On the Robustness of the Botnet Topology Formed by Worm Infection

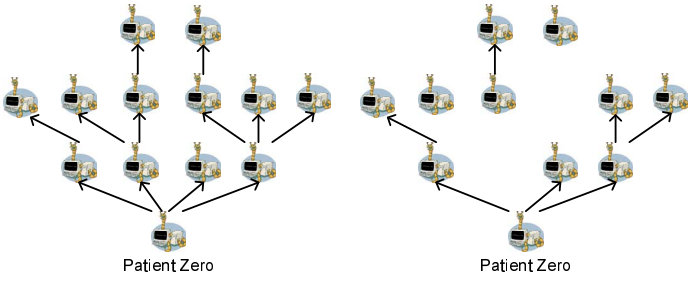
Qian Wang[1], Zesheng Chen[2], Chao Chen[2], and Niki Pissinou[1]
[1]Department of Electrical & Computer Engineering, Florida International University, Miami, Florida 33174
[2]Department of Engineering, Indiana University - Purdue University Fort Wayne, Indiana 46805

***Abstract*—Peer-to-peer botnets formed by worm infection have become a real threat to the Internet and are expected to become rampant in the near future. In our previous work [1], we have analyzed the underlying botnet topology formed by worm infection, without considering potential user defenses. In this paper, we extend the study to characterize the evolution of the botnet structure when users patch or clean part of infected hosts after all vulnerable machines are compromised. Specifically, we examine the number of peers of an infected host and the size of disconnected botnets under random node removal through simulation. We find that when part of infected hosts are patched or cleaned, the distribution of the number of peers follows closely an exponential distribution, whereas the distribution of the size of isolated botnets is power-law. Moreover, we also evaluate a simple countermeasure by botnets that enhances topology robustness through worm re-infection, and show that re-infection can significantly mitigate the effectiveness of patching and cleaning on the botnet structure. We believe that such a study can not only provide better understandings on both the strength and the weakness of botnets, but also better prepare us for future attacks.**

I. Introduction

Since the Morris worm was released in 1988, Internet worms continue to be one of top security threats. For example, the Conficker worm infected 9 to 15 million machines in early 2009 and shut down the service of some critical government and medical networks [2]. Moreover, it constructed a massive peer-to-peer (P2P) botnet. A botnet is a zombie network and is capable of sending denial-of-service attacks, producing spams, and stealing financial information. There are two major types of botnets: Internet relay chat (IRC) based botnets and P2P-based botnets. While IRC-based botnets require central servers for command delivery, P2P-based botnets make use of peer-to-peer systems and can form different command communication networks such as random graphs or power-law topologies [3]. As a result, P2P-based botnets are more resilient to defenses and have plagued the Internet [4].

In our previous work [1], we studied the network structure of P2P-based botnets formed by Internet worm infection. Specifically, we considered that once an infected host compromises another host, they form the "father" and "child" relationship, as shown in Fig. 1(a). Thus, the botnet topology forms a "worm tree", of which the root is the patient zero and the leaves are infected hosts that do not compromise any target. P2P-based botnets formed by worm infection are a real threat. For example, Conficker C uses random scanning to locate peers and forms a P2P botnet through scan-based peer discovery [5], [6]. Thus, the way that Conficker C builds the botnet is in principle the same as worm infection. In our previous work, we found through theoretical analysis that the number of children has asymptotically a geometric distribution with parameter 0.5 and the generation follows closely a Poisson distribution. In this prior work, however, we focused on the process of worm infection and the formation of P2P-based botnets, and did not consider the potential countermeasures from users.



(a) A worm tree. (b) A worm forest.

Fig. 1. P2P-based botnets formed by worm infection.

Users can respond to worm outbreaks by patching or cleaning discovered infected hosts. For example, to counterattack the Conficker worm, Microsoft released a removal guide to clean and patch the Conficker compromised machines after the outbreak of the worm [2]. When an infected host is patched, it becomes invulnerable; and when it is cleaned, it is no longer infectious, but is still vulnerable to worm infection. It is obvious that a patched or cleaned infected host can break its relationships with its father and children in the worm tree. Specifically, when an infected host is patched or cleaned, the corresponding node along with its associated links are removed from the worm tree. As a result, the infection topology is no longer a tree, but a forest, as shown in Figure 1(b). When user countermeasures are considered, therefore, two interesting questions arise: Are patching and cleaning methods effective against P2P-based botnets, and how do user countermeasures affect the botnets formed by worm infection?

To answer these questions, in this paper we extend our previous work to investigate the structure of P2P-based botnets under user countermeasures. Specifically, we consider that a vulnerable host has three states: susceptible, infected, and removed. A susceptible host can become infected through

worm infection. An infected host can either become removed by user patching or become susceptible again by user cleaning. Note that user cleaning is a real method against some worms. For example, a Code-Red infected host becomes susceptible once rebooted [7]. The effectiveness of patching and cleaning against worms has been studied in terms of the total number of infected hosts over time [8], [9]. In this work we focus on the effect of user countermeasures on the P2P-based botnet structure. To characterize the key features of botnet topologies under both worm infection and user countermeasures, we study two important metrics in particular:

- *Number of peers:* For a randomly selected node in the botnet topology, how many peers (*i.e.*, an infected host's father and children) does it have? This metric represents the node degree of individual hosts.
- *Botnet size:* For a randomly selected tree in the forest, how many nodes does it have? This metric represents the size of disconnected botnets after node removal.

These two metrics shed light on the robustness and the effectiveness of formed P2P-based botnets. For example, if a very small number of hosts have a large number of peers and the majority of hosts have none or few peers, such botnets are robust to random defenses, but are vulnerable to targeted defenses (*i.e.*, quarantining the hosts with the largest node degree) [1], [3]. On the other hand, if each host has a similar node degree, then such botnets are robust to both defense schemes [1], [3]. Moreover, the bigger a botnet is, the more effective and dangerous it is [3]. For example, if the forest consists of a collection of small isolated botnets, then its effectiveness is significantly lower than the single connected botnet with the same total number of nodes.

In this paper, we investigate the P2P-based botnet topologies under user patching and cleaning through simulation. We then further study the effectiveness of worm re-infection against user countermeasures. We make the following observations from this research:

- We find that the distribution of the number of peers has an exponential scaling with the decay constant increasing with the number of patched or cleaned hosts. This implies that a small percentage of bots have a large number of peers and the majority of bots have none or few peers. Moreover, the distribution of the disconnected botnet size has a power-law tail with the scaling exponent increasing with the number of patched or cleaned hosts. This reflects that patching or cleaning severely disrupts the single worm tree. We also find that the size of the largest isolated botnet is relatively small. Therefore, P2P-based botnets formed by worm infection are vulnerable to targeted defenses and ineffective due to patching or cleaning.
- We discover that botmasters may potentially enhance the robustness and the effectiveness of P2P-based botnets through worm re-infection.

The remainder of this paper is structured as follows. Section II gives the background of the worm forest and simulation settings. Section III presents the P2P-based botnet structure under user countermeasures. Section IV further studies the effect of worm re-infection against user countermeasures. Finally, Section V concludes this paper.

TABLE I
NOTATIONS USED IN THIS PAPER.

Notations	Definition or explanation
n_0	Total number of vulnerable hosts
r_p	Patching rate: the rate at which an infected or vulnerable machine becomes invulnerable
r_c	Cleaning rate: the rate at which the infection is cleaned on a machine without patching
n_d	Number of hosts that get patched or cleaned
n_r	Number of remaining infected hosts after n_d hosts get patched or cleaned
t_r	Number of trees in the worm forest
$B_{n_0}^{n_d}(i)$	Number of nodes that have i peers
$T_{n_0}^{n_d}(j)$	Number of trees that have j nodes
$b_{n_0}^{n_d}(i)$	Distribution of the number of peers
$t_{n_0}^{n_d}(j)$	Distribution of the botnet size

II. WORM FOREST AND SIMULATION SETTINGS

In this section, we first provide the background of the worm forest and then introduce our simulation settings.

In our previous work [1], we studied the P2P-based botnets topology formed by Internet worm infection without considering user defenses. Specifically, we analyzed the tree structure of P2P-based botnets formed by a wide class of worms starting from patient zero, for which a new victim is compromised by each existing infected host with equal probability. Such worms include well known random-scanning worms, routable-scanning worms, importance-scanning worms, OPT-STATIC worms, and SUBOPT-STATIC worms. Here, we assume that all vulnerable hosts are globally reachable and do not consider the effect of network address translation [10]. In this paper, we construct the worm forest by randomly patching or cleaning hosts in the worm tree studied in [1]. Since most Internet worms spread so fast that existing defense systems cannot respond until they have infected most vulnerable hosts [11], [12], we assume that user patching or cleaning starts when the entire vulnerable population (denoted as n_0) gets infected. We use r_p to denote the patching rate at which a machine is patched and becomes invulnerable, and r_c to denote the cleaning rate at which the infection is cleaned on a machine without patching. Once patched or cleaned, the node and its associated links are then removed from the botnet topology. Suppose that n_d hosts get patched or cleaned, and the number of remaining infected hosts and trees are denoted as n_r and t_r, respectively. We use $B_{n_0}^{n_d}(i)$ $(i = 0, 1, 2, \cdots, n_r - 1)$ to denote the number of nodes that have i peers and $T_{n_0}^{n_d}(j)$ $(j = 1, 2, 3, \cdots, n_r)$ to denote the number of trees that have j nodes. Note that $\sum_{i=0}^{n_r-1} B_{n_0}^{n_d}(i) = n_r$, and $\sum_{j=1}^{n_r} T_{n_0}^{n_d}(j) = t_r$. Moreover, $B_{n_0}^{n_d}(i)$ and $T_{n_0}^{n_d}(j)$ are random variables. Thus, we define $b_{n_0}^{n_d}(i) = \frac{\mathrm{E}[B_{n_0}^{n_d}(i)]}{n_r}$ to represent the distribution of the number of peers and $t_{n_0}^{n_d}(j) = \frac{\mathrm{E}[T_{n_0}^{n_d}(j)]}{t_r}$ to represent the distribution of the botnet size. Note that the worm tree is a special case of the worm forest when $n_d = 0$ (*i.e.,* without user defenses).

For such a tree, we have

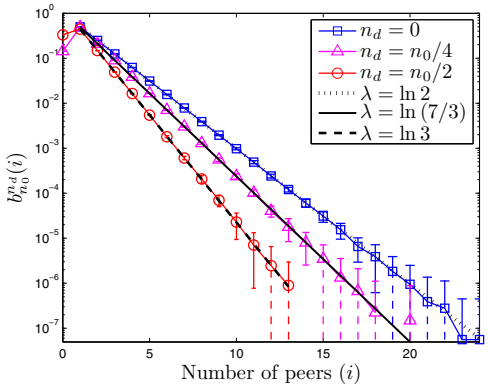
$$\lim_{n_0 \to \infty} b^0_{n_0}(i) = (\tfrac{1}{2})^i, \quad i = 1, 2, 3, \cdots \tag{1}$$

by extending the result in our previous work [1]. While our previous work only considers the number of children, this paper studies the number of peers including both the father and children. Therefore, in P2P-based botnets formed by worm infection without user countermeasures, the distribution of the number of peers has asymptotically a geometric distribution with parameter 0.5, and decreases exponentially with a decay constant of $\ln 2$. Moreover, Since there is only one botnet, we then have the distribution of the botnet size $t^0_{n_0}(n_0) = 1$. Table I summarizes the notations used in this paper.

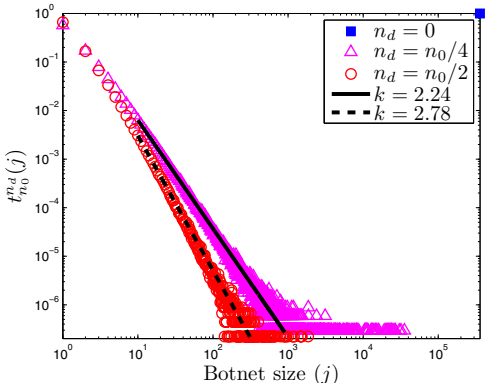
To investigate the P2P-based botnet topology under user patching and cleaning, in this paper we study $b^{n_d}_{n_0}(i)$ and $t^{n_d}_{n_0}(j)$ through simulations. As far as we know, there is no publicly available data to show the real botnet topologies. Moreover, the complex dynamics of patching and cleaning make the botnet structure difficult to be charaterized analytically. Therefore, we apply Monte Carlo simulation. Monte Carlo simulation is widely applied in probability modeling and is the only viable method for the modeling of many complex stochastic systems [13]. Specifically, we simulate a P2P-based botnet formed through worm infection by using and extending the simulator in [14]. The simulator considers a discrete-time system and mimics the random-scanning behavior of infected hosts during each discrete time interval. Moreover, the parameter setting is based on the Code Red v2 worm's characteristics. For example, the vulnerable population is $360,000$ (*i.e.,* $n_0 = 360,000$). A worm selects targets in the IPv4 address space randomly, and a newly infected host is assigned with a scanning rate $s = 358$ scans/min. Detailed information about how the parameters are chosen can be found in Section VII of [15]. We then extend the simulator to mimic the dynamics of user countermeasures and capture the resulting botnet structure. Specifically, after all vulnerable machines get compromised, we randomly patch or clean hosts with $r_p = 2 \times 10^{-5}$/sec or $r_c = 2 \times 10^{-5}$/sec. We also record the information of the number of peers and the botnet size to track the botnet structure. Moreover, we set the time unit to 20 seconds and start our simulation at time tick 0 with patient zero. The simulation results are obtained from 100 independent runs with different seeds.

III. P2P-based Botnet Structure under User Countermeasures

In this section, we present the P2P-based botnet structure under user countermeasures. Specifically, we examine the distributions of the number of peers and the botnet size under three different defense schemes: host patching only, host cleaning only, and host patching/cleaning schemes. The results are shown in Fig.s 2-4. Scaling parameters λ and k are estimated through regression analysis on empirical data by using the Matlab curve fitting toolbox [16], and the coefficient of determination R^2 is very close to 1 for all estimates.



(a) Number of peers.



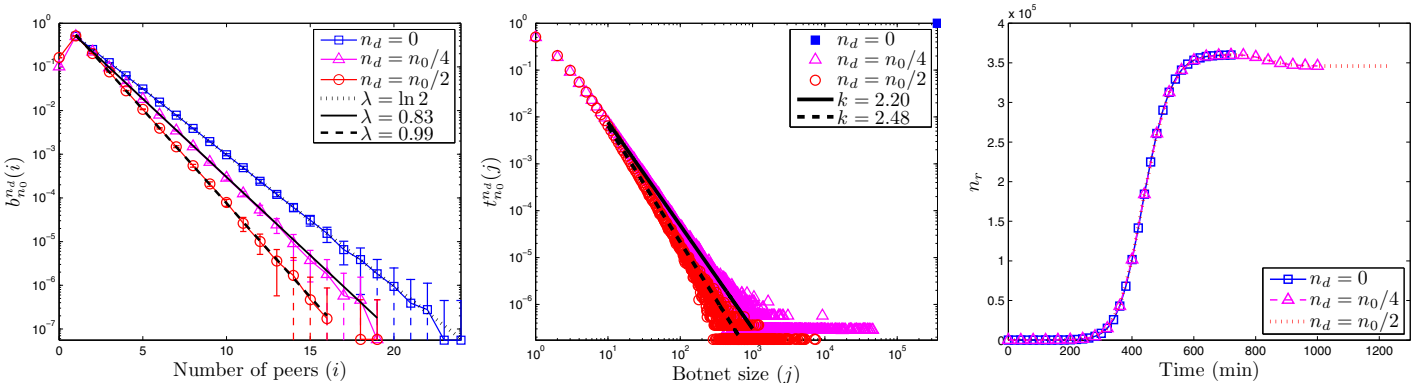
(b) Botnet size.

Fig. 2. Host patching only scheme.

A. Host Patching Only Scheme

Under this defense scheme, we begin to randomly patch infected hosts with $r_p = 2 \times 10^{-5}$/sec after all vulnerable machines get infected. Once patched, an infected host becomes invulnerable, and the node and its associated links are removed from the worm forest. We then examine the P2P-based botnet structure when n_d hosts get patched. The results are shown in Fig. 2.

Fig. 2(a) shows the distribution of the number of peers, comparing the simulation results of $b^{n_d}_{n_0}(i)$ for $n_d = 0$, $n_0/4$, and $n_0/2$ with the exponential scaling obtained through regression. Note that the y-axis uses the log-scale and the error bar represents the standard deviation over 100 runs. The dotted line represents the standard deviation that goes into the negative territory. It can be seen that the distribution of the number of peers has an exponential tail. Specifically, without user defenses (*i.e.,* when $n_d = 0$), $b^0_{n_0}(i)$ can be well approximated by the geometric distribution with parameter 0.5 shown in Equation (1), and therefore decreases exponentially with the decay constant $\lambda = \ln 2$. However, as infected hosts get patched, nodes that do not have any peer emerge in the forest. Moreover, when n_d increases, $b^{n_d}_{n_0}(i)$ still has an exponential tail, but decays faster. This is because when more infected hosts get removed, there are fewer hosts with a large node degree and more hosts becoming isolated nodes without any peer. On one hand, the exponential scaling of $b^{n_d}_{n_0}(i)$



(a) Number of peers. (b) Botnet size. (c) Number of infected hosts.

Fig. 3. Host cleaning only scheme.

implies that after random patching, a small portion of bots still have a large number of peers and the majority of bots have none or few peers. For example, when $n_d = n_0/2$, on average over 99.7% of bots have no more than five peers. On the other hand, an increasing decay constant indicates that the node degree of a bot decreases due to patching. For example, the average node degree decreases from 2 when $n_d = 0$ to 1 when $n_d = n_0/2$. Moreover, through extensive regression analysis, we find that after user patching, in the resulting P2P-based botnet topology, the decay constant $\lambda \approx \ln((n_0 + n_r)/n_r)$, where $n_r = n_0 - n_d$. For example, when half of infected hosts are patched, $b_{n_0}^{n_d}(i)$ decreases exponentially with a decay constant approximately of $\ln 3$.

Fig. 2(b) demonstrates the distribution of the botnet size, comparing the simulation results of $t_{n_0}^{n_d}(j)$ for $n_d = 0$, $n_0/4$, and $n_0/2$ with the power-law tails obtained through regression. Note that the x- and y-axes use the log-scale. It can be seen that when $n_d = 0$, $t_{n_0}^{0}(n_0) = 1$. That is, without patching, worm infection forms a single botnet with n_0 nodes. However, with infected hosts being patched, the distribution of the botnet size has a power-law tail. Moreover, when n_d increases, the scaling exponent k becomes larger. This is because as we patch more infected hosts, the number of trees in the forest increases, whereas the maximum size of trees decreases. For example, when $n_d = n_0/2$, on average there are $90,011$ trees[1] in the forest with an average size of 2 nodes. The average maximum tree size is 622 nodes, comprising less than 0.04% of infected hosts in the forest. Therefore, the size of the largest botnet is relatively small, indicating that patching infected hosts severely disrupts the single botnet formed by worm infection.

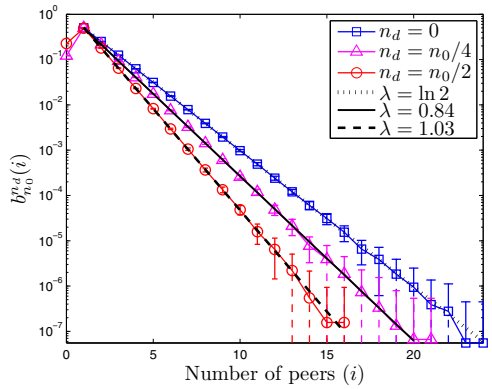
After performing sensitivity analysis on the parameter r_p when n_d is fixed, we find that the patching rate does not affect the botnet structure.

B. Host Cleaning Only Scheme

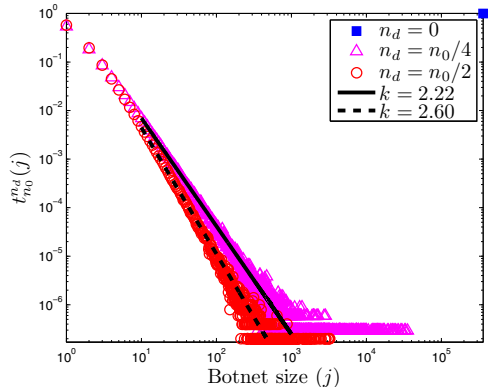
Under this defense scheme, we begin to randomly clean infected hosts with $r_c = 2 \times 10^{-5}$/sec after all vulnerable machines get compromised. Once cleaned, an infected host becomes susceptible, and the host and its associated links are removed from the forest. Note that different from patching, cleaned infected hosts can be compromised again and rejoin the forest. We then examine the P2P-based botnet structure when n_d hosts get cleaned. The results are shown in Fig. 3.

Fig.s 3(a) and (b) show the results of the distributions of the number of peers and the botnet size. It can be seen that $b_{n_0}^{n_d}(i)$ still has an exponential decay and $t_{n_0}^{n_d}(j)$ has a power-law tail. As a result, after user cleaning, a small portion of bots still have a large number of peers, and the majority of bots have none or few peers. For example, when $n_d = n_0/2$, the average node degree of bots is 1.36, and on average about 99.3% of them have a node degree of no more than five. Moreover, cleaning infected hosts severely disrupts the single botnet formed by worm infection. For example, when $n_d = n_0/2$, on average there are $110,740$ disconnected botnets in the forest with an average size of 3 nodes. The average maximum size of the disconnected botnets is $2,954$ nodes, comprising about 0.85% of the remaining infected hosts in the forest. However, compared with the patching only scheme, the exponential and power-law scaling parameters under the host cleaning only scheme are smaller. This is due to the different nature of patching and cleaning. Under the host cleaning only scheme, when n_d hosts are cleaned, some of them get compromised again and rejoin the worm forest. As a result, the number of remaining infected hosts in the forest $n_r > (n_0 - n_d)$. Comparatively, under the host patching only scheme, when n_d nodes are patched, $n_r = n_0 - n_d$. Therefore, as expected, the host cleaning only scheme less disrupts the botnet structure than the host patching only scheme. Moreover, as shown in Fig. 3(c), we find that under the host cleaning only scheme, on average n_r stabilizes at around $345,950$. This happens when the number of nodes being cleaned, $n_r \cdot r_c$, is about the same with the number of susceptible hosts getting infected again, $(n_0 - n_r) \cdot p_i$, where $p_i = n_r \cdot s \cdot \frac{1}{2^{32}}$ is the probability of a susceptible host being compromised. Setting $n_r \cdot r_c = (n_0 - n_r) \cdot n_r \cdot s \cdot \frac{1}{2^{32}}$, we then obtain that the number of nodes in the worm forest will stabilize at $n_r = n_0 - \frac{r_c}{s} \cdot 2^{32}$. For example, with $r_c = 2 \times 10^{-5}$/sec and $s = 358$ scans/min,

[1]We consider that isolated nodes without any peer are a special tree of size one.



(a) Number of peers.



(b) Botnet size.

Fig. 4. Host patching/cleaning scheme.

$n_r = 345,603$, which is very close to our simulation result. In the figure, we also find that n_r is about the same for the cases of $n_d = n_0/4$ and $n_0/2$. However, $b_{n_0}^{n_d}(i)$ and $t_{n_0}^{n_d}(j)$ of the case $n_d = n_0/2$ has larger scaling parameters. This is due to the fact that hosts with a large number of peers might get cleaned, whereas susceptible hosts rejoin the forest as leaves with a node degree of one. As a result, although the number of infected hosts stabilizes at the same level, the host cleaning process decreases the node degree of infected hosts over time and further disrupts the worm forest. Furthermore, we find that the cleaning rate r_c has little effect on the botnet structure when n_d is fixed. On one hand, a smaller cleaning rate corresponds to a larger stabilized botnets population n_r. One the other hand, it takes more time to clean n_d nodes with a smaller cleaning rate.

C. Host Patching/Cleaning Scheme

Under this defense scheme, we consider both user patching and cleaning, which is the case in real world scenarios. Specifically, we begin to randomly clean infected hosts with $r_c = 2 \times 10^{-5}$/sec after all vulnerable hosts get compromised. Meanwhile, susceptible and infected hosts are randomly patched with $r_p = 2 \times 10^{-5}$/sec. We then examine the P2P-based botnet structure when n_d hosts get patched or cleaned. The results are shown in Fig. 4. It is intuitive that the distributions of the number of peers and the botnet size exhibit the combined effects of the host patching only and the host cleaning only schemes. Specifically, the exponential decay constant λ and the power-law scaling exponent k are smaller than those under the host patching only scheme but greater than those under the host cleaning only scheme. For example, when $n_d = n_0/2$, the average node degree of bots is 1.21, and on average about 99.5% of them have no more than five peers. Moreover, on average there are $100,535$ disconnected botnets in the forest with an average size of 2.5 nodes. The average maximum size of the disconnected botnets is $1,636$ nodes, comprising about 0.64% of the remaining infected hosts in the forest.

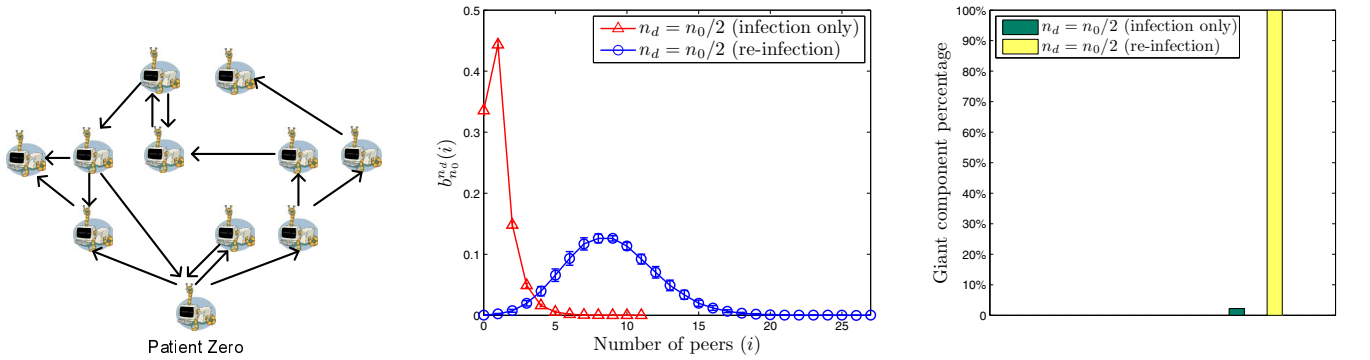
The simulation results of all three defense schemes show that when users patch or clean part of infected hosts, P2P-based botnets formed by worm infection suffer two weaknesses. First, the botnets are highly centralized to a small percentage of the "hub" bots that have a large node degree, and thus vulnerable to targeted defenses [1], [3]. Second, the single botnet formed by worm infection is severly disrupted into a collection of small isolated low-effective botnets.

IV. P2P-based Botnets Formed by Worm Re-infection

In this section, we study a potential countermeasure by future botnets to combat against user patching or cleaning.

A simple potential countermeasure for botmasters to construct more robust and effective P2P-based botnets is through worm re-infection. That is, if an infected host is hit by a worm scan, this host will be further re-infected and become a peer of the infector. As a result, the remaining bots may have a balanced node degree and stay well connected even when some infected hosts get patched or cleaned (see Fig. 5(a)). Note that different from the botnet formed by re-infection discussed in [17], in our P2P-based botnet, there is no exchange of peers between bots. Infected hosts are only peers to their own infectors and infectees.

To show the effectiveness of worm re-infection on P2P-based botnets against user patching or cleaning, we consider the host patching only scheme, which is the worst case scenario. As shown in Section III, under the host patching only scheme, $b_{n_0}^{n_d}(i)$ and $t_{n_0}^{n_d}(j)$ have the largest scaling parameters among the three schemes, and therefore the resulting P2P-based botnets are least robust and effective. In Fig.s 5(b) and (c), we compare the network structure of botnets formed by worm infection only and by worm re-infection when n_d hosts get patched. Here, the vulnerable population n_0 is set to $10,000$. All other parameters ramain the same as the ones used in Section III. Moreover, for worm re-infection, once a vulnerable host gets compromised, it is open for re-infection from the next time tick. We begin to randomly patch infected hosts with $r_p = 2 \times 10^{-5}$/sec when all vulnerable machines get compromised. Once patched, the infected host becomes invulnerable, and the host and its associated links are then removed from the botnet topology. Fig. 5(b) shows the distribution of the number of peers. It can be seen that in the P2P-based botnet formed by worm re-infection, when



(a) Worm re-infection topology. (b) Number of peers (host patching only). (c) Giant component percentage (host patching only).

Fig. 5. P2P-based botnets formed by worm re-infection.

half of infected hosts get patched, $b_{n_0}^{n_d}(i)$ has a bell shape and therefore the node degree of a bot is more evenly distributed. For example, on average 92.56% of bots have a node degree between 5 and 15, and the average node degree of bots is 9. On one hand, such a botnet is resilient to both random and targeted defenses [1], [3]. On the other hand, the P2P-based botnet formed by worm re-infection may have an average node degree similar to other P2P networks [18]. As a result, it may appear to have normal P2P traffic and can potentially avoid detection [3]. Moreover, by further connecting to other bots, the P2P-based botnet formed by worm re-infection stays well connected. In [3], Dagon *et al.* used the giant component or the size of the largest connected botnet to measure the effectiveness. In Fig. 5(c), we show the percentage of the giant component to the available bots. It can be seen that for the botnets formed by worm re-infection, almost all of the remaining bots are connected, whereas the giant component of the botnets formed by worm infection comprises only 2.2% of the remaining infected hosts. Therefore, P2P-based botnets formed by worm re-infection are much more robust and effective than those formed by worm infection only.

V. Conclusions

In this paper, we attempt to characterize the network structure of P2P-based botnets formed by worm infection under user countermeasures. We have shown that when part of infected hosts are randomly patched or cleaned after all vulnerable hosts get compromised, the distribution of the number of peers of a bot has an exponential scaling and the distribution of the size of disconnected botnets has a power-law tail. As a result, a very small percentage of bots have a large number of peers, and the majority of bots have none or few peers. Moreover, patching or cleaning severely disrupts the single botnet formed by worm infection, and the size of the largest isolated botnet is relatively small. Therefore, P2P-based botnets formed by worm infection are vulnerable to targeted defenses and ineffective due to patching or cleaning. We have then applied the observations to design future botnets and found that botmasters can significantly enhance the robustness and the effectiveness of P2P-based botnets through worm re-infection.

Acknowledgment

This work was supported by FIU Dissertation Year Fellowship.

References

[1] Q. Wang, Z. Chen, and C. Chen, "Charaterizing Internet Worm Infection Structure," Preprint. [Online]. Available: http://arxiv.org/abs/1001.1195.

[2] Wikipedia. Conficker. [Online]. Available: http://en.wikipedia.org/wiki/Conficker.

[3] D. Dagon, G. Gu, C. Lee, and W. Lee, "A Taxonomy of Botnet Structures," in *Proc. 23 Annual Computer Security Applications Conference (ACSAC'07)*, Dec. 2007.

[4] P. Wang, L. Wu, B. Aslam, and C. C. Zou, "A Systematic Study on Peer-to-Peer Botnets," in *Proc. International Conference on Computer Communications and Networks (ICCCN)*, Aug. 2009.

[5] P. Porras, H. Saidi, and V. Yegneswaran, "Conficker C P2P Protocol and Implementation," *SRI International Technical Report*, Sept. 2009.

[6] CAIDA. Conficker/Conflicker/Downadup as seen from the UCSD Network Telescope. [Online]. Available: http://www.caida.org/research/security/ms08-067/conficker.xml.

[7] D. Moore, C. Shannon, and J. Brown, "Code-Red: a Case Study on the Spread and Victims of an Internet Worm," in *Proc. ACM SIGCOMM/USENIX Internet Measurement Workshop*, Nov. 2002.

[8] Z. Chen, L. Gao, and K. Kwiat, "Modeling the Spread of Active Worms," in *Proc. IEEE INFOCOM*, Apr. 2003.

[9] C. C. Zou, W. Gong, and D. Towsley, "Code Red Worm Propagation Modeling and Analysis," in *Proc. 9th ACM Conference on Computer and Communication Security (CCS'02)*, Nov. 2002.

[10] M. A. Rajab, F. Monrose, and A. Terzis, "On the Impact of Dynamic Addressing on Malware Propagation," in *Proc. 4th ACM Workshop on Recurring Malcode*, Nov. 2006.

[11] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the Slammer Worm," *IEEE Security and Privacy*, vol. 1, no. 4, pp. 33–39, Jul. 2003.

[12] C. Shannon and D. Moore, "The Spread of the Witty Worm," in *IEEE Security and Privacy*, vol. 2, no. 4, Jul-Aug 2004, pp. 46–50.

[13] M. M. Meerschaert, "Mathematical Modeling (Third Edition)," *Academic Press*, 2007.

[14] C. C. Zou. Internet Worm Propagation Simulator. [Online]. Available: http://www.cs.ucf.edu/~czou/research/wormSimulation/simulator-codered-100run.cpp.

[15] C. C. Zou, W. Gong, D. Towsley, and L. Gao, "The Monitoring and Early Detection of Internet Worms," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 967–974, Oct. 2005.

[16] The MathWorks. Matlab Curve Fitting Toolbox. [Online]. Available: http://www.mathworks.com/products/curvefitting/.

[17] P. Wang, S. Sparks, and C. C. Zou, "An Advanced Hybrid Peer-to-Peer Botnet," in *Proc. First Conference on First Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, Apr. 2007.

[18] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-Peer Networks," in *Proc. 16th International Conference on Supercomputing (ICS'02)*, Jun. 2002.