Monitoring Stealthy Network Conversations with Sampled Traffic

Anirudh Ramachandran*, Srinivasan Seetharaman*,
Nick Feamster*, Anukool Lakhina†
* College of Computing, Georgia Tech
† Department of Computer Science, Boston University

Abstract

Network *conversations*—communication patterns between communities of network endpoints—can reveal important information about the nature of applications and about the participants in a particular application. Logging information about network flows can help network operators expose and monitor these conversations, but many large networks apply sampling to traffic monitoring (*e.g.*, "sampled NetFlow"), a technique that potentially obscures these conversations. This paper studies network-level conversations among botnets, a specific type of network community, as visible in sampled traffic flows. We first characterize the limitations of traffic sampling in exposing these conversations: we compare conversations visible in packet traces of known botnet activity to the same conversations in sampled traffic of two large transit ISPs, each of which uses a different flow sampling rate. We also use controlled experiments to examine the extent to which increasing flow sampling rates beyond the rates commonly used in today's networks can better expose these communities. The results of our study emphasize that, in many cases, monitoring interesting network conversations (*e.g.*, for network security purposes, monitoring peer-to-peer applications, etc.) will require significant advances in network flow monitoring algorithms.

1 Introduction

Communication patterns between Internet end-hosts—network *conversations*—can reveal important information about the nature of applications, such as the location and timing of application traffic. The ability to monitor these conversations can help network operators provision resources and improve performance, but, perhaps even more importantly, tracking these conversations can help network operators identify and monitor security incidents. For example, the characteristics of a conversation (*e.g.*, the volume of traffic for a particular application suddenly increases, or the distribution of this traffic across participants suddenly shifts) may indicate anomalous behavior, such as a denial of service attack or a routing failure. Indeed, even the fundamental *structure* of these conversations—who is communicating with whom at any given time—can reveal *communities* of end-hosts, or groups of hosts that may have common interests (*e.g.*, groups of Web clients) or purposes (*e.g.*, a botnet). Unfortunately, *conventional traffic sampling techniques were not designed to expose this type of traffic, which is often inherently (and intentionally) low-volume*.

This paper examines the effectiveness of using traffic sampling, a passive network-wide measurement technique, to expose network conversations and communities. We examine the extent to which existing passive traffic sampling techniques (*e.g.*, sampled NetFlow [27]) can provide fidelity in exposing the traffic patterns of a single community, and analyze the various invariants involved in reconstructing the complete view of its membership.

We study the conversations of a particular type of community—a *botnet*, which is a network of (typically compromised) machines under the control of a single (commonly malicious) entity. Observing these communication patterns can help network operators tease apart different types of application traffic and isolate communities or groups of end hosts. For instance, because botnet activity is dispersed across a very large number of hosts that are continually communicating with command-and-control; observing this activity in sampled traffic flows over a given time window will reveal structure in the botnet "communication graph" that can help us identify botnet membership. With the ability to passively detect botnet activity, network operators could filter this traffic or even quarantine "botted" hosts.

Given complete fidelity, a network operator's perspective affords the ability not only to observe the existence of any conversation that traverses their network but also to take action against these malicious hosts by deploying filters, notifying authorities, etc. Unfortunately, the data that networks collect is typically very coarse-grained: large transit networks sample only one in every hundredth or thousandth packet that traverses the network [21, 27]. Because sampled traffic flows offer only a coarse picture of network communication patterns, certain communications between bots (*i.e.*, both control traffic and attack traffic) will be lost. Specifically, low-volume flows that are characteristic of certain communities of interest (*e.g.*, botnet "command and control" traces) may escape detection.

To test how sampling can limit the ability to monitor conversation patterns, we start with "ground truth": we use two collected traces that are unequivocally known to contain botnet activity—those collected by hijacking botnet "rallying" behavior and those collected from a large spam sinkhole—to determine the communication patterns that *unsampled* traffic traces would expose. We observe these communication patterns in sampled traffic flow traces from the perspective of two large transit Internet Service Providers (ISPs) to better understand the fraction of each community that is visible in sampled traffic over any given collection time interval. To gain insight on how different sampling rates and techniques assist network operators in monitoring low-volume communication patterns, we perform an experiment that combines our ground truth datasets with appropriate cross-traffic to determine the effects of traffic sampling on the ability to monitor low-volume conversations. We observe from both the ISP flow record measurements and the controlled experiment that most commonly used techniques are unsuitable for detecting these stealthy conversations.

Using botnet detection as an example, this paper motivates the need for reconstructing communication patterns between groups of hosts (*i.e.*, network conversations) and explores the capabilities and limitations of existing traffic sampling techniques for this purpose. Although this paper focuses on identifying botnet-related conversations, our models are sufficiently general; indeed, our findings could likely be applied to *any* detection problems where identifying network conversations might be useful. For example, given the ability to monitor conversation structures, network operators construct graph signatures that are indicative of many types of application traffic (as suggested in previous work [22]), including peer-to-peer and Web traffic, network management applications, etc.

The rest of this paper is organized as follows. Section 2 provides background on traffic sampling and botnets. Section 3 presents the model we use to analyze communication patterns. Section 4 describe the setup of the experiments we use to investigate the effects of sampling. Section 5 describes our data collection techniques and gives an overview of our datasets. In Section 6, we study the effectiveness of traffic sampling for exposing *known* conversations, as observed in packet traces from two large ISPs. Section 7 examines the effectiveness of different sampling rates and techniques for exposing network conversations by means of a controlled experiment using the same ground truth datasets. Section 8 surveys related work in traffic sampling and traffic classification. We conclude in Section 9.

2 Background

This section presents a brief background on sampling-based traffic monitoring techniques and botnets.

2.1 Traffic Sampling

Sampling is an attractive method to scalably collect backbone traffic data. Modern routers have the functionality to sample traffic as it arrives, and only export summaries of the sampled traffic (*e.g.*, using Cisco's NetFlow [27], Juniper Traffic sampling [21], or InMon sFlow [20]). In particular, traffic sampling inspects every n-th packet using a configurable sampling technique, and continuously record statistics associated with the sampled packet's header in a local router cache until either a configured timeout value is reached or the cache is full, at which point the cache is flushed to a collector box. This sampling conserves resources needed to perform passive monitoring.

Claffy *et al.* [9] studied three classes of sampling techniques at a variety of granularities. Each of these sampling methods can be either packet-based or time-based, with the exception of one. We briefly describe each of these commonly used sampling techniques as follows [1]:

1. *Packet-based Deterministic (or systematic) sampling:* This involves deterministically selecting every N^{th} packet of the dataset and ensuring that a flow entry is created for it. However, the first sample, which actually determines the future selections, is randomly selected.
2. *Time-based Deterministic sampling:* This scheme selects one packet every N milliseconds, with the first packet selected randomly from the first time window.
3. *Packet-based Stratified random sampling:* Packets are divided into equal-length bins of N packets each, and one sample is randomly selected from each bin. This scheme offers better variance than simple random sampling [8].
4. *Time-based Stratified random sampling:* The time axis is divided into intervals of length T, and one sample is randomly selected from packets observed in each interval.
5. *Simple random sampling:* This scheme randomly selects a packet with a fixed probability of $1/N$. Random sampling is known to provide unbiased estimators for the population mean, total, and proportion [7].

In addition to these techniques, there have been other efforts to improve sampling, with due consideration to overhead and accuracy. Section 8 describes several improved schemes.

It is worth noting that time-based sampling does not work well with bursty traffic and is vulnerable to diurnal

[1] Some router vendors refer to *packet-based stratified random sampling* as *random sampling*, which is not the same as *simple random sampling*. Throughout the rest of the paper, we adopt the convention as described in the listing above.

patterns in the traffic. Moreover, results in [9] indicate ineffectiveness of timer-based sampling schemes in assessing packet inter-arrival times. Hence, we only analyze the effects of the three widely used packet-based sampling techniques.

2.2 Motivation: Botnets

A *botnet* is a network of typically compromised machines under the control of a single entity. Botnets have been used to perpetrate denial of service (DoS) attacks, for spamming, and for other nefarious activities such as click fraud. Botnets are often controlled by a single "command and control" host, which compromised "bots", or machines, contact for further instructions. This command-and-control is often a single centralized host; the control channel over which bots exchange messages with the command-and-control has historically been Internet Relay Chat (IRC), but various botnets, including the "Bobax" botnet, based on the `Bobax/W32` vulnerability, uses port 80 (HTTP) to communicate with the command-and-control. In this paper, we seek to determine the effectiveness of traffic sampling for exposing both the command-and-control conversations, as well as the conversations corresponding to the actual "attack" (*e.g.*, sending spam, infecting other hosts). We briefly describe the nature of each of these conversations and the challenges in monitoring each of these conversations with traffic sampling.

Command-and-control. A botnet's control conversations may often appear as a "star", with large numbers of bots exchanging control traffic with a single controlling entity. In this case, monitoring network conversations may be able to expose the botnet command-and-control. However, to evade detection, botnets are increasingly adopting more complex strategies to constantly alter the location of the command-and-control host. To achieve this, bots often contact their command-and-control using a DNS name, which can continually be bound to different IP addresses as the command-and-control changes locations. This movement defeats the currently popular techniques for exposing botnet communication today, namely honeypot tools [1, 26]. These migration strategies also pose complications for detecting command-and-control conversations in the presence of traffic sampling. However, given a substantially long collection window, traffic sampling should eventually expose communication between each bot and a single command-and-control host.

Indeed, the topological and temporal structure of the graph may expose characteristics (*e.g.*, a node exchanging periodic IRC traffic with a large number of hosts) that allow a network operator to identify members of the botnet. As the command-and-control moves, however, the structure of the network may not resemble a set of conversations with a single host, but rather a collection of seemingly unrelated conversations. Flow sampling threatens to exacerbate this effect because some conversations between a bot and its command-and-control may not be sampled, so the movement of a command-and-control may make it difficult to detect the movement of large groups of hosts at all.

There have also been conjectures of a gradual change in botnet rallying techniques: a move from the conventional "star" topology to a more advanced peer-to-peer (P2P) communication structure [16, 2]. P2P botnets allow the bot controller a higher degree of anonymity, as there is no single command-and-control (C&C). Ideally, the bot controller needs to communicate with only a few "trusted" bots, *i.e.*, ones which are unlikely to be detected by honeypots (perhaps because their respective Internet Service providers perform little or no botnet monitoring). One could imagine a P2P network in which a bot controller's messages to a botted host is routed through other bots (and responses routed back similarly). Thus, the inspection of traffic at an infected host will not reveal the location of the C&C—this is in stark contrast to how many present-day botnet monitoring systems function [26]. In these cases, it is crucial to monitor network conversations from a network-wide perspective in order to expose the botnet control structure. As such, these network-wide monitoring techniques will prove tremendously useful as botnet control becomes increasingly decentralized.

"Attack" traffic. Botnets generate various types of attack traffic, from spam to denial-of-service attacks to false clicks on Web advertisements ("click fraud"). Besides these commodity traffic, a botnet also generated a substantial amount of infection traffic to recruit newer bots to its network. This is vital for improving its scale of impact.

Our previous work presented a study of the network-level behavior of spammers and found that the majority of observed spamming bots from a Bobax botnet sent very low volumes of spam traffic to a single sinkhole domain [29]. This behavior suggests that network-wide monitoring techniques may be useful in identifying spamming bots, since a transit ISP has the capability of monitoring traffic between all domains whose network paths cross that ISP, and not just a single domain. Nevertheless, sampling may introduce challenges because spamming botnets are not likely to be sending a constant stream of traffic. Further, even a large transit ISP will only capture some fraction of the traffic from the botnet. In this paper, we use the "ground truth" data from a spam sinkhole (described in further detail in Section 5) to study the detriments of sampling on our ability to monitor this specific attack conversation.

Another important class of attack traffic constitutes the infection traffic. Most botnets exploit certain vulnerabilities in the host system to establish itself. After botting a host, the bot program contributes a portion of its resources for spreading to uninfected hosts. This *recruitment* process is specific to the particular vulnerability the

bot exploits. In Section 5, we presents results from our study of the Bobax botnet infection traffic and the effect of sampling in reconstructing the infection pattern.

3 Conversation Model

In this section, we present our model for *conversations*, or communications between host pairs. We first present a definition for a conversation and explain why detecting stealth conversations is a different problem than detecting traditional anomalies. We then present a few motivating examples, explaining how, given the ability to monitor and track conversations, network operators could detect better detect and mitigate new types of emerging threats.

3.1 Preliminaries

We define a *conversation* as the communication between a pair of hosts, represented by the combination of the source IP address, source port number, destination IP address and destination port number. Although a conversation, for all practical purposes, is the same as a flow, we use a different terminology to make a distinction for our motivation behind the network monitoring: We are more concerned with identifying the existence of a particular flow, than with the measurement of detailed flow statistics.

Depending on the type of activity that a network operator wants to monitor, a network operator might attempt to capture conversations of interest in one of three ways: (1) *volume-based accounting*, which attempts to capture as much traffic in the conversations of interest as possible, (2) *conversation-based accounting*, which is concerned with simply collecting traffic from as many unique conversations as possible, and is not concerned with capturing significant traffic volumes from any conversation, or (3) host-based accounting, which attempts to determine as many of the communicating hosts (possibly those infected) as possible. If conversations are *stealthy* (*i.e.*, low-volume, either by design or by nature), then traffic sampling will harm the operator's ability to see these conversations.

3.2 Motivation: Coordinated Conversations

Previous work has observed that communication between hosts can be represented in terms of graph structures, and that these graph structures can be used to identify certain types of applications [22]. Based on this insight, we hypothesize that, by observing network traffic as it crosses a transit network, network operators can generate graph signatures for "applications" (*e.g.*, botnet command-and-control, spammers, etc.) that may help network operators identify and mitigate unwanted traffic. As described in the previous section, monitoring the communication graph structure of a particular application over time may help operators identify suspicious stealth activity. In particular, it would be worthwhile to look for the following indicators of botnet behavior:

- Low-volume, correlated traffic spikes (*e.g.*, as seen with a distributed denial-of-service (DDoS) attack).
- Overlapping communication between botnets, when a single host is a member of multiple botnets
- Migration of command-and-control traffic
- Peer-to-peer communication structures over non-standard ports

3.3 Differences from Anomaly Detection

It is important to note that we are *searching* for specific signatures in sampled traffic, which is distinct from anomaly detection. Anomaly detection is the process of defining normal behavior and then looking for deviations from that normal behavior; an anomaly detection method therefore has no knowledge of specific *signatures* to search in the underlying traffic [25, 24]. However, the network conversations that we seek are typically a set of chained network-wide connections, or even a single connection between two IPs. As such, network conversations from stealthy applications (like botnets) are likely not to disturb normal traffic in any detectable manner.

A primary difference between conversation detection and anomaly detection is that conversation detection is *far more sensitive to sampling* than certain types of anomaly detection. Although many types of anomalies will still be visible in some form after sampling because of their nature, sampling may completely eradicate low-volume, stealthy conversations between pairs of end hosts. Given that we are primarily concerned with identifying *all* pairs of these conversations, sampling the traffic flows would intuitively make it very difficult to capture all of these conversations. In some sense, whereas anomaly detection may be looking for unusual incidents (which, by definition, could show up at any sampling rates), conversation detection is fundamentally a *collection* problem. Ideally, an operator will use a coordinated network-wide sampling strategy for identifying a higher percentage of conversations.

3.4 Probabilistic Analysis

A popular sampling strategy preferred by many routers is simple random sampling, where each packet is sampled at a constant probability, irrespective of earlier communication. In such a scenario, it is trivial to predict the effect of sampling on conversation-based accounting. Let:

$p =$ Sampling probability
$C(x) =$ Number of packets in conversation 'x'
$X =$ Set of all conversations

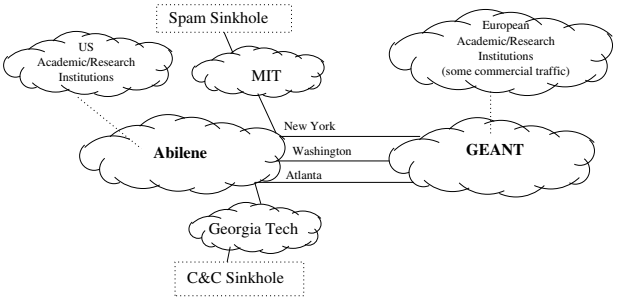


Figure 1: Network topology for experimental setup, showing the location of our spam sinkhole and our command-and-control sinkhole in relation to the Abilene and GEANT networks.

Considering the above definition of the parameters involved, the probability that at least one packet from a conversation 'x' is sampled is given by: $P(X) = [1 - (1 - p)^{C(x)}]$. Given a distribution of the number of packets per conversation and a list of possible conversations, it is easy to determine the expected number of unique conversations sampled: $E(X) = \sum_X P(X)$. Clearly, the lower the sampling probability, the lower is the expected number of conversations sampled. Furthermore, the non-linearity in the relation between $E(X)$ and $C(X)$ indicates that a "mice" conversation with few packets suffers a drastically low probability of being sampled. This explains much of our results in the forthcoming sections.

4 Experimental Setup

In this section, we describe the setup of the two experiments we run to characterize the effectiveness of traffic sampling in capturing two types of conversations: botnet communication with command-and-control, and spam sinkhole traffic from the spamming hosts (most of which are bots [29]) to a spam sinkhole.

To study the effectiveness of using sampled network flow traces to monitor network conversations, we perform two independent experiments. First, we collect both packet traces for two sets of conversations—(1) bot communication with a command-and control sinkhole at Georgia Tech and (2) spamming host communication with a spam sinkhole at MIT—and determine, using flow records collected at routers in the GEANT and Abilene networks over the same time period, how well traffic sampling exposes each conversation that we determine to have crossed the GEANT or Abilene network. Second, to evaluate the effects of higher sampling rates than those that are used in GEANT or Abilene, we perform a controlled experiment by synthesizing a packet trace that is characteristic of what would be observed on an interface at an Abilene router and apply different sampling rates to this trace.

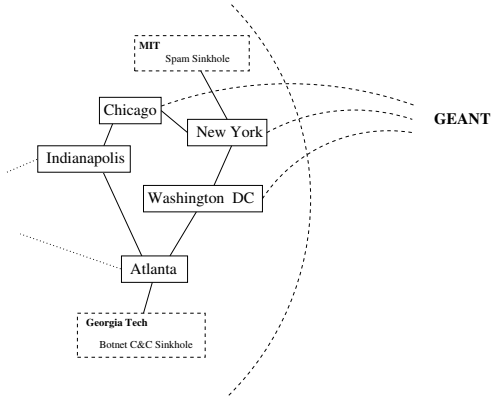


Figure 2: The east coast routers in the Abilene network topology, and their topological relationships to our sinkholes and to the GEANT network.

Figure 1 shows a picture that summarizes the locations of our sinkholes in relation to the Abilene and GEANT networks. The spam sinkhole is located at MIT; traffic that transits the Abilene network will enter and exit the Abilene network at the Abilene router in New York City. Similarly, the botnet command-and-control sinkhole is located at Georgia Tech, and traffic that transits the Abilene network to and from the sinkhole will transit the Abilene router in Atlanta. We ran *rcc* [15] on the Abilene routing configurations to determine both the Abilene IGP topology and the peering locations of Abilene and GEANT. The GEANT and Abilene networks peer in 3 locations: Chicago, New York, and Washington. Thus, for the two experiments we conduct, the conversations will typically pass through at least two routers in the Abilene network: one of the three peering routers, one of the two egress routers, and any intermediate routers along those paths.

Figure 2 shows the IGP graph for Abilene, as measured from router configuration files. In theory, link failures could cause traffic between ingress and egress of Abilene to traverse any number of routers within Abilene, but we expect that most traffic from the GEANT network will traverse no more than 3 routers, as per the shortest paths in the IGP graph.

As Figure 1 illustrates, some of the conversations in our ground truth dataset will traverse both the GEANT and Abilene networks, and other traffic will traverse only the Abilene network. The data captured at our sinkholes is from a diverse set of Internet hosts, many of which do not traverse either GEANT or Abilene, so one of the major challenges with our experiment is to estimate the "ground truth" conversations that the traffic sampled in the Abilene and GEANT networks should catch. In Section 5.3, we describe how we use routing table information to determine from our packet traces which conversations are likely to traverse either the Abilene network or both the GEANT and Abilene networks.

5 Data Collection

In this section, we describe the data used in our analysis. We first describe the three datasets that we used to generate "ground truth"—network conversations that we can say, with high confidence, traversed the ISP from where the passive network flow measurements were collected. Then, we describe the flow measurements collected at 22 routers across the GEANT network [17], which carries both research and academic traffic and some amount of commercial traffic in Europe; and 11 routers in the Abilene network [3], the academic and research network within the United States. Finally, we describe the traffic that we collect from the Georgia Tech campus network for use as cross-traffic in our emulation experiments.

5.1 Ground Truth "Conversation" Data

To quantify the effectiveness of using traffic sampling for capturing network conversations, we use three different datasets to derive ground truth expectations for the traffic that should traverse the network that we are monitoring with traffic sampling. The packet traces of the botnet command-and-control and the traces collected at the spam sinkhole provide information about pairs of communicating hosts. Given knowledge of the routing tables from each bot to both the command-and control and the sinkhole, we can determine with high likelihood the conversations that are likely to traverse the networks where we are monitoring flows.

5.1.1 Command-and-Control Conversations

Our first dataset for botnet-related conversations is the command-and-control traffic itself (*i.e.*, the communication between each of the bots and its controller). We captured a packet-level trace of this conversation by hijacking the authoritative DNS server for the domain running the command and control of the botnet and redirecting it to a machine at a large campus network; more details of this technique are described in previous work [11]. This method was only possible because the Bobax drones contacted a centralized controller using a domain name. For our analysis, we use a packet trace of conversations between hosts infected by the `W32/Bobax` ("Bobax") worm and their command-and-control from November 16, 2005 to December 31, 2005.

This DNS hijacking technique directs control of the botnet to the honeypot, which effectively disables it from spamming for the period of the hijack. Nevertheless, even though this technique disables the botnet, the resulting packet trace allows us to obtain a reasonable estimate for the IP addresses involved in botnet-related communication with a command-and-control host. Because we have redirected the command-and-control traffic to a location on the Georgia Tech network, much of the traffic may cross either the GEANT network or the Abilene network where we have access to flow records (described in Section 5.2).

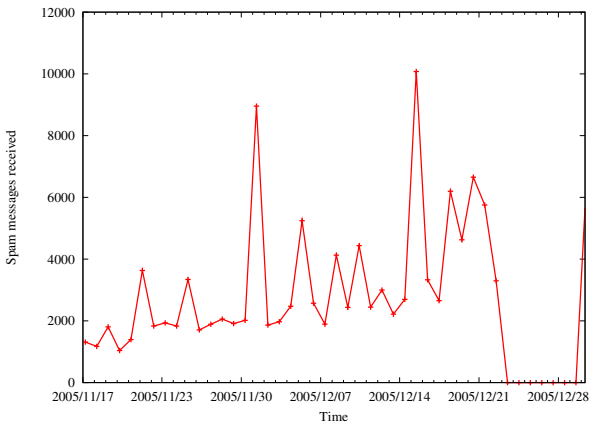


Figure 3: The amount of spam received by the spam sinkhole over the duration of the experiment (November 17, 2005—December 31, 2005)

5.1.2 Spam Conversations

To create a sinkhole for spam traffic, we registered a domain and established a corresponding DNS Mail Exchange (MX) record with *no legitimate email addresses*, which allows us to deduce that all mail received by the mail server is spam. This "sinkhole" has been capturing spam since August 5, 2004. Figure 3 shows the amount of spam that the sinkhole has received per day during November 2005 and December 2005 (the period of time over which we conduct our analysis). In this trace, we cannot determine with absolute certainty the methods that each of the spamming hosts used to send spam to the sinkhole (*i.e.*, with a botnet, with BGP route hijacking, through an open relay, etc.); nevertheless, results from our previous study strongly suggest that the vast majority of this spam—upwards of 80%—is being received from spamming bots, even though we were only able to confirm about 5% of these IP addresses by correlating them with the command-and-control trace [29]. For the purposes of our study, we determine how much of *any* of the traffic to our spam sinkhole can be observed in the sampled traffic, under the assumption that most of this traffic is attack traffic from botnets.

5.2 Traffic Flow Data

Our principal datasets that we mine for network conversations come from two backbone networks: Abilene and GEANT. Abilene spans the continental US, with 11 PoPs, and carries traffic for 200 US universities and research labs. GEANT is a larger network, with 22 PoPs, which connect research networks across all the major European

cities. From both networks, we collected sampled traffic records from all the PoPs for the month of November 2005. The Abilene data is sampled using stratified random packet sampling at a rate of 1 out of 100; the GEANT data is also sampled in a similar manner, but at a higher rate of 1 out of 1000 packets. Also, the IP addresses in the Abilene data are anonymized, but the GEANT data has no such anonymization.

5.3 Calibration: Routing Data

As previously mentioned, to establish a baseline level of communication for what we expect to see in the sampled traffic, we need to determine which of the above conversations are likely to traverse the networks where we have captured flow records (*i.e.*, the GEANT and Abilene networks). Unfortunately, determining whether the *forward* path from each bot to the sinkhole traverses one of these networks requires either access to the routing tables from where each of these bots are located, or the ability to run traceroutes from each of these bots to the respective sinkholes, neither of which is realistic.

Because it is not possible to obtain routing tables from the networks where each of the bots is located, we estimate the bots that are likely to send traffic through the GEANT or Abilene network based on the routes that pass through these networks in the *reverse* direction (*i.e.*, from the respective sinkholes to the bots). This technique, of course, assumes that the routes between the bots and the sinkholes are symmetric, at least to the extent that passing through either the GEANT or Abilene network along the reverse path is a fair indication that the respective network is also traversed along the forward path. We use the routing tables co-located with our spam sinkhole, located at MIT, to estimate whether the IP-level paths from the bots to the sinkhole are likely to cross these networks. We also use this routing table to estimate the IP-level paths from the Bobax command-and-control sinkhole to the command-and-control sinkhole, which is located at Georgia Tech.

5.4 Cross-Traffic Data

To emulate traffic sampling performance and its impact on the amount of information about network conversations preserved in Netflow records, we combine our command-and-control packet captures with background traffic that we expect to be present between Georgia Tech (location of the C&C) and its uplinks. Since the ingress traffic of the Georgia Tech gateway router is roughly constant over corresponding time-intervals across different dates, we believe it is sufficient to "mix" cross-traffic for a given time-interval on a given day of the week (irrespective of the actual date) with command-and-control traffic for the same time-interval/day-of-week combination. We also ensure that the day in question do not suffer anomalous traffic patterns such as flash crowds, network disruptions etc. by comparing the total traffic on the uplink with average traffic rates for that month.

The cross-traffic logs we acquired contain about 2.5 hours of anonymized packet capture logs from the ingress interface of the Georgia Tech border router. The IP addresses are anonymized with a /21 mask due to policy restrictions; however this does not influence our emulation because we have the exact volume and timestamp information, and the IP specifics are just extraneous.

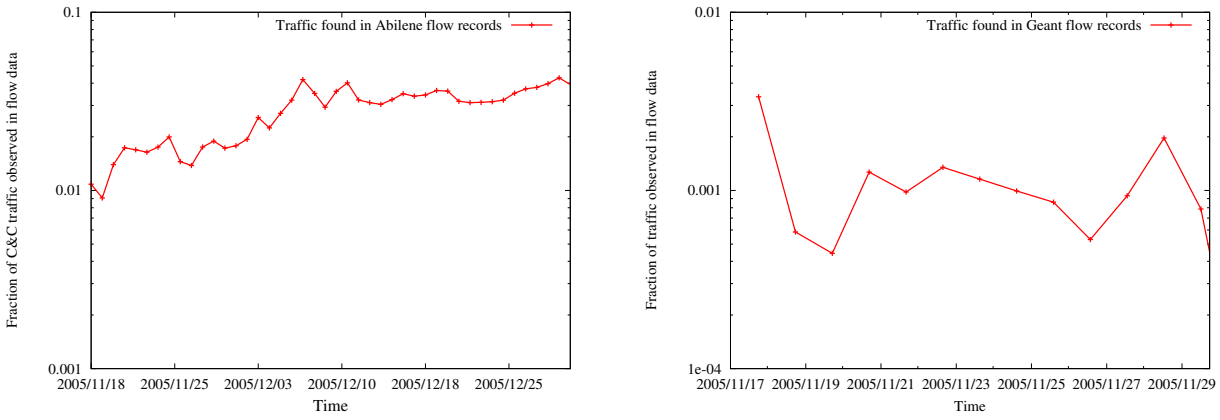
6 Evaluation of Traffic Sampling

In this section, we evaluate the efficiency of traffic sampling in capturing malicious conversations ("ground-truth"). We try to measure whether, and to what extent, different types of conversations appear in sampled traffic obtained at two transit ISPs (each of which uses a different sampling rate). We monitor two types of conversations: (1) "Command-and-Control" packet captures from a sinkhole for a Bobax botnet (Section 5.1.1); and (2) Spam traffic collected at a spam honeypot (Section 5.1.2). Using sampled traffic from two transit ISPs–Abilene and GEANT–we attempt to correlate the volume of conversations that crossed these networks, to the amount that was actually captured by the flow logs.

As mentioned in Section 5, Abilene and GEANT use different rates ($1/100$ and $1/1000$, respectively) to sample packets before the packet headers are inspected and its statistics aggregated. Irrespective of the specific sampling technique used, the order-of-magnitude difference in the sampling rates of Abilene and GEANT enables us to draw conclusions about how quickly the efficiency of traffic sampling for capturing malicious conversations degrades.

6.1 Experimental Setup

To examine the effectiveness of traffic sampling in capturing conversations that are known to exist in the traffic that traverses GEANT, Abilene, or both, we take logs of the sampled traffic during a time period when we have determined certain conversations exist (our "ground truth" data) and determine to what extent traffic sampling can extract these conversations. We first use the routing tables and packet traces to determine ground truth: that is, the complete set of conversations in each of our traces (*i.e.*, botnet command-and-control and spam) that traversed the Abilene network, the GEANT network, or both. Note that, because our sinkholes are located downstream from the Abilene network, there actually exist only two cases for which we must consider ground truth: conversations that traverse both Abilene and GEANT, and conversations that traverse only Abilene.



(a) C&C traffic in Abilene

(b) C&C traffic in GEANT

Figure 4: Fraction of Command-and-Control traffic captured in the flow logs collected from Abilene and GEANT. The GEANT simulation runs for fewer days due to unavailability of flow data.

For both of the network conversations in our experiment, we perform the following sequence of steps:

Step 1 (Determine ground truth) Determine which IP prefixes in the ground truth datasets are likely to transit Abilene and GEANT to and from our sinkholes.

For this purpose, we use the BGP routing tables collected at MIT. Our inferences make two assumptions: First, we assume that if MIT uses Abilene to reach a certain destination, then Georgia Tech will do the same. This assumption is reasonable because MIT and Georgia Tech both have similar relationships with Abilene. Second, we assume that traffic that exits MIT or Georgia Tech via Abilene will also return via Abilene. This assumption is also reasonable because Abilene transits traffic between research and educational networks: if traffic from Georgia Tech or MIT transits Abilene en route to some destination, then that destination is also likely a network that is permitted to use the research network to reach Georgia Tech or MIT (*i.e.*, it is also a member of Internet2).

Step 2 (Extract conversations from flow records) We extract the communicating pair of hosts from each flow record, and perform a union over all routers in the respective network (*i.e.*, Abilene or GEANT), preserving the timestamps for when each conversation was observed.

Given the ability to extract conversations from the sampled flow records in both Abilene and GEANT networks and to compare the resulting observations to our ground truth measurements (*i.e.*, all of the conversations that we have determined are traversing the network), we can evaluate the effectiveness of flow sampling in capturing the conversations that our ground truth data indicates exist in the traces.

To determine whether a given IP address crossed one of these networks *at the time of connection to the sinkhole*, we inspected BGP update logs collected at the sinkholes for *the existence of the network's AS number in the ASPATH attribute of a route announcement matching the IP address under scrutiny*. For instance, to determine whether a conversation from a host with IP address `10.0.0.1` crosses AS number `1`, we perform a longest-prefix match for those route announcements including `10.0.0.1` (*e.g.*, `10.0.0.0/24`), received before the time of the conversation[2]. If the obtained announcement contains the required network's AS number (`1` in our example above) in its ASPATH attribute, we claim that the given conversation likely crossed the network in question.

The next step is to match ground truth data with conversations that are known to have crossed a network, with the actual flow data captured. Owing to the large volume of flow data and packet captures to be analyzed, we applied pruning techniques to expedite our measurements. First, we pruned Abilene and GEANT flow data to only flows which had either source or destination as one of the sinkhole IP addresses and their corresponding ports. Second, we pruned the resulting set to include only those remote IPs that appeared in our ground-truth data sets. After pruning, we match the IP addresses and ports in the pruned flow data with ground-truth data, for comparable unixtimes. Since the flow records could have a slightly different timestamp from the sinkhole logs, we maintain a

[2]We did not set an explicit time bound when looking for a route going back from the timestamp of the conversation. In this analysis, we looked for all routes within a few days before the conversation. We believe this is a reasonable approximation as Abilene, GEANT, and MIT networks are well-managed and usually do not originate unstable or erroneous routes.

low acceptable threshold of error between the two timestamps. We will now discuss the details of the analysis for the two sinkholes, for both Abilene and GEANT.

6.2 Command-and-Control Traffic

The Command-and-Control (C&C) data we used for this analysis was available as packet captures, enabling us to identify the exact timestamp, remote IP address, and port number for each connection. We correlated this data with observed conversations in flow records from both Abilene and GEANT. Our analysis is based on a 45-day packet capture, containing connections from over 2.14 million unique IP addresses, with an average of 7.4 million connections per day.

6.2.1 Abilene

Abilene flow records anonymize the source and destination IP addresses with 21-bit mask. Though this might appear to be a deterrent in accurately matching the two data sets, we are able to increase our confidence by also comparing client port numbers in the sampled traffic traces with the port numbers in our packet capture files. While we acknowledge that there could be a small amount of false positives, we are confident that this number is insignificant compared to the actual number of observed conversations: both our sinkholes are machines at academic institutions, and the usage of allocated IP address at these institutions is not dense. Thus, for a given subnet (and particularly for the /21 subnets at MIT and Georgia Tech containing the sinkholes), it is unlikely that there are many other machines offering the same services as our sinkholes, and *receiving connections from the same IP addresses that are present in the ground truth data (albeit the /21 anonymization) at the same time*.

Figure 4(a) shows the fraction of ground truth conversations captured in the sampled Abilene traffic. We see that the fraction of the total botnet command-and-control conversations that were captured roughly tracks the sampling rate (*i.e.*, $1/100$). This result indicates, as expected, that low sampling rates are likely to be ineffective at capturing much of these conversations.

In addition to observing the fraction of *total* conversations captured (what we have defined as "volume accounting" in Section 3.1), in many cases, identifying the *existence* of a conversation between two unique IP addresses ("conversation-based accounting") may suffice. For example, most of the applications described in Section 3.2 require only conversation-based accounting. Figure 5 shows the performance of traffic sampling for conversation-based accounting (*i.e.*, the fraction of unique IP observed in flow data). As can be seen from the plot, the number of unique conversations observed is significantly higher than the sampling rate, which correlates well

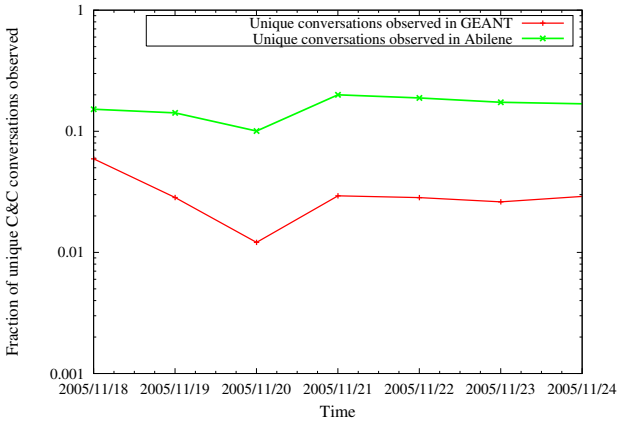


Figure 5: Fraction of unique conversations observed in sampled GEANT and Abilene flow records for a 1 week period

with Figure 7(b) from our experiments in Section 7. However, as we will see in Section 7, as the sampling rate increases, the marginal benefit for capturing additional conversations decreases, and it becomes harder to capture all conversations as long as the traffic is sampled.

6.2.2 GEANT

Our measurements for C&C conversations in the GEANT traffic are similar to those for Abilene, with a few minor differences. First, we use Abilene BGP tables from Abilene's Atlanta router (`abilene_atla`) instead of the Georgia Tech BGP tables. This assumption is justifiable, as GEANT peers with Abilene (see Figure 1), which in turn provides transit for Georgia Tech (and MIT) (see Section 4). Second, we can be certain that the conversations captured by traffic sampling match those in our ground truth dataset, because GEANT does not anonymize IP addresses. Third, our analysis is restricted from November 17, 2005 to November 30, 2005, due to unavailability of GEANT data past November 2005. The fraction of C&C packets actually observed in GEANT traffic is shown in Figure 4(b). Indeed, we note that the values stay around 0.001, which is the traffic sampling rate employed by GEANT.

6.3 Attact Traffic: Spam

In comparison to C&C traffic, spam arrives at a much lower and much more sporadic rate. Over the 45-day duration of the analysis, the sinkhole received 1,559,282 spam messages, of which 38,420 traverse Abilene and 15,708 traverse GEANT. (The analysis used to identify this traffic is similar to that in the previous subsection.) As Figure 6 shows, there is higher variance in values of fractions compared to Figure 4(a), possibly because the actual amount of spam traffic crossing Abilene is much

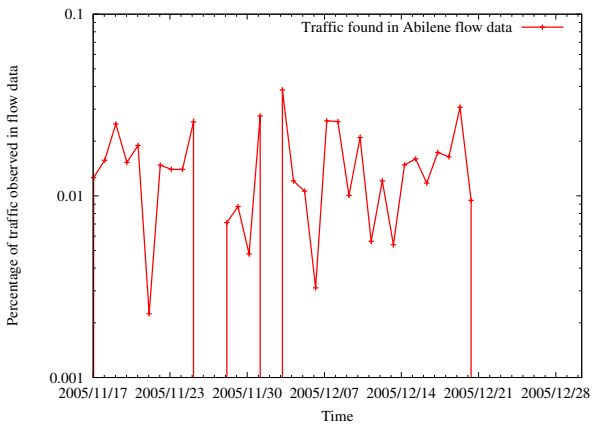


Figure 6: Fraction of spam traffic captured in sampled Abilene traffic, from Nov. 17–Dec. 31, 2005

more sporadic than C&C traffic. Still, the average value of the fraction of total conversations captured is slightly less than $1/100$, Abilene's sampling rate.

Although traffic sampling is ineffective for capturing spam-related network conversations, the sampled traffic flows did reveal some interesting activity: the logs reflected many SMTP conversations (*i.e.*, on port 25) for which there were *no* spam received at the sinkhole. This reflects evidence of port-scans or reconnaissance activity.

6.4 Attack Traffic: Infection

Network operators often use traffic sampling as a means to detect certain types of malicious activity (*e.g.*, portscans, DDoS attacks, etc.). We speculated whether profiling the infection pattern emanating from the known members of the Bobax-based botnet would be useful for detecting other groups of infected hosts that we were not aware of. To do so, we inspected activity on specific ports that are typically used for worm propagation in the Internet. The general motivation behind our approach was to determine if a host is emanating certain suspect traffic chronologically after the suspect traffic was sent to it by known Bobax hosts.

We isolated the traffic in the GEANT flow records that were generated by hosts we know have already been infected by Bobax/W32 (as noted from the botnet C&C data described in Section 5). Table 1 shows the traffic volume and unique conversations that we observed for infection traffic from known Bobax hosts. The table also contrasts the activity by known hosts against the overall traffic and number of conversations using those ports. We observe from the table that the $1/1000$ sampling rate of the GEANT network eliminates all interesting activity on the suspect ports. This experiment illustrates that existing traffic sampling techniques are also likely to be ineffective at detecting conversations involving malware.

Port	Exploit	Volume-based		Conversation-based	
		Known	Overall	Known	Overall
135	RPC/DCOM	1724	485782	213	16909
445	LSASS	3664	745567	1291	82833
5000	LSASS	213	188678	65	32151

Table 1: Analysis of activity on ports used by existing bots to spread the Bobax/W32 infection. We present traffic from known bots and the overall traffic.

7 Analysis of Sampling Techniques

In this section, we explore the effects of different means of traffic sampling.

7.1 Experimental Setup

We would like the flexibility to examine how sampling rates and techniques other than those currently used in existing networks might better expose network conversations. Without direct access to routers in a real network, however, we must devise an alternate mechanism that emulates the effects of applying different sampling rates and techniques in an actual network.

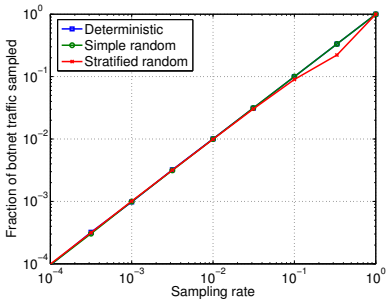
To perform controlled experiments to determine the effects of different sampling rates on the ability to monitor conversations, we mix the packet traces for the conversation of interest, namely the botnet command-and-control trace, with an appropriate estimate of the cross-traffic that would be traversing the ingress router at the same timeframe. Ideally, we would be able to obtain packet traces for all traffic entering the Georgia Tech network coinciding with the time of the actual conversations that we monitored at the end of 2005. Unfortunately, we do not have packet traces from the times when these actually did occur.

As mentioned in Section 2, we only focus on three sampling techniques, namely *deterministic*, *simple random* and *stratified random*. To analyze their effects, we emulate the corresponding sampling on a traffic mix comprised of the traffic captured at our Georgia Tech sinkhole, and an approximate estimate of cross-traffic. To perform this emulation, we adopt the following steps:

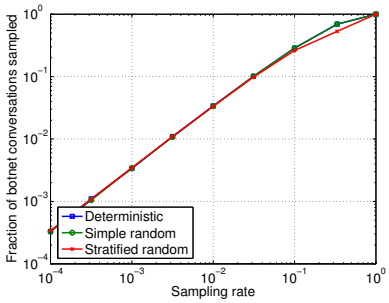
Step 1 (Determine ground truth) Obtain the complete set of botnet command-and-control packets logged at the Georgia Tech sinkhole for a particular timeframe. The packets use one of four different upstream ASes to enter the Georgia Tech network.

Step 2 (Derive cross-traffic) Derive a set of packet traces that represents the plausible cross-traffic at the Georgia Tech border router.

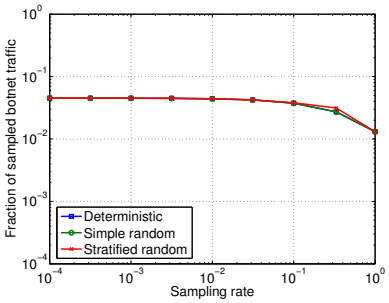
Step 3 (Combine packet traces) Create synthetic traffic by mixing the above two sampled packet traces together (*i.e.*, the conversation traffic and the cross-traffic), keeping the relative timing of packet arrivals from the two packet



(a) Volume-based accounting

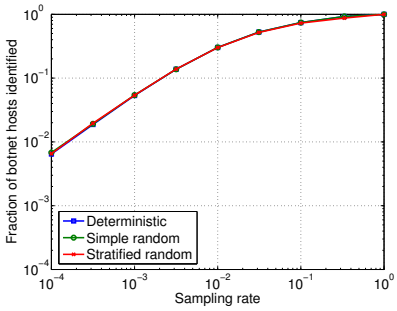


(b) Conversation-based accounting.

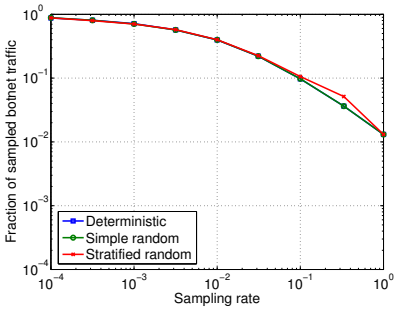


(c) Fraction of botnet traffic samples that belong to a new conversation.

Figure 7: Effect of sampling rate and sampling technique on the volume of botnet command-and-control traffic sampled, and the number of unique conversations observed.



(a) Host-based accounting



(b) Fraction of botnet traffic samples that belong to a new bot

Figure 8: Effect of sampling rate and sampling technique on the number of unique bots observed.

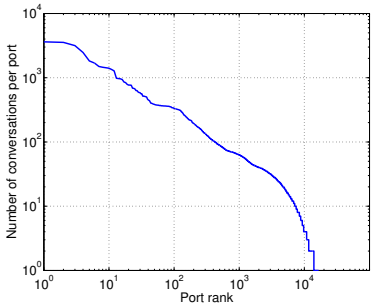


Figure 9: Number of different conversations over each TCP port.

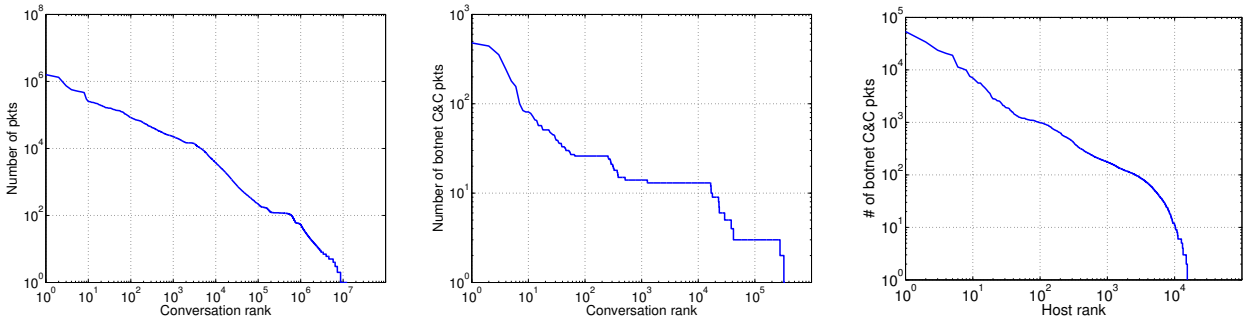
traces. This gives the perspective of overall traffic as seen by any network operator concerned about conversations entering his/her network.

To circumvent the lack of coinciding packet traces, we devise the following workaround to obtain an approximate mix of cross-traffic. We extracted Bobax command-and-control traffic captured on November 22, 2005 from 7:30 to 10 a.m EST, with the appropriate timestamps and interspersed the regular traffic captured at the Georgia Tech ingress router on April 4, 2006 during the same time period, while maintaining the temporal ordering. This is an approximation, but, unfortunately, the actual cross-traffic from Nov 22 is unavailable. Further, we mark Bobax packets to avoid any misinterpretation owing to same IP address. We apply the three packet-based sampling schemes to this traffic mix and determine the resulting number of botnet conversations sampled for various sampling rates.

We perform three experiments with this synthetic packet trace. First, to measure the effectiveness of different sampling rates for exposing the conversations of interest, we subject the packet trace to nine different sampling rates: $1/10000$, $1/3160$ $1/1000$, $1/316$, $1/100$, $1/31$, $1/10$, $1/3$ and 1 (the last serving as an upper bound on performance). As we sample the packet trace, we keep track of each conversation that we capture and record the fraction of total conversations we were able to expose using each traffic sampling rate. Second, to emulate the effects of these flows crossing k different routers in the Abilene network, we perform k independent trials at each sampling rate, keeping track of the union of the conversations exposed by all trials. Finally, to measure the effects of cross-traffic on the ability of using flow sampling to expose these conversations of interest, we downsample or re-populate the cross-traffic measured by the Georgia Tech ingress router, and re-perform the measurements to determine how many conversations from the packet trace are detectable with sampling.

7.2 Traffic Accounting

Figure 7 presents our results for two modes of accounting: general volume-based accounting, where we deter-



(a) Number of packets per conversation for all types

(b) Number of packets per botnet conversation

(c) Number of packets sent by each unique bot

Figure 10: Log-log distribution of packets generated by each host or conversation.

mine the fraction of all botnet traffic that is sampled, and conversation-based accounting, where we determine the fraction of unique conversations that are sampled. Note that each value on the two plots have been averaged over 25 iterations. We observed a total of 1,199,135 original packets during the timeframe of interest, and 15,712 unique conversations during the same timeframe.

In Figure 7(a), we present the fraction of original botnet traffic sampled by each of the three schemes. We observe That all three schemes perform similarly with regards to the amount of botnet traffic sampled. Moreover, all three schemes sample the subset traffic (viz. botnet command-and-control) to the same extent as the overall population. This shows that the schemes are unbiased estimators with respect to the volume of packets.

Figure 7(b) shows that, although the number of unique conversations sampled at low sampling rate is relatively high (a surprising result), the number of unique conversations captured does not increase proportionally to the sampling rate. This result can be explained intuitively, and, in fact, it bears similarity to the "coupon collector" problem: we are using traffic sampling to collect all conversations in the botnet, and each sample effectively "collects" a conversation which is either new or a duplicate. As the sampling rate increases, the likelihood of collecting duplicate conversations increases, and it takes many trials (*i.e.*, samples) to collect the last few conversations.

Figure 7(c) further explains this result: At a sampling rate of $1/100$, each technique samples an average of 12,000 botnet packets in the dataset. These packets belong to about 5,000 unique network conversations on average. At a sampling rate of $1/10000$, though the fraction of botnet traffic sampled is small, most of the packets belong to new conversations, even though absolute number of conversations logged is smaller.

Figure 8 presents our results for the third mode of accounting: host-based accounting, where we determine the fraction of unique botnet hosts that are identified. We see that host-based accouting has a higher success rate relative to conversation-based accounting. Hence, the fraction of hosts identified does not increase linearly, with respect to the sampling rate. Furthermore, Figure 8(b) illustrates how host-based accounting also has a behavior similar to conversation-based accounting, where most of the packets sampled at a low sampling rate belong to a new conversation.

Figure 9 shows the distribution of ports used by various conversations. We see a non-uniform distribution of the ports used by all conversation. This explains why the host-based accounting is not congruent with conversation-based accounting.

We present in Figure 10 the distribution of the number of packets generated by each host or by each conversation. This gives an insight into the $C(x)$ parameter we described in the model. All three figures give further evidence on why it is difficult to use traffic sampling for conversation-based accounting. Figure 10(a) shows that there are a few conversations that transmit heavy volumes of packets, thereby making the distribution of packets per conversation non-uniform. The heavy tail of low-volume talkers makes it difficult to select an appropriate sampling rate. Similarly, figures 10(b) and 10(c) show the non-uniform distribution of botnet packets generated by each conversation and each host respectively.

In summary, our results emphasize that existing sampling techniques are not particularly effective for capturing these low-volume conversations.

7.3 Network-wide Sampling

In the previous subsection, we analyzed the accuracy [3] of detecting network conversations at a single point (typ-

[3] By the term *accuracy*, we refer to the success achieved in logging as many of the network conversations as possible.

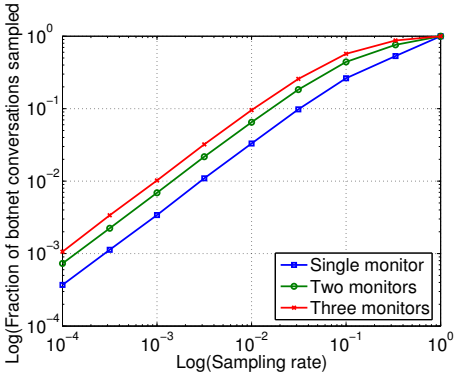


Figure 11: Effect of network-wide sampling on the fraction of unique conversations logged.

ically the ingress router of an AS). We next investigate the effect of sampling at multiple locations of a network with the aim of capturing as many of the conversations as possible, with least sampling overhead.

Previous research has investigated the problem of selecting traffic monitors and setting the optimal sampling rates for each monitor in an effort to achieve a specific measurement task [6]. However, the previous work is still bounded by the tradeoff between the number of small flows that needs to be monitored and the sampling rate. When the resources available for sampling are limited, this tradeoff causes network-wide sampling solutions to also miss certain stealthy conversations.

We emulate a scenario similar to that observed in Abilene or GEANT, where traffic passes through multiple routers, with each router performing stratified random sampling at a fixed rate. Particularly, we examine whether it is possible to capture more unique botnet conversations by monitoring the conversations at more locations. To emulate the cross-traffic at the other monitors, we use the same traces as in our previous experiments, with some additional pre-processing: We divided the whole traffic into bins of 1 second each and vary the timestamp of cross-traffic packets within each bin, which preserves the diurnal pattern, while yielding a different ordering of Bobax and non-Bobax packets. Note that the pre-processing still maintains the ordering within the set of Bobax packets and the set of non-Bobax packets.

Figure 11 shows the accuracy for the cases when the number of monitors that the conversation traverses is 2 or 3. Our results show that monitoring traffic at several locations within the network does not significantly improve the likelihood of capturing botnet conversations. Also, the sampling rate is increased, the marginal benefit of sampling at additional routers decreases. However, the network-wide sampling scenario incurs high resource usage, which increases with the number of monitors. Thus, the overhead incurred in network-wide sampling of botnet traffic is not worth the accuracy.

7.4 Effects of Cross-Traffic

Past research of Internet traffic has showed the existence of diurnal patterns. It is possible, however, that the botnet command-and-control traffic does not exhibit the same diurnal pattern as the cross-traffic does. In that case, the volume of cross-traffic has a bearing on the sampling process. Hence, we examine the effect of different levels of cross-traffic on the number of conversations logged. To emulate different amounts of cross-traffic, we perform a scrambling process similar to the pre-processing described in the previous subsection. During this process, we divide the whole set of Georgia Tech cross-traffic into bins of 1 second each. We either remove a random subset of packets from each bin or add new packets with random timestamps to each bin. In particular, we simulate conditions where the cross-traffic is 50%, 200%, and 400% that of what was actually observed.

We inspected the number of unique network conversations logged with each mix of cross-traffic and observed that the sampling process performs the same under all four levels of cross-traffic. This is because the sampling process depends only on the rate of sampling—the actual value of N—as it determines the periodicity of the selection bin in the case of deterministic sampling and stratified random sampling, or the selection probability in the case of simple random sampling. Hence, the number of packets belonging to the botnet traffic is the same for all levels of cross-traffic. The exact number of unique conversations the sampled packets belong to depends on the cardinality of the hosts (as shown in Figure **??**). To understand this result, consider an example where the dataset has 1000 packets in the botnet traffic and 100,000 packets in the cross-traffic. A sampling rate of 1/100, will yield about 1,000 samples, of which we can expect 10 to be belonging to the botnet traffic. If the cross-traffic doubles, we can still expect 10 samples from the original botnet traffic. This is an artifact of unbiased *packet sampling* (more cross traffic means more samples).

8 Related Work

We briefly survey previous work in two areas: traffic sampling, and traffic classification and detection.

8.1 Traffic Sampling

Previous studies have decried the limitations of traffic sampling, in terms of how it distorts estimates of traffic flow size distributions [18] and short flow characteristics [12]. Others have demonstrated the utility of traffic sampling for detecting high-volume attacks [13, 19]. There have also been various efforts to improve the accuracy and overhead problem of sampling [14, 13, 23, 12, 7], most of which attempt to adapt the sampling rate to

changes in flow characteristics, or attempt a different sampling strategy altogether.

In this paper, however, we are interested in a different question entirely: we are not concerned with traffic volumes, length of flows, and so forth; rather, we only seek to determine *whether two hosts are communicating at all*. Our problem is also unique because we are attempting to evaluate the effect of sampling on capturing a specific subset of traffic—specifically, botnet traffic—that is present within the overall population. The botnet traffic typically represents short length "mice" flows, unlike previous work that focused predominantly on "elephant" flows [14], which makes the problem of capturing network "conversations" all the more interesting. [13] identifies the problem in estimating the number of flows belonging to a certain aggregate within the overall traffic (*e.g.*, the number of SMTP flows) and proposes a hardware extension for purposes of flow counting. Our work tries to address a similar problem, but at a much smaller granularity, and with respect to flows designed to be stealthy in nature.

Recently, there has been focus on improving network-wide sampling by strategically placing and tuning monitors at various locations of the network, in an effort to sample most of the flows pairs in a collaborative manner [6]. Because complete (full) sampling is typically infeasible on high-volume links, however, these techniques must also rely on traffic sampling. Our results in Section 7 demonstrate that even network-wide sampling is still deficient in tracking all existing "mice" flows.

8.2 Traffic Classification and Detection

One application of traffic sampling is to aid in network security by helping network operators detect malicious traffic [4]. Previous traffic classification studies have used network conversations to identify attack traffic [22]. This work observed that it was possible to characterize application-level traffic according to the communication patterns between hosts, and applied this observation to the flows observed on a single link. This work inspired us to explore whether network-wide traffic sampling could be used to construct these types of communication graphs. Similarly, there are protocol-specific studies that help identify the botnet memberships [5, 10, 28]. Typically, they use the patterns in command and control traffic (*e.g.*, IRC) to identify suspect activity and associated hosts.

9 Conclusion

This paper has motivated the need for capturing the existence of network *conversations*—pairs of end hosts that are exchanging traffic. Network conversations represent a new class of network-wide traffic patterns they provide insight into the structure of traffic flows between hosts. This structure can provide insight that helps network operators detect suspicious or unwanted traffic that might otherwise escape notice. Exposing these conversations can help operators detect emerging threats which are fundamentally different than conventional anomalies and attacks that tend to send high volumes of traffic (*e.g.*, DDoS attacks, portscans, etc.).

We have used two botnet conversations as motivating examples: communication with command-and-control and "attack" traffic with victims. Detecting these conversations is a challenging problem, as they are typically designed to be low volume and stealthy in nature. This paper takes the first steps towards studying and measuring this important class of network conversations using existing measurement techniques, using sampled flow records from two ISPs to mine for botnet conversations. However, conversations by their very nature are fine-grained chained connections, and are extremely sensitive to packet sampling. We show that all commonly used techniques are ineffective and unsuitable for detecting these stealthy conversations.

Our results and techniques (joint-analysis of datasets and controlled experimental techniques) open exciting new research directions, given the importance of detecting network conversations, and the inadequacy of current sampling methods for doing so. Thus, we present this paper as a call-for-arms for better sampling strategies, and not just improved sampling rates, in order to capture mice conversations and other subsets of the overall traffic.

References

[1] Botnets mailing list. `http://www.whitestar.linuxbox.org/mailman/listinfo/botnets`.

[2] Botnet uses peer-to-peer control channel, May 2006. `http://www.smoothwall.net/information/news/newsitem.php?id=1013`.

[3] Abilene. `http://abilene.internet2.edu`.

[4] Arbor Networks. `http://www.arbornetworks.com`.

[5] BOLLIGER, J., AND KAUFMANN, T. Detecting Bots in Internet Relay Chat Systems. `ftp://www.tik.ee.ethz.ch/pub/students/2004-So/SA-2004-29.pdf`, 2004.

[6] CANTIENI, G., IANNACCONE, G., THIRAN, P., BARAKAT, C., AND DIOT, C. Reformulating the monitor placement problem: optimal network-wide sampling. Intel Research Technical Report, Feb. 2006.

[7] CHOI, B.-Y., AND BHATTACHARYYA, S. On the Accuracy and Overhead of Cisco Sampled NetFlow. In *Proceedings of ACM SIGMETRICS Workshop on Large Scale Network Inference (LSNI)* (June 2005).

[8] CHOI, B.-Y., PARK, J., AND ZHANG, Z.-L. Adaptive packet sampling for accurate and scalable flow measurement. In *Proceedings of IEEE GLOBECOM* (Dec. 2004), pp. 1448–1452.

[9] CLAFFY, K. C., POLYZOS, G. C., AND BRAUN, H.-W. Application of sampling methodologies to network traffic characterization. In *Proc. ACM SIGCOMM* (San Francisco, CA, Sept. 1993), pp. 194–203.

[10] COOKE, E., JAHANIAN, F., AND MCPHERSON, D. The Zombie Roundup: Understanding, Detecting, and Disrupting Botnets. In *Usenix Workshop on Steps to Reducing Unwanted Traffic on the Internet (SRUTI)* (July 2006).

[11] DAGON, D., ZOU, C., AND LEE, W. Modeling Botnet Propagation Using Time Zones. In *Proceedings of the 13th Annual Network and Distributed System Security Symposium (NDSS '06)* (2006).

[12] DUFFIELD, N., LUND, C., AND THORUP, M. Estimating flow distributions from sampled flow statistics. In *Proc. ACM SIGCOMM* (Karlsruhe, Germany, Aug. 2003), pp. 325–336.

[13] ESTAN, C., KEYS, K., MOORE, D., AND VARGHESE, G. Building a Better NetFlow. In *Proc. ACM SIGCOMM* (Portland, OR, Aug. 2004).

[14] ESTAN, C., AND VARGHESE, G. New directions in traffic measurement and accounting: Focusing on the elephants, Ignoring the mice. *ACM Transactions on Computer Systems 21*, 3 (2003), 270–313.

[15] FEAMSTER, N., AND BALAKRISHNAN, H. Detecting BGP Configuration Faults with Static Analysis. In *Proc. 2nd Symposium on Networked Systems Design and Implementation (NSDI)* (Boston, MA, May 2005), pp. 43–56.

[16] FENDLEY, S. As the bot turns. `http://isc.sans.org/diary.php?storyid=1300`, Apr. 2006.

[17] Geant. `http://www.geant.com`.

[18] HOHN, N., AND VEITCH, D. Inverting sampled traffic. In *Proceedings of ACM SIGCOMM conference on Internet measurement* (2003), pp. 222–233.

[19] HUANG, Y., AND PULLEN, J. Countering Denial of Service Attacks using Congestion Triggered Packet Sampling and Filtering. In *Proceedings of International Conference on Computer Communications and Networks* (2001), pp. 490–494.

[20] Inmon sflow. `http://www.inmon.com/technology`.

[21] Juniper traffic sampling. `http://www.juniper.net`.

[22] KARAGIANNIS, T., PAPAGIANNAKI, K., AND FALOUTSOS, M. BLINC: multilevel traffic classification in the dark. In *Proc. ACM SIGCOMM* (Philadelphia, PA, Aug. 2005), pp. 229–240.

[23] KOMPELLA, R., AND ESTAN, C. The power of slicing in internet flow measurement. In *Proc. ACM SIGCOMM Internet Measurement Conference* (Berkeley, CA, Oct. 2005).

[24] LAKHINA, A., CROVELLA, M., AND DIOT, C. Mining anomalies using traffic feature distributions. In *Proc. ACM SIGCOMM* (Philadelphia, PA, Aug. 2005), pp. 217–228.

[25] LAKHINA, A., PAPAGIANNAKI, K., CROVELLA, M., DIOT, C., KOLACZYK, E. D., AND TAFT, N. Structural analysis of network traffic flows. In *Proc. ACM SIGMETRICS* (New York, NY, June 2004), pp. 61–72.

[26] Nepenthes. `http://nepenthes.mwcollect.org`.

[27] Cisco netflow. `http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html`.

[28] RACINE, S. *Analysis of Internet Relay Chat Usage by DDoS Zombies*. PhD thesis, Swiss Federal Institute of Technology, Zurich, Apr. 2004.

[29] RAMACHANDRAN, A., AND FEAMSTER, N. Understanding the Network-Level Behavior of Spammers. In *Proc. ACM SIGCOMM* (Pisa, Italy, Sept. 2006).