Simulation and Analysis on the Resiliency and Efficiency of Malnets

Jun Li
University of Oregon
lijun@cs.uoregon.edu

Toby Ehrenkranz
University of Oregon
tehrenkr@cs.uoregon.edu

Geoff Kuenning
Harvey Mudd College
geoff@cs.hmc.edu

Peter Reiher
University of California, Los Angeles
reiher@cs.ucla.edu

Abstract

Future network intruders will probably use an organized army of malicious nodes (here called "malnodes", or collectively a "malnet") to deliver many different attacks, rather than recruiting a disorganized set of compromised nodes per attack. However, partly due to the lack of understanding of the resiliency and efficiency a malnet can have, countering malnets has been ineffective.

This paper begins to address this deficiency. Through calculation and simulation for three representative malnets—random, small-world, and Gnutella-like—we show that extremely resilient malnets can be formed to deliver attack code quickly. In particular, we show that disconnecting malnets is possible, but extremely naive approaches such as randomly disinfecting malnodes will not suffice, and effective defenses must either happen very quickly during a second-wave attack, or take effect prior to it.

1. Introduction

An increasingly important problem in network security is the emergence of large numbers of networks of malicious nodes (here called "malnodes", or collectively a "malnet"). A single malnet can be used repeatedly for various nefarious purposes, such as launching DDoS (distributed denial-of-service) attacks, sending spam, or simply stealing computing cycles. Although such networks are not new, recent malnets have increased in number and sophistication.

For example, trinoo, a distributed denial-of-service attack tool, builds a simple three-layer *trinoo network* [1] in which the attacker controls one or more "master" servers, each master controls many "daemons,"

and the daemons are all instructed to coordinate an attack against one or more victim systems (Figure 1(a)).

Botnets and their variants [2] can also be harmful. For example, a botnet can use IRC channels to connect a collection of IRC bots, where each bot is executable (malicious) code on an IRC client [3]. The study in [4] reported two main types of IRC botnet structures: the *Hub-Leaf* structure in which all bots connect through a hub, resulting in a star architecture (Figure 1(b)), and the *Channel* structure in which a bot needs to join an IRC channel to listen to commands issued by the controller (Figure 1(c)). According to [2], security experts identify botnets with 10 to 100 compromised hosts several times a day, and botnets with 10,000 or more hosts weekly. Botnets with 100,000 computers have also been found.

It is also known that malnodes of a worm can form a *worm network* through which an attacker can issue commands and perform remote control [6]. In this worm network, every malnode keeps a list of other worm malnodes, and can create encrypted communication channels with them; therefore, the command from the attacker can be injected into any malnode and then propagated further toward all remaining malnodes. Furthermore, redundancy can be used to keep the worm network connected even if some malnodes are disinfected and thus removed from the network.

In this paper, we generalize all of these networks as "malnets", which are overlay networks of malnodes. A malnet can be built by malicious code (such as a worm or a Trojan horse) during its infection phase. Further overlay construction can continue even if that malicious code stops propagating.

These malnets can be very sophisticated. Once an intruder has recruited and organized its malnode army, the overlay network it has built can serve as a superhighway for code propagation, including the distribution of upgraded versions for maintaining the malnet it-
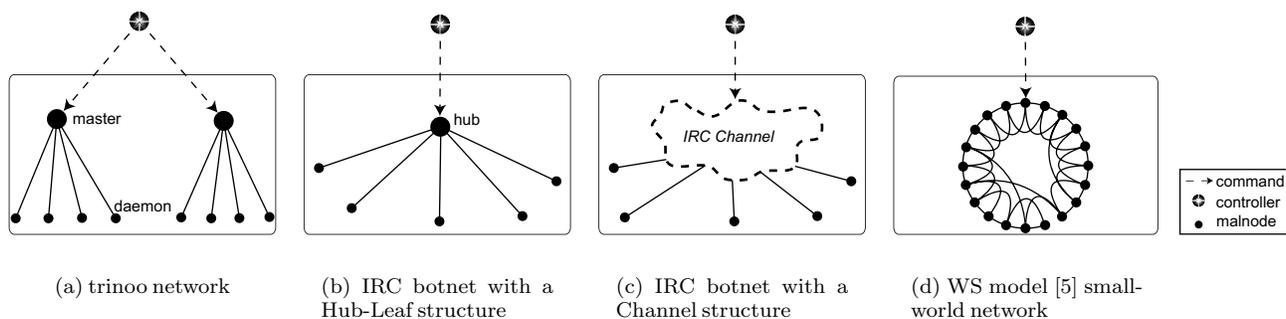
Figure 1. Example malnets.

(a) trinoo network

(b) IRC botnet with a Hub-Leaf structure

(c) IRC botnet with a Channel structure

(d) WS model [5] small-world network

self. In addition to the first wave of attack in which vulnerable nodes are converted to malnodes and recruited into the malnet, second-wave attacks can be launched. The new danger of such a malnet is that it offers the attacker a surreptitious "anytime, anywhere, any flavor" capability: attacks can be launched at a moment's notice, injected at any point in the overlay, and can be crafted to any purpose. Attackers could create a network of compromised machines that would allow them to perform arbitrary, "useful" activities. In addition to sending spam and launching DDOS attacks, for instance, malnets could be used to break cryptographic keys by brute force, store and distribute stolen software or data for fun and profit, or allow malicious "customers" to rent the malnet to perform any desired activity that requires a large number of nodes.

Furthermore, malnets not only can perform dangerous operations, but can also employ mechanisms to manage their own network structures. For example, a malnet might constitute itself as a small-world network (see Figure 1(d)). Recently, malnets have also been found [7] that utilize WASTE [8], an open source P2P communication framework that allows anonymous, secure, and encrypted communication. Malnets can maintain themselves so that they can easily incorporate newly recruited malnodes or forget old departed ones. Moreover, even if they are large networks of many malnodes, malnets can have small diameters so that disseminating commands or new exploits is fast. They can also be made resilient so that disinfecting a subset of malnodes might not disconnect the whole malnet. As a result, those malnets could become hard to defeat.

In this paper, we first justify why it is important to understand the efficiency and resiliency of malnets, and then use calculation and simulation to study the efficiency and resiliency characteristics of three types of malnets. The implications of the simulation results

will also be discussed. We will also present some related work and then conclude the paper.

2. The Importance of Understanding the Efficiency and Resiliency of Malnets

In order to defend against malnets, effective solutions must be designed. Note that the focus of this paper is not to discuss approaches to the difficult problem of defense. Instead, we touch upon defense here only to justify our study of two important properties of malnets: *efficiency* and *resiliency.*

The life cycle of a malnet has many interesting phases, each of which offers defensive opportunities. We can attempt to limit the growth of malnets; we can find and disinfect the malnodes; or we can insulate uninfected machines from the damage that existing malnodes try to inflict.

Limiting the growth of malnets and insulating the uninfected from infection are similar to combating the spread of worms or other malware. These approaches are currently of limited efficacy, and are not expected to be completely effective any time in the near future. At best, these mechanisms will cut down on the size of malnets, but will not eliminate them.

Thus, since malnets can be formed even in the face of defensive measures, it is critical to be able to locate and disinfect malnodes. Malnet-specific approaches such as the following offer the hope of at least some success, either individually or in combination:

- Searching for malnet command listeners (since malnodes need to listen to malnet commands)

- Searching for heartbeats (since neighboring malnodes in sophisticated malnets will probably periodically ensure the liveness of their neighbors)

- Traffic analysis (since malnodes will come alive in second-wave attacks and will exchange data with their neighbors)

- Tracing malnets (since each malnode must maintain certain information about other malnodes)

- Having users inspect their computers and look for malnets

Many defensive approaches will have to rely on the detection of malnodes in the middle of second-wave attacks. Examples include searching for malnet command listeners, traffic analysis, and tracing. As a second-wave attack propagates, the sooner defenders can detect the propagation, the sooner they can search as many listeners as possible, conduct thorough traffic analysis, and trace malnets to find more malnodes. Clearly, understanding the efficiency of malnets is essential when considering the defensive reaction-time requirement. If the spread of malnet commands is sufficiently slow, early detection could act as a warning allowing defenders to effectively preempt the attack. If the spread is very fast, there may not be enough time for a warning to be of much use.

Furthermore, assuming certain malnodes can be found, they should then be cleaned. Unfortunately, past experience suggests that while many owners of infected machines are eager to disinfect them as soon as the problem is discovered or reported to them, a significant number of them are either unaware of the problem, unable to perform disinfection, or unconcerned by the infection. For instance, there are still large numbers of machines infected with CodeRed years after its introduction to the Internet [9, 10]. Therefore, if malnets will be formed anyway (although perhaps to a lesser degree), we probably will not be able to clean all the malnodes on a malnet, so it will be important to know how many malnodes must be disinfected to achieve a certain level of disconnection or partitioning. This naturally leads to studying the resiliency of malnets, which is the focus of the present paper.

In Sections 3 and 4, we try to understand the efficiency and resiliency of malnets by simulating three different types of malnets:

- *Random malnet* where every malnode randomly adds r (a constant) malnodes on average to its neighborhood.

- *Small-world malnet* where a malnet is actually a small-world graph.

- *Gnutella-like malnet* where malnodes are connected through a Gnutella-like peer-to-peer network.

These three malnets actually map to three possible paradigms for forming malnets by attackers. Small-world malnets have strong local clustering characteristics, corresponding to the locality feature when a malnet grows. This is particularly true when the infecting code first explores vulnerabilities in the same subnet or administrative domain before attempting to propagate to other places.

In contrast, random malnets do not rely on local knowledge; instead, every malnode infects a certain number of victims throughout the entire Internet. As an example construction paradigm for random malnets, every vulnerable node might choose 1–5 other nodes throughout the network with the same vulnerability (here, the amount of probing needed to discover these nodes is irrelevant to our analysis).

Gnutella-like malnets have a more stringent formation requirement: a two-level hierarchy will be formed, where "ultrapeer" nodes have a relatively high connectivity to each other, and every leaf node connects to a small number (say 3) of ultrapeers.

3. Efficiency Analysis of Malnets

When the controller of a malnet wishes to launch a second-wave attack, how long does it take for the attack code or command to reach all the malnodes through the malnet? We answer this question by studying random malnets, small-world malnets, and Gnutella-like malnets in the following. As the dissemination speed is directly related to the distance between two malnodes, our study will focus on the diameter of a given malnet.

3.1. Random malnet

Through simulation, we estimated how long it would take for exploit code or a command from an attacker to reach all the nodes on a malnet. Figure 2(a) shows the average and maximum diameter of random malnets in which every malnode has an average of four neighbors, i.e. $r = 4$. (4 is a typical number; for example, [6] reports the average node degree to be 4–5.5 in a 1-million-node worm network formed through permutation scanning.) We have found that the maximum and average hop count between any two malnodes closely follow a logarithmic trend with respect to the number of malnodes. Assuming the trend is sustained when a malnet has 1 million nodes, at most 17 hops are needed between any two nodes. Plugging in the average latency between two nodes on the Internet and considering factors such as congestion, our study shows that starting from any member node, the mas-

| (a) random malnet (average node degree 4) | (b) small-world malnet (average node degree 4) | (c) Gnutella-like malnet (only ultrapeer level presented, average node degree 30 for ultrapeers) |
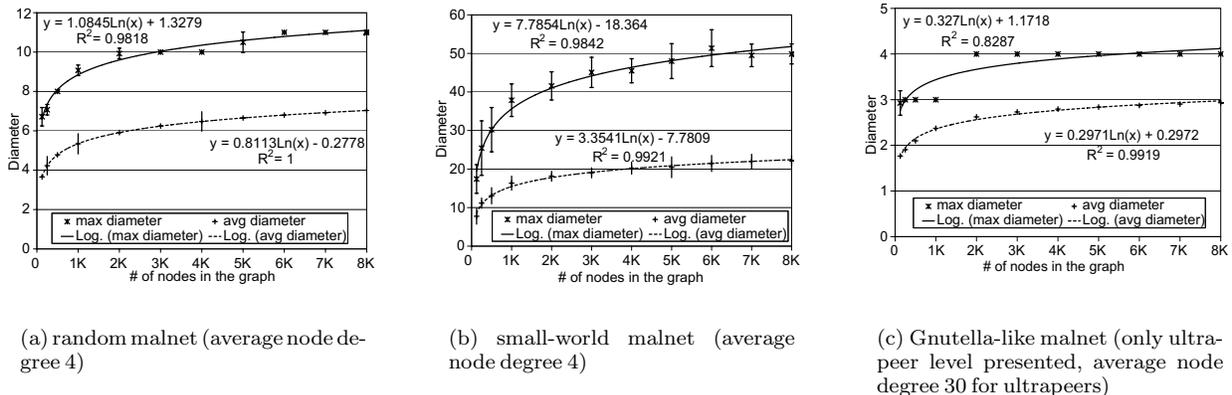
Figure 2. Malnet diameter evaluations.

ter of a malnode army could disseminate a 1-megabyte exploit to 1 million malnodes in less than six minutes [11].

3.2. Small-world malnet

Using methods and formulas from Comellas et al. [12], we have calculated the diameters of some *deterministic* small-world-based malnets. These deterministic small-world graphs are created by starting with a "circulant" graph where nodes are arranged in a ring topology with edges to the closest r neighbors, half in the clockwise direction and half in the counter-clockwise direction. For a network of 8,192 nodes arranged in a circulant graph where every node has degree 6, the diameter is 1,366. A hub graph can then be overlaid on top of the circulant graph to create shortcuts between distant clusters of nodes. If we overlay a complete hub graph with 265 nodes (3.2% of the total) on top of the above circulant graph, the diameter is reduced to 11. Since it is not reasonable to have each hub node connect to every other hub node, we recalculated the diameter when a double loop graph is used as the hub graph. This results in a larger diameter of 21 when applied to the above circulant graph, but only increases the degree of hub nodes by 4 as opposed to 264 $(265 - 1)$ when using a complete graph.

We also introduced random factors in our simulation to study the diameters of small-world malnets. We followed the methods described by Watts and Strogatz [5] in forming small-world graphs in the simulation. First, a circulant graph is created with each node connected to its r closest neighbors, the same as above. Then a node is chosen, and the edge that connects to

its nearest clockwise neighbor is considered. This edge has probability p of being reconnected to any random node in the ring, as long as the new connection would not be a duplicate. This process continues on for the rest of the nodes in a clockwise manner, then repeats for the second closest neighbor, and so forth. This process is done until each edge has been considered once. For our small-world simulations we set $p = 0.03$.

Figure 2(b) shows the average and maximum diameter of small-world malnets from the simulation. Here, if the trend reported in the figure continues, when there are 1 million malnodes and every node has four neighbors on average, the maximal distance between any two nodes would be 90, while the average maximal distance from one node to another would be 39.

3.3. Gnutella-like malnet

A malnet could be built using Gnutella-like peer-to-peer techniques. To analyze the diameter of such a malnet, we studied the measurement results of the real Gnutella. As reported in [13], which contains the most recent Gnutella measurements, Gnutella has grown to have around 800,000 peers with a network diameter of 11. However, the vast majority of peers are reachable within 6 hops or less. Such short paths stem from the use of the two-level hierarchy. In current implementations of Gnutella, every ultrapeer tends to have around 30 ultrapeer neighbors, and every leaf node connects to a small number of ultrapeers (around 3).

Figure 2(c) shows the average and maximum diameters of the ultrapeer level of Gnutella-like malnets, where we evaluated random graphs as the ultrapeer-level connection graphs and each ultrapeer had an average of 30 neighbors. Note that the distance between

two leaf nodes is the distance between their ultrapeer nodes, plus two. If the trend is taken out to 1 million nodes with $\frac{1}{14}$ being ultrapeers, and if every leaf has one ultrapeer, the total number of links will be approximately the same as that in a 1-million-node random malnet where the average node degree is 4. The maximum diameter will be 4.8 between ultrapeers or 6.8 between leaf nodes. This is consistent with the results reported by previous researchers: in [13], 99.5% of the ultrapeer nodes were within 5 hops of each other.

4. Resiliency Analysis of Malnets

Do we really have to monitor malnets closely to learn their structures and properties in order to effectively disrupt them? While it is clear that any achievable degree of disinfection should reduce the damage a malnet can do, can we just *randomly* disinfect malnodes or convert some of them into firewalls or filters to effectively disrupt the malnet, at least with a high probability? For example, the trinoo network, often used for DDoS attacks, does not offer much resiliency. With two masters and five daemons, for instance, randomly dropping two nodes has a 52.4% probability of dropping a master node.

We have designed a simulation to study the resiliency of random malnets, small-world malnets and Gnutella-like malnets, aiming to answer two questions: (1) For a given malnet, if x nodes are randomly disinfected, what percentage of remaining nodes will remain connected? (2) What is the impact of a malnet's size on its resiliency?

Here, we assume that a malnet does not attempt to reconnect any nodes after the disinfection. Also, since disinfection is often about cleaning up malnodes, not links between malnodes, our resiliency study will focus on dropping nodes, not links, from a malnet.

We first define a resiliency metric, the *maximal reachability* of a graph, which is the percentage of nodes that belong to the largest partition of the graph. It indicates the percentage of malnodes that are reachable by the attacker if the exploit injection point is from the largest partition.

4.1. Random malnet

Despite the lack of directed organization, a random malnet can have good connectivity. For different values of node degree r, we tested twenty 10,000-node malnets; for each network, we tested different values of x; and ten different random cuts were measured for each value of x. Figure 3(a) shows the maximum reachabilities for those 10,000-node graphs with x nodes cut.

For example, when 1,024 nodes are randomly cut and $r = 2$, the maximum reachability will decrease from 100% to roughly 43.30%; but if $r = 4$, the maximum reachability will only drop from 100% to 99.80%. Furthermore, while a higher value of r clearly leads to a higher maximum reachability, Figure 3 shows that a small value of r can already lead to high reachability, and thus high resiliency. With $r = 10$, even if 80% of the malnodes were disinfected, 70% of the remainder would remain connected and ready to receive new exploits.

4.2. Small-world malnet

As in Section 3.2, we first conduct a deterministic analysis of small-world malnets to calculate the number of nodes that must be removed to disconnect small-world graphs. Recall that in small-world graphs as considered by Comellas et. al [14], when a node has r neighbors there will be $\frac{r}{2}$ in either direction around the circulant graph, and a hub graph is overlaid on top of the circulant graph. Assuming that nodes have degree r, it is easy to see that if $\frac{r}{2}$ nodes on both sides of a given node v are removed then v will become disconnected. If v is part of the hub graph, another r_h nodes must be removed (r_h is the degree of nodes in the hub graph). So to disconnect a single node, either r or $r+r_h$ nodes must be removed. As an upper bound, suppose we want to ensure there are no connected components larger than the distance between nodes which make up the hub graph. We can simply take out blocks of $\frac{r}{2}$ nodes, with each block centered around a hub. Taking our example from Section 3.2 with 8,192 nodes of degree 6 and a hub graph consisting of 265 nodes, we should remove $3 \times 265 = 795$ nodes to create disconnected networks of size $\lceil \frac{8192-795}{265} \rceil = 28$. It is important to remember that we are assuming the removal of nodes is focused, rather than being random.

To compare this analysis with our statistics on random malnets, we also ran simulations. We again adopted the Watts and Strogatz model for the simulation, as in Section 3.2. Figure 3(b) shows the small-world malnet reachability results, where each malnet has 10k nodes but r varies. Note that for the small-world malnet we cannot consider graphs with $r = 2$, since otherwise there could be no clustering while maintaining connectivity. A graph must have a high level of clustering to be considered small-world, so our simulations start with $r = 4$ as a minimum. The figure shows that in general, the maximal reachability begins to decrease when 1,000 nodes, or 10% of the total nodes in a malnet, are dropped. Dropping around 7,000-8,000 nodes will almost completely partition the malnet, in

(a) Reachability after disinfection of a random malnet (10,000-node graphs with varying node degree)

(b) Reachability after disinfection of a small-world malnet (10,000-node graphs with varying node degree)

(c) Average reachability for random malnets, small-world malnets, and Gnutella-like malnets, with varying N
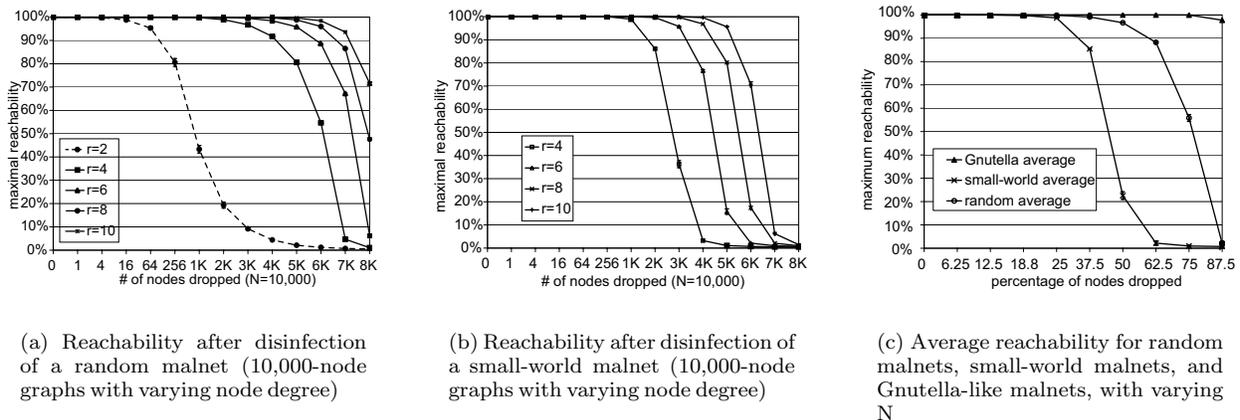
Figure 3. Malnet resiliency evaluations.

contrast to only needing to drop 795 carefully selected nodes from the 8,192-node deterministic small-world malnet.

4.3. Gnutella-like malnet

A previous study [15] found the Gnutella overlay to be very robust, requiring an estimated 60% of the overlay's nodes to be randomly dropped before the network fragments. More interestingly, the study also found that the Gnutella network fragments rapidly when the highest-degree nodes were removed (as opposed to random node failures). This can be attributed to earlier observations that the degree of Gnutella nodes exhibits a power-law distribution. However, more recent studies of modern Gnutella do not show a power-law distribution [13], suggesting that the current overlay has a higher resiliency.

Our simulation also shows that Gnutella-like malnets stay very well connected even after a high percentage of nodes is randomly dropped. For example, Figure 3(c) shows that even dropping 75% of the nodes will still leave the remainder unpartitioned, and dropping 87.5% will leave about 97% still connected.

4.4. Malnet size impact on resiliency

We also studied the impact of the size of a malnet on its resiliency and found that it is minimal. Figure 3(c) shows the results for malnets with a constant value of r ($r = 6$ for small-world and random networks and $r = 30$ for Gnutella-like malnets) but different sizes of N, where $N = 1,024$ to $16,384$ in powers of two. Clearly, with the same percentage of nodes cut, the maximal reachability will be approximately the same

for malnets of different sizes. This implies that even malnets with a relatively modest number of nodes, say 1,000, can still be very well connected.

5. Discussion

We have presented results for three different types of malnets. Section 3 shows that for the same number of malnodes and the same average node degree, small-world malnets have larger diameters than those of random malnets, whereas small-world malnets tend to be easier to form. This is largely due to the clustering feature of small-world graphs. The diameter and amount of clustering present in a small-world graph varies with the probability p that an edge will be reconnected to a random node. When $p = 0$ there is no reconnection and the graph remains a regular circulant graph with a high amount of clustering and large diameter. As p increases, the clustering and diameter decrease; when $p = 1$, the graph becomes a random graph.

As N increases, Gnutella-like malnets achieve even smaller diameters than random malnets by exploring more sophisticated formation procedures, even when the same number of links is used. By having a more densely connected core, malnodes can in general get even closer to one another.

The resiliency evaluation in Section 4 further shows that randomly dropping nodes will typically not be able to effectively partition a malnet. Selective dropping, on the other hand, can be much more effective, as shown in Sections 4.2 and 4.3.

Moreover, even though small-world malnets are less resilient and efficient compared to the other two types of malnets, note that they are already fast in that the

master of a malnode army could still disseminate a 1-megabyte exploit or upgrade to a million malnodes, starting from any member node, in less than 30 minutes. They are also resilient in that randomly disinfecting malnodes cannot effectively disconnect the remaining nodes in a small-world malnet.

Although it is true that Gnutella-like malnets may be hard for attackers to code, the chance of such malnets happening is non-trivial. In addition, malnodes could use the existing Gnutella network for covert communications, rather than creating their own Gnutella-like overlay. In the latter case the malnet overlay would be composed of mostly neutral nodes, which means that instead of "dropping" malicious nodes, we would have to filter or block them at the neutral points, which must be able to handle both the second-wave attack traffic and the legitimate Gnutella traffic.

Superficially, some of the results presented so far would seem to suggest that disconnecting a malnet is a hopeless task. However, we must emphasize that these results address the question of *random* malnode disinfections. Rather than proving the task impossible, our results indicate that disconnection may be possible, but that extremely naive approaches will not suffice. Meanwhile, as discussed in Section 3, there is a very real and serious threat from malnets because of the speed with which new attacks can be disseminated. Thus, it is critical that we find ways to combat these networks.

6. Related Work

Studies focusing on malicious networks have been rare. In terms of distributing information to a large number of recipients, malnets share some features with legitimate distribution mechanisms, such as broadcasting, multicasting, content-delivery networks, etc. Malnets are especially similar to legitimate delivery services based on overlay networks. There are tree-structured overlays for delivery service that are often regarded as application-level multicast [16, 17, 18, 19, 20, 21]. There are also non-tree-structured overlays; for example, Bullet [22] provides high-bandwidth data dissemination through an overlay mesh, and Revere [23] supports large-scale security update delivery through resilient self-organized overlay networks. A malnet can in principle have the same form as any of above. The essential difference is that a malnet program often tries to infect unprotected and uninfected machines to form or expand its network, without any prior agreement. This difference makes it possible to design solutions such as honeypots that can lure malnet programs and even join malnets to disrupt their operations, or at least gather information on them [7].

There has also been significantly related research on graph-theoretical models. One is the rich area of the structure of interconnection networks, including those evolving in a fairly random manner [24, 25, 26, 5], and those exhibiting controlled growth patterns [12]. Some researchers have also conducted studies on the resiliency of small world networks, but these are not directly applicable to our research. For example, research in [27] created networks of degree 20 and compared the resiliency of small-world networks to that of random networks; but in investigating disconnection, they considered the removal of graph edges, whereas we remove nodes.

7. Conclusions

Our paper addresses a major problem in the modern Internet: the threat of malicious "armies" of co-opted computers being used for illegal or even warlike activities. Many different types of malnets have appeared, and more sophisticated ones can be expected in the future. In fact, some highly sophisticated malnets are already feasible today and could be designed by an attacker of no more than moderate sophistication. Most component parts of such malnets have already been observed in released worms or other malicious code. The remaining parts resemble other popular and well-known programs (such as peer-to-peer file-sharing networks) and are not tremendously challenging to implement.

Prior work in the area has not developed effective methods for disabling these armies, partly due to the lack of understanding of the resiliency and efficiency features a malnet can have, and partly due to the lack of proper countermeasures to deal with these features. As an important step in addressing this deficiency, we have combined theoretical analysis and simulation to characterize the resiliency and speed of three representative types of malnets, corresponding to different paradigms of forming malnets. We have shown that it is not feasible to counteract such malnets by simply randomly dropping nodes, even if a large percentage of malnodes is dropped from the network, *i.e.*, disinfected or even converted to a firewall or filter node. Instead, the malnodes to be dropped need to be selected in a sophisticated way. Moreover, since second-wave attacks of malnets can be launched quickly, an effective defense must either happen very quickly during a second-wave attack, or take effect prior to the attack.

Future work on malnets must combine both theory and practice to address the problem at the source. A radically new approach must be designed to defend

against malicious network intruders so that they can be shut down in an effective and efficient manner. To use a military analogy, rather than building tall walls around our city and trying to withstand a long siege, we need to locate the enemy commander and cut the lines of communication between him and his troops.

8. Acknowledgments

References

[1] Computer Emergency Response Team, "Distributed denial of service tools," http://www.cert.org/incident_notes/IN-99-07.html, 1999, cERT IN-1999-07.

[2] D. Geer, "Malicious bots threaten network security," *IEEE Computer*, vol. 38, no. 1, pp. 18–20, January 2005.

[3] C. Kalt, "RFC 2810: Internet Relay Chat: Architecture," RFC 2810, IETF, April 2000.

[4] J. Jones, "BotNets: Detection and mitigation," Federal Computer Incident Response Center, Tech. Rep. FCIRC, February 2003.

[5] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, pp. 440–442, 1998.

[6] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the Internet in your spare time," in *Proceedings of the 11th USENIX Security Symposium (Security '02)*, 2002. [Online]. Available: citeseer.nj.nec.com/staniford02how.html

[7] Honeynet Project, "Know your enemy: Tracking botnets," http://www.honeynet.org/papers/bots, March 2005.

[8] WASTE Development Team, "WASTE: Anonymous, secure, encrypted sharing," http://waste.sourceforge.net.

[9] B. N. Chun, J. Lee, and H. Weatherspoon, "Netbait: A distributed worm detection service," Intel Research Berkeley, Tech. Rep. IRB-TR-03-033, September 2003, http://netbait.planet-lab.org/.

[10] S. Costello, "Nimda, Code Red still crawling, threatening the Net," http://www.nwfusion.com/news/2002/0506code.html, May 2002.

[11] P. Reiher, J. Li, and G. Kuenning, "Midgard worms: Sudden nasty surprises from a large resilient zombie army," UCLA Computer Science Department, Tech. Rep. UCLA-CSD-040019, April 2004.

[12] F. Comellas, J. Ozón, and J. G. Peters, "Deterministic small-world communication networks," *Information Processing Letters*, vol. 76, no. 1-2, pp. 83–90, 2000.

[13] D. Stutzbach and R. Rejaie, "Characterizing today's Gnutella topology," http://www.cs.uoregon.edu/~reza/PUB/tr04-02.pdf, Department of Computer Science, University of Oregon, Tech. Rep. CIS-TR-04-02, December 2004.

[14] F. Comellas, M. Mitjana, and J. G. Peters, "Epidemics in small-world communication networks," http://citeseer.ist.psu.edu/623969.html, School of Computing Science, Simon Fraser University, Tech. Rep. SFU-CMPT-TR 2002-09, October 2002.

[15] S. Saroiu, K. P. Gummadi, and S. D. Gribble, "Measuring and analyzing the characteristics of Napster and Gnutella hosts," *Multimedia Syst.*, vol. 9, no. 2, pp. 170–184, 2003.

[16] Y. D. Chawathe, "Scattercast: An architecture for Internet broadcast distribution as an infrastructure service," Ph.D. dissertation, University of California, Berkeley, 2000, chair: Eric A. Brewer.

[17] Y. H. Chu, S. G. Rao, and H. Zhang, "A case for end system multicast," in *Measurement and Modeling of Computer Systems*, 2000, pp. 1–12.

[18] P. Francis, "YOID: Your own Internet distribution," http://www.icir.org/yoid/, April 2000.

[19] J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr., "Overcast: Reliable multicasting with an overlay network," in *Proceedings of the Fourth Symposium on Operating System Design and Implementation (OSDI)*, October 2000, pp. 197–212.

[20] D. Pendarakis, S. Shi, D. Verma, and M. Waldvogel, "ALMI: An application level multicast infrastructure," in *Proceedings of the 3rd USENIX Symposium on Internet Technologies and Systems (USITS '01)*, San Francisco, CA, USA, March 2001, pp. 49–60.

[21] S. Q. Zhuang, B. Y. Zhao, A. D. Joseph, R. H. Katz, and J. D. Kubiatowicz, "Bayeux: An architecture for scalable and fault-tolerant wide-area data dissemination," in *Proceedings of NOSSDAV*, June 2001.

[22] D. Kostić, A. Rodriguez, J. Albrecht, and A. Vahdat, "Bullet: High bandwidth data dissemination using an overlay mesh," in *SOSP '03: Proceedings of the nineteenth ACM symposium on Operating systems principles.* ACM Press, 2003, pp. 282–297.

[23] J. Li, P. Reiher, and G. Popek, "Resilient self-organizing overlay networks for security update delivery," *IEEE Journal on Selected Areas in Communications, special issue on Service Overlay Networks*, 2003.

[24] P. Erdős and A. Rényi, "On random graphs. I," *Publicationes Mathematicae Debrecen*, vol. 6, pp. 290–297, 1959.

[25] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[26] B. M. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas in Communication*, vol. 6, no. 9, pp. 1617–1622, December 1988.

[27] K. Sun and Q. Ouyang, "Distance distribution and reliability of small-world networks," *Chinese Physics Letter*, vol. 18, pp. 452–454, Mar. 2001.