On Collection of Large-Scale Multi-Purpose Datasets on Internet Backbone Links

Farnaz Moradi, Magnus Almgren, Wolfgang John, Tomas Olovsson, Philippas Tsigas
Computer Science and Engineering
Chalmers University of Technology
Göteborg, Sweden
{moradi,almgren,johnwolf,tomasol,tsigas}@chalmers.se

ABSTRACT

We have collected several large-scale datasets in a number of passive measurement projects on an Internet backbone link belonging to a national university network. The datasets have been used in different studies such as in general classification and characterization of properties of Internet traffic, in network security projects detecting and classifying malicious traffic and hosts, and in studies of network-level properties of unsolicited e-mail (spam) traffic. The *Antispam* dataset alone contains traffic between more than 10 million e-mail addresses.

In this paper we describe our datasets, the data collection methodology including experiences in collecting and processing data on a large scale. We have in particular selected a dataset belonging to an anti-spam project to show how a practical analysis of highly privacy-sensitive data can be done, in this case containing complete e-mail traffic. Not only do we show that it is possible to collect large datasets, we also show how to solve different issues regarding user privacy and give experiences from how to work with large datasets.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network Monitoring; C.2.2 [**Network Protocols**]: Applications (SMTP, FTP, etc.)

General Terms

Measurement

Keywords

Internet Measurement, Large-Scale Datasets, E-mail traffic, Spam

1. INTRODUCTION

Access to real-life large-scale datasets is in many cases crucial for understanding the true characteristics of network traffic and application behavior. The collection of large datasets from backbone Internet traffic is therefore very important for such analysis although the data collection projects in themselves face several challenges [13]. Not only is mere physical access to optical Internet backbone links needed, but also rather expensive equipment in order to deal with the large data volumes arriving at high speeds. Adding to the complexity, the collected data traces must be desensitized due to privacy issues because they may contain privacy-sensitive data. This anonymization process must be done in such a way so that a satisfactory analysis to answer the research question still can be performed, without leaking any sensitive user data. Packets also need to be reassembled into application level "conversations" so that, finally and maybe the most challenging part, methods and algorithms suitable for analysis of massive data volumes can be run. Finding these scalable methods is difficult.

We have over the years performed several data collection projects where large datasets have been gathered and analyzed. Different projects have had different goals with the data collection and for each project, unique tools have been developed and used. In this paper we describe the data collection procedure and the challenges we have faced with dealing with high-speed data collection and give examples of how data have been used in different projects. In particular, we describe a current project, the Antispam project, aiming for spam detection mechanisms on the network level where characteristics of SMTP traffic are collected and analyzed. Not only does this involve collection of vast amounts of e-mail traffic but the data collected is also highly sensitive so that automated ways to handle message privacy are essential. We also describe a method that could be deployed for analyzing the large-scale *Antispam* dataset. This method allows us to find distinguishing characteristics of legitimate and unsolicited e-mails which could be used for complementing current anti-spam tools.

The rest of this paper is organized as follows. Section 2 presents our methodology for data collection, including challenges we encountered and the solutions we deployed. Section 3 introduces the large-scale datasets collected during different years for different projects. Section 4 describes the collection of a particular dataset, the Antispam dataset, and in Section 5 we shift focus to describe the analysis of this Antispam dataset and how we compare unsolicited with legitimate e-mails. In Section 6 we present related work by comparing other sources of data collection with our collection method and resulting datasets. Finally, Section 7 con-

Badgers'11, 10-APR-2011, Salzburg, Austria

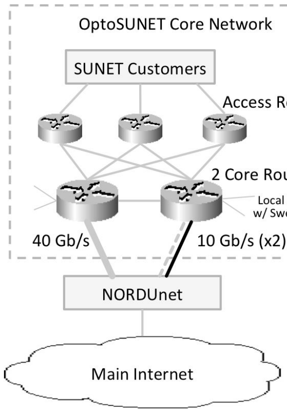

Figure 1: OptoSUNET core topology. All SUNET customers are via access routers connected to two core routers. The SUNET core routers have local peering with Swedish ISPs, and are connected to the international commodity Internet via NORDUnet. SUNET is connected to NORDUnet via three links: a 40 Gbps link and two 10 Gbps links. Our measurement equipment collects data on the first of the two 10 Gbps links (black) between SUNET and NORDUnet.

cludes the paper.

2. DATA COLLECTION METHODOLOGY

In this section, we describe the current measurement setup used to collect data, as well as some of the challenges encountered and our solutions to these. It is vital to understand the underlying data collection platform to correctly be able to use the resulting datasets. As with any experimental platform, problems do occur, but as we show below, these can sometimes be compensated for either in the collection phase or in the analysis stage.

2.1 Current Measurement Setup

We collect backbone traffic on an OC-192 (10 Gbps in each direction) link in the core-backbone of SUNET, the Swedish University Network (SUNET),[1] which not only serves as a backbone for university traffic but also for a substantial number of student dormitories, research institutes, as well as some museums and government agencies. It contains a large amount of exchange traffic with commercial companies.

Its current version, *OptoSUNET*, is a star structure over leased fiber, with a central exchange point in Stockholm. OptoSUNET connects all SUNET customers redundantly to a core network in Stockholm, as depicted in Figure 1. Traffic routed to the international commodity Internet is carried on three links between SUNET and NORDUnet, where NORDUnet peers with Tier-1 backbone providers, large CDNs (Content Distribution Networks) and other academic networks.

We use two high-end rack mounted systems (Linux) as measurement systems, one for outgoing and one for incoming traffic. At the core network in Stockholm, we apply optical splitters to tap the two OC-192 links. Each optical splitter, tapping either the inbound or outbound OC-192 link, is attached to an Endace DAG6.2SE card in one of the measurement nodes. The cards are capable of collecting data on PoS and 10 Gbit-Ethernet links with bandwidths of up to 10 Gbps. We usually collect network data simultaneously for both directions.

Depending on the project (see Table 1) and its goal with the data collection, we then perform some pre-processing of the raw data before transferring them for further analysis and storage at the processing platform at Chalmers University. This pre-processing ranges from *anonymization* (see Section 2.2.3) of the data (all projects), to spam categorization (the *Antispam* project). The experimental infrastructure is further described in [7].

2.2 Challenges and Solutions

We categorized the problems we encountered and our solutions in three clusters, *general problems* relating to the setup of the system, problems related to the *collection process*, and finally problems related to the *pre-processing of the dataset* before the final analysis and storage.

2.2.1 General Challenges

One of the most difficult problems we faced was actually not of a technical nature, but involved gaining access to the network infrastructure in the first place. Fortunately, there is a long tradition of work between our department and SUNET, so there was already a basis for trust. Furthermore, we also consulted an ethical vetting board, and based on their feedback we could proceed with the measurements. However, as will be described below, the required *anonymization of user data* is very important and permeates many of our decisions on what kind of data we can collect and how it can be analyzed.

A more technical problem involved the equipment. At the time of purchase (2004), we faced problems with finding systems with enough internal bus bandwidth to cope with full link-speed data collection. Captured data should be received by the network card, be moved to main memory, and then be written to disk in speeds up to about 1 GB/s. The used high-end RAID system with six striped disks offered around 0.4 GB/s disk throughput, which turned out to suffice due to the large over-provisioning of the 10 Gbps link by SUNET. The network architecture changed somewhat during 2007 when one (parallel) 40 Gbps and one 10 Gbps link were added. Unfortunately, equipment for collecting data from all links was too expensive to acquire for our projects (each direction would require 5 times as much hardware).

Finally, there are limitations in using real-life datasets, that are not specific to our datasets. To mention one is that the measurements give us snapshots of traffic from a single vantage point during a limited time period. The results should thus be extended with similar data from other times and locations.

2.2.2 Collection Challenges

Regarding the collection phase of large datasets, the first problem we must cope with is the sheer volume of traffic.

[1] http://www.sunet.se/

At heavily loaded links, data may arrive with up to 1 GB/s. Even if this theoretical maximum is rarely reached due to over-provisioning of the links, different data reduction methods must be applied to further decrease the load on the measurement nodes. However, these methods must not influence the real-time nature of the traffic capture, i.e. we must be able to keep up with the traffic. We partially solved this by having very well-defined experimental plans with clear goals of exactly what we should capture to do our analysis.

For example, for datasets spanning long time intervals, we only capture flow summaries instead of individual packets, while we collect short snapshots (10 or 20 minutes) of packet traces to investigate protocol properties. To allow for a more dynamic traffic capture, we currently investigate real-time computations on the DAG cards so that decisions can be taken in real time based on more complex traffic properties.

Moreover, any dataset we collect and analyze should be safely stored so that others can repeat the analysis or compare results. This poses additional archiving requirements on the measurement system.

To ensure sanity of the resulting data, we apply consistency and sanity checks immediately after collection, allowing us to document both measurement related problems (e.g. measurement card failures) and network-related anomalies (e.g. network maintenance tasks by SUNET). These include inspection of time stamps, frame types, packet header information, etc. With these sanity checks we can improve the system in the longer term, but they can also be applied during the analysis phase to explain certain traffic behavior. For sanity checking, we use existing tools such as CAIDA's CoralReef [15] and Endace's dagtools. Additionally, we developed our own software and modified publicly available software to suit our needs. The use of our own methods and programs requires substantial effort, but gives us complete control of the quality of the data.

As an example of trace insanity, we have experienced some cases of completely garbled data, most likely occurring due to hardware problems in the DAG cards loosing framing synchronization. These traces have been discarded immediately. To reduce this problem we now restart the cards in regular intervals, which in turn may lead to some missing packets in the second between such data collection periods. We are currently installing new equipment, including new DAG cards, which should eventually solve this issue.

During normal operation, we have not detected any packet loss. However, during the collection of one of the datasets, the Malbone dataset, there have been a few short, but immense traffic surges, where traffic was increasing from the normal rate of $<200k$ to $>400k$ packets per second. During these surges, our nodes could not keep up with the speed and dropped packets, which was logged by the measurement cards. As the information was logged, it was easy to accommodate in the analysis stage. We have also detected some minor errors with the IP header checksums (1 out of 300 million frames) and 1 out of 191 million frames were discarded due to receiver errors.

Finally, the measurements are done over an operational large network, meaning that parameters change over the course of the data collection, both in a longer perspective with planned upgrades as well as with transient failures of certain equipment.

2.2.3 *Pre-processing Challenges*

As can be seen from Figure 1, we collect data from one link (out of three) with two separate systems to collect traffic in two directions, meaning that we have two datasets with unidirectional traffic traces. The traffic is load balanced between the links and, according to SNMP statistics collected during the last measurement campaign, we see around one third of all incoming traffic and 15% of all outbound traffic. This effect introduces an observed routing asymmetry, as we can sometimes only see the traffic going in one direction of a TCP connection [8].

Assembling bidirectional TCP flows requires very good time synchronization between the two systems. During measurements, the DAG cards are thus synchronized with each other using DUCK Time Synchronization [4], allowing synchronization within $\pm 30ns$, which suffices for trace assembly.

A key processing step is also desensitization of the data, i.e. removing any privacy-sensitive information. Besides our responsibilities as ethical researchers, this is also one of the requirements of the ethical vetting board (see Section 2.2.1). As a basic step, we discard sensitive payload and anonymize IP addresses in the resulting trace based on the prefix-preserving CryptoPAN [21]. Throughout all measurements campaigns we use a single, unique encryption-key to allow us to track specific hosts and IP ranges between all measurements. Unfortunately, this anonymization step is not without penalty but may influence the type of analysis method we can use as well as restricting the refinement of the result. As an example, even if we find a very aggressive host spreading malware within SUNET, we cannot inform the owner due to the anonymization of the data. Furthermore, there still exists challenges to improve the anonymization before datasets potentially could be shared to others on a large scale. Other desensitization steps performed in projects include payload removal (MonNet), and e-mail anonymization (Antispam), which is detailed further in Section 4.3.

2.2.4 *Summary*

To summarize, our analysis is network centric and does not consider the end hosts in detail. Overall, it was not a trivial effort to initially setup the data collection platform but it took both time (years) and effort. As with any experimentally collected data, it is important to understand the limitations of the experiment setup for correct analysis of the data. Some problems, given their careful documentation in the collection phase, can be accommodated for in the analysis stage.

3. DATASETS

We have collected several large-scale datasets on the Internet backbone links. The first traces were collected in 2005 and we still have active projects collecting data from the links. The datasets differ in the information they contain, reflecting the type of research question we want to investigate. As we detailed in Section 2, we simply cannot collect

Table 1: Datasets Overview

Dataset	Location	Collection Period	Collection Duration	Number of traces	Number of Packets $/10^9$
MonNet I	GigaSUNET	2006-04	20 minutes	74	10.8
MonNet II	GigaSUNET	2006-09 to 2006-11	10 minutes	277	27.9
MonNet III	OptoSUNET	2008-12 to 2009-11	10 minutes	151	33.0
Malbone	OptoSUNET	2010-03 to 2010-10	24 hours	34	12 (flows)
Antispam	OptoSUNET	2010-03	24 hours	14	0.8 (SMTP only)

"all" information but need to reduce it both in regard to storage as well as for anonymization purposes.

The datasets are summarized in Table 1. Below we briefly describe the characteristics of each dataset. We then describe the collection and use of the *Antispam* dataset in detail as a case study.

3.1 The MonNet Datasets

The largest datasets until today were collected within the MonNet project to classify and understand the characteristics of Internet traffic and to see how it changes over time. Later analysis of this dataset also included finding malicious traffic in order to see how and to what extent protocols are abused.

The MonNet datasets represent 95 hours of backbone traffic, collected on 156 different days mainly during 2006 and 2009. The first set of data was collected during 80 days from September to November 2006 as 277 randomly selected 10-minute snapshots. When recording these traces, payload beyond transport layer was removed. About 27.9 billion IPv4 frames containing around 480 million flows were collected and analyzed. The size of the dataset was almost 20 TB in size (headers only). A second (slightly larger) dataset was collected during 2009 where 33 billion packets were collected. Traffic analysis from this data set reveals inbound traffic from $2,270,000$ distinct IP addresses to $360,000$ unique internal addresses.

The datasets collected in the MonNet project have been studied in detail. Initial studies investigated protocol features of packet headers [10] and packet header anomalies in order to discuss potential security problems, such as incorrect use of IP fragmentation [9]. Additional flow-level analysis of the MonNet data allowed investigation of trends and changes in connection behavior of Internet traffic over time, e.g. how the popularity of p2p traffic has caused a change in Internet traffic patterns in the last few years [11, 12, 22].

3.2 The Malbone Dataset

The objective of the Malbone project is to measure and understand larger communication patterns among hosts over a longer time period. This may include normal as well as more malicious behavior.

For more than six months, a 24h snapshot of all flows was regularly collected once a week. The dataset contains a total of 12 billion flows for both directions. In Table 2, we have summarized all unique IPs we found during a single collection day to give an idea of the scale of the traffic passing by the measuring point.

This dataset also contains metadata, including, for example, hosts known to aggressively spread malware at the time of the collected snapshots. By using the flow data together with this information, we can then make more targeted types of analysis of hosts, despite their addresses being anonymized.

The analysis of this dataset is still in its infancy, but some results documenting malicious behavior of scanning hosts has been published as well as particulars of the timing behavior of hosts. [1].

3.3 The Antispam Dataset

In the Antispam project SMTP traffic was collected to permit the study of the differences in traffic characteristics between spam and legitimate traffic. The goal is to find methods for early detection of spamming nodes on the Internet as close to the source as possible. This method should be an alternative to spam removal in the receiving hosts. There is a clear need for moving the defense against spam as close to the spammers as possible, in order to reduce problems such as the amount of unwanted traffic and waste of mail server resources.

Within this project, during 14 days in March 2010, more than 797 million SMTP packets (filtered on TCP port 25 in the hardware) were passively captured. More than 627 million packets were incoming packets to SUNET and the rest were outgoing. We aggregated these packets into 34.9 million incoming and 11.9 outgoing SMTP flows. The captured flows on the incoming direction were originating from $2,300,660$ distinct IP addresses and were destined to $569,591$ internal distinct IP addresses. The outgoing flows were sent from $10,795$ to $1,943,919$ distinct IP addresses.

The main challenges in this project relate to the highly privacy-sensitive data as well as how to analyze the characteristics of this type of traffic on a large scale. This project is described more in detail in Section 4.

4. ANTISPAM DATASET COLLECTION

In this section we use the dataset *Antispam* as a case study to concretely illustrate the collection and analysis of a large-scale dataset. Such an e-mail dataset can be studied for better understanding of the behavior of spam traffic, often a means to propagate malicious content. Research such as [19] has suggested that spam mainly originates from botnets. These botnets are also most likely active in other malicious activities on the Internet. Therefore, detecting spam close to its source instead of just discarding it by the receivers can also lead to detection of other malicious traffic from the same origin.

4.1 SMTP Data Collection

In order to analyze characteristics of e-mail traffic, SMTP packets were passively collected during two consecutive weeks of measurements in March 2010.

Table 2: Unique hosts during the data collection 2010-04-01

	Inside SUNET		Outside SUNET	
Incoming Link	Destination IPs	970,149	Source IPs	24,587,096
Outgoing Link	Source IPs	23,600	Destination IPs	18,780,894

To overcome the storage problem described in Section 2.2.2, we used a hardware filter to only capture traffic to and from *port 25* using the *crl_to_dag* utility of CoralReef [15]. This still resulted in more than 183 GB of SMTP data, divided into two unidirectional datasets (see Section 2.2.3).

The captured packets belonging to a single flow were then aggregated to allow the analysis of complete SMTP sessions. To reconstruct the sequence of packets into flows, we used the tcpflow program,[2] which understands sequence numbers and correctly compensates for problems such as out-of-order packets and retransmitted packets.

The collected data contains both TCP flows with destination port 25 (*SMTP request*) and TCP flows with source port 25 (*SMTP reply*). As each *SMTP request* flow corresponds to an SMTP session, it can carry one or more e-mails; thus we had to extract each e-mail from the flows by examining the SMTP commands. The resulting extracted e-mail transaction contains the (1) SMTP commands containing the e-mail addresses of the sender and the receiver(s), (2) e-mail headers, and (3) the e-mail content. Each *SMTP reply* contains the corresponding response code to an SMTP request command, and by also including these in the analysis one can gain a better insight into the behavior of the receiving mail servers.

4.2 E-mail Classification

After the collection phase, (1) the dataset is pruned of all unusable e-mail traces, (2) the remaining e-mail transactions are classified into either being *accepted* or *rejected*, and finally (3) the e-mails in the *accepted* category are refined into either being *spam* or *ham*. These three steps are described in detail below.

Before any classification, we begin by discarding all unusable traces. For example, flows with no payload are mainly scanning attempts and should not be considered in the classification. Also, SMTP flows missing the proper commands are excluded from the dataset as they most likely belong to other applications using port 25. Encrypted e-mail communications cannot be analyzed, and were also eliminated.[3] Any e-mail with an empty sender address is a notification message, such as a non-delivery message [14]; it does not represent a real e-mail transmission and is also excluded. Finally, any e-mail transaction that is missing either the proper starting/ending or any intermediate packet is considered as incomplete and one might decide to leave out these e-mails when analyzing the dataset. Possible reasons for having incomplete flows include transmission errors and measurement hardware limitations caused by the framing synchronization problem (Section 2.2.3).

The remaining e-mail transactions are then classified as *accepted*, i.e. those e-mails that are delivered by the mail servers, or *rejected*. An e-mail transaction can fail at any time before the transmission of the e-mail data (header and content) due to rejection by the receiving mail server. Therefore, *rejected* e-mails are those that do not finish the SMTP command exchange phase and consequently do not send any e-mail data. The rejections are mostly because of spam prefiltering strategies deployed by mail servers including blacklisting, greylisting, DNS lookups, and user database checks.

Examining SMTP replies sent by the receiving mail servers has no effect on the classification of accepted e-mails. However, they could have been consulted for finding the reasons of e-mail rejections. In our dataset, due to asymmetric routing (see Section 2.2.3) only approximately 10% of the flows are symmetric, where both the e-mail and the corresponding mail server reply are available in the collected traces. Therefore, we have decided to not further classify the rejected e-mail transactions. However, existing responses can always be queried if required in the analysis.

Finally, we discriminate between *spam* and *ham* in our dataset. As we have captured the complete SMTP flows, including IP addresses, SMTP commands, and e-mail contents, we can establish a ground truth for further analysis of *only* the spam traffic properties and a comparison with the corresponding legitimate e-mail traffic. We deploy the widely-used spam detection tool called SpamAssassin[4] to mark e-mails as spam and ham. SpamAssassin uses a variety of techniques for its classification, such as header and content analysis, Bayesian filtering, DNS blocklists, and collaborative filtering databases.[5]

We would like to stress that these classification steps are carried out automatically after the data collection. As we describe in the next section, the contents of the e-mails are then discarded and all other user data desensitized before we can manually analyze the dataset.

4.3 Anonymization

The final pre-processing step of the Antispam dataset is to desensitize any user data. As mentioned in Section 2.2.3, any real data collection is in general privacy sensitive and large scale e-mail collection even more so. For that reason, we complete the pre-processing with a complete anonymization step. For the anti-spam project where we study traffic characteristics of ham versus spam traffic, we actually have little use of the full contents of the e-mails after they have been properly labeled. On the contrary, given that we reduce the size of the dataset significantly by throwing away user data, it actually gets easier for us to process and store.

[2] `http://www.circlemud.org/~jelson/software/tcpflow/`

[3] Around 3.8% of the flows carried encrypted SMTP sessions.

[4] `http://spamassassin.apache.org`

[5] The well-trained SpamAssassin applied to our dataset was in use for a long time at our university, incurring an approximate false positive rate of less than 0.1%, and an detection rate of 91.3% after around 94% of the spam being rejected by blacklists.

Immediately after the classification, we started by discarding the body of the e-mails as well as the subject of the e-mail and the names of the sender and receiver(s). The IP addresses in the packet headers and payload are anonymized in a prefix-preserving fashion using CryptoPAN [21], similarly to all of our other projects.

Finally, we are left with the sensitive data carried in the SMTP requests and replies, namely e-mail addresses and host/domain names. These form a structure of the underlying communication pattern and cannot simply be discarded but should instead be anonymized. We have introduced the following approach for performing domain-preserving anonymization:

- First, each e-mail address is divided into the user name and the domain name (i.e. user@domain).
- The user name is local to each domain and is simply hashed using a secure hash function.[6]
- The domain name, consisting of one or more dot-separated components, is split into its parts, and a secure hash function is applied separately to each component.
- The outputs of the hash function is then re-encoded into printable ASCII characters.
- Finally, the hashed items are appended to each other to form an *anonymized* e-mail address or domain name. This anonymized name then replaces the original one in the dataset.

Hashing each domain name component individually allows us to generate domain preserving anonymized addresses and names. This gives us the possibility to study the behavior of e-mail traffic originating from the same domain and to compare them with traffic from other domains.

Once the sensitive data was discarded, the resulting anonymized dataset had a size of 37 GB.

4.4 Summary

The anti-spam dataset was collected in a similar fashion to the other datasets (Section 2.1). However, as the collection also included packet payloads, this dataset required a more complete pre-processing step before any manual analysis could be performed. Automatic extraction of e-mail transactions from SMTP sessions, classification of the e-mails, extracting followed by discarding the e-mail bodies, finding and replacing all IP, e-mail addresses, and host/domain names inside the headers with a corresponding anonymized version, etc. are just a number of challenges associated with the collection of this type of traffic that we had to overcome.

5. ANTISPAM DATASET ANALYSIS

In the previous sections we described the necessary automatic pre-processing of the Antispam dataset before the analysis could start. In this section we change focus and present our analysis methodology of the dataset.

As we stated before, the goals of the Antispam project is to study the statistical characteristics of e-mail traffic and finding the distinguishing properties of spam and legitimate e-mails. Understanding these properties is necessary for the development of new spam detection mechanisms to detect spam already on the network level as close to its source as possible. In this section, we present some overall statistical properties of the collected e-mail traffic, and briefly describe an approach to spam mitigation we have developed.

Table 3: Antispam dataset statistics

	Incoming ($/10^6$)	Outgoing ($/10^6$)
Packets	626.9	170.1
Flows	34.9	11.9
Distinct srcIPs	2.30	0.01
Distinct dstIPs	0.57	1.94
SMTP Replies	2.84	9.14
E-mails	23.5	0.90
Ham	1.15	0.19
Spam	1.43	0.16
Rejected	17.3	0.35
Unusable	3.64	0.20

5.1 Overall E-mail Traffic Characteristics

After the exclusion of unusable flows described in Section 4.2, we ended up with 24.4 million e-mails and approximately 12 million SMTP replies. The e-mails contained $10,544,647$ distinct e-mail addresses in the SMTP headers from $532,825$ distinct domains. The unusable e-mails were then discarded.

After e-mail classification, more than 17.6 million e-mails in our dataset were classified as *rejected* and only around 2.6 million incoming and 350 thousand outgoing e-mails were classified as *accepted*. This observation is similar to what was observed in [20] where the logs of a university mail server was analyzed. In this study more than 78% of the SMTP sessions were rejected by pre-acceptance strategies deployed by the mail server to filter out spamming attempts. Table 3 shows the dataset statistics for our e-mail data captured in each direction.

5.2 E-mail Analysis for Spam Mitigation

One approach to spam detection is to conduct a social network based analysis of e-mail communication. This approach was first proposed in [3] and has since then gained a large interest. In such analysis, an e-mail network based on e-mail communication is generated and then graph-theoretical analysis is applied. By using e-mail addresses as nodes and letting edges symbolize any e-mail exchange, an e-mail network captures the social interactions between e-mail senders and receivers. Even though our dataset has been anonymized, we can still generate an equivalent e-mail network to the originally collected traffic due to the properties of the anonymization process. In [17] we study the structural properties of such a network generated from one week of traffic.

Any type of analysis of large datasets is challenging from both a memory and computational time requirement perspective, but we also faced some additional challenges in our graph-theoretical analysis. Many of the standard graph-theoretical functions used for analysis of graph structures are very computationally expensive. For instance, the calculation of the average shortest path length between all the nodes in the network (a measure of the graph connectivity) is computationally prohibitive for larger graphs. One

[6] The secure hash function is a one-way function, which takes a secret cryptographic key as input.

method to reduce the complexity is to use *sampling*, but the interpretation of the results must then be done with caution [2].

The generated e-mail network from two weeks contains $10,544,647$ nodes and $21,537,314$ edges. To the best of our knowledge this is the largest e-mail dataset that has been used for studying the characteristics of e-mail networks. We used the `networkx`[7] package in python to create and analyze the structure of the constructed e-mail network. This package tries to load the whole graph into main memory to increase the performance. However, loading the complete graph based on two weeks of e-mail traffic was not possible, despite the fact that our processing machine has 16 GB of memory. In order to reduce the required memory we used methods such as mapping e-mail addresses to integer labels.

We also built more specific e-mail networks based on a subset of the data according to the e-mail classification into the described categories of *rejected*, *accepted/ham*, and *accepted/spam*. For example, a *spam e-mail network* is an e-mail network containing only the e-mail addresses sending and receiving spam as nodes with the edges representing any spam communication. By comparing the generated e-mail networks, many structural differences are revealed between networks built from legitimate e-mails and unsolicited traffic. A remarkable observation from our study [17] is that the structure of a ham network exhibits similar properties to that of online social networks, Internet topology, or the World Wide Web. A spam network, on the contrary, has a different structure as well as a rejected traffic network. This does in turn, given the large number of spam e-mails, affect the structural properties of the *complete* e-mail network. Our observations suggest that these distinguishing properties could potentially be exploited for detection of spamming nodes on the network level.

Our research so far has thus led to two important findings. First, we have observed differences in the characteristics of spam and ham traffic, which could lead to spam detection methods complementing current antispam tools. The acquired knowledge from our analysis of the data also provides us with the means to produce realistic models of e-mail traffic. These models could in turn be used to generate synthetic datasets as an alternative to the costly collection and challenging distribution of the large-scale original data.

6. RELATED WORK

In this section existing sources of data collection that can be deployed for performing security-related research are introduced and compared with our collection methodology.

To study malicious traffic, methods such as *distributed sensors*, *honeypot networks*, *network telescopes/darknets*, as well as *passive measurements* can be deployed for data collection. Network telescopes monitor large, unused IP address spaces (darkspaces) on the Internet[16], and are typically only traffic sinks which attract unsolicited traffic without responding to them. Distributed sensors are usually placed at diverse geographical and logical network locations by some companies including antivirus companies, allowing them to summarize wide-area trends by correlating sensor data. However, they introduce a serious bias, as the users obviously care about security. Networks of honeypots collect a large aggregation of traffic behavior from dedicated, unprotected but well monitored hosts, but passive honeypots are not very suitable for analysis of normal user responses.

Our approach, passive measurements on large-scale links, is generally viewed as the best way to study Internet traffic, as it includes real behavioral responses from a diverse user population.

Research attempts to characterize and analyze spam have used a wide range of different datasets, such as data extracted from *users' mailboxes*, *mail server log files*, *sinkholes*, and *network flows*.

Collecting sent and received e-mail headers in one user's mailbox is used in [3], but this collection methodology does not scale and any such dataset is limited to an individual user. Mail server SMTP log files, on the other hand, contain information about more users but are usually limited to incoming e-mails to a single domain. Such datasets have been used, for example by Gomes et al. [6] where eight days of SMTP log files of incoming e-mails to a university mail server was used after a pre-filtering phase and categorization by SpamAssassin.

Spam collected at sinkholes (honeypots) are usually not restricted to a single domain, as these can either just receive spam passively, or imitate an open relay which spammers can exploit to relay spam. However, as described above, sinkholes, does not include the normal user's behavior and do not provide the possibility of comparing characteristics of spam and ham. Ramachandran and Feamster [19] collected spam e-mails from two sinkholes and complemented their traces with other sources of data such as external log files of legitimate e-mails, BGP routing information, IP blacklists, etc. Pathak et al. [18] collected spam during three months from an open relay sinkhole together with information about the sending host such as TCP fingerprints, IP blacklists, etc.

Collection of flow-level data at gateway routers can lead to very large datasets; however, no ground truth and limited possibility of validating the findings are its main shortcomings. Schatzmann et al. [20] have studied NetFlow data captured during 3 months at the border router of a national ISP, and complemented their dataset with the log of a university mail server to discriminate between rejected spam and ham flows. Ehrlich et al. [5] have collected large network flow datasets from a router connecting their network to other ISPs and used local IP blacklists and whitelists to distinguish spam from ham.

Our Antispam dataset, which was passively collected on an Internet backbone link, is not limited to a single user or domain. Not only does it give us the possibility of studying the flow-level characteristics of e-mail traffic, but it also shows which flows carry spam or ham traffic, a property which is difficult to accurately determine without consulting the e-mail content.

[7] `http://networkx.lanl.gov/`

7. CONCLUSIONS

We have described a number of large-scale datasets collected on a high-speed backbone link. The datasets have been far from trivial to collect, and for that reason we shared the challenges we faced as well as our solutions for processing the large-scale data.

To exemplify the analysis process, we used the *Antispam* dataset to concretely discuss the collection and analysis of a large-scale dataset. This included our methodology for anonymization, i.e. the removal of any user-sensitive data in such a way that also allowed accurate traffic analysis, as well as a discussion of applying graph-theoretical thechniques to the generated e-mail network. To the best of our knowledge, this e-mail network is the largest that has been used to study the characteristics of such networks. We could find clear differences in the communication patterns of spam and ham traffic, something that we suggest can be used to both discriminate between them on the network level and to create more complete simulation models.

The described type of data collection is necessary for such analysis since most other contemporary data collection approaches either lack participants' e-mail addresses or do not have any legitimate traffic.

We believe that the collection of large-scale datasets such as the datasets presented in this paper is crucial for understanding the behavior of the Internet and its applications. Security research in particular needs contemporary Internet traffic in order to show the usefulness and correctness of security mechanisms and algorithms.

8. ACKNOWLEDGMENTS

This work was supported by .SE – The Internet Infrastructure Foundation, SUNET, and the Swedish Civil Contingencies Agency (MSB). The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257007.

9. REFERENCES

[1] M. Almgren and W. John. Tracking malicious hosts on a 10gbps backbone link. In *15th Nordic Conference in Secure IT Systems (NordSec 2010)*, 2010.

[2] P. R. V. Boas, F. A. Rodrigues, G. Travieso, and L. da F Costa. Sensitivity of complex networks measurements. *Statistical Mechanics*, 2010.

[3] P. O. Boykin and V. P. Roychowdhury. Leveraging social networks to fight spam. *Computer*, 38(4), 2005.

[4] S. Donnelly. Endace dag timestamping whitepaper, `endace,http://www.endace.com/`, 2007.

[5] W. K. Ehrlich, A. Karasaridis, D. Liu, and D. Hoeflin. Detection of spam hosts and spam bots using network flow traffic modeling. In *Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*, LEET'10, pages 7–7, 2010.

[6] L. H. Gomes, C. Cazita, J. M. Almeida, V. Almeida, and W. Meira, Jr. Workload models of spam and legitimate e-mails. *Perform. Eval.*, 64(7-8), 2007.

[7] W. John. *Characterization and Classification of Internet Backbone Traffic*. PhD dissertation, Chalmers University of Technology, 2010.

[8] W. John, M. Dusi, and k. claffy. Estimating routing symmetry on single links by passive flow measurements. In *Proc. of the Wireless Communications and Mobile Computing Conference*, 2010.

[9] W. John and T. Olovsson. Detection of malicious traffic on backbone links via packet header analysis. *Campus-Wide Information Systems*, 25(5), 2008.

[10] W. John and S. Tafvelin. Analysis of internet backbone traffic and header anomalies observed. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 111–116, 2007.

[11] W. John and S. Tafvelin. Differences between in- and outbound Internet Backbone Traffic. In *TERENA Networking Conference (TNC)*, 2007.

[12] W. John, S. Tafvelin, and T. Olovsson. Trends and Differences in Connection-behavior within Classes of Internet Backbone Traffic. In *Passive/Active Measurement (PAM)*, 2008.

[13] W. John, S. Tafvelin, and T. Olovsson. Passive Internet measurement: Overview and guidelines based on experiences. *Computer Communications*, 33(5), 2010.

[14] J. Klensin. Simple Mail Transfer Protocol. RFC 5321 `http://www.ietf.org/rfc/rfc5321.txt`, 2008.

[15] D. Moore, K. Keys, R. Koga, E. Lagache, and k. claffy. The CoralReef Software Suite as a Tool for System and Network Administrators. In *USENIX LISA*, 2001.

[16] D. Moore, C. Shannon, G. Voelker, and S. Savage. Network Telescopes. Tech.rep., CAIDA, 2004.

[17] F. Moradi, T. Olovsson, and P. Tsigas. Analyzing the social structure and dynamics of e-mail and spam in massive backbone internet traffic. Technical report, Chalmers University of Technology, no. 2010-03, 2010.

[18] A. Pathak, Y. C. Hu, and Z. M. Mao. Peeking into spammer behavior from a unique vantage point. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, pages 3:1–3:9, 2008.

[19] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. *SIGCOMM*, 36(4), 2006.

[20] D. Schatzmann, M. Burkhart, and T. Spyropoulos. Inferring spammers in the network core. In *Passive and Active Measurement Conference*, 2009.

[21] J. Xu, J. Fan, M. Ammar, and S. B. Moon. On the design and performance of prefix-preserving ip traffic trace anonymization. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, IMW '01, pages 263–266, New York, NY, USA, 2001. ACM.

[22] M. Zhang, M. Dusi, W. John, and C. Chen. Analysis of udp traffic usage on internet backbone links. In *Proceedings of the 2009 Ninth Annual International Symposium on Applications and the Internet*, pages 280–281, 2009.