The Use of White Holes to Mislead and Defeat Importance Scanning Worms

Guofei Gu[1], Zesheng Chen[1], Phillip Porras[2], Wenke Lee[1]

Abstract

Recently, a new self-learning worm propagation strategy was introduced in [4, 2], which we refer to as the *importance scanning* method. Under the importance scanning approach, a worm employs an address sampling scheme to search for the underlying group distribution of (vulnerable) hosts in the address space through which it propagates. The worm exploits this information to increase the rate at which it locates viable addresses during its search for infection targets. In this paper, we introduce a strategy to combat the importance scanning propagation technique. We propose the use of *white hole* networks, which employ several existing components to dissuade, slow, and ultimately halt the propagation of importance scanning worms. We demonstrate how the white hole approach can be an effective defense, even when the deployment of this countermeasure represents a very small fraction of the address space population.

1 Introduction

Worms are becoming smarter in selecting their victims. In recent years we have seen an evolution in the approaches of scanning strategies from naive random-scanning techniques, to much faster and more evasive propagation methods. While many worms have used random scanning techniques with notable success [12, 26], in general the random scan method is relatively inefficient in searching for victim hosts within the Internet, and its indiscriminate nature makes it highly subject to passive detection [9, 13, 14]. A key observation is that a random search algorithm is ill-suited for seeking targets when those targets reside in a space at (predictably) non-uniformly distributed locations.

With respect to Internet address occupation, the existing research points out that the real (and also vulnerable) machine distribution in the whole IP space is not uniform [22, 25, 3], and this point has also been noted in some worm studies [26]. Recently, researchers proposed several new propagation strategies for worms that apply knowledge of the Internet structure. For example, some have proposed propagation strategies to take advantage of routing space information [24], address sampling to uncover group distributions in the address space prior to address scanning [2], and self-learning using importance-scanning [4].

In particular, importance scanning techniques attempt to uncover the distribution of live IP addresses (or even real vulnerable machines), and then focus their infection efforts on these targets to both achieve a higher scan-to-infection rate and to help evade passive monitors by avoiding the indiscriminate scanning of unused IP addresses. In [2], worms use a two-step infection cycle: a sampling of various network segments followed by a spreading phase to those segments that appear to contain *live* subnets. In the first phase, the worm sample-scans addresses from an address segment, and upon completion will spread with an affinity to those segments that appear to contain targets of interest (e.g., a population of live subnets). In [4], a self-learning worm is proposed, which estimates the vulnerability distribution very early in the infection stage (instead of before spreading). After an initial infection cycle, the worm attacker (such as a botnet-like worm) will estimate the distribution according to existing victim information. Then all the worms will use this vulnerable-host distribution estimate to adjust their scan probability distribution.

Importance scanning poses some disturbing challenges for worm defense research. One implication of these network-structure-aware infection strategies is that they are by design intended to avoid low-

[1] Georgia Institute of Technology

[2] Computer Science Laboratory, SRI International

occupancy address segments, including darknets that are instrumented with passive worm detection tools, such as Kalman-filter-based detection [13], victim number-based detection [14]), Internet Motion Sensor [9], or network telescopes [21]. Second,these worms are shown to provide a faster infection rates than other contemporary naive propagation strategies, suggesting a future of more virulent malware epidemics that combine speed and stealthy behavior.

In this paper we observe that the predictable affinity of importance scanning worms toward densely populated networks can also be viewed as a potential vulnerability. We explore the design space of what we refer to as *white holes*, which are systems that co-occupy populated network segments to increase the difficulty with which legitimate hosts can be targeted. A white hole can turn a legitimate live network segment into a segment that looks anomalously dense to a worm attempting to avoid honeynets, and can proactively mask the location of legitimate co-located addresses.

We introduce a defensive white hole approach that can be constructed using components from existing techniques, and analyze their ability to hinder importance scanning worms. Further, we examine how the incorporation of LaBrea-like mechanisms [18] can make a white hole an effective offensive tool to trap importance scanning worms, and conclude that the very affinity criteria that allows these worms to accelerate their infection rate also increases their susceptibility to our white hole countermeasures. We demonstrate how LaBrea mechanisms are far more effective in countering importance scanning worms, even when those countermeasures are deployed to a small ratio of the address space. We also discuss some challenging design issues and limitations in Section 5.

2 White Hole Design

As intelligent as firewalls, content filters, and address translation systems have become, it generally remains a difficult challenge to keep the existence of a live subnet or network segment invisible to attackers. In recognition of this reality, the alternative approach explored here is to hide the hosts of a live network segment within a population of seemingly live phantom addresses. The objective of a white hole service is to blend live targets in among phantom addresses the way a tree may be blended into a forest, or a needle into a needle stack. White holes present interactive responses to worm probes such that from the worm's point of view, the density of responses in the network segment obfuscate the worm's ability to successfully identify potential targets. A successful white hole deployment will effectively prevent the worm from accurately measuring the address distribution of the network segment.

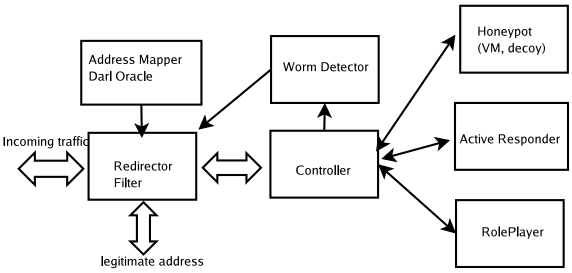


Figure 1: Architecture of a White Hole

We can assemble a white hole in the network by combining several existing techniques. Figure 1 shows the architecture of the white hole components, which include an address mapper, redirector, controller, worm detector, active responder, RolePlayer, VM honeypot, and a decoy honeypot. The following briefly summarize the purpose and function of each component in the white hole architecture.

- **Address mapper**: Actively collects and updates the unused IP/port segment of the network that the white hole will occupy. This can be done using an agent-manager based architecture (similar to SNMP network management). A similar technique is used in [27], in which an active mapping method builds profiles of the network topology and the TCP/IP policies of all hosts on the network. Recently, Cooke et al. proposed a Dark Oracle [8], which discovers dark addresses by actively participating in allocation, routing, and policy systems, and demonstrated successful operation in several networks. Unused ports can also be observed and emulated, or a strategic subset of service ports can be selected and emulated to enhance the realism of the white hole.

- **Redirector:** Redirects all incoming traffic to unused IP/ports, as specified by the address mapper, to the controller. This component is the first line classifier at the edge router of the protected network. All incoming packets targeting legitimate end hosts are passed through without disruption.

- **Controller:** Decides how redirected traffic will be handled within the white hole space. Streams may be directed to any of the available active responder components (including honeypots) within the white hole or filtered in cases of overload.

- **Active responder:** Includes a simple active responder, stateless role player, virtual machine honeypot, and physical decoy honeypot to interact with incoming worm scans. Potential active responders include applications such as iSink [10], or Honeyd [30] for handling simple scan responses. These scans may contribute most of the traffic. For simple scanning as shown in [2], a simple connection responder may be enough. If the connection needs further interaction, simple connection response may be insufficient to follow further application-level dialog. In such cases, traffic may be redirected to **VM-based honeypots**, or even **decoy honeypots** (real physical machines), to emulate full application response. In this way, we can capture the sampling attacks discussed in [4]. [17] discusses efficient approaches to deploy honeyfarms that can support large numbers of virtual machines. While this partially solves the scalability problem, we may also leverage the similarity of worm dialogs to more efficiently scale to larger worm scan volumes, as discussed by Cui et al. in the RolePlayer system [7]. RolePlayer can achieve the goal of protocol independent adaptive replay of application dialog in a stateless (memory efficient) way. That is, one can use such a lightweight technique to learn existing (captured) worm dialog and then mimic the dialog to provide quick response for further similar connections.

- **Worm Detector:** Includes among other techniques a detection algorithm such as threshold random walk (TRW) [16, 15]. We also envision white hole collaboration, allowing detectors from different white hole spaces to corroborate scanning patterns, similar to the Worminator [1] and DOMINO [11] architectures. Every detector will record scan addresses in a Bloom filter. By exchanging these Bloom filters, we can achieve a privacy-preserving way for distributed attacker scan detection. Also note when using TRW, we can assign different weights on scans inside white holes and scans between different white holes because the later case is more likely a malicious scanner.

The white hole operates by preventing an importance scanning worm from analyzing the group distribution statistics of the legitimate network in which it is co-located. In the critical initial sampling stage of an importance scanning worm, the worm initiator sends sampling scans to the Internet [2], or waits for a certain number of initial infected hosts to report their distribution information [4]. In the first case, response from white hole spaces will be considered as live hosts. In the case of [4], white holes will use RolePlayers to mimic infected hosts and report to the attacker, thus white hole addresses will also be considered as live vulnerable hosts. In both instances, the white holes significantly disrupts the ability of the worm to accurately assess the live address distribution in the white-hole-protected network.

We are also interested in using incoming white hole scans to detect the worm initiator[1], potentially to help filter scans to legitimate addresses within the protect network segment. For the propagation strategy in [2], one approach is to employ Bloom filters to capture common source scanning addresses to the white hole space. For the propagation strategy in [4], in which the attacker waits for existing victims to report information, we can detect the attacker by observing numerous outgoing connections to a common target address in an destination-address Bloom filter. Once a worm initiator is detect, the redirector can use this information to drop scans to legitimate addresses within the protected network. We can also envision sharing bloom filters among among white hole spaces, similar to that of Worminator [1].

Furthermore, we can use a LaBrea [18] like technique in white holes to stick TCP worms. In Section 3.2, we demonstrate that white holes will attract importance scanning worms to enter LaBrea-like network segments earlier in its infection phase, and throughout the epidemic with higher probability. We find that this sticking defense strategy is extremely effective when combined with a white hole to attract the importance scanning worms.

[1] in the paper we do not necessarily assume the detector works in the second (spreading) stage, although that will definitely improve our performance to defeat the worm.

3 Mislead and Defeat Importance Scanning Worms

Before our analysis, we list the notation used in the paper in Table 1. Note there is a slightly difference between worm strategies in [4] and [2]. [4] uses live vulnerable host distribution, while [2] just uses live host distribution. This will not make a fundamental difference in our general analysis framework and results. Specifically, our following analysis is based on [4] (worm using live vulnerable distribution).

Using AAWP (Analytical Active Worm Propagation) model [5], we can model the propagation of a worm as

$$I(t+1) = I(t) + (N - I(t))[1 - (1 - \frac{1}{\Omega})^{sI(t)}]$$

where $I(t)$ is the number of infected hosts at time t, N is the total number of vulnerable hosts on Internet, Ω is the total number of addresses in the scanning space, s is the worm scanning rate.

When a worm begins to spread, $I(t) << N$ and $sI(t) << \Omega$, thus, the AAWP model can be approximated as

$$I(t+1) = I(t) + N\frac{sI(t)}{\Omega} = (1+\alpha)I(t),$$

where $\alpha = \frac{sN}{\Omega}$ is the infection rate [24], which represents the average number of infected vulnerable hosts per unit time by a single worm victim during the early stage of worm propagation. By using address distribution information, the worm can increase its infection rate success. For example, [24] mentioned that a BGP and Class-A routing worm can speed up this infection rate by 3.5 and 2.2 times compared to a regular worm that scans the whole IPv4 space uniformly.

3.1 Infection Rate and Misleading Effect Analysis

The Internet is partitioned into m groups. As shown in [6], the infection rate of an importance-scanning worm is

$$\alpha = sN\sum_{i=1}^{m}\frac{p_g(i)p_g^*(i)}{\Omega_i},$$

Especially, when $p_g^*(i) = p_g(i)$,

$$\alpha = \frac{sN}{\Omega} \times \Omega\sum_{i=1}^{m}\frac{(p_g(i))^2}{\Omega_i},$$

where $\Omega = 2^{32}$. Therefore, importance-scanning worms can increase the infection rate with the factor of $\Omega\sum_{i=1}^{m}\frac{(p_g(i))^2}{\Omega_i}$, compared to random-scanning worms.

Let H_i denote the event that the ith group deploys a white hole that covers U_i white hole addresses.

$$H_i = \begin{cases} 1, & if\ the\ i th\ group\ deploys\ a\ white\ hole \\ 0, & otherwise \end{cases}$$

When a white hole is introduced, from the view of a worm, the number of vulnerable hosts increases from N to $N+U$ (remember all white hole addresses will appear live and vulnerable to the worm in its estimation at first stage, we refer to this as misleading U). When we consider the case where detection and blocking (in the sampling phase) is available (and many networks deploy address blacklisting), we can provide much less real vulnerable information to the worm (we refer to this as misleading N). Thus, for the worm, the final vulnerable hosts are estimated as $N\beta+U$, where β is the correct estimation probability of real vulnerable hosts. With the help of detector and wide deployment of address blacklisting, we could keep β very small.

Thus, a worm estimates the vulnerable-host distribution as following

$$\hat{p}_g(i) = \frac{N_i\beta + U_iH_i}{N\beta + U} \tag{1}$$

When $p_g^*(i) = \hat{p}_g(i)$, and for simplicity, we assume that the white hole is deployed only in group k where $U_k >> N_k$,

$$\begin{aligned} \alpha &= \frac{sN}{\Omega} \times r \times \Omega\sum_{i=1}^{m}\frac{(p_g(i))^2}{\Omega_i} + (1-r)sN\frac{p_g(k)}{\Omega_k} \\ &\approx \frac{sN}{\Omega} \times r \times \Omega\sum_{i=1}^{m}\frac{(p_g(i))^2}{\Omega_i}, \end{aligned}$$

where $r = \frac{N\beta}{N\beta+U}$ and we ignore the last item ($p_g(k)$ is very small as assumed). Therefore, the white hole decreases the infection rate with the factor of $\frac{N\beta+U}{N\beta}$. When $U >> N\beta$, the worm is slowed down through the false information of the vulnerable-host distribution. In fact, we find that even using a relatively *small* white hole, we can still efficiently mislead and defeat and importance-scanning worm.

For importance-scanning worms, their propagation using distribution information can be modeled as following:

$$I_i(t+1) = I_i(t) + (N_i - I_i(t))[1 - (1 - \frac{1}{\Omega_i})^{sI_tp_g^*(i)}]$$

Table 1: Notation used in the paper

N	total number of vulnerable hosts on Internet
N_i	number of vulnerable hosts in group i
m	total number of groups on Internet
$I(t)$	number of infected hosts at time t
$I_i(t)$	number of infected hosts at time t in group i
Ω	total number of addresses in the scanning space
Ω_i	number of addresses in group i
s	scanning rate
α	infection rate
β	correct estimation probability of real vulnerable hosts
$p_g(i)$	percent of the live vulnerable hosts in group i
$p_g^*(i)$	probability of a worm scan hitting group i
U	total number of addresses used by all white holes
U_i	number of addressed covered by the white hole in group i
$K_i(t)$	average number of scans at time t in group i
$K(t)$	total number of scans at time t
$e_i(t)$	average number of newly infected hosts at time t in group i
$e(t)$	total number of newly infected hosts at time t

We show the analytical and simulation results using Matlab in Figure 2. Here we use the real distribution of the Witty worm [26] as the underlying real vulnerable distribution. We simulate a Witty-like worm, with an initial hitlist of ten and scanning rate at 1,200 per unit time.

Figure 2(a)(b) show the results when we only mislead U (as if we do not use worm detectors). Figure 2(a) considers the group size as a /8 network, and (b) for a group size at /16. We can see that in both cases we slow down the importance scanning worm using a variable size of white hole addresses (from 1,200 to 48,000). The performance with group size /16 is worse than /8 when only using misleading. This is because /16 distribution information is definitely more accurate than /8 distribution information. Thus, using a more detailed distribution information will make the worm spread faster. That is why although we use a white hole covering the same space in two cases, the worm still propagates faster in the /16 scenario than in the /8 scenario.

Figure 2(c)(d) shows the cases when we mislead both N and U, with $\beta = 0.1$. This will be much better than only misleading U. From (c) we see that white hole covering only 48,000 addressed can greatly impact the worm's infection growth rate.

3.2 Using Sticking in White Holes

We now consider the effects of incorporating a LaBrea-like service into the white hole to defend against an importance scanning worm (note this only works on TCP worms). We modify the AAWP model:

$$K(t+1) = K(t)\left(1 - p_g^*(k)\frac{U}{\Omega_k}\right) + se(t)$$

$$K_i(t+1) = K(t+1)p_g^*(i)$$

$$e_i(i+1) = (N_i - I_i(t))\left[1 - (1 - \frac{1}{\Omega_i})^{K_i(t+1)}\right]$$

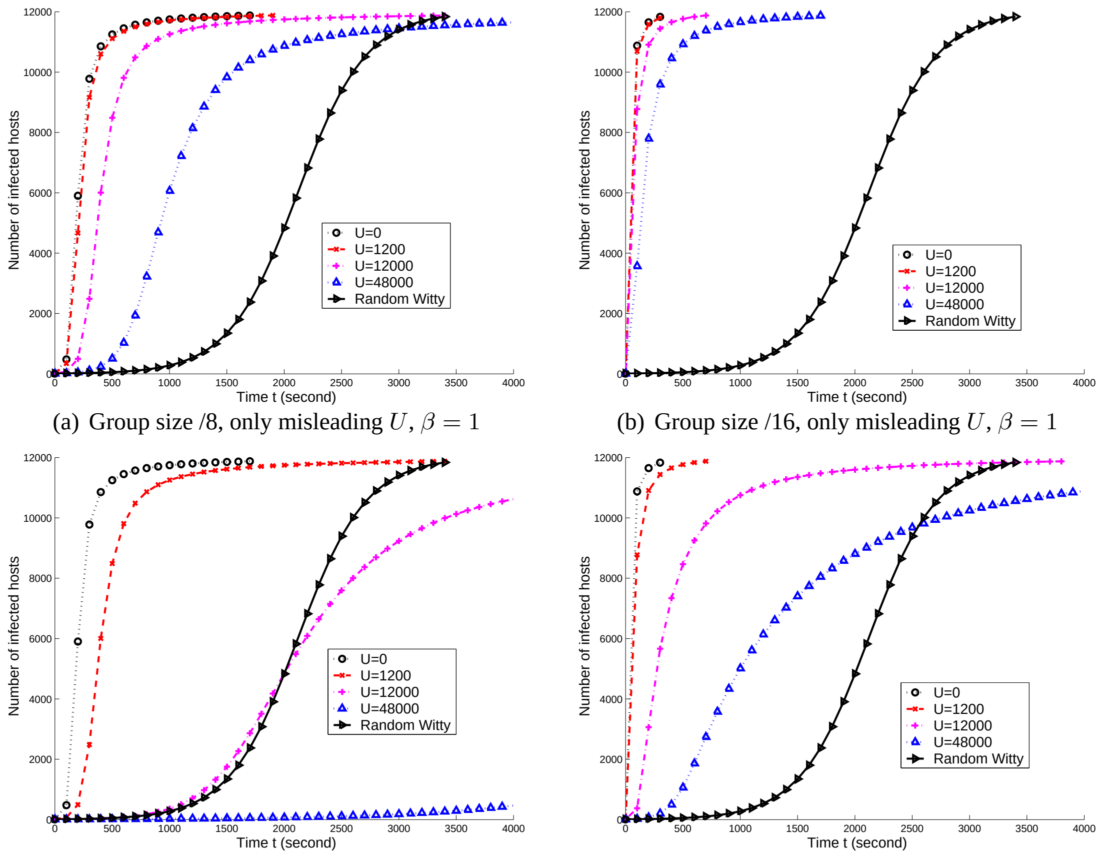
$$e(t+1) = \sum_{i=1}^{m} e_i(t+1)$$

$$I_i(t+1) = I_i(t) + e_i(t+1)$$

$$I(t+1) = \sum_{i=1}^{m} I_i(t+1).$$

where $K_i(t)$ and $e_i(t)$ denote the average number of scans and newly infected hosts at time t in group i (white hole is deployed in group k as before). The results are shown in Figure 3.

From Figure 3(a)(b), we observe that the group size at /16 actually has a better performance than the group size at /8 (opposite to the results from using just misleading as shown in Figure 2). This is because the probability of the white hole being scanned is not changed when using /8 or /16, but the probability for group with size /16 is much reduced



(a) Group size /8, only misleading U, $\beta = 1$

(b) Group size /16, only misleading U, $\beta = 1$

(c) Group size /8, misleading both N and U, $\beta = 0.1$

(d) Group size /16, misleading both N and U, $\beta = 0.1$

Figure 2: Misleading importance-scanning worm using white holes. For simplicity, only *one* white holes is deployed.

than the group with /8. Thus, very likely, before the /16 group is hit, worm is already stuck within the white holes.

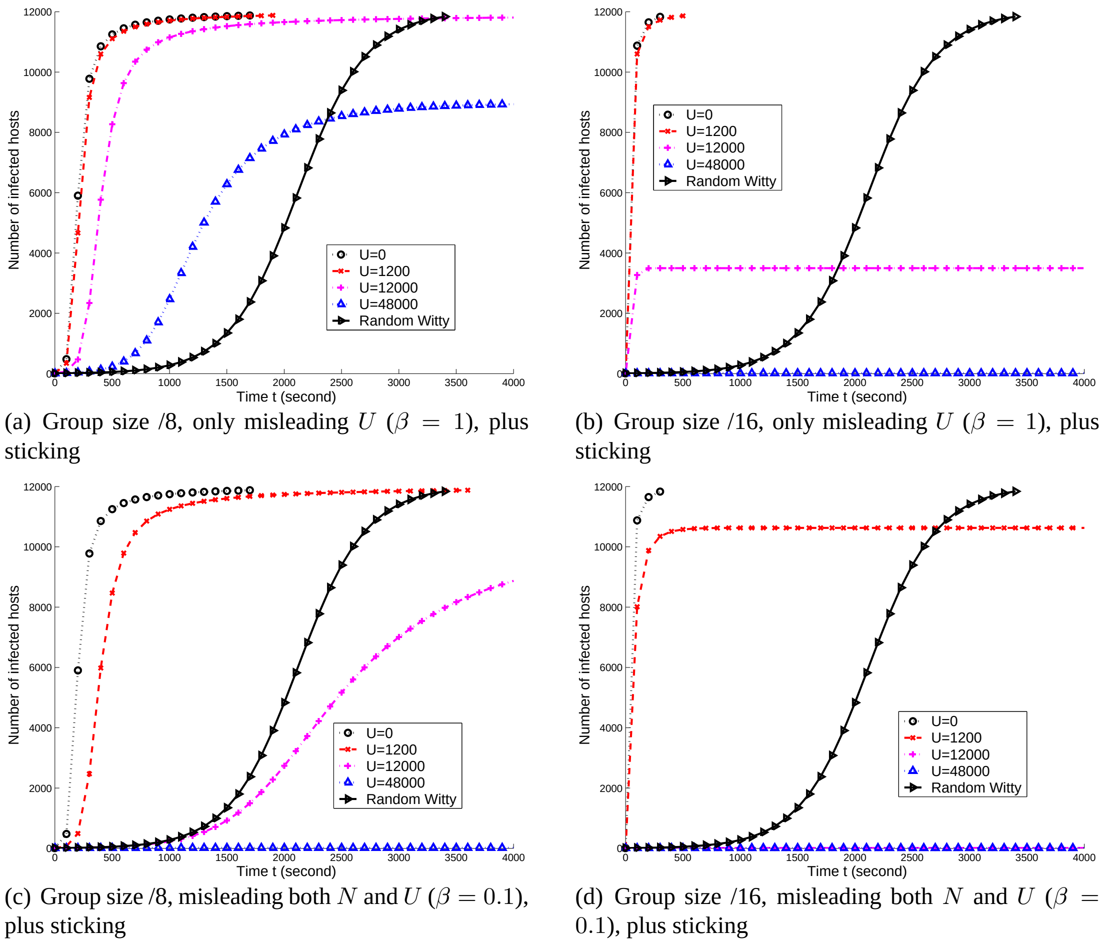
The results suggest that combining a LaBrea-like technique is extremely effective in the context of importance scanning worms in comparison to other worm propagation strategies. For non-importance scanning worms, Chen et al.[5] found that one needs at least 2^{18} LaBrea hosts to effectively defend against an active worm. Here we find that a single white hole covering only 12,000 addresses (Figure 3(d)) is effective in halting the worm. Even without misleading N, we still can use white holes covering 48,000 to defeat the worm efficiently (Figure 3(b)). This is because the bias of importance-scanning will mislead most of the scan efforts onto the white hole space, using the worm's own affinity to densely populated network segments against it to lure in into the LaBrea countermeasure.

4 Related Work

There is an abundance of work covering the use of unused space for worm detection and defense. Most of this research involves passive monitoring techniques, such as Internet Motion Sensor [9], telescope [21], iSink [10]. Some of these systems also employ simple active response to TCP connections, but do not handle further request after TCP handshaking. Their primary purpose is to record and analyze incoming traffic.

There are also approaches that detect worm outbreaks through the use of monitored traffic. Zou et al. propose a Kalman filter based detection [13] for efficient worm early warning. Wu et al. propose a victim number-based approach [13] to detecting the exponential increasing scans by worms.

Honeypot techniques are used to lure attacks, and their functionality can range from simple connection acknowledgement and traffic collection, to full interaction with attackers. Several honeypot projects, such as honeynet[20], honeyd [30], honeyfarm [17], show great potential value for Internet malware study



(a) Group size /8, only misleading U ($\beta = 1$), plus sticking

(b) Group size /16, only misleading U ($\beta = 1$), plus sticking

(c) Group size /8, misleading both N and U ($\beta = 0.1$), plus sticking

(d) Group size /16, misleading both N and U ($\beta = 0.1$), plus sticking

Figure 3: Misleading importance-scanning worm using white holes. The white holes also stick incoming connection (LaBrea-like). For simplicity, only *one* white holes is deployed.

and defense.

Openfire [29] is perhaps the most similar to our white holes, but with different focus and purpose. The white hole technique is designed in the context of addressing importance-scanning worms, with the objective of misleading and ultimately defeating them when coupled with LaBrea-like countermeasures. White holes use several different response and detection techniques in operation. While Openfire focuses on using real decoy machines to reduce general attacks on (relative small) legitimate networks.

5 Discussion and Limitation

There are several challenging issues with the white hole approach, many of them are our future work.

White hole dissuasion vs. attraction: Attackers can fingerprint the existence of white holes by observing that almost all IP/ports in the protected network segment are responsive to connect attempts, which can be a director indicator of a potential probe monitoring system [19]. However, rather than this being a problem, we think it actually provides stronger motivation for adoption of the technique by a wider audience. The white hole effectively masks the legitimate network as a potential low interest address segment rather than a high interest target. We also observe a cumulative affect as more address spaces employ white holes in their networks, which further aids in disrupting worms based on importance scanning. That is, U increases, and the probability of β decreases. Alternatively, white hole owners can also configure their response strategy to closely mimic real network distribution and operation with the intent of making the white hole space operate with characteristics similar to a legitimate network. Here the intent is to construct a network that will produce a higher than average attraction from importance scanning worms, which can be used both the better study the worm attack strategies, and to deploy countermeasures such as a LaBrea system to halt the worm's progress.

Distributed deployment strategy: When we deploy multiple white holes on the Internet, we could employ strategies to deploy the distribution according to our real vulnerable distribution. We plan to study the effect of distribution of white holes in the future (similar to study in [3]).

Scalability: Existing techniques such as [17] demonstrate positive progress on deploying large number of VM-based honeypot. The simple and stateless design of role player also shows positive potentials. We keep most of the components simple, thus we believe scalability is not a big issue, and the simple design will also bring better network pressure tolerance. We will verify this in the future.

Attack tolerance: Worms may collect distribution information using approaches other than sampling, e.g., through address harvesting (SSH, Email, IM, etc.), other channel/out-of-band, to fingerprint live (even vulnerable) hosts. However, these approaches are much slower than sampling, and they are not easy to achieve the whole picture of the Internet. Second, smart evasive worm, such as VM detection [28], honeypot-aware [23], or traffic learning, can identify whether they are within a white hole or not. Of course, future study of defense is definitely needed in this arm race. However, in the *sampling* phase, the *primary* target of the importance scanning attacker is to be stealthy to avoid detection. Honeypot-aware technique[23] will involve more anomaly clues and yield higher risk of being detected.

LaBrea Resistance: Worms may eventually adapt to detect and escape tarpit mechanisms. That is, instead of achieving sustained sticking TCP worms, we should assume we can only stick for a certain time. We plan to do simulations in the future to find out the effect of different sticking capabilities. We should keep in mind that besides LaBrea-like sticking, we have several other defense choices, e.g., address blacklisting, automatic signature generation, etc (a taxonomy on defense techniques is in [31]). Finally, there is a debate on the legacy of using LaBrea-like sticking technique [18], which is non-technical issue out of our control.

We should acknowledge that our proposed white hold strategy is a first step toward addressing emerging network-aware worms, and is a non-trivial component to design and deploy, depending on the depth of features one would want to incorporate. However, we also note that the key features envisioned in white holes represent an integration of existing techniques. We also note that launching of a successful importance scanning worm is also a non-trivial activity.

6 Conclusion

In this paper we propose the design of white holes as a method to respond to a new generation of worm propagation strategies that seek to learn the address distribution statistics of the networks they are attacking. We propose the use of white holes to produce anomalous densities that are characteristic of naive honeynets that will be ignored by worms, in the spirit of hiding trees within a forest. We can also use the detection capabilities within the white hole to dynamically protect co-located legitimate addresses.

We also suggest that the density analysis of importance scanning worms can be used against them, and propose the incorporation of LaBrea-like mechanisms into a white hole that tries to mimic dense legitimate dense networks. We observe that such an approach can rapidly trap the importance scanning worm to a far greater degree than other propagation strategies. Our current assessment of this approach motivates us to continue our study of more strategies to actively mislead and defeat future network-aware worms. Our next step is to implement and deploy such a white hole.

References

[1] M. Locasto, J. Parekh, A. Keromytis, S. Stolfo. Towards Collaborative Security and P2P Intrusion Detection. *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security*, June 2005.

[2] Moheeb Rajab, Fabian Monrose and Andreas Terzis. Fast and Evasive Attacks: Highlighting the Challenges Ahead. *To appear in proceedings of the 9th International Symposium on Recent Advances in Intrusion Detection (RAID)*, Sept, Germany, 2006. (Available as JHU Computer Science Technical Report HiNRG-RMT-112205)

[3] Moheeb Abu Rajab, Fabian Monrose, Andreas Terzis. On the Effectiveness of Distributed Worm Monitoring. *Proceeding of the 14th Usenix Security Symposium.* August, 2005.

[4] Zesheng Chen and Chuanyi Ji. A Self-Learning Worm Using Importance Scanning. Proceedings of the ACM CCS Workshop on Rapid Malcode (WORM'05), October, 2005.

[5] Zesheng Chen, Lixin Gao, and Kevin Kwiat. Modeling the Spread of Active Worms. *Proceeding of INFOCOM.* 2003.

[6] Zesheng Chen and Chuanyi Ji. Optimal worm-scanning method using vulnerable-host distributions. *To appear in the International Journal of Security and Networks (IJSN),* special issue on “Computer & Network Security.”

[7] Weidong Cui, Vern Paxson, Nicholas Weaver, Randy H. Katz. Protocol-Independent Adaptive Replay of Application Dialog. Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, February 2006.

[8] Evan Cooke, Michael Bailey, Farnam Jahanian, Richard Mortier. The Dark Oracle: Perspective-Aware Unused and Unreachable Address Discovery. *Proceedings of 3rd Symposium on Networked Systems Design and Implementation (NSDI’06),* May 2006.

[9] Michael Bailey, Evan Cooke, Farnam Jahanian, Jose Nazario, and David Watson. The Internet Motion Sensor: A distributed blackhole monitoring system. *Proceedings of Network and Distributed System Security Symposium (NDSS’05),* February 2005.

[10] V. Yegneswaran, P. Barford and D. Plonka. On the Design and Utility of Internet Sinks for Network Abuse Monitoring. *Proceedings of Symposium on Recent Advances in Intrusion Detection (RAID),* October, 2004.

[11] V. Yegneswaran, P. Barford, and S. Jha. Global Intrusion Detection in the DOMINO Overlay System. *Proceedings of Network and Distributed Security Symposium (NDSS).* February, 2004.

[12] Cliff C. Zou, Weibo Gong, Don Towsley. Code Red Worm Propagation Modeling and Analysis. *Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS’02),* Nov. 2002

[13] Cliff Zou, Lixin Gao, Weibo Gong, and Don Towsley. Monitoring and Early Warning of Internet Worms. *Proceedings of ACM Conference on Computer and Communications Security (CCS),* October, 2003.

[14] Jiang Wu, Sarma Vanagala, Lixin Gao, and Kevin Kwiat. An Effective Architecture and Algorithm for Detecting Worms with Various Scan Techniques. *Proceedings of the ISOC Network and Distributed System Security Symposium (NDSS),* February, 2004.

[15] Nicholas Weaver, Stuart Staniford, and Vern Paxson. Very Fast Containment of Scanning Worms. *Proceedings of the 13th USENIX Security Symposium,* August 2004.

[16] Jaeyeon Jung, Vern Paxson, Arthur W. Berger, and Hari Balakrishnan. Fast Portscan Detection Using Sequential Hypothesis Testing. *Proceeding of the IEEE Symposium on Security and Privacy,* May 2004.

[17] M. Vrable, J. Ma, J.Chen, D. Moore, E. Vandekieft, A. Snoeren, G. Voelker, and S. Savage. Scalability, Fidelity and Containment in the Potemkin Virtual Honeyfarm. *Proceedings of ACMSIGOPS Operating System Review,* 39(5):148-162, 2005.

[18] LaBrea Tarpit Project, http://labrea.sourceforge.net/.

[19] John Bethencourt, Jason Franklin, and Mary Vernon. Mapping Internet Sensors with Probe Response Attacks. *Proceedings of the 14th USENIX Security Symposium,* August 2005.

[20] Honeynet project. Know your enemy: Learning about Security Threats. Pearson Education, 2004.

[21] David Moore. Network Telescopes: Observing Small or Distant Security Events. *Proceedings of the 11th USENIX Security Symposium,* Invited Talk, August 2002.

[22] Y. Pryadkin, R. Lindell, J. Bannister, R. Govindan. An Empirical Evaluation of IP Address Space Occupancy, *USC/ISI Technical Report ISI-TR-598,* November, 2004.

[23] Cliff C. Zou and Ryan Cunningham. Honeypot-Aware Advanced Botnet Construction and Maintenance. *To appear in the International Conference on Dependable Systems and Networks (DSN),* June 2006

[24] Cliff C. Zou, Don Towsley, Weibo Gong, and Songlin Cai. Routing Worm: A Fast, Selective Attack Worm based on IP Address Information. *Proceedings of the 19th ACM/IEEE/SCS Workshop on Principles of Advanced and Distributed Simulation (PADS’05),* June 2005.

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

[26] C. Shannon and D. Moore. The spread of the Witty worm. *Proceedings of the IEEE Symposium on Security and Privacy,* May, 2004.

[27] U. Shankar and V. Paxson, Active Mapping: Resisting NIDS Evasion Without Altering Traffic. *Proceedings of the IEEE Symposium on Security and Privacy,* May, 2003.

[28] T. Holz, F. Raynal. Detecting honeypots and other suspicious environments. *Proceedings of the Sixth Annual IEEE Systems, Man and Cybernetics (SMC) Information Assurance Workshop (IAW’05),* June 2005.

[29] Kevin Borders, Laura Falk, and Atul Prakash. OpenFire: Opening Networks to Reduce Network Attacks on Legitimate Services. University of Michigan Technical Report CSE-TR-517-06, May, 2006.

[30] Niels Provos. A Virtual Honeypot Framework. *Proceedings of the 13th USENIX Security Symposium,* August, 2004.

[31] David Brumley, Li-Hao Liu, Pongsin Poosankam, and Dawn Song. Design Space and Analysis of Worm Defense Strategies. *Proceedings of the 2006 ACM Symposium on Information, Computer, and Communication Security (ASIACCS),* 2006.