On the effectiveness of service registration-based worm defense

Jin-Ho Kim
Telecommunication R&D Center
Samsung Electronics
Email: jin_ho.kim@samsung.com

Hyogon Kim
Department of Computer Science
and Engineering
Korea University
Email: hyogon@korea.ac.kr

Saewoong Bahk
School of Electrical Engineering
and Computer Science
Seoul National University
Email: sbahk@snu.ac.kr

***Abstract*— Existing Internet worm research focuses either on worm detection inside an AS, or on prevention of Internet-wide worm epidemic. But of more practical concern is how to repel worm infiltration attempts at the AS boundary. In this paper, we analyze the efficacy of the general perimeter defense system operating on service registration information. When such system finds incoming packets targeting an unregistered service, it intercepts the packets and relays them to the signature generation module. While the signature is extracted, the system blocks the infiltration through blacklisting. Finally, upon the signature generation, content filtering based on the signature takes over, replacing blacklisting. Since the effectiveness of such systems depends on the type of worm, we analyze the effectiveness against the following practical worm types: random scanning TCP worms, random-start sequential scanning TCP worms, and UDP worms.**

I. Introduction

A worm refers to the self-replicating and self-propagating code. Since the Morris worm of 1988, it has become the focus of attention with Code Red [1]–[3] in 2001. The nearly homogeneous OS and software environments of today provide a rich breeding ground for worms. Since the Code Red worm, more have appeared with increasingly larger damages. Worms can destroy or export sensitive data, format hard disk, make the infected host a zombie for a future denial-of-service (DoS) attack, install a secret backdoor, install a spammer [4], and even cause DoS simply through propagation activity [5].

Researchers are working on this problem but the progress is slow because worm is fast. In case of virus the defense is easier since the propagation is slow, giving ample time to anti-virus vaccine industry to react. However, worms are automated and the epidemic can mature even in the matter of a few minutes [5], which would beat any human-intervened reaction. Therefore, the first and foremost objective of worm defense system design is the fast reaction, through which the damage can be minimized.

Existing worm defense systems either focus on minimizing the worm epidemic from the perspective of the global Internet welfare [6], or focus on pinpointing infected hosts inside the AS [7]–[9]. But from the viewpoint of an AS, it is most important that it is not infected in the first place. As the primary defense mechanisms such as firewalls are usually placed only at AS perimeter, worms become significantly hard to cope with once the infiltration is made.

In this paper, we analyze the effectiveness of the generalized perimeter defense system operating on service registration information. When such system finds incoming packets targeting an unregistered service, it intercepts the packets and relays them to the signature generation module. While the signature is extracted, the system blocks the infiltration through blacklisting. Finally, upon the signature generation, the more accurate content filtering based on the signature takes over, replacing blacklisting. For convenience, we will call such system SWORD (Service registration-based WORm Defense) in this paper. Since the effectiveness of SWORD systems depends on the type of worm, we analyze its performance against the following practical worm types: random scanning TCP worms (*e.g.* CodeRed), random-start sequential scanning TCP worms [9], and UDP worms.

II. Service-based worm defense

A. Need for service registration

An inherent property of worm propagation activity is scanning. One way to positively identify the scanning activity is to find packets destined to an unassigned IP address and/or non-existent "service" (combination of IP addresses and transport-layer ports) [8], [10]. Since legitimate traffic scarcely targets unassigned addresses or services, we can determine the traffic destined to it as some sort of attack, if not a worm. Suppose an externally residing worm w scans the given AS in a uniform and random manner. For this AS, let us denote the set of hosts that are vulnerable to w as $V(w)$, and the set of unassigned IP addresses as U, and the size of its IP address space as T. We identify the traffic as the worm as soon as it hits U. Then the probability of detection before infection is:

$$d(w) = \sum_{k=0}^{\infty} \left(\frac{T - |U| - |V(w)|}{T} \right)^k \cdot \frac{|U|}{T} = \frac{|U|}{|U| + |V(w)|}.$$

Here, we notice that maximizing $|U|$ leads to the maximum $d(w)$ for a given $V(w)$. Namely, the $|U|$ term significantly increases by additionally considering the set of offered services. Table I shows the impact of such enhancement. It is obtained from scanning the campus networks with a /16 address space at two different universities in 2004 and 2005, respectively. In both cases, we notice that there are far fewer

TABLE I

NETWORK SCANNING RESULTS

	Port	Service	$T - \|U\|$	$\|U\|$	$\frac{\|U\|}{T} \cdot 100\%$
Univ.1 (2004)	N/A	ICMP ping	17,064	48,472	73.96%
	21 (TCP)	ftp	2,916	62,620	95.55%
	22 (TCP)	ssh	892	64,644	98.64%
	25 (TCP)	smtp	1,300	64,236	98.02%
	80 (TCP)	http	2,610	62,926	96.02%
Univ.2 (2005)	N/A	ICMP ping	5,473	60,063	91.65%
	21 (TCP)	ftp	639	64,897	99.02%
	22 (TCP)	ssh	643	64,893	99.02%
	25 (TCP)	smtp	302	65,234	99.54%
	80 (TCP)	http	1,648	63,888	97.49%

legitimate targets $T - |U|$ under the consideration of services than just unassigned IP addresses. Since $\frac{|U|}{T} \leq \frac{|U|}{|U|+|V|}$ (*i.e.*, not all servicing hosts are vulnerable), $d(w)$ can be larger. Furthermore, since many services are automatically installed along with the OS in the ignorance of the user, the number of servicing hosts could be further reduced if we let them register the service only when they manifestly express their desire to serve external hosts.

B. Model of the analyzed system

The SWORD system is composed of the perimeter firewall and a worm signature generator [11]–[14]. These components interwork in the following manner:

1) The firewall detects scanning packet(s).
2) The suspected worm-transporting packet is intercepted and passed to the signature generator.
3) The firewall blacklists the source host that allegedly sent the attack packet(s).
4) The signature generator extracts the signature from the attack packet(s) and forwards it to the firewall.
5) After installing the new signature, the firewall starts content filtering against the worm.

In order to intercept the suspected worm-transporting packet(s) (step (2)), the worm interceptor that could be on either the firewall or the signature generator uses a spoofed connection in case of TCP (UDP case is addressed in Section V). Fig. 1 depicts the intercept process. Notice that we have a step to confirm the verity of an incoming SYN packet by sending a spoofed SYN/ACK and waiting for an ACK. It is in order to prevent the blacklisting from being exploited in the DoS attack. The design of the signature generation system is beyond the scope of this paper. The objective of this work is to analyze the effectiveness and requirements of the SWORD systems under the assumption that a signature generation system exists. For detailed discussion on on-line signature generation, refer to [11]–[14]. Below, we analyze SWORD using the Analytical Active Worm Propagation (AAWP) model [10].

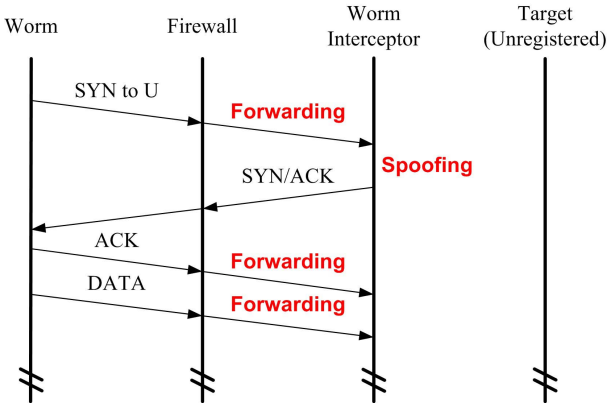

Fig. 1. Worm interception process.

III. RANDOM SCANNING TCP WORMS

We start with the basic random scanning worms that transport worm payload using TCP. Table II is the list of parameters used in the analysis. The hitlist refers to the list of vulnerable hosts to be used for the initial worm deployment. Initially, no hosts inside the AS are compromised yet, so they do not appear in the hitlist. The worm performs uniform and random scanning over the entire Internet, and when a vulnerable host is scanned at $t = i$, it starts operation at $t = i + 1$. Since U and R are mutually disjoint and collectively exhaustive, we have $|U| + |R| = 2^{32-L}$. Since V is the subset of vulnerable hosts in R, we have $|R| \geq |V|$. Finally, we assume $|V| \ll N$.

TABLE II

PARAMETERS FOR ANALYSIS

Parameter	Connotation
N	Number of vulnerable hosts in the global Internet
H	Size of hitlist
s	Scanning rate
L	AS size, /L network
U	Set of unregistered hosts in the AS
R	Set of registered hosts in the AS
V	Subset of vulnerable hosts in R

Let I_i and B_i denote the number of infected hosts on the global Internet and the number of blacklisted hosts by the AS at $t = i$, respectively. Since we have $|V| \ll N$, we can ignore the contribution of $|V|$ to N. Applying the AAWP model, we get:

$$\begin{aligned} I_{i+1} &= I_i + (N - I_i)\left[1 - \left(1 - \frac{1}{2^{32}}\right)^{s \cdot I_i}\right], \ I_0 = H. \\ B_{i+1} &= B_i + (I_i - B_i)\left[1 - \left(1 - \frac{|U|}{2^{32}}\right)^{s}\right], \ B_0 = 0. \quad (1) \end{aligned}$$

Let u_i and v_i denote the probability that at tick i any infected host scans U, and any infected but non-blacklisted host scans V, respectively. Similar to Eq. (1) we get:

$$u_{i+1} = 1 - \left(1 - \frac{|U|}{2^{32}}\right)^{s \cdot I_i}, \ u_0 = 0. \quad (2)$$

$$v_{i+1} = 1 - \left(1 - \frac{|V|}{2^{32}}\right)^{s \cdot (I_i - B_i)}, \ v_0 = 0. \quad (3)$$

The probability that the AS first detects the worm at tick i and the expected time it takes until the detection t_D [10] are,

respectively,

$$d_i = \left[\prod_{k=0}^{i-1}(1-u_k)\right] \cdot u_i, \quad t_D = \sum_{k=1}^{j+1} k \cdot d_k.$$

where j is the tick at which I_i no longer increases. Similarly, the probability that the AS is first infected at tick i and the expected time until the infection t_V is given by:

$$c_i = \left[\prod_{k=0}^{i-1}(1-v_k)\right] \cdot v_i, \quad t_V = \sum_{k=1}^{j+1} k \cdot c_k.$$

Now, the probability that the *reaction window* W, *i.e.*, the critical time duration between the detection and the infection, is exactly m ticks is

$$P[W = m] = \sum_{i=1}^{j+1} d_i c_{i+m}. \tag{4}$$

Notice W is the time budget given to the signature generation module.

Fig. 2 uses Eq. (4) to generate the CDFs of detection, infection and reaction times for $N = 400,000$, $H = 100$, $s = 10$, $L = 16$, $|V| = 200$, and $|R| = 2,000$. These parameter values are mostly taken from the CodeRed v2 epidemic. One exception is s which is set larger so as to make the epidemic more virulent. If vulnerable hosts are evenly distributed over the Internet address space, the expected number of vulnerable hosts inside this AS is $\frac{N}{2^L}$, and using the above parameter values it is $400,000/2^{16} \approx 6.1$. Another exception is our choice of $|V| \gg 6.1$, which will put SWORD under significantly higher stress (with higher success probability for scanning attempts).[1] The figure shows that the detection under SWORD is very fast so the CDF of W is close to that of the infection time.

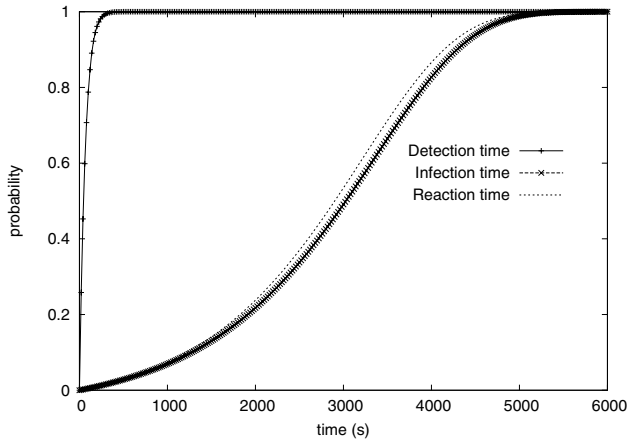

Fig. 2. CDF of detection, infection, and reaction times in the enhanced CodeRed v2 scenario.

In our setting, if the signature generation time t_{SIG} is 60s, the probability that the AS succeeds in defense and it does not get infected is 99.4%. For $t_{SIG} = 600s$, the probability goes down to 95.9%. If t_{SIG} further increases to 2,400 seconds, the probability is as low as 67.2%. This tells us that the signature generation delay is pivotal in the success of the content filtering phase. Fig. 3 shows t_D gets shorter for smaller $|R|$,

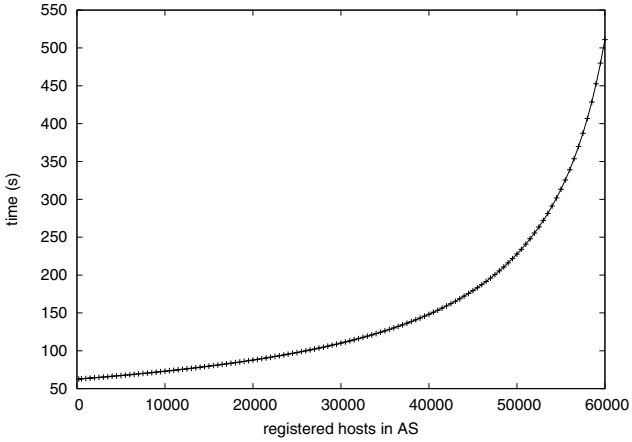

Fig. 3. Average detection time as a function of the number of registered hosts in the enhanced CodeRed v2 scenario.

which strengthens the defense due to enlarged $|U|$. And Fig. 4 shows the maximum time budget for signature generation for a given $|V|$. For instance, `Defense probability 95%` shows the maximum time that can be given to the signature generation module for 95% defense probability under $|V|$. We notice it is a fast decreasing function of $|V|$. Comparing the curves, we also notice that the defense probability increases as the signature generation time decreases.

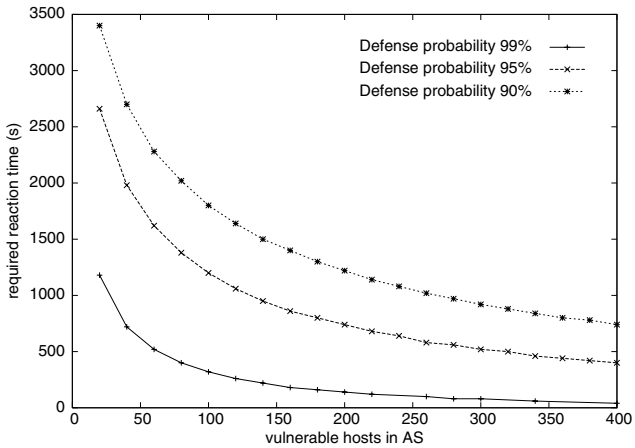

Fig. 4. Signature generation time as a function of vulnerable population in the enhanced CodeRed v2 scenario.

Fig. 5 shows t_V and t_D as functions of L. From the relation between `Infection time` and `Infection time (No blacklisting)` we notice that the effect of blacklisting is very small for mid- to small- networks. In particular, class-B networks (/16) reap only marginal value from it, let alone class-C networks (/24; not shown). So in order to maximize the effect of blacklisting, we must raise the B_i/I_i value through

[1]The reason is twofold. First, in practice there is a significantly large unused portion of all the IP addresses. Second, depending on the given worm the number of vulnerable hosts can be larger in the AS than the global average.

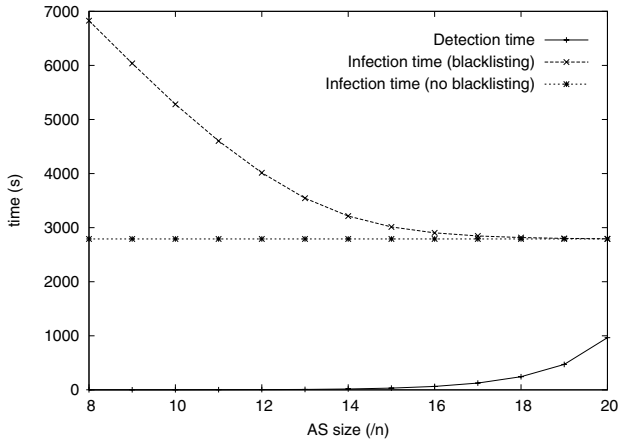

Fig. 5. Average times depending on network size in the enhanced CodeRed v2 scenario.

the cooperation with other AS's or ISPs.

The main reason that we should keep and strengthen blacklisting despite the miserable performance against random scanning worms is the existence of other worm types. In particular, it is essential to defend the AS against random-start sequential-scanning worms such as Blaster [9]. Without blacklisting, even if we detect such a worm as soon as it executes random-start, the infection can happen within a short amount of time since the worm sequentially scans the AS. For the AS, the interval between the points in time axis where two consecutive scans fall becomes much smaller, and we can have only a very small window W for signature generation. Blacklisting is the only conceivable first-line defense in this case. Below, we discuss the performance of SWORD against the random-start sequential-scanning worms.

IV. Random-start sequential-scanning worms

Random-start sequential-scanning worm first generates a random address a, and then scans for a vulnerable service by incrementing the address, *i.e.*, $a+1,\ a+2,\ \ldots$. Unlike in random scanning, W becomes extremely small in this case, so we must block the infected host quickly. As there is little time for signature generation, blacklisting is the best first-aid, although it may not be effective against random scanning worms (see Fig. 5).

Suppose an infected host outside the AS scans the AS starting from $a \in A$, where A is the address space of the AS. Although the scanning itself is sequential, whether a scanned host (to be precise, *service*) is in U or V is generally random within the AS address space, so we can model the detection and infection probability distributions to be uniform random. Let t_{BL} denote the time that takes a firewall to blacklist a host after detecting an access to an unregistered service, t_I denote the time that takes a worm to infect a particular host after it scans the host, and $t_S = 1/s$ denote the inter-scan time. Each time duration is measured at the firewall.

First of all, if the worm scans a host in V first, the defense is more likely to fail than the case it hits U first and is immediately detected. So for the moment let us consider the case where the worm hits U first. Then, the successful blocking probability is given by

$$b_1 = \sum_{i=0}^{\infty} \left(\frac{T - |U| - |V|}{T} \right)^i \cdot \frac{|U|}{T} = \frac{|U|}{|U| + |V|}. \quad (5)$$

Unfortunately, even this does not hold unless t_{BL} is small enough. Therefore, at the minimum we require $t_{BL} < t_S + t_I$ to have the probability in Eq. (5). Here, $t_I > t_{RTT}$ for TCP-transported worms, where t_{RTT} is the round-trip time for the TCP connection between the infector and the infectee. The requirement is depicted in Fig. 6. If it is met, we can preempt even an ongoing infection.

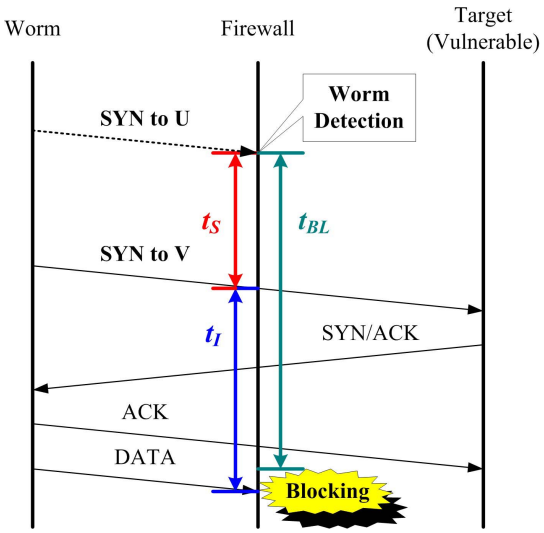

Fig. 6. The requirement on t_{BL} for successful defense.

Now, we come to the case where the scans hit V first. As we mentioned above, the defense is more likely to fail in this case because the SWORD system is given less time to deploy defense. An interesting fact in this case, however, is the defense is more likely to succeed if the scanning rate is higher. Fast scanning worms typically generate multiple threads, and each thread independently performs scanning. Thus even when a scan is not finished by one thread, another thread can start a scan. The consequence is that t_S can be very small. If t_{BL} is also small enough, we just might succeed in preempting the ongoing infection. Namely, if we have

$$t_{BL} < t_I - t_S \quad (6)$$

we can succeed in defense if we detect the immediately following a scan to U after a scan to V. In order to verify this claim, let us consider the following cases:

1) $t_I - t_S \le t_{BL} < t_S + t_I$: In this case, we do not satisfy the condition of Eq. (6). So the blocking probability b_1 is as given in Eq. (5).
2) $t_I - 2 \cdot t_S \le t_{BL} \le t_I - t_S$: In this case, even if a scan *could* hit V and start infection process, it can be aborted

if we can blacklist the source address in the next scan. Fig. 7 shows the situation, and the blocking probability is computed as follows.

$$\begin{aligned} b_2 &= \sum_{i=0}^{\infty} \left(\frac{T - |U| - |V|}{T} \right)^i \cdot \frac{T - |U|}{T} \cdot \frac{|U|}{T} + \frac{|U|}{T} \\ &= \frac{T + |V|}{|U| + |V|} \cdot \frac{|U|}{T} \\ &= \left(1 + \frac{|V|}{T} \right) \cdot b_1 \geq b_1. \end{aligned} \quad (7)$$

In other words, the blocking probability increases as t_S decreases relative to t_{BL}.

3) $t_I - k \cdot t_S \leq t_{BL} < t_I - (k-1) \cdot t_S$, $k \geq 2$: If we generalize Eq. (7) for k we can get the recurrence relation as follows:

$$\begin{aligned} b_k &= \frac{|U|}{T} + \frac{T - |U|}{T} \cdot b_{k-1} \\ &= b_{k-1} + \frac{|U|}{T} (1 - b_{k-1}) \geq b_{k-1}. \end{aligned}$$

Solving the recurrence relation using Eq. (5) and (7), we get

$$b_k = 1 - (1 - b_1) \cdot \left(1 - \frac{|U|}{T} \right)^{k-1}, \quad k \geq 2.$$

With a given t_{BL}, increasing k means smaller t_S. The fact that b_k increases with k means that the blocking performance improves as the scanning speed increases.

In essence, faster scanning speed means that SWORD gets to see more scans in a given time interval, and it raises the chance that one of them hits U, triggering blacklisting. The earlier the trigger, the bigger the blocking probability for a given t_{BL}.

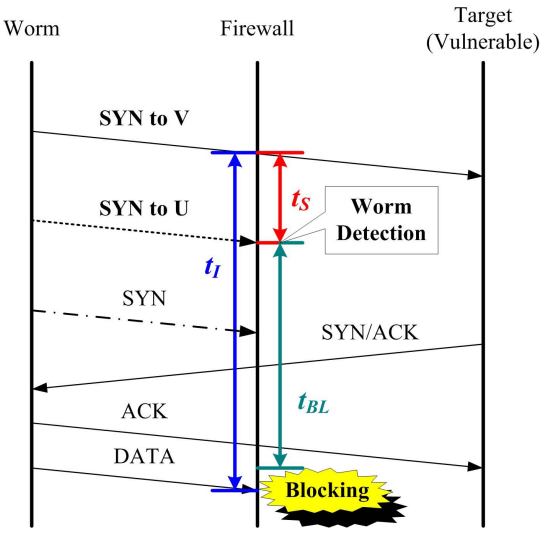

Fig. 7. The case of fast scanning.

V. UDP WORMS

In case of TCP we can assert the validity of the source address in a SYN packet by responding with a SYN/ACK and receiving an ACK. In UDP, however, source spoofing is possible thus blacklisting is practically useless. So SWORD skips the source validation for UDP packets and directly forwards the intercepted scanning packet to the signature generator.

Let us first consider random-scanning UDP worms. Since SWORD cannot conduct the defense with blacklisting in case of UDP worms, so much less time is given for signature generation. Let v_i^{UDP} be the probability that at least one infected host scans V at tick i. We get v_i^{UDP} as follows.

$$v_{i+1}^{UDP} = 1 - \left(1 - \frac{|V|}{2^{32}} \right)^{s \cdot I_i}, \quad v_0^{UDP} = 0. \quad (8)$$

Notice the $I_i - B_i$ term in Eq. (3) changed to I_i here due to the absence of blacklisting. But there is no change in the equation for worm detection, and it is the same as Eq. (2). As to the detection probability, Fig. 5 shows that t_V even without blacklisting is significantly longer than t_D so that if the signature generation can be done quickly we can still block the random-scanning UDP worms.

Random-start sequential-scanning UDP worms must be more difficult to block. No such worm has been reported yet, but since the inter-scan time is extremely short, producing the signature must be done in lesser amount of time. Let t_C be the time it takes the content filtering begins to take effect, starting from the detection instant. Since UDP worms can infect a vulnerable host with the first scanning packet, we let $t_I = 0$. That is, if we do not block the first attacking host we fail the defense. For $k \cdot t_S \leq t_C < (k+1) \cdot t_S$, the probability c_k of succeeding in defense against an attacker is:

$$\begin{aligned} c_k &= \sum_{i=0}^{\infty} \left(\frac{T - |U| - |V|}{T} \right)^i \cdot \frac{|U|}{T} \cdot \left(1 - \frac{|V|}{T} \right)^k \\ &= \frac{|U|}{|U| + |V|} \cdot \left(1 - \frac{|V|}{T} \right)^k. \end{aligned} \quad (9)$$

We notice c_k exponentially decreases with k. Fig. 8 shows c_k for $L = 16$, $|V| = 200$, $|R| = 2,000$ as a function of k. For 95% success probability we should have $t_C < 15 \cdot t_S$ and for 90% probability we should have $t_C < 33 \cdot t_S$. Namely, t_{SIG} should be extremely small to block random-start sequential-scanning UDP worms. But there are also bright sides. First, since there are not many services using UDP, we have a small $|V|$, which increases the defense probability (see Eq. (9)). Also, known UDP worms attack with a single UDP packet, so we can relatively easily generate the signature. Therefore, if we take account of these characteristics of UDP worms we can make a signature generation system sufficiently effective against random-start sequential-scanning UDP worms.

VI. LOCAL SCANNING WORMS

Worms do not always scan the entire Internet space (*e.g.* CodeRed II and Nimda) . Rather they perform so called the

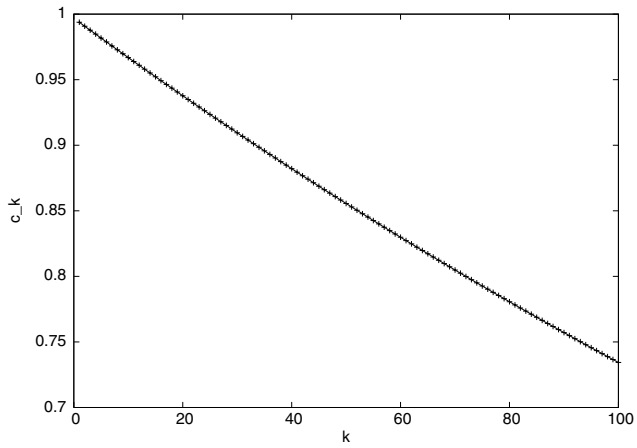

Fig. 8. Defense success probability vs. content filtering deployment time in the units of inter-scan times.

local scan, a.k.a. subnet scan, where they mix the scans for /8 common prefix, /16 common prefix, and the entire IP space [15], [16]. The reason is that if a host is infected, neighboring hosts in the address space are likely to share the same OS or softwares, *i.e.*, same vulnerability, being administered by the same authority. Also, since the worm is already inside the network it can evade perimeter defense more easily [10]. These are the very reasons why we focus on the defense against the worm infiltration from outside in the first place. Once we fail to block the infiltration the local scanning worm spreads quickly inside the AS, and we still need the intra-AS defense as discussed in [7].

VII. Related work

Since the Code Red worm of 2001, worm research has been very active recently. The areas of focus are worm propagation modeling, worm detection and defense from the global Internet perspective, and worm detection and defense within an AS. Modeling the worm propagation was facilitated by the introduction of epidemiological models [17], [18]. Although these models help delineate the general dynamics of the Internet worm epidemic, they do not precisely characterize the detail. So Chen *et al.* [10] propose the Analytical Active Worm Propagation(AAWP) model. Moore *et al.* [6] classify the worm defense into prevention, treatment, and containment. In the containment category, they compare blacklisting and content filtering. In particular, the paper discusses how effective these techniques will be when the part or whole of Internet cooperate.

Worms are easiest to detect in the proximity of the infected host. Several works [7], [8], [19] propose a technique to detect and respond to a worm by observing the traffic inside an AS. But the system requires meticulously deployed firewalls to block internal traffic inside the AS when the AS has been infected.

This paper departs from these two branches of work, since it models and analyzes a defense system that blocks the worm infiltration from outside the AS, using blacklisting and content filtering.

VIII. Conclusion

The most effective way to block warm infiltration is content filtering. But for previously unknown worms, we need to extract the signature from the attacking packet. In this paper, we explored a generic perimeter defense system that detects infiltration attempts based on the service registration information. We showed that having services registered greatly increases the worm blocking power of the system. We also analyzed the timing requirement for the signature generation module in such a system, against the three most popular worm types. We believe that the analysis in this paper can provide a concrete guideline as to the signature generation timing requirement for the on-line worm signature generation systems design, which has been a focus of attention recently.

References

[1] E. H. Spafford, "The Internet Worm Program: An Analysis," Purdue Technical Report CSD-TR-823, 1988.
[2] D. Moore and C. Shannon, "The Spread of the Code-Red Worm (CRv2)," CAIDA, July 2001.
[3] eEye Digital Security, ".ida Code Red Worm," July 2001, http://www.eeye.com/html/Research/Advisories/AL20010717.html.
[4] J. Markoff, "Experts say money is motive for SoBig virus," SFGate.com, August 2003, http://www.sfgate.com/cgi-bin/article.cgi?file=/chronicle/archive/2003/08/26/BU249135.DTL.
[5] CAIDA, "Analysis of the Sapphire Worm," http://www.caida.org/analysis/security/sapphire/, Jan. 30, 2003.
[6] D. Moore, C. Shannon, G. M. Voelker, and S. Savage, "Internet Quarantine: Requirements for Containing Self-Propagating Code," Proceedings of IEEE INFOCOM, April 2003.
[7] S. Staniford, "Containment of Scanning Worms in Enterprise Networks," Silicon Defense, October 2003. http://www.silicondefense.com/research/researchpapers/scanContainment/
[8] T. Toth and C. Kruegel, "Connection-history based anomaly detection," Proceedings of IEEE Workshop on Information Assurance and Security, June 2002.
[9] eEye Digital Security, "Blaster Worm Analysis," August 2003, http://www.eeye.com/html/Research/Advisories/ AL20030811.html
[10] Z. Chen, L. Gao and K. Kwiat, "Modeling the Spread of Active Worms," Proceedings of IEEE INFOCOM, April 2003.
[11] J. Newsome and D. Song, "Dynamic taint analysis for automatic detection, analysis, and signature generation of exploits on commodity software," In Proceedings of the 12th Annual Network and Distributed System Security Symposium (NDSS '05), February 2005.
[12] L. Zhou, L. Zhang, F. McSherry, N. Immorlica, M. Costa, and S. Chien, "A First Look at Peer-to-Peer Worms: Threats and Defenses," in proceedings of IPTPS 2005.
[13] J. Newsome, B. Karp, and D. Song, "Polygraph: Automatically Generating Signatures for Polymorphic Worms," in proceedings of IEEE Symposium on Security and Privacy, 2005.
[14] X. Jiang, D. Xu, H. J. Wang, and E. H. Spafford, "Virtual Playgrounds for Worm Behavior Investigation," in Proceedings of the 8th International Symposium on Recent Advances in Intrusion Detection, 2005.
[15] eEye Digital Security, "CodeRedII Worm Analysis," August 2001, http://www.eeye.com/html/Research/Advisories/ AL20010804.html
[16] CERT, "Nimda Worm," CERT Advisory CA-2001-26, September 2001. http://www.cert.org/advisories/CA-2001-26.html.
[17] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the Internet in Your Spare Time," 10th USENIX Security Symposium, August 2002.
[18] J. Kephart, D. Chess, and S. White, "Computers and Epidemiology," IEEE Spectrum, May 1993.
[19] S. Staniford-Chen, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, and D. Zerkle, "GrIDS: A Graph-Based Intrusion Detection System for Large Networks," Proceedings of the 19th National Information Systems Security Conference, 1996.