly, for N(m σ) , m is randomly selected in (12,000, 75,000) and σ is randomly selected in (0.2, 100). We simulate different C-Worm attacks by varying the values of m and σ . The detection sampling window Ws is set to 5 min and the detection sliding window Wd is set to be incremental from 80 to 800 min. The incremental selection of Ws from a comparatively small window to a large window can adaptively reflect the worm scan traffic dynamics caused by the C-Worm propagation at various speeds. We choose the setting of the detection sampling window to be short enough in order to provide enough sampling accuracy as prescribed by Nyquist's sampling theory. Also, we choose the detection sliding window to be long enough to capture adequate information for spectrum-based analysis [63].

In practice, since detection systems analyze port scanv traffic blended with the nonworm scan traffic, we replay the real-world traces as nonworm scan traffic (background noise to attack traffic) in our simulations. In particular, we used the ISC real-world trace (Shield logs data set) from 01/01/2005 to 01/15/2005. Note that SANs ISC, maintained by the SANs Institute, have gained popularity among the Internet security community in recent years. ISC collects firewall and Intrusion detection system logs, which indicate port-scan trends from approximately 2,000 organizations that monitor up to one million IP addresses. We choose the scan traffic logs for port 8080, as an example, for profiling the non-worm scan traffic.

In order to provide the creditability of such data, we did the following effort before using the data in our experiments. First, we

had the 15 days traces from 01/01/2005 to 01/15/2005 provided by SANs ISC. We checked with the SANs Website and found that there were no worm attack incidents within those 15 days. Second, we obtained the statistical profile of traffic traces, including the mean value and standard deviation of traffic rates. Based on the statistical profile, we set a threshold, which is the summary of mean value and four times that of the standard derivation, and filtered out some data, which had unusual large values. Third, we conducted our evaluation 15 times based on data randomly combined with different dates. The results, we showed in the paper, are the mean values of experimental results from different rounds.

5.2 Performance of Detection Schemes

We evaluate our proposed spectrum-based detection scheme by comparing its performance with three existing representative traffic volume-based detection schemes. The first scheme is the volume mean-based (MEAN) detection scheme, which uses mean of scan traffic to detect worm propagation [20]; the second scheme is the trend-based (TREND) detection scheme, which uses the increasing trend of scan traffic to detect worm propagation [19]; and the third scheme is the victim number variance-based (VAR) detection scheme, which uses the variance of the scan traffic to detect worm propagation [21]. We define our spectrum-based detection scheme as SPEC. We evaluate two types of SPEC: One has no knowledge of any C-Worm attacks or C-Worm scan traffic (denoted by SPEC(W)) and the other has knowledge of C-Worm attacks through an offline training process (denoted by SPEC). For the offline training, we use 1,000 worm attacks that include both the C-Worm (800 C-Worm attacks) and PRS worms (200 PRS worm attacks). For fairness, we set the detection parameters for our SPEC scheme and the other three detection schemes, so that all detection schemes achieve a similar false positive rate (PF) below 1 percent. In the following sections, we first evaluate the performance of our spectrum-based detection scheme in the context of detecting C-Worm attacks. We then evaluate the performance of our spectrum-based detection scheme in the context of detecting traditional PRS worms, followed by performance comparison between traffic-volume-based detection and traffic-distribution-based detection.

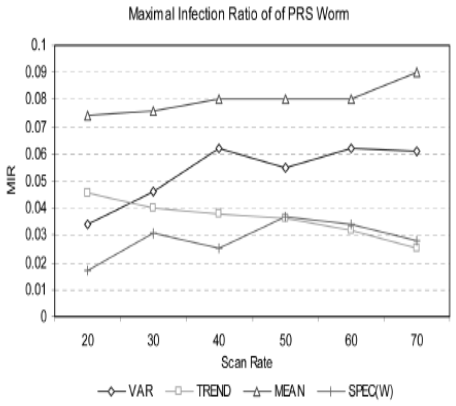
5.2.1 Detection Performance for C-Worm Attacks

Table 2 shows the detection results of different detection schemes against the C-Worm. The results have been averaged over 500 C-Worm attacks. From this table, we can observe that existing detection schemes are not able to effectively detect the C-Worm and their detection rate (PD) values are significantly lower in comparison with our spectrum-based detection schemes (SPEC and SPEC(W)). For example, SPEC achieves the detection rate of 99 percent, which is at least 3-4 times more accurate than detection schemes, such as VAR and MEAN that achieve detection rate values of only 48 percent and 14 percent, respectively. Our SPEC and SPEC(W) detection schemes also achieve good DT performance in addition to the high detection rate values indicated above. In contrast, the detection time of existing detection schemes have relatively larger values. As a consequence of the detection time values, we can see that the C-Worm propagation is effectively contained by SPEC and SPEC(W), as demonstrated by the lower values of MIR for the SPEC and SPEC(W). Since the

detection rate values for the existing detection schemes are relatively small, obtaining low values of MIR for those schemes are not as significant as those for SPEC and SPEC(W). Furthermore, we can notice that the detection performance of the SPEC(W) is worse than the SPEC. This is because the SPEC(W) lacks offline training knowledge for the C-Worm scan traffic. Nonetheless, the SPEC(W) still performs much better than existing detection schemes.

5.2.2 Detection Performance for Traditional PRS Worms

We evaluate the detection performance of different detection schemes for traditional PRS worm attacks. The detection performance results have been averaged over 500 PRS worm



Maximal infection ratio of detection schemes against PRS worm.

attacks. We observe that both our SPEC and SPEC(W) schemes achieve 100 percent detection rate (P_D) while detecting traditional PRS worms in comparison with the existing worm detection schemes that have been specifically designed for detecting the traditional PRS worms. In view of emphasizing the relative performance of our SPEC and SPEC(W) schemes with the existing worm detection schemes, we plot the MIR and DT results in Figs. 7 and 8 for different scan rates S. We can observe from these figures that the MIR and DT results of our spectrum-based scheme (shown only for SPEC(W)) are comparable or better than the existing worm detection schemes. For a mean scan rate of 70/min, our SPEC(W) scheme achieves a detection time of 1;024 min, which is faster than that of VARandMEAN schemes, whose values are 1;239 and 1;161 min, respectively. For the same mean scan rate of 70/min, SPEC(W) achieves a maximal infection ratio of 0.03, which is comparable to TREND's MIR value and is less than 50 percent of the MIR value for the VAR and MEAN detection schemes. The effectiveness of our spectrum-based scheme is based on the fact that traditional PRS worm scanning traffic shows a constantly rapid increase. Thus, SFM values are relatively small due to PSD concentration at the low-frequency bands in the case of the traditional PRS worm scanning.

6 Remarks

In this paper, we studied a new class of smart-worm called C-Worm, which has the capability to camouflage its propagation and further avoid the detection. Our investigation showed that, although the C-Worm successfully camouflages its propagation in the time domain, its camouflaging nature inevitably manifests as a distinct pattern in the frequency domain. Based on observation, we developed a novel spectrum-based detection scheme to detect the C-Worm. Our evaluation data showed that our scheme achieved superior detection performance against the C-Worm in comparison with existing representative detection schemes. This paper lays the foundation for ongoing studies of "smart" worms that intelligently adapt their propagation patterns to reduce the effectiveness of countermeasures.

Acknowledgments

The authors thank the anonymous reviewers for their invaluable feedback. This work was supported in part by the US National Science Foundation (NSF) under grant No. CNS-0916584, CAREER Award CCF-0546668, and the Army Research Office (ARO) under grant No. AMSRD-ACC-R 50521-CI; by the NSF under grant Nos. 0963973 and 0963979 and by the University of Macau, and Macao Science and Technology Development Foundation. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies. The authors would like to acknowledge Ms. L. Archer for her dedicated help to improve the paper.

References

[1] D. Moore, C. Shannon, and J. Brown, "Code-Red: A Case Study on the Spread and Victims of an Internet Worm," Proc. Second Internet Measurement Workshop (IMW), Nov. 2002.
[2] D. Moore, V. Paxson, and S. Savage, "Inside the Slammer Worm," Proc. IEEE Magazine of Security and Privacy, July 2003.
[3] CERT, CERT/CC Advisories, http://www.cert.org/advisories/,2010.
12 IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL. 8, NO. 3, MAY/JUNE 2011
[4] P.R. Roberts, Zotob Arrest Breaks Credit Card Fraud Ring, www.eweek.com/article2/0,1895,1854162,00.asp, 2010.
[5] W32/MyDoom.B Virus, http://www.us-cert.gov/cas/techalerts/ TA04-028A.html, 2010.
[6] W32.Sircam.Worm@mm, http://www.symantec.com/avcenter/venc/data/w32.sircam.worm@mm.html, 2010.
[7] Worm.ExploreZip, http://www.symantec.com/avcenter/venc/ data/worm.explore.zip.html, 2010.
[8] R. Naraine, Botnet Hunters Search for Command and Control Servers, http://www.eweek.com/article2/0,1759,1829347,00.asp, 2010.
[9] T. Sanders, Botnet Operation Controlled 1.5m PCs Largest Zombie Army Ever Created, tp://www.vnunet.com/vnunet/news/2144375/botnet-operation-ruled-million, 2005. [10] R. Vogt, J. Aycock, and M. Jacobson, "Quorum Sensing and Self-Stopping Worms," Proc. Fifth ACM Workshop Recurring Malcode (WORM), Oct. 2007.
[11] S. Staniford, V. Paxson, and N. Weaver, "How to Own the Internet in Your Spare Time," Proc. 11th USENIX Security Symp. (SECURITY), Aug. 2002.
[12] Z.S. Chen, L.X. Gao, and K. Kwiat, "Modeling the Spread of Active Worms," Proc. IEEE INFOCOM, Mar. 2003.
[13] M. Garetto, W.B. Gong, and D. Towsley, "Modeling Malware Spreading Dynamics," Proc. IEEE INFOCOM, Mar. 2003.
[14] C.C. Zou, W. Gong, and D. Towsley, "Code-Red Worm Propagation Modeling and Analysis," Proc. Ninth ACM Conf. Computer and Comm. Security (CCS), Nov. 2002.
[15] Zdnet, Smart Worm Lies Low to Evade Detection, http://news. zdnet.co.uk/internet/security/0,39020375,39160285,00.htm, 2010.
[16] J. Ma, G.M. Voelker, and S. Savage, "Self-Stopping Worms," Proc. ACM Workshop Rapid Malcode (WORM), Nov. 2005.
[17] M.G. Kang, J. Caballero, and D. Song, "Distributed Evasive Scan Techniques and

Countermeasures," Proc. Int'l Conf. Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), July 2007.
[18] C. Wright, S. Coull, and F. Monrose, "Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis," Proc. 15th IEEE Network and Distributed System Security Symp. (NDSS), Feb. 2008.
[19] C. Zou, W.B. Gong, D. Towsley, and L.X. Gao, "Monitoring and Early Detection for Internet Worms," Proc. 10th ACM Conf. Computer and Comm. Security (CCS), Oct. 2003.
[20] S. Venkataraman, D. Song, P. Gibbons, and A. Blum, "New Streaming Algorithms for SuperSpreader Detection," Proc. 12th IEEE Network and Distributed Systems Security Symp. (NDSS), Feb. 2005.
[21] J. Wu, S. Vangala, and L.X. Gao, "An Effective Architecture and Algorithm for Detecting Worms with Various Scan Techniques," Proc. 11th IEEE Network and Distributed System Security Symp. (NDSS), Feb. 2004.
[22] Dshield.org, Distributed Intrusion Detection System, http:// www.dshield.org/, 2005.
[23] SANS, Internet Storm Center, http://isc.sans.org/, 2010.
[24] C.C. Zou, W. Gong, and D. Towsley, "Worm Propagation Modeling and Analysis under Dynamic Quarantine Defense," Proc. First ACM CCS Workshop Rapid Malcode (WORM), Oct. 2003.
[25] C.C. Zou, D. Towsley, and W. Gong, "Modeling and Simulation Study of the Propagation and Defense of Internet E-Mail Worm," IEEE Trans. Dependable and Secure Computing, vol. 4, no. 2, pp. 105-118, Apr.-June 2007.
[26] C. Zou, D. Towsley, and W. Gong, "Email Worm Modeling and Defense," Proc. 13th Int'l Conf. Computer Comm. and Networks (ICCCN), Oct. 2004.
[27] W. Yu, S. Chellappan, C. Boyer, and D. Xuan, "Peer-to-Peer System-Based Active Worm Attacks: Modeling and Analysis," Proc. IEEE Int'l Conf. Comm. (ICC), May 2005.
[28] Dynamic Graphs of the Nimda Worm, http://www.caida.org/ dynamic/analysis/security/nimda, 2010.
[29] S. Staniford, D. Moore, V. Paxson, and N. Weaver, "The Top Speed of Flash Worms," Proc. Second ACM Conf. Computer and Comm. Security (CCS) Workshop Rapid Malcode (WORM), Oct. 2004.
[30] Y. Li, Z. Chen, and C. Chen, "Understanding Divide-Conquer- Scanning Worms," Proc. Int'l Performance Computing and Comm. Conf. (IPCCC), Dec. 2008.
[31] D. Ha and H. Ngo, "On the Trade-Off between Speed and Resiliency of Flash Worms and Similar Malcodes," Proc. Fifth ACM Workshop Recurring Malcode (WORM), Oct. 2007.
[32] Y. Yang, S. Zhu, and G. Cao, "Improving Sensor Network Immunity under Worm Attacks: A Software Diversity Approach," Proc. ACM MobiHoc, May 2008.
[33] L. Martignoni, D. Bruschi, and M. Monga, "Detecting Self- Mutating Malware Using Control Flow Graph Matching," Proc. Conf. Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), July 2006.
[34] R. Perdisci, O. Kolesnikov, P. Fogla, M. Sharif, and W. Lee, "Polymorphic Blending Attacks," Proc. 15th USENIX Security Symp. (SECURITY), Aug. 2006.
[35] Linux.com, Understanding Stealth Scans: Forewarned is Forearmed, http://security.itworld.com/4363/LWD010321vcontrol3/page1. html, 2010.

[36] Solar Designer, Designing and Attacking Port Scan Detection Tools, http://www.phrack.org/phrack/53/P53-13, 2006.
[37] J.Z. Kolter and M.A. Maloof, "Learning to Detect Malicious Executables in the Wild," Proc. 10th ACM SIGKDD, Aug. 2004.
[38] X. Wang, W. Yu, A. Champion, X. Fu, and D. Xuan, "Detecting Worms via Mining Dynamic Program Execution," Proc. IEEE Int'l Conf. Security and Privacy in Comm. Networks (SECURECOMM), Sept. 2007.
[39] W. Yu, X. Wang, D. Xuan, and D. Lee, "Effective Detection of Active Worms with Varying Scan Rate," Proc. IEEE Int'l Conf. Security and Privacy in Comm. Networks (SECURECOMM), Aug. 2006.
[40] A. Lakhina, M. Crovella, and C. Diot, "Mining Anomalies Using Traffic Feature Distribution," Proc. ACM SIGCOMM, Aug. 2005.
[41] V. Yegneswaran, P. Barford, and D. Plonka, "On the Design and Utility of Internet Sinks for Network Abuse Monitoring," Proc. Symp. Recent Advances in Intrusion Detection (RAID), Sept. 2003.
[42] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, and D. Watson, "The Internet Motion Sensor: A Distributed Blackhole Monitoring System," Proc. 12th IEEE Network and Distributed Systems Security Symp. (NDSS), Feb. 2005.
[43] D. Moore, "Network Telescopes: Observing Small or Distant Security Events," Proc. Invited Presentation at the 11th USENIX Security Symp. (SECURITY), Aug. 2002.
[44] J. Jung, V. Paxson, A.W. Berger, and H. Balakrishnan, "Fast Portscan Detection Using Sequential Hypothesis Testing," Proc. 25th IEEE Symp. Security and Privacy (S&P), May 2004.
[45] H. Kim and B. Karp, "Autograph: Toward Automated, Distributed Worm Signature Detection," Proc. 13th USENIX Security Symp. (SECURITY), Aug. 2004.
[46] M. Cai, K. Hwang, J. Pan, and C. Papadopoulos, "Wormshield: Fast Worm Signature Generation with Distributed Fingerprint Aggregation," IEEE Trans. Dependable and Secure Computing, vol. 4, no. 2, pp. 88-104, Apr.-June 2007.
[47] R. Dantu, J.W. Cangussu, and S. Patwardhan, "Fast Worm Containment Using Feedback Control," IEEE Trans. Dependable and Secure Computing, vol. 4, no. 2, pp. 119-136, Apr.-June 2007.
[48] K. Ogata, Modern Control Engineering. Pearson Prentice Hall, 2002.
[49] J.B. Grizzard, V. Sharma, C. Nunnery, B.B. Kang, and D. Dagon, "Peer-to-Peer Botnets: Overview and Case Study," Proc. USENIX Workshop Hot Topics in Understanding Botnets (HotBots), Apr. 2007.
[50] P. Wang, S. SParka, and C. Zou, "An Advanced Hybrid Peer-to-Peer Botnet," Proc. USENIX Workshop Hot Topics in Understanding Botnets (HotBots), Apr. 2007.
[51] D.J. Daley and J. Gani, Epidemic Modeling: An Introduction. Cambridge Univ. Press, 1999.
[52] D. Bruschi, L. Martignoni, and M. Monga, "Detecting Self-Mutating Malware Using Control Flow Graph Matching," Proc. Conf. Detection of Intrusions and Malware and Vulnerability Assessment (DIMVA), July 2006.

[53] MetaPHOR, http://securityresponse.symantec.com/avcenter/ venc/data/w32.simile.html, 2010.
[54] P. Ferrie and P.S. Zmist, "Zmist Opportunities," Virus Bull., http://www.virusbtn.com, 2010.
[55] J. Bethencourt, D. Song, and B. Waters, "Analysis-Resistant Malware," Proc. 15th IEEE Network and Distributed System Security Symp. (NDSS), Feb. 2008.
[56] M. Sharif, J. Giffin, W. Lee, and A. Lanzi, "Impeding Malware Analysis Using Conditional Code Obfuscation," Proc. 15th IEEE Network and Distributed System Security Symp. (NDSS), Feb. 2008.
[57] I.V. Popov, S.K. Debray, and G.R. Andrews, "Binary Obfuscation Using Signals," Proc. 17th USENIX Security Symp. (SECURITY), July 2008.
YU ET AL.: MODELING AND DETECTION OF CAMOUFLAGING WORM 13
[58] M. Christodorescu and S. Jha, "Testing Malware Detectors," Proc. 2004 ACM SIGSOFT Int'l Symp. Software Testing and Analysis (ISSTA), July 2004.
[59] X. Wang, W. Yu, X. Fu, D. Xuan, and W. Zhao, "iloc: An Invisible Localization Attack to Internet Threat Monitoring Systems," Proc. 27th IEEE INFOCOM, Apr. 2008.
[60] J. Bethencourt, J. Frankin, and M. Vernon, "Mapping Internet Sensors with Probe Response Attacks," Proc. 14th USNIX Security Symp., July/Aug. 2005.
[61] Y. Shinoda, K. Ikai, and M. Itoh, "Vulnerabilities of Passive Internet Threat Monitors," Proc. 14th USNIX Security Symp., July/Aug. 2005.
[62] S. Soundararajan and D.L. Wang, "A Schema-Based Model for Phonemic Restoration," Technical Report OSU-CISRC-1/04-TR03, Dept. of Computer Science and Eng., The State Univ., Jan. 2004.
[63] N.S. Jayant and P. Noll, Digital Coding of Waveforms. Prentice Hall, 1984.

