8-2011

Challenges in Experimenting with Botnet Detection Systems

Adam J. Aviv
University of Pennsylvania

Andreas Haeberlen
University of Pennsylvania, ahae@cis.upenn.edu

Challenges in Experimenting with Botnet Detection Systems

Abstract
In this paper, we examine the challenges faced when evaluating botnet detection systems. Many of these challenges stem from difficulties in obtaining and sharing diverse sets of real network traces, as well as determining a botnet ground truth in such traces. On the one hand, there are good reasons why network traces should not be shared freely, such as privacy concerns, but on the other hand, the resulting data scarcity complicates quantitative comparisons to other work and conducting independently repeatable experiments. These challenges are similar to those faced by researchers studying large-scale distributed systems only a few years ago, and researchers were able to overcome many of the challenges by collaborating to create a global testbed, namely PlanetLab. We speculate that a similar system for botnet detection research could help overcome the challenges in this domain, and we briefly discuss the associated research directions.

Disciplines
Computer Sciences

Comments
Aviv, A. & Haeberlen, A., Challenges in Experimenting with Botnet Detection Systems, *4th USENIX Workshop on Cyber Security Experimentation and Test (CSET'11)*, Aug. 2011

Challenges in Experimenting with Botnet Detection Systems

Adam J. Aviv Andreas Haeberlen
University of Pennsylvania

Abstract

In this paper, we examine the challenges faced when evaluating botnet detection systems. Many of these challenges stem from difficulties in obtaining and sharing diverse sets of real network traces, as well as determining a botnet ground truth in such traces. On the one hand, there are good reasons why network traces should not be shared freely, such as privacy concerns, but on the other hand, the resulting data scarcity complicates quantitative comparisons to other work and conducting independently repeatable experiments.

These challenges are similar to those faced by researchers studying large-scale distributed systems only a few years ago, and researchers were able to overcome many of the challenges by collaborating to create a global testbed, namely PlanetLab. We speculate that a similar system for botnet detection research could help overcome the challenges in this domain, and we briefly discuss the associated research directions.

1 Introduction

Botnets continue to be a major threat to the well-being of the Internet, and there is every reason to believe that they will continue to be a threat. Therefore, the arms race between botnet controllers and researchers designing detection systems will also continue, at least for the foreseeable future. As is often the case in computer security, it is not likely that this battle will ever be won by the "white-hat" side. Botnet controllers are continually developing more resilient and more covert ways to control their bots, while researchers must continue to play catch up by improving their techniques, by repairing compromises, and by mitigating or taking down botnet control structures [12, 40]. All the while, the botnet problem persists, and new botnets are emerging constantly.

However, it seems that the botnet battle is being fought on a particularly uneven playing field. Most botnets are global phenomena: botnet controllers have, at the touch of a button, ready access to vulnerable hosts in thousands of administrative domains all across the Internet, and they can easily send data back and forth between bots in different domains. Given this global nature of botnets, it would be natural to expect that the most successful botnet detectors would also be global, in the sense that they would combine data from many different domains. However, building and experimenting with such a detector is currently not realistic, for at least two reasons.

First, most administrative domains consider any detailed information about their networks to be a business secret, and researchers (unlike botnet controllers) can only obtain such information through explicit collaboration agreements, which need to be manually negotiated with every single domain. Second, network traces – the most detailed and potentially most useful type of information – can contain sensitive information and are therefore treated like information plutonium: something to be mined only when absolutely necessary, to be carefully controlled, and never to be shared with outsiders or even others within the same organization. There are good reasons for this cautious treatment of network traces: recent scandals [29, 2] have demonstrated that even carefully anonymized data can accidentally leak private information. Nevertheless, it creates an asymmetry between botnet controllers and researchers: the former have access to plenty of information, while the latter can consider themselves lucky if they can, after hours of careful negotiation, obtain a small trace from their own organization.

Not only does this dearth of data hamper the effectiveness of botnet detection, it is also a major obstacle to botnet research. Since real traces are difficult to obtain, researchers must be content with evaluating their systems on a small set of traces, which is risky because high heterogeneity on the Internet [32, 9] is not well represented by limited data sets. Alternatives, such as synthetic traces [42, 44] or botnet emulators [22], are available but have their own limitations with respect to re-

alism and potential biases. Since traces are not easily shared with other researchers, important scientific principles, such as a robust comparisons to other work or independent reproducibility, are difficult to follow in practice.

In this paper, we take a step back from discussing the design of specific botnet detection systems, and we examine the challenges faced by the field as a whole in evaluating the performance of such systems. We conclude that these challenges mostly stem from difficulties in obtaining and sharing the basic 'raw material' of botnet detection: real traces of network traffic, and ground truths about infections. This leaves researchers with an inherent disadvantage in the botnet arms race and, looking forward, it has alarming implications for network security: botnets could exploit these data-sharing difficulties to become harder to detect! For example, a future botnet could craft its command-and-control channel to cross as many administrative domains as possible [18]. An individual domain would only be able to see many small, disconnected segments, each of which would look innocuous by themselves. Without data sharing across domains, such a botnet would be difficult to detect.

We also observe that many of the challenges faced by botnet detection researchers are similar to those faced by researchers studying large-scale distributed systems just a few years ago. Proposed distributed systems were meant to be deployed globally on varied machines and networks, but evaluation was notoriously difficult because individual researchers rarely had access to sufficiently large and diverse sets of wide-area linked machines. Researchers were forced to use small-scale deployments and simulations instead. This changed with the advent of PlanetLab [30], a global research testbed run in collaboration with research groups around the globe.

Hypothetically, a global, collaborative testbed for botnet research – akin to PlanetLab – could be used to overcome many of the challenges we point out here. Building such a testbed would not be trivial; indeed, several research problems would have to be addressed first (*e.g.*, to reliably protect sensitive information), and there are significant logistical and organizational challenges that would have to be overcome as well. Nevertheless, there is evidence that both the research problems [26, 35, 48] and the logistical problems [30] are tractable.

2 Challenges

2.1 Theory: An ideal botnet experiment

To illustrate the challenges botnet research is facing today, we first describe an idealized scenario. Suppose Alice is a botnet researcher who has developed a new botnet detector. What are the questions we would like Alice to answer in her evaluation? These would probably include:

- **Does the detector work in a realistic setting?** Ideally, we would like Alice to deploy her detector in the environment for which it was designed, *e.g.*, a federation of several ISPs.
- **How well does the detector work?** We would like Alice to measure both recall and precision, *i.e.*, how many botnets are reported by her detector, and how many of its reports correspond to actual botnets.
- **Can the detector find state-of-the-art botnets?** Since botnets continue to evolve, we would like Alice to test her detector with, and report results for, very recent malware and very recent traces.
- **How does the detector compare to prior work?** Ideally, we would like Alice to compare her detector's performance to that of the best existing detectors.
- **Can we independently repeat the experiment?** Ideally, Alice would publish her detector's source code and/or a precise description of the algorithm, as well as all the traces from her experiments.

Note that none of these expectations are unusual, and similar questions are being answered in other areas of computer science and in related disciplines.

2.2 Practice: Botnets in the real world

In practice, the ideal botnet experiments described above are not realistic because of several challenges facing botnet research (and, more generally, research on network-based intrusion detection, *e.g.*, worm/virus detection [3, 10] or stepping-stone detection [50]):

- **Multiple administrative domains:** The Internet is controlled by many different organizations, which have different goals, interests and policies, and which tend to be guarded about data sharing.
- **Heterogeneity:** Different networks can have widely different characteristics; for example, academic and corporate networks differ considerably [32]. It is difficult to capture the diversity of the Internet with a small number of network traces.
- **Lack of ground truth:** Given a host within a network trace, it is difficult to establish whether or not it is part of a botnet. This is particularly true for hosts in other administrative domains where researchers cannot directly investigate.
- **Privacy concerns:** Network traces contain sensitive information about the actions and communications of the users of the network; thus, it is difficult to persuade network operators to collect them, let alone share them with a third party.

	Academic traces only	At least one other trace
Overlay methodology	[13] [49] [15] [36] [46] [47] [41] [23] [6] [7]	[28] [25] [24] [14]
Other methodology	[36] [11] [5]	[20] [14] [45]

Table 1: Methodologies used in the botnet literature. Note that some papers appear in two rows if multiple experiments were performed.

These challenges substantially complicate botnet experiments and evaluation. Due to the thousands of administrative domains and the high degree of network heterogeneity, it is would be very difficult for Alice to set up a realistic experiment, particularly if her botnet detector is designed to operate at large commercial ISPs and/or requires collaboration between multiple ISPs. A lack of ground truth also limits Alice's ability to estimate the precision and recall of her detector. Given any trace, it is hard to be sure how many botnets are present in it, if any. Finally, the privacy concerns result in a general scarcity of traces, which discourages or prevents sharing in many cases. A larger side effect of the privacy issues and lack of sharing is that it complicates scientific practices of quantitative comparisons and independent repeatability.

As a consequence, researchers are forced to approximate the "ideal" experiment from Section 2.1, *e.g.*, by using synthetic traces, or by extrapolating from a small number of sample traces. However, this experimental methodology can carry considerable risks, which we discuss in the next section.

3 Obtaining test traces

3.1 Best practices

Today, most experiments with botnet detection systems rely on synthetic traces. Although methods exist for generating synthetic network traffic [42, 44], these techniques typically focus on reproducing high-level traffic characteristics and do not accurately capture details – such as packet payloads – that are crucial for many detectors. Instead, researchers typically synthesize traces by mixing traffic collected through measurements, *e.g.*, one trace with known botnet traffic and another with benign background traffic. In this paper, we refer to this approach as the *overlay methodology*. This methodology is an attractive technique to researchers because it provides a sense of ground truth, and thus a way to estimate precision and recall. If the traces are benign, *i.e.*, contain no botnet traffic, any report on a background host is a false positive, and any *missing* report is a false negative.

The overlay methodology requires two sets of traces: botnet traffic traces and benign background traffic traces. The botnet traces are usually obtained from a honeypot [1] or by gathering botnet binaries from repositories [8, 43, 37] and then executing them in a controlled setting. The background traces are usually collected from the network at the researcher's own institution, typically a campus network. The botnet traces are then combined with the background traces in one of two ways; either by mapping the botnet traffic to hosts that are not present in the background trace, or, more commonly, by adding the botnet traffic to existing hosts. Typically, each synthetic trace contains traffic from a single botnet. Table 1 shows a survey of papers from the field, as well as the methodology used in each. The publications in the table all proposed detection systems or methods (*e.g.*, automated botnet signature generators [45, 36]) and performed an evaluation on the detector in synthetic or live settings.

Despite its advantages, the overlay methodology also carries risks. The synthetic traces do not necessarily represent what happens in real networks; for example, concurrent infections of different kinds of malware on a single host [40]. It is also difficult to ascertain that the background trace is truly benign and does not contain any botnet traffic. Finally, the focus on academic networks can potentially bias the results. We discuss these risks in more detail in the rest of this section.

3.2 Challenge: Realism

To reliably estimate the performance of a botnet detector using synthetic traces, it is essential that the traces realistically reflect the environment in which the detector is likely to be deployed. If the traces are unrealistic, the detection task can be too hard or too easy, resulting in an under- or overestimation of the detector's performance.

Controlled environments: Honeypot traces do provide a good estimate of the traffic generated by bots, but they are gathered in an artificial environment, *e.g.*, on machines purposely designed to be infected, or in a controlled lab environment. As a consequence, it is difficult to be sure that the botnet behaves in the same way as it would in the wild. Some modern botnets are adaptive; for example, certain spamming bots check their blacklist status regularly [34] because their value to the botnet is proportional to their ability to send spam [40]. Hence, a bot on a blacklist may not directly engage in spamming, but may instead assume other roles in the botnet, which can be easier or harder to detect. Experimenters may create additional artifacts by removing the harmful parts of the bots before generating the trace: this prevents harm to others but could reduces the realism of the resulting trace.

Mixing artifacts: Since botnet trace and background trace are typically collected in different environments,

they do not necessarily fit together. For example, if DHCP frequently reassigns IP addresses in the background trace, a botnet trace from a honeypot with static IP addresses does not match well. This discrepancy could affect detectors that rely on clustering hosts based on network features [13, 46] or on event sequences and timing [14]. Challenges related to DHCP and other effects have been pointed out in the botnet literature before, particularly in estimating the size of botnets [19, 33]. The only reliable way to avoid such artifacts would be to deploy the botnet directly on the network from which the background trace is collected, but this would obviously be unethical.

Multimorbidity: In real networks, it is not uncommon for hosts to be infected by more than one botnet or malware [40]. This could present an additional challenge for botnet detectors; for example, the communication graphs, which are used by some detectors [7, 28] to identify bots based on their control channel, could overlap. Additionally, networks experience multiple simultaneous infections across hosts, and researchers do not typically embed multiple botnets into background traces in experiments. Among the papers we examined that use the overlay methodology (center row in Table 1), none described performing experiments using synthetic traces with multiple botnet infections.

To some extent, these challenges are inherent in the overlay methodology. They could be avoided by using collected traces with botnet infections directly, but, as discussed earlier, this is difficult, *e.g.*, due to privacy concerns and a lack of ground truth.

3.3 Challenge: Representativeness

The performance of most botnet detectors is likely to depend on the characteristics of the network in which they are deployed, as well as the characteristics of the prevalent botnets. Due to the high degree of heterogeneity in both Internet networks and botnet behavior, it would be desirable to experiment with a large variety of scenarios.

Focus on academic networks: As shown in Table 1, the overwhelming majority of the traces that are currently used for botnet experiments come from academic networks. This is not surprising, given that many of the papers come from academia and, due to privacy concerns, it is extremely difficult for experimenters to obtain traces from networks other than their local one (or, indeed, even from there). However, it is known that academic networks differ from, *e.g.*, corporate networks in terms of their performance characteristics [32]. Thus, if a detector is evaluated only using academic traces, it is difficult to estimate how it would perform in a corporate network – the performance could be better, or worse.

Scale: Some botnet detectors, *e.g.*, [28], are designed to run in large-scale deployments across administrative domains to take advantage of information from multiple vantage points. For detectors of this type, it is virtually impossible to collect a representative set of traces because this would require coordinating with, and probably entering into contracts with, tens or hundreds of network operators across the Internet. Some traces from the Internet backbone are available, *e.g.*, from CAIDA [4], but due to the limited number of usable public traces, this probably underestimates the benefits that could be gained from a real large-scale deployment.

3.4 Challenge: Generality

Different botnets have different characteristics. Therefore, it is desirable to experiment with a wide variety of botnet traces and background traces.

Botnet overfitting: Among the papers we examined (see Table 1), the average number of botnet traces used was 5.25, and the median was 4, with 7 papers using traces from only one or two botnets (while [36] used as many as 19 botnet binaries). With a limited number of traces, it is hard to estimate how general the botnet detectors are. There is considerable value in building a detector that is effective for a specific class of botnets, or even just a single botnet; however, there is a certain risk that a very specific detector could be circumvented by the botnet designers with relatively small changes to the botnet code (as suggested in [39]).

Artifact overfitting: As described in Section 3.2, mixing artifacts may result from the overlay methodologies, and these artifacts pose another risk factor for certain detectors. If there are subtle differences between the botnet trace and the background trace, *e.g.*, the presence of SNMP packets or encryption in one but not the other, there is a danger that the detector may focus on these artifacts rather than on the botnet traffic itself. This is particularly risky for detectors that rely on machine learning [23], or detectors, such as [46], that create a background model and look for anomalies with respect to that model. As a consequence, detectors may need to rely on manual verification by a human expert, *e.g.*, such as detectors that generate automated rules, like [45].

Both risks could be avoided by using a greater variety of traces for evaluation. However, as we have pointed out earlier, this is difficult in practice due to privacy concerns.

3.5 Risk: False positives & negatives

In the ideal case, the background trace used to synthesize test traces would be completely free of botnet traffic. However, this is difficult to achieve in practice. In fact, traces collected at tier-1 or tier-2 ISPs, as in [28], will

almost inevitably contain some botnet traffic. The same is true for traces collected from campus networks, which will likely also contain some infections. In practice, this traffic is difficult to remove entirely because some botnets use obfuscation or encryption [16]. This creates two potential problems.

Lack of verification: If the botnet detector produces a report for a host in the background trace, it is difficult to determine whether the report is a false positive. The experimenter would need access to the corresponding hosts to check for infections. This is particularly problematic when the traces are from another administrative domain (no access to hosts and/or anonymization) or is used long after collection (environment changes).

To illustrate this challenge, we relate an example from TĀMD [46], a botnet detection system that clusters hosts based on common destinations, payloads, and platforms. In particular, the common-destination alerts are based on identifying a cluster of hosts that send traffic to "suspicious" subnets, *i.e.*, those that are communicated with rarely or not at all. In one experiment, a large IRC botnet trace was overlaid onto background traffic, but the detection rate was less than expected because some of the hosts chosen to be bots were detected visiting a *different* suspicious site in the background traffic, causing them to form a separate cluster from the other "bots". Further analysis of the suspicious cluster was infeasible because the background traffic had been purposely anonymized before processing, and payloads were not fully provided.

Experimenters can avoid this issue by running a live deployment, as was done in the case of [14, 36, 11], which has the nice side-effect of potentially reporting live infections; however, this requires a major development effort to produce a deployment-grade implementation, especially if the monitoring framework itself does not exist yet and has to be developed as well.

False negatives: Despite best efforts, there is always a possibility that the background trace contains some botnet traffic that has not been identified. If the detector does not report those corresponding hosts, it should be counted as a false negative, but this is, of course, impossible to quantify. One possible remedy would be to re-evaluate the trace once better detectors have become available and/or more information is known about botnets.

4 Sharing traces

4.1 Challenge: Repeatability

It is good scientific practice to ensure that experiments can be repeated independently by any interested third party. However, this is inherently difficult to achieve for botnet experiments because they almost inevitably involve benign background traffic, which in turn can contain private information and thus cannot easily be shared. Out of the 14 papers we examined that use background traces (see Table 1), only two rely exclusively on public traces (from CRAWDAD [21]); one paper [28] uses public CAIDA [4] data in addition to a non-public trace. Two other papers [24, 25] do not explicitly specify where or when the traces were collected.

To some extent, this problem could be mitigated by relying more on public traces, *e.g.*, from [4, 21, 31], but due to privacy concerns, the rarity of public traces render it hardly feasible to perform a comprehensive evaluation based entirely on them. Authors of published papers often do make traces and binaries available upon request – indeed, the authors of this paper have received some traces in this way – but this is typically based on existing trust relationships. It would be better to make the traces available to a wider audience, *e.g.*, by uploading them to public repositories or by publishing them on a project web page. But, it is not always possible to share all information collected while performing botnet research, *e.g.*, due to legal restrictions. For example, the data collected while performing a botnet takeover in [12] includes legally sensitive information, such as credit card numbers and bank account passwords, and cannot be released.

4.2 Challenge: Comparability

A related challenge is that it is very difficult to estimate how much new detectors improve overall botnet detection. A quantitative comparison to existing work would be easiest if there were a standard methodology, or even a widely accepted benchmark for evaluating botnet detectors. However, no such methodology or benchmark exists today, presumably due to the pervasive privacy concerns and the resulting difficulties for data sharing. Indeed, out of the 18 papers we examined in Table 1, only 4 contain any quantitative comparisons to prior work, and among these, one compared their detector to prior work that was done by some of the same authors. Among the others, 10 contained only qualitative comparisons, and 5 contained no comparison at all. (To be fair, a quantitative comparison was not applicable in all settings, for example in [20].)

It is clear that a quantitative comparison of botnet detectors is a difficult undertaking. There are many reasons for this, including: (1) the fact that different detectors are sometimes designed for different deployment scenarios, *e.g.*, AS-local versus Internet-wide; (2) the difficulties in reproducing prior experiments (Section 4.1), which complicate the verification that the other detectors are set up and configured properly; (3) the ever-continuing evolution of botnets, which puts yesterday's detectors at an inherent disadvantage when tasked with detecting today's

botnets; and, finally, (4) the fact that different detectors focus on different types of botnets. Despite these challenges, a quantitative comparison would be useful for understanding gains in new detectors.

5 What can be done?

Given the arguments presented so far, one possible conclusion is that these challenges are inherent to botnet research, and this is the best one can do. We do not agree with this conclusion. We believe that this is merely the best one can do *individually*. In the following section we sketch an approach that could potentially overcome these challenges through collaboration.

5.1 The PlanetLab analogy

The current situation in botnet research has some similarities to the situation in research on large-scale distributed systems a few years ago. Large-scale distributed systems and botnet detectors share several of the challenges we have described in Section 2.2; in particular, most of them are designed to run in a heterogeneous environment with multiple administrative domains. Another parallel is that realistic experiments would have required resources – hundreds of nodes connected by a wide-area network – that no individual research team had available. As a consequence, evaluations were mostly performed in simulation, or on small-scale deployments in the lab. This was known to be suboptimal; for example, wide-area links have complex behaviors, such as routing changes and transient anomalies, that can affect the performance of the system considerably [9] but are difficult to reproduce in the lab.

The situation was transformed with the advent of PlanetLab [30]. By pooling their resources, researchers around the world created a shared planetary-scale testbed, in which organizations that contributed nodes could in return access the other nodes for their own experiments. While PlanetLab may not be perfect [38], it was a giant step forward in terms of scale and realism, and PlanetLab-based evaluations, despite their imperfections, have become a standard in the community.

5.2 A PlanetLab for botnet research?

It seems that a similar approach could potentially be used to overcome the challenges in botnet research. If several different organizations, *e.g.*, universities or research labs, were to allow each other to deploy novel detectors on their own networks, this would enable more comprehensive experiments and, due to the larger view, potentially better detectors.

The big challenge, of course, is privacy. No organization would allow third-party software access to traces from its network if there was any possibility that private data could leak out. This is a formidable challenge, indeed; nevertheless, a solution does not seem infeasible. To illustrate the point, we consider a strawman solution at one extreme in the design space.

In this strawman design, each participating organization would deploy a machine that would receive NetFlow traces from the organization's border routers but would be prevented (through MAC filters or physically cut wires) from sending network traffic of its own. Other organizations could send software packages to be deployed in a VM on this machine, but they could not receive any communication from it unless it was inspected, and declassified, by a network administrator. Manual declassification obviously does not scale, but communication could be infrequent, *e.g.*, a list of results at the end of an one-month deployment. Researchers could perform testing in their own local domains, where they could have more direct access.

As an incentive for deployment, the system could generate regular reports for the local domain's network administrators, who would thus gain access to the bleeding edge of botnet detection technology. In return, they could provide the experimenters with some "ground truth", *e.g.*, by rating how helpful each report was to them.

5.3 "This will never work!"

Recall that the above design is merely a strawman, whose purpose is to show that a solution is not *impossible*. Distributed honeypot efforts, such as [17], and public repositories, such as PREDICT [31], suggest that researchers are willing and ready to collaborate on this issue. Of course, an actual implementation would require a major design effort, similar to Peterson's PlanetLab Central, but it could also build upon existing techniques such as Bunker [26], SC2D [27], or collaborative security [35].

6 Conclusion

In this paper, we have outlined several challenges that researchers face when evaluating new botnet detection systems, including multiple administrative domains, Internet heterogeneity, lack of ground truth, and privacy concerns. Ideally, botnet detectors would be evaluated using diverse, real-world network traffic that is representative of the conditions in which the detectors would be deployed, *e.g.*, across a federated system. But, there are serious privacy concerns about releasing the requisite traffic to researchers and others. Even when researchers do receive network traces, a ground truth is hard to determine,

so reporting performance statistics for the detector is very difficult.

As a result, researchers have relied on synthetic traces generated using an overlay methodology, where botnet traffic is mixed in with benign background traffic. This methodology can lead to a number of pitfalls, which could cause researchers to over- or underestimate the performance of their detector. A larger consequence of this methodology is that the background traces used in experiments are not easily shared due to privacy concerns, complicating basic scientific practices, such as performance comparisons and experimental reproducibility.

We observe that many of these issues are similar to those faced by researchers experimenting with large-scale distributed systems before the advent of PlanetLab. We propose a strawman system similar to PlanetLab that, through collaboration, could ameliorate many of the experimental challenges in botnet detection research. There are several research problems that would need to be addressed before such a system could become a reality, but there is evidence that these problems are solvable.

Acknowledgments

We thank Sean Peisert for shepherding this paper, and Jonathan Smith, Benjamin Pierce, Angelos Keromytis, Micah Sherr, and the anonymous reviewers for their helpful comments. This research was supported in part by ONR Grant N00014-09-1-0770 and by US National Science Foundation grants CNS-1054229 and CNS-1065060.

References

[1] P. Bächer, T. Holz, M. Kötter, and G. Wicherski. Know Your Enemy: Tracking Botnets. Technical report, The Honeynet Project, Aug. 2008.

[2] M. Barbaro and T. Zeller. A face is exposed for AOL searcher no. 4417749. *The New York Times*, Aug. 2006. http://www.nytimes.com/2006/08/09/technology/09aol.html.

[3] V. Berk, G. Bakos, and R. Morris. Designing a framework for active worm detection on global networks. In *1st IEEE International Workshop on Information Assurance (IWIAS)*, Mar. 2003.

[4] CAIDA. http://www.caida.org/.

[5] H. Choi, H. Lee, and H. Kim. BotGAD: detecting botnets by capturing group activities in network traffic. In *4th International ICST Conference on Communication System Software and Middleware (COMSWARE)*, June 2009.

[6] H. Choi, H. Lee, H. Lee, and H. Kim. Botnet Detection by Monitoring Group Activities in DNS Traffic. In *7th IEEE International Conference on Computer and Information Technology (CIT)*, Oct. 2007.

[7] B. Coskun and S. Dietrich. Friends of An Enemy: Identifying Local Members of Peer-to-Peer Botnets Using Mutual Contacts. In *26th Annual Computer Security Applications Conferenec (ACSAC)*, Dec. 2010.

[8] Cyber-TA. http://cyber-ta.org/.

[9] M. Dischinger, A. Haeberlen, I. Beschastnikh, K. P. Gummadi, and S. Saroiu. SatelliteLab: Adding heterogeneity to planetary-scale network testbeds. In *ACM SIGCOMM Conference*, Aug 2008.

[10] D. R. Ellis, J. G. Aiken, K. S. Attwood, and S. D. Tenaglia. A behavioral approach to worm detection. In *ACM Workshop on Rapid Malcode (WORM)*, Oct. 2004.

[11] J. Goebel and T. Holz. Rishi: Identify bot contaminated host by IRC nickname evaluation. In *1st USENIX Workshop on Hot Topics in Understanding Botnets (HotBots)*, Apr. 2007.

[12] B. S. Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your botnet is my botnet: Analysis of a botnet takeover. In *16th ACM conference on Computer and Communications Security (CCS)*, Nov. 2009.

[13] G. Gu, R. Perdisci, J. Zhang, and W. Lee. BotMiner: Clustering analysis of network traffic for protocol- and structure-independent botnet detection. In *17th USENIX Security Symposium*, July 2008.

[14] 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*, Aug. 2007.

[15] G. Gu, J. Zhang, and W. Lee. BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic. In *16th Network and Distributed System Security Symposium (NDSS)*, Feb. 2008.

[16] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F. Freiling. Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm. In *1st USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, Apr. 2008.

[17] Honey@home. http://www.honeyathome.org/.

[18] M. Jelasity and V. Bilicki. Towards automated detection of peer-to-peer botnets: on the limits of local approaches. In *2nd USENIX Conference on Large-scale Exploits and Emergent Threats (LEET)*, Apr. 2009.

[19] C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, and S. Savage. The Heisenbot Uncertainty Problem: Challenges in Separating Bots from Chaff. In *1st USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, Apr. 2008.

[20] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. In *1st USENIX Workshop on Hot Topics in Understanding Botnets (HotBots)*, Apr. 2007.

[21] D. Kotz and T. Henderson. CRAWDAD: A community resource for archiving wireless data at Dartmouth. http:

//crawdad.cs.dartmouth.edu/.

[22] C. P. Lee. *Framework for Botnet Emulation and Analysis*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, May 2009.

[23] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer. Using Machine Learning Techniques to Identify Botnet Traffic. In *31st Annual IEEE Conference on Local Computer Networks (LCN)*, Nov. 2006.

[24] W. Lu, G. Rammidi, and A. A. Ghorbani. Clustering botnet communication traffic based on n-gram feature selection. *Computer Communications*, 34(3):502–514, Mar. 2011.

[25] W. Lu, M. Tavallaee, and A. A. Ghorbani. Automatic discovery of botnet communities on large-scale communication networks. In *4th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, Mar. 2009.

[26] A. G. Miklas, S. Saroiu, A. Wolman, and A. D. Brown. Bunker: a privacy-oriented platform for network tracing. In *6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Apr. 2009.

[27] J. C. Mogul and M. Arlitt. SC2D: An alternative to trace anonymization. In *SIGCOMM Workshop on Mining Network Data (MineNet)*, Sept. 2006.

[28] S. Nagaraja, P. Mittal, C. Y. Hong, M. Caesar, and N. Borisov. BotGrep: Finding P2P Bots with Structured Graph Analysis. In *USENIX Security Symposium*, Aug. 2010.

[29] A. Narayanan and V. Shmatikov. Robust deanonymization of large sparse datasets. In *29th IEEE Symposium on Security and Privacy*, May 2008.

[30] L. Peterson, A. Bavier, M. E. Fiuczynski, and S. Muir. Experiences building PlanetLab. In *7th Symposium on Operating Systems Design and Implementation (OSDI)*, Nov. 2006.

[31] Protected Repository for the Defense of Infrastructure Against Cyber Threats. http://www.predict.org.

[32] H. Pucha, Y. C. Hu, and Z. M. Mao. On the impact of research network based testbeds on wide-area experiments. In *6th ACM SIGCOMM Conference on Internet Measurement (IMC)*, Oct. 2006.

[33] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis. My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging. In *1st USENIX Workshop on Hot Topics in Understanding Botnets (HotBots)*, Apr. 2007.

[34] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using DNSBL counter-intelligence. In *2nd USENIX Conference on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)*, July 2006.

[35] J. Reed, A. J. Aviv, D. Wagner, A. Haeberlen, B. C. Pierce, and J. M. Smith. Differential privacy for collaborative security. In *3rd European Workshop on System Security (EuroSec)*, Apr. 2010.

[36] K. Rieck, G. Schwenk, T. Limmer, T. Holz, and P. Laskov. Botzilla: Detecting the "phoning home" of malicious software. In *25th ACM Symposium on Applied Computing (SAC)*, Mar. 2010.

[37] Shadowserver. http://shadowserver.org/.

[38] N. Spring, L. Peterson, A. Bavier, and V. Pai. Using PlanetLab for network research: myths, realities, and best practices. *SIGOPS Operating Systems Review*, 40(1):17–24, Jan. 2006.

[39] E. Stinson and J. C. Mitchell. Towards systematic evaluation of the evadability of bot/botnet detection methods. In *2nd USENIX Workshop on Offensive Technologies (WOOT)*, July 2008.

[40] B. Stone-Gross, T. Holz, G. Stringhini, and G. Vigna. The underground economy of Spam: A botmaster's perspective of coordinating large-scale Spam campaigns. In *4th USENIX Workshop on Large-Scale Exploits and Emerging Threats (LEET)*, Mar. 2011.

[41] W. T. Strayer, R. Walsh, C. Livads, and D. Lapsley. Detecting Botnets with Tight Command and Control. In *31st IEEE Conference on Local Computer Networks (LCN)*, Nov. 2006.

[42] K. V. Vishwanath and A. Vahdat. Swing: Realistic and responsive network traffic generation. *IEEE/ACM Transactions on Networking*, 17(3):712–725, June 2009.

[43] VX Heavens. http://vx.netlux.org/.

[44] M. C. Weigle, P. Adurthi, F. Hernández-Campos, K. Jeffay, and F. D. Smith. Tmix: a tool for generating realistic tcp application workloads in ns-2. *SIGCOMM Computer Communication Review*, 36:65–76, July 2006.

[45] P. Wurzinger, L. Bilge, T. Holz, J. Goebel, C. Kruegel, and E. Kirda. Automatically Generating Models for Botnet Detection. In *14th European Symposium on Research in Computer Security (ESORICS)*. Sept. 2009.

[46] T.-F. Yen and M. K. Reiter. Traffic aggregation for malware detection. In *5th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, July 2008.

[47] T.-F. Yen and M. K. Reiter. Are Your Hosts Trading or Plotting? Telling P2P File-Sharing and Bots Apart. In *30th International Conference on Distributed Computing Systems (ICDCS)*, June 2010.

[48] N. Zeldovich, S. Boyd-Wickizer, and D. Mazières. Securing distributed systems with information flow control. In *5th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, Apr. 2008.

[49] U. Zhang, X. Luo, R. Perdisci, G. Gu, W. Lee, and N. Feamster. Boosting the scalability of botnet detection using adaptive traffic sampling. In *6th ACM Symposium on Information, Computer and Communcations Security (ASIACCS)*, Mar. 2011.

[50] Y. Zhang and V. Paxson. Detecting stepping stones. In *9th USENIX Security Symposium*, Aug. 2000.