# A Taxonomy of Botnet Structures

David Dagon, Guofei Gu, Christopher P. Lee, Wenke Lee
Georgia Institute of Technology
{dagon@cc., guofei@cc., chrislee@, wenke@cc.}gatech.edu

## Abstract

*We propose a taxonomy of botnet structures, based on their utility to the botmaster. We propose key metrics to measure their utility for various activities (e.g., spam, ddos). Using these performance metrics, we consider the ability of different response techniques to degrade or disrupt botnets.*

*In particular, our models show that targeted responses are particularly effective against scale free botnets and efforts to increase the robustness of scale free networks comes at a cost of diminished transitivity. Botmasters do not appear to have any structural solutions to this problem in scale free networks. We also show that random graph botnets (e.g., those using P2P formations) are highly resistant to both random and targeted responses.*

*We evaluate the impact of responses on different topologies using simulation and demonstrate the utility of our proposed metrics by performing novel measurements of a P2P network. Our analysis shows how botnets may be classified according to structure and given rank or priority using our proposed metrics. This may help direct responses and suggests which general remediation strategies are more likely to succeed.*

## 1 Introduction

Malware authors routinely harness the resources of their victims, creating networks of compromised machines called botnets. The attackers' ability to coordinate the victim computers presents novel challenges for researchers. To fully understand the threat posed by such networks, we must identify classes of botnet topologies, their potential uses, and the challenges each class presents for detection and remediation.

We believe that it is inadequate to simply enumerate the botnets we have seen to date in the wild. Botnets have proven to be very dynamic. For example, researchers have observed changes in botnet sizes, which have trended from large networks (100K+ victims) to numerous smaller botnets (1-5K+ victims) [53]. Likewise, we have seen a rapid transition from centralized botnets (e.g., IRC) to distributed organizational structures (e.g., P2P) [60]. We expect that botnets will continue to be a dynamic, evolving threat.

We must therefore consider the structural and organizational *potential* of botnets. Similar to how previous work detailed key aspects of individual classes of worms [57], this paper provides a taxonomy of botnet organization, and their utility for various malicious activity. We believe that future botnet research will share a common goal of reducing the utility of botnets for botmasters. This raises important questions: How are botnets utilized? What metrics should be used to measure the effectiveness of remediation on such networks?

Recent work by Rajab, et al. [47] noted the need for the botnet research community to better define metrics. Their study examined problems in estimating botnet populations. This paper argues that other metrics (bandwidth, communications efficiency, robustness) require a similar thoughtful examination.

This paper therefore proposes a taxonomy of botnet topologies, based on the utility of the communication structure and their corresponding metrics. Section 2 details metrics for measuring botnet uses, and describes the structural organization of botnets. In Section 3, we demonstrate how to perform measurement of selected metrics, and analyze experimental response techniques designed to address particular classes of botnets. We note how our work relates to other areas of inquiry in Section 4. Since this area of research is new and rapidly changing, we conclude with suggestions for future work in Section 5.

Our contribution is the following: we identify a small number of likely structural forms for botnets, based on a utilitarian analysis. We propose metrics for measuring a botnet's effectiveness, efficiency, and robustness. Our analysis of models and real world observations suggests that some botnet structures are more resilient than others to different types of remediation efforts. This analysis can guide future inquiry into how to best address the botnet problem.

1063-9527/07 $25.00 © 2007 IEEE
DOI 10.1109/ACSAC.2007.44

## 2 Botnet Taxonomy

The evolving and evasive nature of botnets requires researchers to anticipate possible topologies. An interesting early contribution in this area is [13], which listed three topologies (centralized, peer-to-peer, and random) for botnets, and roughly evaluated performance metrics in terms of high, medium and low performance.

To more fully understand the threat, we expand on [13] and propose a taxonomy of possible botnet topologies and how to measure their utilization in various malicious activities.

### 2.1 Purpose and Goals

Taxonomies are most useful when they classify threats in dimensions that correspond to potential defenses [30, 31]. As [29] noted: "[a]n important and sensible goal for an attack taxonomy ... should be to help the defender."

Our botnet taxonomy will help researchers identify what types of responses are most effective against botnets. Our design goals are similar to [57]: (a) assist the defender in identifying possible types of botnets, (b) describe key properties of botnet classes, so researchers may focus their efforts on beneficial response technologies.

Our taxonomy is driven by possible responses, and not detection. There is some initial work in botnet detection [13–15, 17, 18, 20]. Further, the considerable body of literature on worm detection has identified detection techniques that can be adapted to botnet detection [9, 21, 26, 44, 58, 61–63]. We therefore leave for future work a classification of botnet detection techniques.

### 2.2 Key Metrics for Botnet Structures

Naively, one could suppose that bots will organize according to various regular network topologies such as star, mesh, or bus networks. These topologies are useful for formal analysis of discrete network properties, but do not let us describe the utility of large complex botnets.

Instead, we need to pay attention to key *discriminators* that let one compare important attributes of botnets. We identify three important measures of botnets: effectiveness, efficiency, and robustness. We acknowledge there are other characteristics the botmaster may desire, but these are not easily designed into the topology of a victim network. For example, botmasters may desire anonymity from their botnet (e.g., to carry out anonymous attacks); however, this property is not inherently obtained from any single topology, and depends more on the application-layer design of a botnet's messaging system.

Table 1 lists a few botnet uses, and key relevant metrics. More than one metric can be relevant to a botnet use, and botnets certainly have multiple uses. However, the table lists key metrics critical to the botnet's specified function.

### 2.3 Measuring Botnet Effectiveness

The *effectiveness* of a botnet is an estimate of overall utility, to accomplish a given purpose. While botmasters may innovate new uses of botnets, the ability of a botnet to meet existing uses such as spam, ddos, warez distribution and phishing is roughly approximated by size and bandwidth. Both of these terms require elaboration.

We agree with [47], that "botnet size" must be a qualified term. Here, we do not use size to mean the total population count, such as that usually used in worm epidemiology studies [37–39, 50]. Instead, we mean the "giant" component of the botnet, or largest connected (or online) portion of the graph [10, 42]. Botnets are of course more powerful if they have large infected population, but the giant component lets us directly measure the damage potentially caused by certain botnet functions.

In the case of DDoS, the giant component, $S$, lets us measure the largest number of bots that can receive instructions and participate in an attack. This contrasts with the total population of all infected victims, which may not always be reachable by the botmaster, e.g., because of diurnal variations [16].

A related measure is the average amount of bandwidth that a bot can contribute, denoted as $B$. Estimating bandwidth along a single link is a complex problem, and the subject of numerous investigations in the networking community [6, 25]. To estimate the cumulative bandwidth of an entire botnet presents an even more challenging task. For example, one could measure the bandwidth between bots, between a bot and the botmaster, or between any bot and a third party (e.g., a DDoS victim). By average bandwidth, $B$, we mean the cumulative available bandwidth in a bot that a botmaster could generate from the various bots (e.g., for DDoS) under ideal circumstances. Such a measurement of course varies with the distribution of bandwidth available to each member of the botnet, the probability that any victim is "on-line" at any given time, and the amount of bandwidth already being consumed by the victims themselves (e.g., for normal use).

We roughly classify three types of bots according to their transit categories: those using modems (type 1), those using DSL/cable (type 2), and those using 'high-speed' networks (type 3). While bandwidth within each class is highly variable in itself, we believe this grouping is a reasonable first approximation because they are standard in industry–e.g., many commodity databases already map connection classes according to these categories [34]. The probability of a bot belonging to type $i$ is denoted as $P_i$. According to [24], a reasonable distribution for US-based bots could be esti-

| Major Botnet Utilities | Key Metrics | Suggested Variables | Comment |
|---|---|---|---|
| Effectiveness | Giant portion | $S$ | Large numbers of victims increases the likelihood of high-bandwidth bots. Diurnal behavior favors $S$ over total population. |
| | Ave. Avail. Bandwidth | $B$ | Average bandwidth available at any time, because of variations in total victim bandwidth, use by victims, and diurnal changes. |
| Efficiency | Diameter | $l^{-1}$ | Bots sending messages to each other and coordinating activities require efficient communications. |
| Robustness | Local transitivity | $\gamma$ | Bots maintaining state (e.g., keycracking or mirroring files) require redundancy to guard against random loss. Highly transitive networks are more robust. |

**Table 1. Botnet Uses and Relevant Metrics**

mated as $P_1 = 0.3, P_2 = 0.6, P_3 = 0.1$. Similar distributions could be inferred for a global population.

Let us denote the average maximum network bandwidth within each type as $M_i$, the average normal usage of bandwidth within each type is $A_i$. Thus, the average available bandwidth could be used by a botmaster on a bot is $M_i - A_i$. We simplify our measurement by assuming a botmaster would not use even more bandwidth, since this would interfere with the victims existing use, and the disruption might alert them to the infection.

We also need to consider the diurnal sensitivity of these networks. More complete diurnal models of bot behavior were presented in [16]. However, to avoid modeling diurnal changes in numerous time zones, we can use a simplified metric based on the estimated number of hours a victim is online per day (and therefore capable of participating in the botnet). We assign different weights (denoting the distribution of time hosts are online each day) to each class of bots. For example, if we assume average online hours per day for a bot using modem is 2, for a bot with DSL/cable is 6, and for a bot with high-speed is 24, then we have the probability vector $\vec{W} = [2/32, 6/32, 24/32] = [0.0625, 0.1875, 0.75]$. We selected these numbers based on [43]; however, our analysis considers other ranges of values.

Using the simplified bandwidth estimation for each bot, and a simplified diurnal model, we can express the average available bandwidth of a bot as:

$$B = \sum_{i=1}^{3} (M_i - A_i) P_i W_i \quad (1)$$

In Section 3, we suggest the utility of this metric by comparing different botnets. The weights and distribution of hosts in each class are of course variable. To understand their sensitivity, we evaluated the weighted bandwidth for different ranges of estimates.

Figure 1 shows the weighted bandwidth, with different variations in diurnal sensitivity. We can see in Figure 1(a), that the final average weighted bandwidth is around 20Kbps for a single bot, for the values fixed in that plot. With approximately 50,000 such bots in a botnet, the botmaster can utilize about 1Gbps bandwidth on average at any time.[1] The parameters for the plots in Figure 1 are drawn from data measurements described in Section 3.

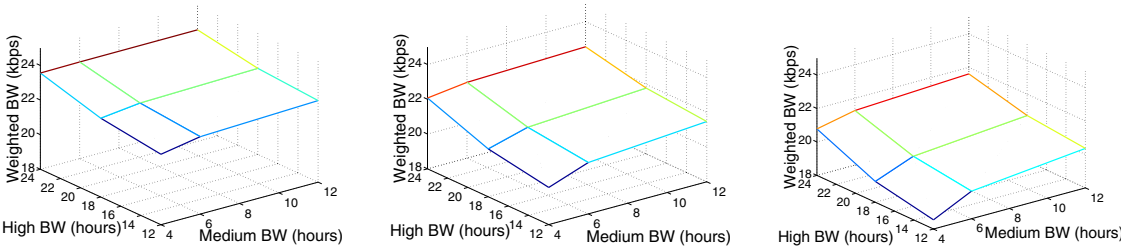
The plots reveal the sensitivity of this metric to the diurnal variation in users. Compare for example Figure 1(a), where low bandwidth users are presumed online for only two hours, to Figure 1(c), where six hours is fixed instead. For diurnal weighing above 6 hours/day, variation in the online hours for the medium and high-bandwidth users does not result in much variation in the overall bandwidth, as shown in Figure 1(a). However, in Figure 1(c), the online variation of the other classes has a significant impact on bandwidth particularly when higher-speed users are "always on" and have a diurnal weight of 1. This suggests that botnets with many low-speed connections experience less variation when the lower-speed connections minimize their time online. In Section 3, we further compare estimated bandwidth of two botnets.

## 2.4 Measuring Botnet Efficiency

Botmasters and security researchers may also be concerned about the *efficiency* of a botnet. Whether used to forward command-and-control messages, update bot executable code, or gather host-based information (e.g., keylogging and data exfiltration), a botnet may be evaluated by its communication *efficiency*.

We propose *network diameter* as one means of expressing this efficiency. By network diameter, we mean the average geodesic length of a network, $l$. This measures the average length of the shortest edge connecting any two nodes in the network. If $l$ is large, the dynamics of the network (communications, information, epidemics) is slow. The reader

[1] We repeat again the caution noted above: our *available bandwidth* metric does not measure the bandwidth between any two points. Rather it measures the amount of traffic the botmaster may reasonably generate using his network.




(a) Fixed online hours for Type1 at 2, varying other two types.

(b) Fixed online hours for Type1 at 4, varying other two types.

(c) Fixed online hours for Type1 at 6, varying other two types.

**Figure 1. Weighted bandwidth and diurnal sensitivity. Low-bandwidth bots have a significant effect on average bandwidth when they are online for more than $\approx$ 4 hours. Figures (a) through (c) fix the diurnal weight of low-bandwidth bots at 2, 4 and 6 hours. Only at the extreme, plot (c), does average bandwidth change significantly. This impact is seen when high- and medium bandwidths bots have less than 24-hour/day connectivity.**

may recall that in Milgram's famous paper, social networks were shown to have short average geodesic lengths, approximately $\log N$, or $l \approx 6$ ("six degrees of separation") for general society [36], while the web has a larger estimated length, $l \approx 17$ [3].

As in [23], we use the inverse geodesic length, $l^{-1}$, instead of $l$, defined as:

$$l^{-1} = \left\langle \frac{1}{d(v,w)} \right\rangle = \frac{1}{N(N-1)} \sum_{v \in \mathcal{V}} \sum_{w \neq v \in \mathcal{W}} \frac{1}{d(v,w)'} \tag{2}$$

This way, if bots $v$ and $w$ are disconnected, the distance $d$ is zero. Further, the inverse length is normalized, ranging from 0 (no edges) to 1 (fully connected). In the context of botnets, $l^{-1}$ refers to the overlay network of bot-to-bot connections created by the malware, instead of the physical topology of the Internet. Thus, bot victims on the same local network (one hop away) may be several edges apart or even unconnected in the overlay bot network created by the malware.

This metric is also relevant to robustness because with each message passed through a botnet, there is a probability of detection or failure. Some researchers have already investigated zombie detection via stepping stone analysis, or the detection of messages being relayed through victim proxies [59]. It is difficult to express this chance of detection precisely, since botnet identification is a new, developing field. But at a high level, botnet detection techniques will generally rely on the chance of intercepting (i.e., detecting and corrupting or halting) a message between two bots in a network. Assume that bots $u$ and $v$ are connected through $n$ possible paths, $P_1, \ldots P_n$, and that each node in the path can be recovered (cleaned) with probability $\alpha$. If $\epsilon_i$ is the chance that path $P_i$ is corrupted, quarantined or blocked, then all paths between $u$ and $v$ are blocked with probability:

$$\prod_{i=1}^{n} \epsilon_i \leq (1-\alpha)^n \tag{3}$$

While bots $u$ and $v$ are connected through *some* path with probability $1-(1-\alpha)^n$, the chance of failure increases with $\alpha$ (i.e., as detection technologies improve). Section 3 characterizes the performance of $l^{-1}$ under increasing link decay.

We expect that in the future, botnet researchers will propose many techniques to detect, disrupt, or interfere with botnet messaging. Network diameter, $l^{-1}$ is therefore a basic, relevant metric to determine how many opportunities network administrators have to observe, disrupt or measure messaging.

The incentive of the botmaster is to increase $l^{-1}$, which yields a more efficient/robust botnet, at least for selected uses noted in Table 1. Under an ideal $l^{-1} = 1$, every bot can talk directly to every other bot. Since a botnet with more interconnections has more short paths, it passes messages quickly, and provides fewer detection opportunities.

### 2.5 Measuring Botnet Robustness

A final category of botnet use can be expressed in the *robustness* of such networks. Bots routinely lose and gain new members over time. If victim machines are performing state-sensitive tasks (e.g., storing files for download, or sending spam messages from a queue), a higher-degree of connection between bots provides fault tolerance and recovery.

To some degree this metric correlates with an improved redundancy. $l^{-1}$ already indicates robustness in some sense.

But we more precisely capture the robustness of networks using local transitivity to measure *redundancy*. Local transitivity measures the likelihood that nodes appear in "triad" groups. That is, given two node pairs, $\{u, v\}$ and $\{u, w\}$, that share a common node, $u$, local transitivity measures the chance that the other two, $v$ and $w$, also share an edge. A clustering coefficient $\gamma$, measures the average degree of local transitivity [56], in a neighborhood of vertices around node $v$, $\Gamma_v$. If $E_v$ represents the number of edges in $\Gamma_v$, then $\gamma_v$ is the clustering coefficient of node $v$. Where $k_v$ represents the number of vertices in $\Gamma_v$, then we have:

$$\gamma_v = \frac{E_v}{\binom{k_v}{2}}, \gamma = \langle\gamma\rangle = \frac{1}{N}\sum_{v\in\mathcal{V}}\gamma_v. \quad (4)$$

The average clustering coefficient $\langle\gamma\rangle$ measures the number of triads divided by the maximal number of possible triads. Just like $l^{-1}$, $\gamma$ ranges from $[0, 1]$, with 1 representing a complete mesh.

Local transitivity is an important measure for certain botnet uses. Warez (stolen programs) and key cracking require reliable, redundant storage, particularly since botnets exhibit strongly diurnal properties. To ensure uninterrupted key cracking, or that file resources are always available, botmasters routinely designate multiple victims to store identical files. (For examples, consult [12].) Botmasters could use quorum systems in addition to simple backups. However, the transitivity measure $\gamma$ index generally captures the robustness of a botnet.

## 2.6 Botnet Network Models

To measure the robustness of different botnet architectures, we must further specify the types of response actions available to network administrators. In a general sense, botnets can suffer random and targeted responses. Random failures correspond to patching by normal users, diurnal properties of computers being powered off at night, and other random failures in a network. Targeted responses are those that select "high value" machines to recover or patch. These response types all correspond to actions directed at botnet vertices. Edge-oriented responses (e.g., quarantine, null routing) have been considered elsewhere, e.g., [64].

Expanding on the general categories of botnets noted in [13], we consider different types of graphs studied in the extensive literature on complex networks. Our taxonomy uses the major models from that field. For a comprehensive overview of complex network mechanics, see [4].

### 2.6.1 Erdös-Rényi Random Graph Models

To avoid creating predictable flows, botnets can be structured as random graphs. In a random graph, each node is connected with equal probability to the other $N - 1$ nodes. Such networks have a logarithmically increasing $l^{-1}$. The chance a bot has a degree of $k$ is the binomial distribution:

$$Pr(k) = \binom{N-1}{k} p^k (1-p)^{N-1-k} \quad (5)$$

Particularly for large networks like botnets, it makes sense to limit the degree $k$ to a maximum number of edges, $L$. For our analysis below, we select an average $\langle k\rangle$ appropriate to botnets, instead of $\langle k\rangle \approx 2L/N$ used by others studying general network complexity problems [23]. Without such a limitation, a pure Erdös-Rényi random botnet would potentially create individual bots with hundreds of edges, even for small (5K victim) botnets. Large numbers of connections on a client host are highly unusual, even for P2P software [33, 49]. So, unless the victim is a rare high-capacity server, botmasters would keep $\langle k\rangle$ small, say $\langle k\rangle \approx 10$. In Section 3, we measure the degree of connection in an unstructured P2P botnet, to confirm that $\langle k\rangle$ will have fairly low values.

One difficulty in random graphs is easily overcome by certain types of botnets. Since each node has a probability $Pr(k)$ of being connected to each vertex, the creation of the graph requires some central collection (or record) of vertices. That is, each bot must either know or learn the address of all the other bots, in order to have a chance of sharing an edge. Because such a list may be discovered by honeypot operators, botmasters have an incentive to not create such a centralized master list, and some bots, e.g., those created by the Zindos worm [32], take explicit steps to limit the number of victim addresses stored in one place.

This creates a technical problem for botnets that propagate through traditional (e.g., scanning, mass-mailing) techniques. The first victims will not know the address of subsequent victims, and have a $Pr(k)$ biased towards zero. One solution is for the attacker to keep track of victims joining their botnet, generate a desired topology overlay, and transmit the edge sets to each bot.

Botmasters can easily select a desired $\langle k\rangle$ to generate such a network. For example, they may select $\langle k\rangle \leq 10$, so that bots appear to have flow behavior similar to many peer-to-peer applications [33, 49]. A botmaster could of course select a higher $\langle k\rangle$, even one close to $N$ to create a mesh, but such structures quickly exhaust bot resources, and may be easily detected by network administrators.

If existing botnets are not available to generate a random graph, one solution was proposed by [13], where bots could randomly scan the Internet to find fellow bots. Although noisy, this approach provides a last-resort technique for botnet creation. Assuming random scanning up to $L$ connections, the resulting botnet would have a poisson $k$ distribution, and both the clustering and diameter properties of a random graph.

#### 2.6.2 Watts-Strogatz Small World Models

Another topology botnets can use is a Watts-Strogatz network. In such a network, a regional network of local connections is created in a ring, within a range $r$. Each bot is further connected with probability $P$ to nodes on the opposite side of the ring through a "shortcut". Typically, $P$ is quite low, and the resulting network has a length $l \approx \log N$. See [4] for further discussion of small world networks.

Intuitively, we can imagine a botnet that spreads by passing along a list of $r$ prior victims, so that each new bot can connect to the previous $r$ victims. To create shortcuts in the small world, bots could also append their address to a growing list of victims, and with probability $P$ connect back to a prior bot. As noted in Section 3, we have witnessed only a few anecdotal botnets that create prior victim lists, e.g., Zindos [32]. To frustrate remediation and recovery, the lists are typically small $r \approx 5$. In the case of propagation-created botnets, botmasters may prudently use $P = 0$, to avoiding transmitting a lengthy list of prior victims. Otherwise, a bot would have to append its address to a growing list of IPs forwarded to each new victim. As noted above, if a botmaster desired to have shortcuts in a small world botnet, they could instead just use an existing botnet.

#### 2.6.3 Barabási-Albert Scale Free Models

The previous botnet structures are characterized by variations in clustering, and each node exhibits a similar degree, $k \approx \langle k \rangle$. In contrast, a Barabási-Albert network is distinguished by degree distribution, and the distribution of $k$ decays as a power law. Many real-world networks have an observed power-law distribution of degrees, creating a so-called scale free structure.

Scale-free networks contain a small number of central, highly connected "hubs" nodes, and many leaf nodes with fewer connections. This has a significant impact on the operation of the network. As discussed in Section 3, random node failures tend to strike low-degree bots, making the network resistant to random patching and loss. Targeted responses, however, can select the high degree nodes, leading to dramatic decay in the operation of the network. This phenomenon is explored in many articles, e.g., [5].

Researchers have noted that bots tend to organize in scale free structures, or even star topologies [11, 15, 17]. For example, botnets might use IRCd [27] for coordination, which explicitly uses a hub architecture.

#### 2.6.4 P2P Models

In a P2P model, there are structured and unstructured topologies [45, 48]. For example, a structured P2P network might use CHORD [52], or CAN [48], while an unstructured P2P might use the hub-and-spoke networks created under gnutella or kazaa [45].

The unstructured P2P networks tend to have power-law link distributions [45]. We therefore treat this type of P2P network as a Barabási-Albert (scale free) model in our analysis. Similarly, structured P2P networks are similar to random networks, in the sense that every node has almost the same degree.

In Section 3, we observe new P2P-based botnets, and perform some measurements on their structures. Since our selected metrics concern only basic botnet properties (length, giant, and local transitivity), we can treat these networks as random or scale free in our analysis. We encourage others to refine these models to identify distinct P2P botnet features that distinguish them from random and scale free networks. For the metrics proposed in this work, however, we will address P2P botnets as special cases of the previous categories.

## 3 Taxonomy-Driven Botnet Response Strategies

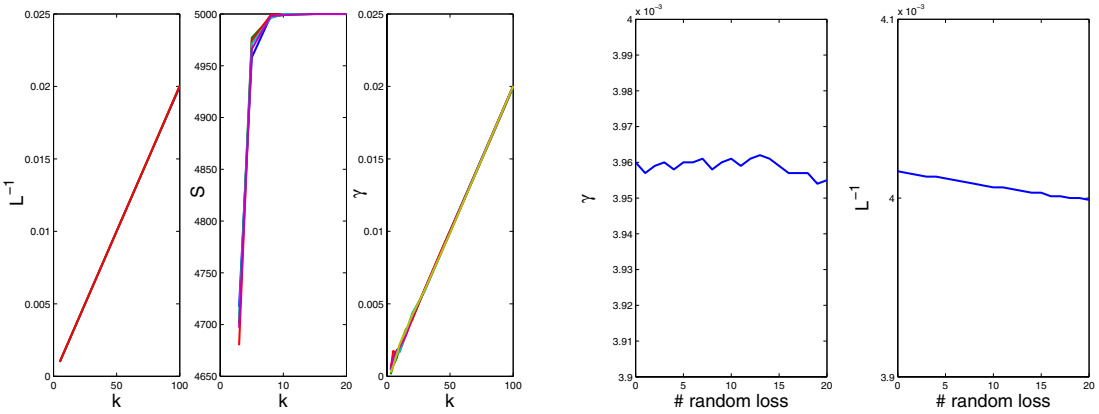
The previous discussion of botnet organization suggests the need for diverse response strategies. To guide future research in this developing area, we model different responses to each botnet category. Our analysis confirms the prevailing wisdom [13] that command-and-control is often the weak link of a botnet. We confirm our model with an empirical analysis of a real-world botnet response. Significantly, our analysis also shows that targeting the botnet C&C is not always an effective response. Some botnets will require new response strategies that research must provide.

### 3.1 Erdös-Rényi and P2P Models

For ranges appropriate to botnets, we evaluate the relationship between node degree, $k$, and the diameter of the botnet, expressed as $l^{-1}$. We assume that, to evade trivial detection, botnets will attempt to limit $\langle k \rangle$ to some value similar to P2P. Empirical studies of P2P systems reveal very low median link scores (e.g., $k \approx 5.5$) [33, 49]. Figure 2(a) plots $\langle k \rangle$ against $l^{-1}$ for realistic values, $k \leq 20$. Others have noted that for increasing average degrees, $\langle k \rangle$, random Erdos-Renyi models have logarithmically increasing diameters [23]. However, in Figure 2(a), realistic values of $k$ show a *linear* relationship to $l^{-1}$.

We also note that giant, $s$, improves significantly with increases in $k$, enabling connections with most of the botnet when $k \approx 10$ for a 5K botnet. This agrees with the general principle noted in Eqn. (3), where logarithmically connected networks enjoy nearly universal broadcasting.

Local transitivity, $\gamma$, also increases logarithmically with $k$. But for a range of small values of $k$, typical of botnets, it shows a linear increase. This means that each additional




(a) Node Degree in Random Botnets (b) Random Loss in Botnets

**Figure 2. (a) Changes in length $l^{-1}$, giant ($s$), and local transitivity ($\gamma$) in response to changes in critical values of $k$, for 5K victim botnet. (b) Effect of loss on random networks.**

value of $k$ equally improves the general robustness of the botnet. We also note a slight flare at the base of the $\gamma$ plot for Figure 2(a), for very low values of $k$. Intuitively, this means botnets with a very low average degree have difficulty forming triads, but this is quickly overcome as $k$ increases. Botmasters therefore have incentives to increase $k$.

Our current analysis, however, shows that for botnets using a random topology, random loss (e.g., infrequent user patching or anecdotal cleanup) will not diminish the number of triads in the botnet. We also omit plotting the performance of random networks under targeted responses. Targeting nodes can at best remove a few nodes with $k$ slightly higher than $\langle k \rangle$. The result is asymptotically the same as random loss.

The work in [55] is a good example of a hybrid botnet with a random graph structure formed using a technique similar to Erdös-Rényi graph, through the use of a peer list. They also confirm the robustness of such networks against targeted and random attacks. The work in [54] is also a good example of botnets created using a random graph structure.

In section 2 we noted that structured P2P networks are very similar to random networks, at least in terms of the metrics we care about: length, giant and transitivity. Structured P2P networks in fact have a constant $k$ (often set equal to the log $N$ size of the network), so they are slightly more stable than purely random networks. Thus, changes in $\gamma$ and $s$, and $l^{-1}$ are constant with the loss of random nodes.

Clearly botnets with random topologies (including structured P2P networks) are therefore extremely resilient, and deserve further study. We speculate that the most effective response strategies will include technologies to remove large numbers of nodes at once. Detecting and cleaning up large numbers of victims (perhaps at the host level) appears to be the most viable strategy. Likewise, strategies that disrupt the ability of the network to maintain indices may be fruitful, as suggested by the P2P index poisoning research in [51].

## 3.2 Watts-Strogatz Models

There are some experimental botnets [32] that use small world structures, but overall they do not appear to have a high utility value, using the metrics we have proposed. The average degree in a small world is $\langle k \rangle \approx r$, or the number of local links in a graph. Thus, random and targeted responses to a small world botnet produce the same result: the loss of $r$ links with each removed node. Thus, the key metrics for botnets, $s, \gamma, l^{-1}$ all decay at a constant rate in a small world.

We presumed that shortcut links in a small world botnet are not used ($P = 0$), but even if present, they would not affect $\gamma$ with $r \geq 4$. That is, if the number of local links is large enough to form triads, the absence of shortcuts does not significantly increase the number of triads (which are already formed by $r$ local neighbors).

There may be other benefits (e.g., propagation stealth or anonymity), for which we have not proposed a utility metric. But overall, small world botnets do not have benefits different from random networks. In other domains, researchers have noted that small world graphs are essentially random [23].

Our investigation of experimental of botnet structures only reveals one representative of the Watts-Strogatz model: the Zindos [32] worm. We speculate that the poor utility scores in the face of targeted and random loss may explain

this phenomena. An equally likely explanation is hinted at by Zou, et al., in [60], where the authors noted the desire of botmasters to avoid revealing a lists of confederate botnet members to honeypot operators.

### 3.3 Barabási-Albert and P2P Models

While random networks present a challenge, at least scale free networks provide some good news for researchers. Figure 3(a) plots the change in diameter and transitivity against changes in the "core" size of the botnet, $C$. The "core" of a scale free botnet is the number of high-degree central nodes–the routers and hubs used to coordinate the soldier bots. As more core nodes are added, the diameter of the scale free botnet stays nearly constant for small regions of $C$. Intuitively, splitting a hub into smaller hubs does not significantly increase the length of the overall network.

The local minima in Figure 3(a) has an intuitive explanation. If we have a single hub in a scale free network, $C = 1$, many of the added leaf nodes have a good chance of forming triads. The scale-free generation algorithm we chose prefers high degree nodes, and tends to form many triads when there are few hubs.

As we increase $C$, we create several high degree hubs that attract distinct groups of leaf nodes. This creates many "squares", where hubs are connected to each other, and leaves are connected to each other. But transitivity is only measured locally (in triads, and not other polygon paths). Thus, increasing $C$ diminishes $\gamma$ slightly. As we increase $C$ more, we observe a tendency for the hubs themselves to form triads, so $\gamma$ grows logarithmically.

Can botmasters avoid this drop in transitivity? We suspect not, if they wish to maintain a "normal" degree count, relative to other applications. In Figure 3(c), we compare changes in $\gamma$ against core size using different link counts for leaf nodes. If nodes have more links, $m \approx 16$, the loss in $\gamma$ shallows out. But increasing the link count of nodes can help anomaly detection algorithms that examine link degrees (e.g., flow log analysis). This reveals a curious mix of incentives. On the one hand botmasters would like to have $C >> 1$, since a single core node is too easily removed. But increasing $C$ just a little drops local transitivity. To recover the loss in transitivity, botmasters would have to increase link counts to rates far in excess of average P2P degree counts.

Responses to scale free botnets are more effective. As expected, random losses in scale free botnets are easily absorbed. Figure 3(b) shows that random patching has almost no affect on a botnet diameter or the frequency of triad clusters. Intuitively, because of the power law distribution of node degrees, random losses tend to affect low-degree nodes (e.g., the leaves), and not important nodes (e.g., hubs).

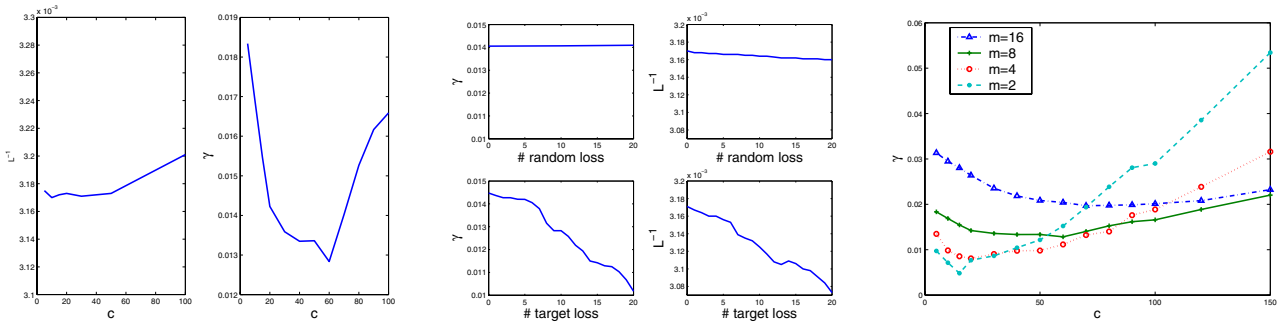
Targeted responses, however, can select key nodes for response. This results in a dramatic increase in diameter, and loss of transitivity. This suggests that researchers should focus on technologies that allow targeted responses to high-degree nodes in botnets. Figure 3(b) validates the intuitive idea that by removing a botnet C&C, the network quickly disintegrates into a collection of discrete, uncoordinated infections.

As noted in [47], measuring aspects of botnets presents a challenge to researchers. To demonstrate the practicality of our proposed metrics, we measured the average link degree in an unstructured P2P botnet. We selected the nugache worm [41], and measured the degree of connections between neighbors in the network mesh. Nugache uses a link encrypted, peer-to-peer filesharing protocol, WASTE [1], and uses several hard-coded IP addresses to request a list of peers to from [41]. After connecting to peers, the bot discovers more peers and continue to form new connections. The resulting botnet is an unstructured P2P network, which tends to create a scale-free form. Thus, although nugache spreads by P2P systems, the resulting mesh is a scale-free network.

Since we believe our data collection technique is somewhat unusual, we describe it in some detail. We note that obtaining precise measurements is, of course, nearly impossible given the distributed nature of nugache. We therefore ran multiple instances of the nugache worm in a modified version of WINE [2], which guaranteed that each copy would obtain a unique IP when a network socket is allocated under `bind()` system calls. Thus, using a single multi-homed machine, we "controlled" hundreds of nugache nodes and were able to observe their connections to the rest of the victims in the wild. (This is similar to the use of numerous heavy-weight honeypots to track botnets, noted in [13].) We ran two such "batch WINE runs" for several weeks, creating hundreds of nodes, and measured the connections degree among our subsample of the overall population.

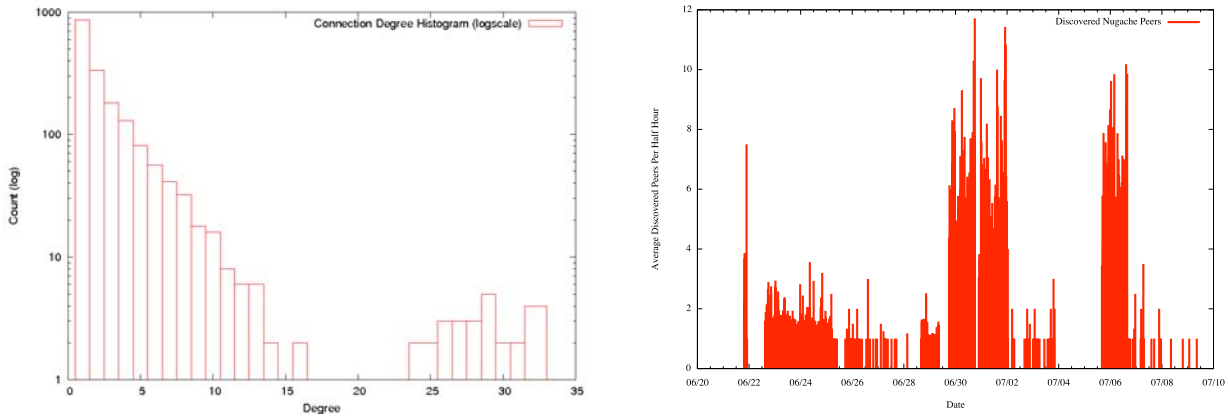
Figure 4(a) shows the distribution of link degrees found in the Nugache sample. The vast majority of victims maintained less than 6 links to other victims. There are a few nodes with a very high degree, $\approx 30$. This suggests a scale-free network typical of unstructured P2P networks. Our sampling technique unfortunately could not inject nodes into the inner ring of the nugache network (created from the hard-coded peers), where we would expect to observe a very high link degree.

If we had contacted the owners of the low-degree nugache nodes we observed, or otherwise caused their remediation and cleanup, our impact on the network's utility would have been negligible, according to our analysis. Our model above shows that random losses in scale free networks (and




(a) Scale Free Properties (b) Loss in Scale Free Networks (c) Scale Free Networks and Transitivity

**Figure 3. (a) Changes in diameter and transitivity vs. core size, for a 5K scale free botnet. (b) Loss in scale free networks. (c) Changes in link count for leaves in a scale free network.**




(a) Nugache Worm Link Distribution (b) Nugache Victim Joins over Time

**Figure 4. Measurements of (a) link degree in Nugache and (b) joins observed over time.**

unstructured P2P networks) do not significantly degrade the network. Figure 3(b) shows that random losses fail to significantly reduce either the diameter or transitivity values.

Of course, we were unable to measure the entire population, $s$, of the nugache network using our data collection technique. Figure 4(b) illustrates the problem. This figure plots the rate of new `SYN+ACK` connections observed by our batch WINE nodes. This is therefore a rough measure of the rate of new link creation, which may or may not correspond to the rate of new victims being recruited. (That is, a new `SYN+ACK` may represent an old nugache victim we've just discovered, or a new victim joining for the first time.) Since we did not catch nugache in its early formation, or successfully inject our honeypots into the inner ring of high-degree nodes, we saw only a small number of potentially new victims over the study period.

As the authors in [47] noted, measuring population values is a complex undertaking. We believe our analysis shows that our proposed metrics are both practical and useful. However, we leave for future work the design of effective data collection techniques for P2P networks (whether structured or unstructured). Given the often stealthy creation of such networks, we expect this may remain a challenging problem for researchers.

### 3.4 Empirical Analysis

Our taxonomy also suggested that available bandwidth $B$ is a useful metric for botnet utility. We again note that bandwidth estimation for end-to-end hosts is a complex task. Nonetheless, to show the utility of our proposed metric, we estimated the available average bandwidth in two botnets.

Using techniques described in [16], we measured one

botnet of approximately 50,000 unique members in February 2005, and estimated the bandwidth of 7,326 bots chosen in a uniformly random manner. Likewise, we measured the bandwidth of a 3,391 member subsample from a 48,000 member botnet in January of 2006.

We used the `tmetric` [7] tool to perform the bandwidth estimation. `tmetric` essentially uses successively larger probes to estimate the bandwidth to a host. We used a high-capacity link (OC-48) close to our network's core routers, so that we were more likely to measure the end host's available bandwidth, rather than any limitations in our internal network. Dozens of probes sent over minutes were used to obtain an average. Again, we note that the networking community has developed far more sophisticated techniques to estimate bandwidth *end-to-end*. We believe our simple measurements were useful to quickly obtain a first order approximation of the average bandwidth in an entire distributed *network*.

Figure 5(a) and (b) show the distribution of bandwidth, with min/max and average bandwidth values observed during the probes. Table 2 shows the average available bandwidth (that the botmaster can utilize) from a single bot. Using Eq(1) and without considering the diurnal sensitivity, we can calculate the average available bandwidth for botmaster to use on one bot is around 53.3004 Kbps. For data set 2, the average is 34.8164 Kbps, a few less than the first case.

But when accounting for diurnal sensitivity, and assuming the average online times for each class of bots is [2, 4, 24] hours, then the final average bandwidth for botmaster on one bot is 22.7164 Kbps. If a botnet has a size of 50K, then on average the botmaster consistently has more than 1Gbps bandwidth on average at anytime. This suggests the botnet could easily launch a successful denial of service attack on almost any web site. (Indeed, during our period of observation, the 50K member botnet did DDoS several websites that only had 100Mbs transit.) For data set 2, the weighted bandwidth is 14.6378 Kbps– comparatively lower.

The metric therefore reveals something counter-intuitive about botnets. Just looking at the sampled bandwidth in Figure 5(a) and (b), it seems that the botnets have roughly the same maximum bandwidth, and the same number of bots, and therefore have the same general utility from a DDoS perspective. When accounting for diurnal changes in populations, however, the second botnet (plotted in Figure 5(b)) has approximately half the average available bandwidth, despite having only 2,000 less members than the other network. If network administrators had to select between these two botnets and prioritize a single response effort, the simple bandwidth estimate $B$ shows a higher utility in the botnet in Figure 5(a).

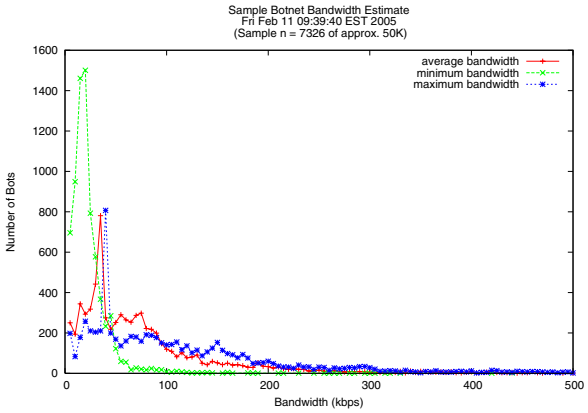
Our bandwidth estimate metric may have other uses besides priority ranking botnets. This exercise suggests that diurnal changes in botnet membership can significantly affect a botnet's utility as a DDoS vehicle. We leave for future work an analysis of how this metric can be leveraged in a targeted attack on a botnet. That is, we speculate that responders might significantly reduce a botnet's DDoS potential by targeting the "high-speed" members of a botnet. The bandwidth $B$ metric should let researchers measure their progress in such a response, and tell them how many more high-speed members must be removed, relative to the mix of low-speed members, for a given estimated diurnal usage pattern.

## 4 Related Work

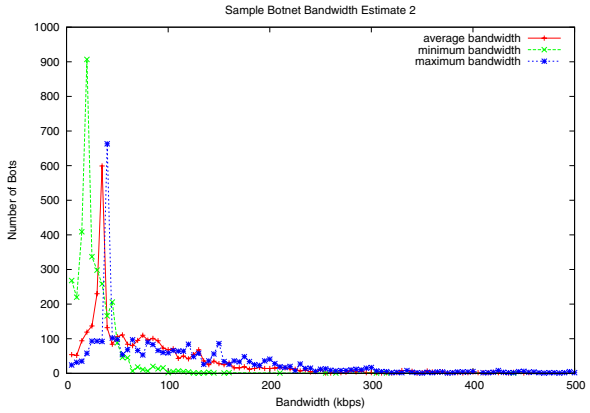
Our work fits into the larger body of literature addressing the statistical mechanics of complex networks [4]. Others have studied the brittle nature of scale-free networks and resilience of random networks in other contexts [5, 23, 40]. Our work adapts these findings to the particular domain of botnets. The topology of networks under active decay was analyzed in [40]. Many of the results in [40] anticipate our own. The authors took a fascinating look at all domains of network structures (e.g., including terror cells, and global history), and not just computer networks. By restricting our analysis to botnets, we identified several unique and interesting phenomena not considered in [40]. For example, the authors in [40] suggest a strategy of splitting high-degree nodes to avoid targeted responses. This is analogous to increasing $C$ in scale free networks, discussed in 3. Since we focused on the botnet domain, we were able to further observe that this results in a degraded transitivity.

Botnet research is still maturing. The work in [13] anticipated many of the general categories of botnets analyzed in Section 2, including the difficulty in responding to different type of botnet taxonomies. The models and empirical data we presented in Section 2 flesh out and formalize the intuitive discussion in [13]. Recently, advanced botnets with complex network structures have been studied. Vogt, et al. [54] presented a super-botnet, the network of many independent, small botnet, which is a special case of a random graph botnet. Wang, et al. [55] introduced an advanced hybrid peer-to-peer botnet. Grizzard, et al. [19] provided an overview of P2P botnet and a case study of a specific bot.

There have been several works on botnet measurement. In [17, 46], the authors used honeynets to track existing IRC-based botnets and report a few simple statistics about botnets. Rajab, et al., [47], argue that the estimation of botnet size is actually hard in practice, and call for further research on the measurement of botnets. We believe our analysis in Sections 2 and 3 help with this problem. Wang, et al. [55], propose two metrics, connection ratio and degree ratio, to measure the resilience of removing mostly-connected bots from a botnet. In this paper, we propose




(a) Botnet 1: sampling 7,326 of approximately 50K




(b) Botnet 2: sampling 3,380 of 48k.

**Figure 5. An estimate of bandwidth usage in two sampled botnets. Just examining the maximum bandwidth, the botnets appear to have roughly the same distribution of high, medium and low-speed bots, and therefore appear to pose the same DDoS threat potential. The analysis below, however, shows how diurnal changes significantly reduce the average available bandwidth of (b), compared to (a).**

| Bot Bandwidth Type | Low (std) | Medium (std) | High (std) |
|---|---|---|---|
| Dataset 1: Average Max BW | 28.2356 (11.9612) | 119.1708 (54.2837) | 601.7158 (989.2654) |
| Average usage BW | 19.2395 (8.5739) | 74.3089 (34.4838) | 364.8714 (636.2601) |
| Average available BW | 8.9961 | 44.8619 | 236.8444 |
| Dataset 2: Average Max BW | 33.9266 (9.3649) | 116.0036 (51.0478) | 432.4184 (354.3628) |
| Average usage BW | 27.9144 (8.8397) | 86.2721 (33.3334) | 280.6805 (229.9276) |
| Average available BW | 6.0122 | 29.7315 | 151.7379 |

**Table 2. Average and standard deviation of bandwidth observed in two botnets, plotted in Figure 5**

more metrics, and not only measure the robustness, but also the effectiveness and efficiency of a botnet for the botmaster.

Researchers have attempted to study the botnet problem in a systematic way. Barford and Yegneswaran [8] codify the capabilities of malware by dissecting four widely-used Internet Relay Chat (IRC) botnet codebases. Each codebase is classified along seven dimensions including botnet control mechanisms, host control mechanisms, propagation mechanisms, exploits, delivery mechanisms, obfuscation and deception mechanisms. Trend Micro [35] also proposed a taxonomy of botnet threats, along dimensions such as attacking behavior, command and control model, rally mechanism, communication protocol, evasion technique, and other observable activities. Our taxonomy is different from this existing work. It is a use-driven taxonomy focused on the botnet structure. We study the problem from specific aspects such as the structure and the utility metrics of the botnets.

Our taxonomy and discussion of general response options presumes a sensitive detection system. We have not considered detection of botnets, and urge further research. We note preliminary detection work in misuse systems [22], and IRC traces [11]. Significantly, this early work focuses on tracking *individual* bots (e.g., to obtain a binary) and not the *network* cloud of coordinated attackers addressed in our study. In [13,17], researchers focused on countering bot*nets* (as opposed to individual bots), which used honeypots and broad sensors to track and infiltrate botnets. Recently, there are several works on the botnet detection problem. BotHunter [20] is a bot detection system using IDS-Driven Dialog Correlation according to defined bot infection dialog model. Rishi [18] uses the similarity of nick name to detect botnet channel. Karasaridis, et al. [28], proposed to detect botnet command and control through passive network flow record analysis.

## 5 Conclusion

Botnets present significant new challenges for researchers. The fluid nature of this problem requires researchers anticipate future botnet strategies and design effective response techniques. To assist in this effort, we proposed key metrics to measure botnet utility for various activities, and presented a taxonomy of botnets based on topological structure.

Our analysis shows that random network models (either direct Erdös-Rényi models or structured P2P systems) give botnets considerable resilience. Such formations resist both random and targeted responses. Our analysis also showed that targeted removals on scale free botnets offer the best response.

We have demonstrated the utility of this taxonomy and proposed metrics by using both simulation and real-world botnet experiments. We also performed some novel measurements of a P2P botnet to demonstrate the utility of our proposed metrics.

In our future work, we will refine our metrics of botnet utilities, explore effective techniques for more accurate estimations of these metrics in real-world botnets. We will also identify metrics that measure difficulties in detection, and the evasive potential of botnets.

## Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. CCR-0133629 and CNS-0627477, and by the U.S. Army Research Office under Grant No. W911NF0610042. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation and the U.S. Army Research Office.

## References

[1] Waste: Anonymous, secure, encrypted sharing. `http://waste.sourceforge.net/index.php?id=projects`, 2007.

[2] WineHQ: Windows API Implementation for Li5Dnux. `http://www.winehq.com/`, 2007.

[3] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(509), 1999.

[4] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1), 2002.

[5] R. Albert, H. Jeong, and A.-L. Barabási. Error and attack tolerance of complex networks. *Nature*, 406:378–382, 2000.

[6] M. Allman and V. Paxson. On estimating end-to-end network path properties. In *ACM Special Interest Group on Data Communication (SIGCOMM '99)*, volume 29, 1999.

[7] M. Bacarella. TMetric bandwidth estimation tool. `http://michael.bacarella.com/projects/tmetric/`, 2007.

[8] P. Barford and V. Yegneswaran. An inside look at botnets. In *In Series: Advances in Information Security*. Springer Verlag, 2006.

[9] V. Berk, R. Gray, and G. Bakos. Using sensor networks and data fusion for early detection of active worms. In *Proceedings of the SPIE AeroSense*, 2003.

[10] B. Bollobás. *Random Graphs*. Academic Press, 1985.

[11] D. Brumley. Tracking hackers on IRC. `http://www.doomdead.com/texts/ircmirc/TrackingHackersonIRC.htm`, 2003.

[12] E. Calimbo. Packetnews: The ultimate irc search engine. `http://www.packetnews.com/`, 2007.

[13] E. Cooke and F. Jahanian. The zombie roundup: Understanding, detecting, and disrupting botnets. In *Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI '05)*, 2005.

[14] D. Dagon. The network is the infection. `http://www.caida.org/projects/oarc/200507/slides/oarc0507-D\agon.pdf`, 2005.

[15] D. Dagon, A. Takar, G. Gu, X. Qin, and W. Lee. Worm population control through periodic response. Technical report, Georgia Institute of Technology, June 2004.

[16] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS'06)*, 2006.

[17] F. C. Freiling, T. Holz, and G. Wicherski. Botnet tracking: Exploring a root-cause methodology to prevent distributed denial-of-service attacks. Technical Report ISSN-0935-3232, RWTH Aachen, April 2005.

[18] J. Goebel and T. Holz. Rishi: Identify bot contaminated hosts by irc nickname evaluation. In *USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, 2007.

[19] J. B. Grizzard, V. Sharma, C. Nunnery, B. B. Kang, and D. Dagon. Peer-to-peer botnets: Overview and case study. In *USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, 2007.

[20] G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee. Bothunter: Detecting malware infection through ids-driven dialog correlation. In *16th USENIX Security Symposium (Security'07)*, 2007.

[21] G. Gu, M. Sharif, X. Qin, D. Dagon, W. Lee, and G. Riley. Worm detection, early warning and response based on local victim information. In *20th Annual Computer Security Applications Conference (ACSAC)*, 2004.

[22] C. Hanna. Using snort to detect rogue IRC bot programs. Technical report, October 2004.

[23] P. Holme, B. J. Kim, C. N. Yoon, and S. K. Han. Attack vulnerability of complex networks. *Phys. Rev.*, E65(056109), 2002.

[24] J. Horrigan. Broadband adoption at home in the united states: Growing but slowing. `http://web.si.umich.edu/tprc/papers/2005/501/TPRC\%20Horrigan\%20Broadband.2005b.pdf`, 2005.

[25] M. Jain and C. Dovrolis. End-to-end available bandwidth: Measurement. methodology, dynamics, and relation with tcp. In *Special Interest Group on Data Communication (SIGCOMM '02)*, 2002.

[26] X. Jiang, D. Xu, H. J. Wang, and E. H. Spafford. Virtual playgrounds for worm behavior investigation. Technical Report CERIAS Technical Report (2005-24), Purdue University, February 2005.

[27] C. Kalt. Internet relay chat: Architecture. `http://www.faqs.org/rfcs/rfc2810.html`, 2000.

[28] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. In *USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, 2007.

[29] K. Killourhy, R. Maxion, and K. Tan. A defense-centric taxonomy based on attack manifestations. In *International Conference on Dependable Systems and Networks (ICDS'04)*, 2004.

[30] C. E. Landwehr, A. R. Bull, J. P. McDermott, and W. S. Choi. A taxonomy of computer program security flaws, September 1994.

[31] U. Lindqvist and E. Jonsson. How to systematically classify computer security intrusions. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 154–163, 1997.

[32] LURHQ. Zindos worm analysis. `http://www.lurhq.com/zindos.html`, 2004.

[33] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In *ICS '02: Proceedings of the 16th international conference on Supercomputing*, pages 84–95, New York, NY, USA, 2002. ACM Press.

[34] MaxMind LLC. Maxmind - ip geolocation and online fraud prevention. `http://www.maxmind.com/`, 2007.

[35] T. Micro. Taxonomy of botnet threats. Technical report, Trend Micro White Paper, November 2006.

[36] S. Milgram. The small world problem. *Psychology Today*, 2(60), 1967.

[37] D. Moore. Code-red: A case study on the spread and victims of an internet worm. `http://www.icir.org/vern/imw-2002/imw2002-papers/209.ps.gz`, 2002.

[38] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver. Inside the slammer worm. *IEEE Magazine on Security and Privacy*, 1(4), July 2003.

[39] D. Moore, C. Shannon, G. M. Voelker, and S. Savage. Internet quarantine: Requirements for containing self-propagating code. In *Proceedings of the IEEE INFOCOM 2003*, March 2003.

[40] S. Nagarja and R. Anderson. The topology of covert conflict. Technical Report UCAM-CL-TR-637, University of Cambridge, July 2005.

[41] J. Nazario. Botnet tracking: Tools, techniques, and lessons learned. In *Black Hat*, 2007.

[42] M. Newman, S. Strogatz, and D. Watts. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev.*, E64(026118), 2001.

[43] Nielsen NetRatings. Average web usage. `http://www.nielsen-netratings.com/reports.jsp?section=pub_reports&report=usage&period=weekly`, 2007.

[44] J. J. Parekh. Columbia ids worminator project. `http://worminator.cs.columbia.edu/`, 2004.

[45] L. Qin, C. Pei, E. Cohen, L. Kai, and S. Scott. Search and replication in unstructured peer-to-peer networks. In *16th ACM International Conference on Supercomputing*, 2002.

[46] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM on Internet Measurement (IMC)*, pages 41–52, 2006.

[47] M. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. My botnet is bigger than yours (maybe, better than yours): Why size estimates remain challenging. In *USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, 2007.

[48] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Proceedings of the ACM Conference of the Special Interest Group on Data Communication (SIGCOMM)*, pages 161–172, August 2001.

[49] M. Ripeanu, I. Foster, and A. Iamnitchi. Mapping the gnutella network: Properties of large-scale peer-to-peer systems and implications for system design. *IEEE Internet Computing Journal*, 6(1), 2002.

[50] C. Shannon and D. Moore. The spread of the witty worm. *Security & Privacy Magazine*, 2(4):46–50, 2004.

[51] A. Singh, T.-W. Ngan, P. Druschel, and D. Wallach. Eclipse attacks on overlay networks: Threats and defenses. In *Proceedings of INFOCOM'06*, April 2006.

[52] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California, August 2001.

[53] R. Vogt and J. Aycock. Attack of the 50 foot botnet. Technical report, Department of Computer Science, University of Calgary, August 2006.

[54] R. Vogt, J. Aycock, and M. Jacobson. Army of botnets. In *Proceedings of NDSS'07*, 2007.

[55] P. Wang, S. Sparks, and C. C. Zou. An advanced hybrid peer-to-peer botnet. In *USENIX Workshop on Hot Topics in Understanding Botnets (HotBots'07)*, 2007.

[56] D. Watts and S. Strogatz. *Nature*, 393(440), 1998.

[57] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. A taxonomy of computer worms. In *2003 ACM Workshop on Rapid Malcode (WORM'03)*. ACM SIGSAC, October 2003.

[58] Y. Xie, H.-A. Kim, D. R. O'Hallaron, M. K. Reiter, and H. Zhang. Seurat: A pointillist approach to network security, 2004.

[59] Y. Zhang and V. Paxson. Detecting stepping stones. In *Proceedings of the 9th USENIX Security Symposium*, August 2000.

[60] C. Zou and R. Cunningham. Honeypot-aware advanced botnet construction and maintenance. In *International Conference on Dependable Systems and Networks (DSN)*, pages 199–208, June 2006.

[61] C. Zou, D. Towsley, W. Gong, and S. Cai. Routing worm: A fast, selective attack worm based on ip address information. Technical Report TR-03-CSE-06, Umass ECE Dept., November 2003.

[62] C. C. Zou, L. Gao, W. Gong, and D. Towsley. Monitoring and early warning for internet worms. In *Proceedings of 10th ACM Conference on Computer and Communications Security (CCS'03)*, October 2003.

[63] C. C. Zou, W. Gong, and D. Towsley. Code red worm propagation modeling and analysis. In *Proceedings of 9th ACM Conference on Computer and Communications Security (CCS'02)*, October 2002.

[64] C. C. Zou, W. Gong, and D. Towsley. Worm propagation modeling and analysis under dynamic quarantine defense. In *Proceedings of ACM CCS Workshop on Rapid Malcode (WORM'03)*, October 2003.