# Revealing Botnet Membership Using DNSBL Counter-Intelligence

Anirudh Ramachandran, Nick Feamster and David Dagon
College of Computing, Georgia Institute of Technology
{avr, feamster, dagon}@cc.gatech.edu

## ABSTRACT

Botnets—networks of (typically compromised) machines—are often used for nefarious activities (*e.g.*, spam, click fraud, denial-of-service attacks, etc.). Identifying members of botnets could help stem these attacks, but *passively* detecting botnet membership (*i.e.*, without disrupting the operation of the botnet) proves to be difficult. This paper studies the effectiveness of monitoring lookups to a DNS-based blackhole list (DNSBL) to expose botnet membership.

We perform *counter-intelligence* based on the insight that botmasters themselves perform DNSBL lookups to determine whether their spamming bots are blacklisted. Using heuristics to identify which DNSBL lookups are perpetrated by a botmaster performing such reconnaissance, we are able to compile a list of likely bots. This paper studies the prevalence of DNSBL reconnaissance observed at a mirror of a well-known blacklist for a 45-day period, identifies the means by which botmasters are performing reconnaissance, and suggests the possibility of using counter-intelligence to discover likely bots. We find that bots are performing reconnaissance on behalf of other bots. Based on this finding, we suggest counter-intelligence techniques that may be useful for early bot detection.

## 1. Introduction

Internet malice has evolved from pranks conceived and executed by amateur hackers to a global business involving significant monetary gains for the perpetrators [19]. Examples include: (1) unsolicited commercial email ("spam"), which threatens to render email useless by immensely decreasing the signal-to-noise ratio of traffic [17]; (2) denial of service attacks, which have become common [12], and (3) click fraud, whereby a group of attackers send bogus "clicks" for online advertisements that mimic legitimate request patterns, swindling advertisers out of large sums of money [4].

Botnets are a root cause of these problems [8], since they allow attackers to distribute tasks over thousands of hosts distributed across the Internet. A botnet is network of compromised hosts ("bots") connected to the Internet under the control of a single entity ("botmaster", "controller", or *command and control*) [5]. The large cumulative bandwidth and relatively untraceable nature of spam from bots makes botnets an attractive choice for large-scale spamming. Previous work provides further background on botnets [5, 6].

If network operators and system administrators could reliably determine whether a host is a member of a botnet, they could take appropriate steps towards mitigating the attacks they perpetrate. Although previous work has described an *active* detection technique using DNS hijacking technique and social engineering [6], there are few efficient methods to *passively* detect and identify bots (*i.e.*, without disrupting the operation of the botnet). Indeed, detecting botnets proves to be very challenging: a victim of a botnet attack can typically only observe the attack from a single network, from which point the attack traffic may closely resemble the traffic of legitimate users. Regrettably, the state-of-the-art in botnet identification is based on user complaints, localized honeypots and intrusion detection systems, or through the complex correlation of data collected through darknets [13].

We propose a set of techniques to identify botnets using *passive* analysis of DNS-based blackhole list (DNSBL) lookup traffic. Many Internet Service Providers (ISPs) and enterprise networks use DNSBLs to track IP addresses that originate spam, so that future emails sent from these IP addresses can be rejected. For the same reason, botmasters are known to sell "clean" bots (*i.e.*, not listed in any DNSBL) at a premium. This paper addresses the possibility of performing *counter-intelligence* to help us discover identities of bots, based on the insight that *botmasters themselves must perform "reconnaissance" lookups to determine their bots' blacklist status*.

The contributions of this paper include:

**1. Passive heuristics for counter-intelligence.** We develop heuristics to distinguish DNSBL reconnaissance queries for a botnet from legitimate DNSBL traffic (either offline or in real-time), to identify likely bots. These heuristics are based on an enumeration of possible lookup techniques that botmasters are likely to use to perform reconnaissance, which we detail in Section 2. Unlike previous detection schemes, our techniques are *covert* and do not disrupt the botnet's activity.

**2. Study of DNSBL reconnaissance techniques.** We study the prevalence of DNSBL reconnaissance by analyzing logs from a mirror of a well-known blackhole list for a 45-day period from November 17, 2005 to December 31, 2005. Section 4 discusses the prevalence of the different types of reconnaissance techniques that

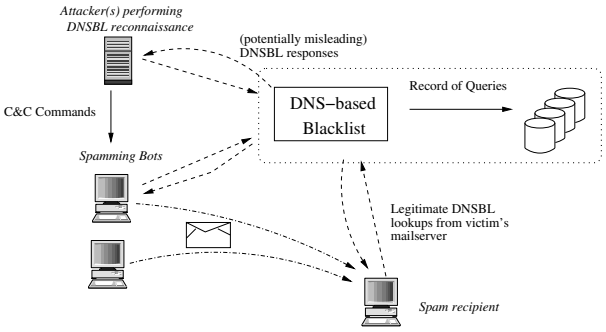


**Figure 1: DNSBL-based Spam Mitigation Architecture.**

we observed. Much to our surprise, we find that bots are performing reconnaissance on behalf of other (possibly newly infected) bots. Although some bots perform a large number of reconnaissance queries, it appears that much of the reconnaissance activity is spread across many bots each of which issue few queries, thus making detection more difficult.

**3. Identification of new bots.** We analyze DNSBL queries that are likely being performed by botmasters to identify "clean" bots. Such reconnaissance usually precedes the use of bots in an attack, suggesting the possibility that this DNSBL counter-intelligence can be used to bolster responses. Section 3 demonstrates the possibility of such early warning. To validate our detection scheme, we correlate the IP addresses of these likely bots with data collected at a botnet sinkhole (sinkholing technique explained in previous work [6]) over the same time period (this dataset has been used as "ground truth" for botnet membership in previous studies [6, 17]).

**4. DNSBL-based countermeasures.** Our heuristics could be used to detect reconnaissance in real-time. This ability potentially allows for active countermeasures, such as returning misleading responses to reconnaissance lookups, as shown in Figure 1. We revisit this topic in Section 5.

## 2. Model of Reconnaissance Techniques

This section describes our model for DNSBL reconnaissance techniques (*i.e.*, the techniques that botmasters may be using to determine whether bots have been blacklisted). Our goal in developing these models and heuristics is to distinguish DNSBL queries issued by botmasters from those performed by legitimate mail servers.[1]

### 2.1 Properties of Reconnaissance Queries

Our detection heuristics are based on the construction of a *DNSBL query graph*, where an edge in the graph from node $A$ to node $B$ indicates that node $A$ has issued a query to a DNSBL to determine whether node $B$ is listed. After constructing this graph, we develop detection heuristics based on the expected *spatial* and *temporal* characteristics of legitimate lookups versus reconnaissance-based lookups. These characteristics hold primarily in cases when members of the botnet are not performing queries on behalf of each other, a case that makes detecting reconnaissance more difficult, as we explain in Section 2.2.3. As we describe below, our detection heuristics exploit both spatial and temporal properties of the DNSBL query graph.

**Property 1 (Spatial relationships)** *A legitimate mail server will perform queries* and *be the object of queries. In contrast, hosts performing reconnaissance-based lookups will only perform queries; they will* not *be queried by other hosts.*[2]

In other words, legitimate mail servers are likely to be queried by other mail servers that are receiving mail from that server. On the other hand, a host that is not itself being looked up by any other mail servers is, in all likelihood, not a mail server. We can use this observation to identify hosts that are likely performing reconnaissance: lookups from hosts that have a high *out-degree* in the DNSBL query graph (*i.e.*, hosts that are performing many lookups) but have a low *in-degree* are likely unrelated to the delivery of legitimate mail. To quantify this effect, we define the *lookup ratio*, $\lambda$, of some node $n$ as follows:

$$\lambda_n = \frac{d_{n,out}}{d_{n,in}}$$

where $d_{out}$ is the number of distinct IP addresses that node $n$ queries, and $d_{in}$ is the number of distinct IP addresses that issue a query for node $n$.[3] This metric is most effective when hosts performing reconnaissance are disjoint from hosts that are actually used to spam, which appears to the case today.However, as reconnaissance techniques become increasingly more sophisticated (as we describe in Section 2.2.3), this metric may become less useful. Still, we find that this metric proves to be quite useful in detecting many instances of DNSBL-based reconnaissance.

The *temporal arrival pattern* of queries at the DNSBL by hosts performing reconnaissance may differ from temporal characteristics of queries performed by legitimate hosts. We expect this to be the case because, whereas legitimate DNSBL lookups are driven by the arrival of actual email, reconnaissance queries will not reflect any realistic arrival patterns of actual email.

**Property 2 (Temporal relationships)** *A legitimate mail server's DNSBL lookups reflect actual arrival patterns of real email messages: legitimate lookups are typically driven automatically when emails arrive at the mail server and will thus arrive at a rate that mirrors the arrival rates of emails. Reconnaissance-based lookups, on the other hand, will not mirror the arrival patterns of legitimate email.*

We may be able to exploit the fact that email traffic tends to be diurnal [9] to tease apart DNSBL lookups that are

driven by actual mail arrival from those that are driven by reconnaissance. Discovering reconnaissance activity using this method is a topic for future work.

## 2.2 Reconnaissance Techniques

In this section, we describe three classes of DNSBL reconnaissance techniques that may be performed by botmasters: *single-host, or third-party, reconnaissance*; *self-reconnaissance*; and *reconnaissance using other bots*. For each case, we describe the basic mechanism, the heuristics that we can use to detect reconnaissance in each of these cases, and how each technique may complicate detection.

### 2.2.1 Third-party Reconnaissance

In *third-party reconnaissance*, a botmaster performs DNSBL lookups from a single host for a list of spamming bots; this host may be the command-and-control of the botnet, or it might be some other dedicated machine. In any case, we hypothesize that the machine performing the lookups in these cases is not likely to be a mail server. Single-host reconnaissance, if performed by a machine other than a mail server, is easily detected, because the node performing reconnaissance will have a high value of $\lambda_n$.

Once detected, single-host reconnaissance may provide useful information to aid us in revealing botnet membership. First, once we have identified a single host performing such lookups, the operator of the DNSBL can monitor the lookups issued by that host over time to track the identity of hosts that are likely bots. If the identity of this querying host is relatively static (*i.e.*, if its IP address does not change over time, or if it changes slowly enough so that its movements can be tracked in real-time), the DNSBL operator could take active countermeasures, such as intentionally returning incorrect information about bots' status in the blacklist, a possibility we discuss in more detail in Section 5.

### 2.2.2 Self-Reconnaissance

Single-host reconnaissance is simple, but it is susceptible to detection. To remain more stealthy, and to distribute the workload of performing DNSBL reconnaissance, botmasters may begin to distribute these lookups *across the botnet itself*. A simple (albeit sub-optimal) way to distribute these queries is to have a bot perform reconnaissance on its own behalf ("self-reconnaissance"); in other words, each bot could issue a DNSBL query to itself (*i.e.*, to determine whether it was listed) before sending spam to the victim.

In this case, identifying a reconnaissance-based DNSBL query is fairly straightforward, because, except in cases of misconfiguration, a legitimate mail server is unlikely to issue a DNSBL lookup for itself. Even though this technique has the advantage of distributing the load of reconnaissance across the botnet, we did not observe this technique being used in practice, likely because a self-query is a dead giveaway.

### 2.2.3 Distributed Reconnaissance

A more stealthy way to distribute the operation across the botnet is to have each bot perform reconnaissance on behalf of other bots either in the same botnet or in other botnets. For instance, note that Property 1 is unlikely to hold: in this case, the nodes performing reconnaissance will also be queried by other mail servers to which they send spam. As a result, these nodes are likely to have a high $d_{n,in}$, unlike nodes performing single-host reconnaissance. Ultimately, detecting this type of reconnaissance activity may require mining temporal properties (*e.g.*, Property 2).

Although using the botnet itself for DNSBL reconnaissance is more discreet than performing this reconnaissance from a single host, a network operator who positively identifies a small number of bots (*e.g.*, starting with a small hit-list of known bots, probably by using a honeynet with known infected machines). As discussed in Section 4, if this *seed list* of bots performs queries for other hosts, it is likely that these machines are also bots.

We suspected that this mode of reconnaissance would be uncommon, possibly because of the complexity involved in implementing and operating such a system (*e.g.*, keeping track of nodes in the looked-up botnet, disseminating this information to the querying nodes etc.). Much to our surprise, we did witness this behavior; we present these results in Section 4.

# 3. Data and Analysis

This section describes our data collection and analysis. We first describe our DNSBL dataset and its limitations. Then, we describe how this dataset is used to construct the DNSBL query graph described in Section 2.

## 3.1 Data Collection and Processing

Our study primarily involves two datasets collected from the same time period (November 17, 2005 to December 31, 2005): (1) the DNSBL query logs to a mirror of a large DNSBL, and (2) the logs of bot connections to a sinkhole for a Bobax botnet [2]. Unlike most botnets, the Bobax bot is designed solely for spamming [1], increasing the likelihood that a query for known Bobax host is the consequence of the querying mail server having received spam from that host.

To verify whether the scheme we propose is indeed able to discover *additional* bots, we compared the IP addresses in the DNSBL query graph against the IP addresses of spammers in a large spam corpus collected at a spam honeypot (the setup of this honeypot is described in our earlier work [17]).

## 3.2 Analysis and Detection

In this section, we describe how the DNSBL query graph is constructed. Definitions for the terminology used in our algorithm follow: (1) $B$, the set of IP addresses that attempted to connect to the Bobax sinkhole during the observation period (November 17, 2005–

```
CONSTRUCTGRAPH()
create empty directed graph G

/* Parsing */
for each DNSBL query:
    Identify querier and queried

    /* Pruning */
    if querier ∈ B or queried ∈ B then
        add querier and queried to G if they
            are not already members of G
        if there exists an edge E(querier, queried) ∈ G then
            increment the weight of E(querier, queried)
        else
            add E(querier, queried) to G with weight 1
```

**Figure 2: Algorithm to construct a DNSBL query graph**

December 31, 2005); (2) $querier$, the IP address of the host that performs a given DNSBL query; (3) $queried$, the IP address of the host that is looked up in a DNSBL query; and (4) $G$, the DNSBL query graph constructed as a result of the algorithm.

The graph construction algorithm takes as input a set of DNSBL query logs (we use `tcpdump` for packet captures) and the set $B$ and outputs a directed graph $G$. The algorithm, summarized in Figure 2, consists of two main steps: *parsing* and *pruning*. As the algorithm suggests, we prune DNSBL queries to only include edges which have at least one end (either $querier$ or $queried$) present in the set $B$. Pruning is performed for efficiency reasons: the full DNSBL query logs mostly contain queries from legitimate mail servers. Using $B$ to prune the complete query graph allows us to concentrate on a subgraph which has a higher percentage of reconnaissance lookups than the unpruned graph. We recognize that our analysis will overlook reconnaissance activity where both the $querier$ or $queried$ nodes are not members of $B$. To address this shortcoming, we perform a *query graph extrapolation* after the algorithm is run. In this step, we make a second pass over the DNSBL query logs and add edges if at least one of the endpoints of the edge (*i.e.*, either $querier$ or $queried$) is already present in the graph. Query graph extrapolation is repeated until no new edges are added to $G$.

We then compute $\lambda_n$ for each node in the graph (Property 1), which allows us to identify nodes involved in reconnaissance techniques described in Section 2. Although the results in Section 4 suggest that some bots have large values of $\lambda_n$, techniques that use a large number bots to look each other up may be undetectable with this metric. We are developing techniques based on Property 2 to further improve our detection.

## 4. Preliminary Results

This section presents preliminary results using Property 1 to identify DNSBL reconnaissance activity on the observed DNSBL query graph. We emphasize that the reconnaissance being performed by bots is distinctly under the radar as far as total DNSBL traffic is concerned: the pruned traffic amounts to less than 1% of the total DNSBL traffic. In this section, we present two surprising results: First, botnets are being used to perform DNSBL reconnaissance on behalf of bots in other botnets, which has implications for botnet detection. Second, the distribution of these queries across bots suggests that some DNSBL reconnaissance activities may be detectable in real-time, which has implications for early detection and mitigation.

| *Node #* | ASN of Node | Out-degree | known spammers |
|---|---|---|---|
| 1 | Everyone's Internet (AS 13749) | 36,875 | 12 |
| 2 | IQuest (AS 7332) | 32,159 | 7 |
| 3 | UUNet (AS 701) | 31,682 | 5 |
| 4 | UPC Broadband (AS 6830) | 26,502 | 8 |
| 5 | E-xpedient (AS 17054) | 19,530 | 4 |

**Table 1: AS numbers of hosts which have the highest out-degrees. The last column shows the number of hosts queried by this node that are known spammers (verified using logs from our spam sinkhole).**

Attempts to validate our hypotheses from Section 2 resulted in some interesting discoveries, including the discovery of new bots. We initially expected that most DNSBL lookups would be third-party lookups, as described in Section 2.2.1, and that we would be able to validate the queried nodes as being known bots. Instead, we discovered the opposite: the nodes with the highest values of $\lambda_n$ in the pruned graph were *known* bots, while *the queried nodes in the graph were new, previously unknown bots*. Further, using data from our spam sinkhole [17], we found that some of these nodes were Windows machines and confirmed spam originators. This finding suggests that, in general, it may be possible to start with a set of known bots and use the DNSBL graph to "bootstrap" the discovery of new bots.

Table 1 shows five of the top queriers (*i.e.*, high out-degree nodes), *all* of which are known bots from our Bobax trace. Even more interesting is the fact that a few IP addresses queried by these nodes actually sent spam to our spam honeypot. Moreover, nearly all of IP addresses that sent spam to our honeypot were *not* present in our list of known bots. Due to the fact that our honeypot only captures a small portion of the Internet's spam, the fraction of total reconnaissance queries that we can confirm as spamming bots is small. Still, we believe it strongly suggests evidence of a known bot performing DNSBL reconnaissance on a distinct (and possibly newly compromised) botnet.

Figure 3 shows the distribution of out-degrees for all querying nodes present in the pruned DNSBL query graph. The long tail also confirms that bots already have the capability to distribute these queries, which is cause for concern. Our view of DNSBL queries is narrow (most querying nodes are geographically close to the DNSBL mirror), so we expect that more vantage points of DNSBL lookups would reveal other prominent "play-

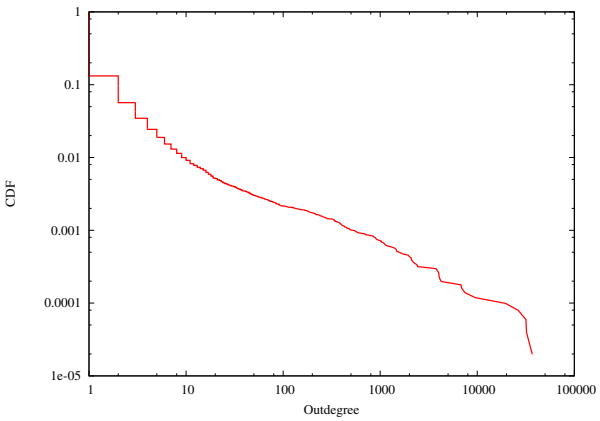


**Figure 3: CDF of the distribution of out-degrees for querying IP addresses.**

ers". The fact that the prominent players in our analysis were also bots suggests that these nodes may also be obvious candidates for the mitigation techniques described in Section 5.

## 5. Countermeasures

In Section 4, we found that the known bots in our Bobax trace were not the targets of lookups, but instead were issuing lookups for other, possibly newly compromised bots. This finding suggests a possible technique that could be used for the discovery of new bots, even without an initial list of suspects: an initial set of suspect IP addresses could be constructed by establishing a spam trap, which according to both previous work [17] and the observations in this paper, appear to be largely bots. Alternatively, a suspect node could be detected simply by identifying nodes in the DNSBL query graph with a high value of $\lambda_n$. Beginning with this initial suspect list, an operator may be able to conclude that, not only are the nodes that this node is querying likely bots, but also the node itself is likely a bot. If there are other high-degree nodes also querying the same bots, a detection algorithm might be able to "walk" the DNSBL graph (*e.g.*, from parent to parent) to discover multiple distinct botnets.

We believe that using such techniques to aggressively monitor botnet-based DNSBL reconnaissance may prove to be useful for mitigating spam: as noted in our previous work [17], most bots send a very low volume of spam to any single domain; thus, reporting a bot to blacklists *after* the spam is received may not be effective.

With the ability to distinguish reconnaissance queries from legitimate queries, a DNSBL operator might be able to mitigate spam more effectively. We speculate one possibility as follows: an operator could tune the behavior of the blackhole list server to mislead a botmaster, using a class of techniques we call *reconnaissance poisoning*. On one hand, the DNSBL could trick the botmaster into thinking that a particular bot was "clean" (*i.e.*, unlisted) when in fact it was listed, which would induce the botmaster to unwittingly send spam from blacklisted machines. On the other hand, the DNSBL could also reply to a reconnaissance query with an indication that a host was listed, even though it was not listed, thereby discouraging a botmaster from using a machine that would likely be capable of successfully sending spam.

Of course, active countermeasures such as reconnaissance poisoning do run the risk of false positives: if we mistakenly attribute a legitimate DNSBL query to a reconnaissance-based query, we could mislead a legitimate mail server into either mistakenly accepting spam that would have otherwise been rejected or, more regrettably, rejecting legitimate email. Such techniques could also be defeated if the botmaster queries multiple blacklist providers that maintain independent lists. Investigating the extent to which our detection metrics are subject to false positives, as well as the extent to which these false positives interfere with a legitimate mail server's filtering techniques, is part of our ongoing work.

## 6. Related Work

Botnets have been in use as vehicles of cybercrime for quite some time, but studies on how they spread, and techniques to counter them, are relatively scarce. Previous research has traced the history of botnets [18, 21, 22] and common modes of botnet operation [5]. This section briefly discusses previous botnet detection techniques and previous research on DNSBL traffic analysis.

Previous work has identified bots by examining the communication protocols used by botnets (*e.g.*, for "rallying"), most notably Internet Relay Chat (IRC) [7, 23]. Some have suggested the use of such protocols to identify and remediate botnets. For example, researchers have joined IRC-based botnets and enumerated victims using IRC commands [8]; others have used network traffic to identify IRC zombies [16]. Some researchers have identified bot victims by observing the unwanted traffic they generate, *e.g.*, the `RST` storms or backscatter generated by DDoS attacks using forged source addresses [15].

Studies show that many botnets are IRC-based [5, 22], though other protocols are being used [14]. Attempts have been made to detect such botnets using misuse-detection or basic intrusion detection analysis [3, 10]. Dagon *et al.* used DNS redirection to monitor botnets [6]. In contrast, the detection techniques described in this paper are more discreet because they do not require direct communication with any component of the botnet.

Jung *et al.* found that 80% of spam sources in their analysis were listed in at least one of seven popular blacklists [11], which correlates well with our independent previous study [17]. To the best of our knowledge, this paper presents the first study that uses direct analysis of DNSBL logs to infer other types of network behavior.

## 7. Conclusion

This paper has developed techniques and heuristics for detecting DNSBL reconnaissance activity, whereby botmasters perform lookups against the DNSBL to deter-

mine whether their spamming bots have been blacklisted. We first developed heuristics for counter-intelligence based on several possible ways we figured reconnaissance was being performed. We then studied the prevalence of each of these reconnaissance techniques. Much to our surprise, we found that bots were in fact performing reconnaissance on IP addresses for bots in other botnets. Based on this finding, we have outlined possibilities for new botnet detection techniques using a traversal of the DNSBL query graph, and we have suggested techniques that DNSBL operators might use to more effectively stem the spam originating from botnets. We are investigating the effectiveness of these detection and mitigation techniques as part of our ongoing work.

## Acknowledgments

We thank Randy Bush, Wenke Lee, and Merrick Furst for feedback on some of the ideas in this paper. This work is supported in part by NSF grant CCR-0133629 and Office of Naval Research grant N000140410735. The contents of this work are solely the responsibility of the authors and do not necessarily represent the official views of NSF or the U.S. Navy. Data used in this paper was obtained as part of an information disclosure granted by the Georgia Tech Research Corporation, ref. Record of Invention GTRC ID 3828. Please consult the authors regarding its citation or use.

## Notes

[1]DNSBL queries issued by mail servers are often performed by directly querying the DNSBL, rather than relying on a local resolver. For example, SpamAssassin [20] implements its own recursive DNS resolver. Hosts performing reconnaissance are also unlikely to query DNSBLs using local resolvers. Thus, in both cases, the querying IP address observed at the DNSBL correctly reflects the end-host performing the query.

[2] This heuristic assumes that networks generally use the same host for both inbound and outbound mail servers. Although this configuration is common, some large networks separate the hosts responsible for inbound and outbound mail servers. In this case, queries from the inbound mail server might be misinterpreted as a reconnaissance attempt.

[3]When $d_{n,in}$ is zero (which is commonly the case), we can simply consider $\lambda_n$ to be a very large number.

## REFERENCES

[1] Bobax trojan analysis. `http://www.lurhq.com/bobax.html`, March 2005.

[2] Symantec Security Alert–W32.Bobax.D worm. `http://www.sarc.com/avcenter/venc/data/w32.bobax.d.html`.

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

[4] CNN Technology News. Expert: Botnets No. 1 emerging Internet threat. `http://www.cnn.com/2006/TECH/internet/01/31/furst/`, Jan. 2006.

[5] E. Cooke, F. Jahanian, and D. McPherson. The Zombie Roundup: Understanding, Detecting and Disrupting Botnets. In *Usenix Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, June 2005.

[6] 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.

[7] S. Dietrich, N. Long, and D. Dittrich. Analyzing distributed denial of service attack tools: The shaft case. In *Proceedings of the LISA 2000 System Administration Conference*, December 2000.

[8] 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.

[9] L. H. Gomes, C. Cazita, J. Almeida, V. Almeida, and W. Meira. Characterizing a Spam Traffic. In *Proc. ACM SIGCOMM Internet Measurement Conference*, Taormina, Sicily, Italy, Oct. 2004.

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

[11] J. Jung and E. Sit. An Empirical Study of Spam Traffic and the Use of DNS Black Lists. In *Proc. ACM SIGCOMM Internet Measurement Conference*, pages 370–375, Taormina, Sicily, Italy, Oct. 2004.

[12] S. Kandula, D. Katabi, M. Jacob, and A. Berger. Botz-4-Sale: Surviving Organized DDoS Attacks That Mimic Flash Crowds. In *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI)*, Boston, MA, May 2005.

[13] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen. Real-time and forensic network data analysis using animated and coordinated visualization. In *Proceedings of the 6th IEEE Information Assurance Workshop*, 2005.

[14] B. Krebs. Bringing botnets outof the shadows. `http://www.washingtonpost.com/wp-dyn/content/article/2006/03/21/AR2006032100279.html`, 2006.

[15] D. Moore, G. M. Voelker, and S. Savage. Inferring internet denial-of-service activity. In *Proceedings of the 2001 USENIX Security Symposium*, 2001.

[16] S. Racine. Analysis of internet relay chat usage by ddos zombies. `ftp://www.tik.ee.ethz.ch/pub/students/2003-2004-Wi/MA-2004-01.pdf`, 2004.

[17] A. Ramachandran and N. Feamster. Understanding the Network-Level Behavior of Spammers. In *Proc. ACM SIGCOMM*, Pisa, Italy, Sept. 2006.

[18] P. Ramneek. Bots & Botnets: An Overview. `http://www.giac.com/practical/GSEC/Ramneek_Puri_GSEC.pdf`, 2003.

[19] S. Schechter and M. Smith. Access for sale. In *2003 ACM Workshop on Rapid Malcode (WORM'03)*. ACM SIGSAC, October 2003.

[20] SpamAssassin, 2005. `http://www.spamassassin.org/`.

[21] SwatIt. Bots, drones, zombies, worms and other things that go bump in the night. `http://swatit.org/bots/`, 2004.

[22] Virus Bulletin 2005 Paper on 'Bots and Botnets'. `http://arachnid.homeip.net/papers/VB2005-Bots_and_Botnets-1.0.2.pdf`.

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