Detecting Spammers with SNARE: Spatio-temporal Network-level Automatic Reputation Engine

Shuang Hao, Nadeem Ahmed Syed, Nick Feamster, Alexander G. Gray, Sven Krasser *
College of Computing, Georgia Tech *McAfee, Inc.
{shao, nadeem, feamster, agray}@cc.gatech.edu, sven_krasser@mcafee.com

Abstract

Users and network administrators need ways to filter email messages based primarily on the reputation of the sender. Unfortunately, conventional mechanisms for sender reputation—notably, IP blacklists—are cumbersome to maintain and evadable. This paper investigates ways to infer the reputation of an email sender based solely on network-level features, without looking at the contents of a message. First, we study first-order properties of network-level features that may help distinguish spammers from legitimate senders. We examine features that can be ascertained without ever looking at a packet's contents, such as the distance in IP space to other email senders or the geographic distance between sender and receiver. We derive features that are *lightweight*, since they do not require seeing a large amount of email from a single IP address and can be gleaned without looking at an email's contents—many such features are apparent from even a single packet. Second, we incorporate these features into a classification algorithm and evaluate the classifier's ability to automatically classify email senders as spammers or legitimate senders. We build an *automated* reputation engine, *SNARE*, based on these features using labeled data from a deployed commercial spam-filtering system. We demonstrate that *SNARE* can achieve comparable accuracy to existing static IP blacklists: about a 70% detection rate for less than a 0.3% false positive rate. Third, we show how *SNARE* can be integrated into existing blacklists, essentially as a first-pass filter.

1 Introduction

Spam filtering systems use two mechanisms to filter spam: content filters, which classify messages based on the contents of a message; and sender reputation, which maintains information about the IP address of a sender as an input to filtering. Content filters (e.g., [22, 23]) can block certain types of unwanted email messages, but they can be brittle and evadable, and they require analyzing the contents of email messages, which can be expensive. Hence, spam filters also rely on *sender reputation* to filter messages; the idea is that a mail server may be able to reject a message purely based on the reputation of the sender, rather than the message contents. DNS-based blacklists (DNSBLs) such as Spamhaus [7] maintain lists of IP addresses that are known to send spam. Unfortunately, these blacklists can be both incomplete and slow-to-respond to new spammers [32]. This unresponsiveness will only become more serious as both botnets and BGP route hijacking make it easier for spammers to dynamically obtain new, unlisted IP addresses [33, 34]. Indeed, network administrators are still searching for spam-filtering mechanisms that are both *lightweight* (i.e., they do not require detailed message or content analysis) and *automated* (i.e., they do not require manual update, inspection, or verification).

Towards this goal, this paper presents *SNARE* (Spatio-temporal Network-level Automatic Reputation Engine), a sender reputation engine that can accurately and automatically classify email senders based on lightweight, network-level features that can be determined early in a sender's history—sometimes even upon seeing only a single packet. *SNARE* relies on the intuition that about 95% of all email is spam, and, of this, $75 - 95\%$ can be attributed to botnets, which often exhibit unusual sending patterns that differ from those of legitimate email senders. *SNARE* classifies senders based on *how* they are sending messages (i.e., traffic patterns), rather than *who* the senders are (i.e., their IP addresses). In other words, *SNARE* rests on the assumption that there are lightweight network-level features that can differentiate spammers from legitimate senders; this paper finds such features and uses them to build a system for automatically determining an email sender's reputation.

SNARE bears some similarity to other approaches that classify senders based on network-level behavior [12, 21,

24, 27, 34], but these approaches rely on inspecting the message contents, gathering information across a large number of recipients, or both. In contrast, *SNARE* is based on *lightweight* network-level features, which could allow it to scale better and also to operate on higher traffic rates. In addition, *SNARE* is *more accurate* than previous reputation systems that use network-level behavioral features to classify senders: for example, *SNARE*'s false positive rate is an order of magnitude less than that in our previous work [34] for a similar detection rate. It is the first reputation system that is both as accurate as existing static IP blacklists and automated to keep up with changing sender behavior.

Despite the advantages of automatically inferring sender reputation based on "network-level" features, a major hurdle remains: We must identify *which features* effectively and efficiently distinguish spammers from legitimate senders. Given the massive space of possible features, finding a collection of features that classifies senders with both low false positive and low false negative rates is challenging. This paper identifies thirteen such network-level features that require varying levels of information about senders' history.

Different features impose different levels of overhead. Thus, we begin by evaluating features that can be computed purely locally at the receiver, with no information from other receivers, no previous sending history, and no inspection of the message itself. We found several features that fall into this category are surprisingly effective for classifying senders, including: The AS of the sender, the geographic distance between the IP address of the sender and that of the receiver, the density of email senders in the surrounding IP address space, and the time of day the message was sent. We also looked at various aggregate statistics across messages and receivers (e.g., the mean and standard deviations of messages sent from a single IP address) and found that, while these features require slightly more computation and message overhead, they do help distinguish spammers from legitimate senders as well. After identifying these features, we analyze the relative importance of these features and incorporate them into an automated reputation engine, based on the *RuleFit* [19] ensemble learning algorithm.

In addition to presenting the first automated classifier based on network-level features, this paper presents several additional contributions. First, we presented a detailed study of various network-level characteristics of both spammers and legitimate senders, a detailed study of how well each feature distinguishes spammers from legitimate senders, and explanations of why these features are likely to exhibit differences between spammers and legitimate senders. Second, we use state-of-the-art ensemble learning techniques to build a classifier using these features. Our results show that *SNARE*'s performance is at least as good as static DNS-based blacklists, achieving a 70% detection rate for about a 0.2% false positive rate. Using features extracted from a single message and aggregates of these features provides slight improvements, and adding an AS "whitelist" of the ASes that host the most commonly misclassified senders reduces the false positive rate to 0.14%. This accuracy is roughly equivalent to that of existing static IP blacklists like SpamHaus [7]; the advantage, however, is that *SNARE* is *automated*, and it characterizes a sender based on its sending *behavior*, rather than its IP address, which may change due to dynamic addressing, newly compromised hosts, or route hijacks. Although *SNARE*'s performance is still not perfect, we believe that the benefits are clear: Unlike other email sender reputation systems, *SNARE* is both automated and lightweight enough to operate solely on network-level information. Third, we provide a deployment scenario for *SNARE*. Even if others do not deploy *SNARE*'s algorithms exactly as we have described, we believe that the collection of network-level features themselves may provide useful inputs to other commercial and open-source spam filtering appliances.

The rest of this paper is organized as follows. Section 2 presents background on existing sender reputation systems and a possible deployment scenario for *SNARE* and introduces the ensemble learning algorithm. Section 3 describes the network-level behavioral properties of email senders and measures first-order statistics related to these features concerning both spammers and legitimate senders. Section 4 evaluates *SNARE*'s performance using different feature subsets, ranging from those that can be determined from a single packet to those that require some amount of history. We investigate the potential to incorporate the classifier into a spam-filtering system in Section 5. Section 6 discusses evasion and other limitations, Section 7 describes related work, and Section 8 concludes.

2 Background

In this section, we provide background on existing sender reputation mechanisms, present motivation for improved sender reputation mechanisms (we survey other related work in Section 7), and describe a classification algorithm called *RuleFit* to build the reputation engine. We also describe McAfee's TrustedSource system, which is both the source of the data used for our analysis and a possible deployment scenario for *SNARE*.

2.1 Email Sender Reputation Systems

Today's spam filters look up IP addresses in DNS-based blacklists (DNSBLs) to determine whether an IP address is a known source of spam at the time

of lookup. One commonly used public blacklist is Spamhaus [7]; other blacklist operators include SpamCop [6] and SORBS [5]. Current blacklists have three main shortcomings. First, they only provide reputation at the granularity of IP addresses. Unfortunately, as our earlier work observed [34], IP addresses of senders are dynamic: roughly 10% of spam senders on any given day have not been previously observed. This study also observed that many spamming IP addresses will go inactive for several weeks, presumably until they are removed from IP blacklists. This dynamism makes maintaining responsive IP blacklists a manual, tedious, and inaccurate process; they are also often coarse-grained, blacklisting entire prefixes—sometimes too aggressively—rather than individual senders. Second, IP blacklists are typically incomplete: A previous study has noted that as much as 20% of spam received at spam traps is not listed in any blacklists [33]. Finally, they are sometimes inaccurate: Anecdotal evidence is rife with stories of IP addresses of legitimate mail servers being incorrectly blacklisted (e.g., because they were reflecting spam to mailing lists). To account for these shortcomings, commercial reputation systems typically incorporate additional data such as SMTP metadata or message fingerprints to mitigate these shortcomings [11]. Our previous work introduced "behavioral blacklisting" and developed a spam classifier based on a single behavioral feature: the number of messages that a particular IP address sends to each recipient domain [34]. This paper builds on the main theme of behavioral blacklisting by finding better features that can classify senders earlier and are more resistant to evasion.

2.2 Data and Deployment Scenario

This section describes McAfee's TrustedSource email sender reputation system. We describe how we use the data from this system to study the network-level features of email senders and to evaluate *SNARE*'s classification. We also describe how *SNARE*'s features and classification algorithms could be incorporated into a real-time sender reputation system such as TrustedSource.

Data source TrustedSource is a commercial reputation system that allows lookups on various Internet identifiers such as IP addresses, URLs, domains, or message fingerprints. It receives query feedback from various different device types such as mail gateways, Web gateways, and firewalls. We evaluated *SNARE* using the query logs from McAfee's TrustedSource system over a fourteen-day period from October 22–November 4, 2007. Each received email generates a lookup to the TrustedSource database, so each entry in the query log represents a single email that was sent from some sender to one of McAfee's TrustedSource appliances. Due to the volume

Field	*Description*
timestamp	UNIX timestamp
ts_server_name	Name of server that handles the query
score	Score for the message based on a combination of anti-spam filters
source_ip	Source IP in the packet (DNS server relaying the query to us)
query_ip	The IP being queried
body_length	Length of message body
count_taddr	Number of To-addresses

Figure 1: Description of data used from the McAfee dataset.

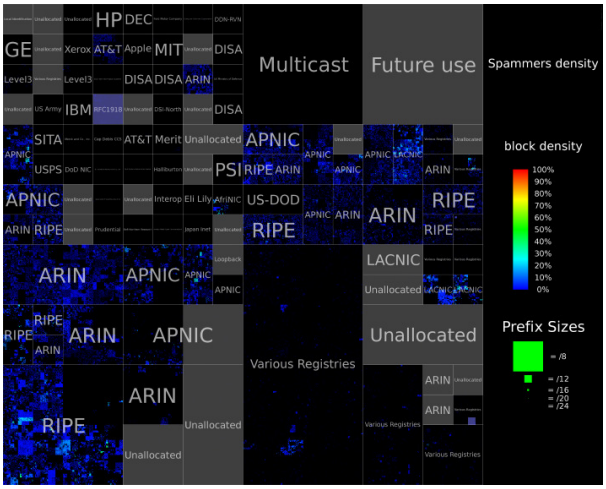


Figure 2: Distribution of senders' IP addresses in Hilbert space for the one-week period (October 22–28, 2007) of our feature study. (The grey blocks are unused IP space.)

of the full set of logs, we focused on logs from a single TrustedSource server, which reflects about 25 million email messages as received from over 1.3 million IP addresses each day. These messages were reported from approximately 2,500 distinct TrustedSource appliances geographically distributed around the world. While there is not a precise one-to-one mapping between domains and appliances, and we do not have a precise count for the number of unique domains, the number of domains is roughly of the same order of magnitude.

The logs contain many fields with *metadata for each email message*; Figure 1 shows a subset of the fields that we ultimately use to develop and evaluate *SNARE*'s classification algorithms. The `timestamp` field reflects the time at which the message was received at a TrustedSource appliance in some domain; the `source_ip` field reflects the source IP of the machine that issued the DNS query (i.e., the recipient of the email). The `query_ip`

field is the IP address being queried (i.e., the IP address of the email sender). The IP addresses of the senders are shown in the Hilbert space, as in Figure 2[1], where each pixel represents a /24 network prefix and the intensity indicates the observed IP density in each block. The distribution of the senders' IP addresses shows that the TrustedSource database collocated a representative set of email across the Internet. We use many of the other features in Figure 1 as input to *SNARE*'s classification algorithms.

To help us label senders as either spammers or legitimate senders for both our feature analysis (Section 3) and training (Sections 2.3 and 4), the logs also contain *scores* for each email message that indicate how McAfee scored the email sender based on its current system. The `score` field indicates McAfee's sender reputation score, which we stratify into five labels: certain ham, likely ham, certain spam, likely ham, and uncertain. Although these scores are not perfect ground truth, they do represent the output of both manual classification and continually tuned algorithms that also operate on more heavy-weight features (e.g., packet payloads). Our goal is to develop a fully automated classifier that is as accurate as TrustedSource but (1) classifies senders *automatically* and (2) relies only on lightweight, evasion-resistant network-level features.

Deployment and data aggregation scenario Because it operates only on network-level features of email messages, *SNARE* could be deployed either as part of TrustedSource or as a standalone DNSBL. Some of the features that *SNARE* uses rely on aggregating sender behavior across a wide variety of senders. To aggregate these features, a monitor could collect information about the global behavior of a sender across a wide variety of recipient domains. Aggregating this information is a reasonably lightweight operation: Since the features that *SNARE* uses are based on simple features (i.e., the IP address, plus auxiliary information), they can be piggybacked in small control messages or in DNS messages (as with McAfee's TrustedSource deployment).

2.3 Supervised Learning: RuleFit

Ensemble learning: *RuleFit* Learning ensembles have been among the popular predictive learning methods over the last decade. Their structural model takes the form

$$F(\mathbf{x}) = a_0 + \sum_{m=1}^{M} a_m f_m(\mathbf{x}) \qquad (1)$$

Where $\mathbf{x}$ are input variables derived form the training data (spatio-temporal features); $f_m(\mathbf{x})$ are different functions called ensemble members ("base learner") and M is the size of the ensemble; and $F(\mathbf{x})$ is the predictive output (labels for "spam" or "ham"), which takes a linear combination of ensemble members. Given the base learners, the technique determines the parameters for the learners by regularized linear regression with a "lasso" penalty (to penalize large coefficients a_m).

Friedman and Popescu proposed *RuleFit* [19] to construct regression and classification problems as linear combinations of simple rules. Because the number of base learners in this case can be large, the authors propose using the rules in a decision tree as the base learners. Further, to improve the accuracy, the variables themselves are also included as basis functions. Moreover, fast algorithms for minimizing the loss function [18] and the strategy to control the tree size can greatly reduce the computational complexity.

Variable importance Another advantage of *RuleFit* is the interpretation. Because of its simple form, each rule is easy to understand. The relative importance of the respective variables can be assessed after the predictive model is built. Input variables that frequently appear in important rules or basic functions are deemed more relevant. The importance of a variable x_i is given as importance of the basis functions that correspond directly to the variable, plus the average importance of all the other rules that involve x_i. The *RuleFit* paper has more details [19]. In Section 4.3, we show the relative importance of these features.

Comparison to other algorithms There exist two other classic classifier candidates, both of which we tested on our dataset and both of which yielded poorer performance (i.e., higher false positive and lower detection rates) than *RuleFit*. Support Vector Machine (SVM) [15] has been shown empirically to give good generalization performance on a wide variety of problems such as handwriting recognition, face detection, text categorization, etc. On the other hand, they do require significant parameter tuning before the best performance can be obtained. If the training set is large, the classifier itself can take up a lot of storage space and classifying new data points will be correspondingly slower since the classification cost is $O(S)$ for each test point, where S is the number of support vectors. The computational complexity of SVM conflicts with *SNARE*'s goal to make decision quickly (at line rate). Decision trees [30] are another type of popular classification method. The resulting classifier is simple to understand and faster, with the prediction on a new test point taking $O(\log(N))$, where N is the number of nodes in the trained tree. Unfortunately, decision trees compromise accuracy: its high false positive rates make it less than ideal for our purpose.

[1] A larger figure is available at `http://www.gtnoise.net/snare/hilbert-ip.png`.

3 Network-level Features

In this section, we explore various spatio-temporal features of email senders and discuss why these properties are relevant and useful for differentiating spammers from legitimate senders. We categorize the features we analyze by increasing level of overhead:

- *Single-packet features* are those that can be determined with no previous history from the IP address that *SNARE* is trying to classify, and given only a *single packet* from the IP address in question (Section 3.1).
- *Single-header and single-message features* can be gleaned from a single SMTP message header or email message (Section 3.2).
- *Aggregate features* can be computed with varying amounts of history (i.e., aggregates of other features) (Section 3.3).

Each class of features contains those that may be either purely local to a single receiver or aggregated across multiple receivers; the latter implies that the reputation system must have some mechanism for aggregating features in the network. In the following sections, we describe features in each of these classes, explain the intuition behind selecting that feature, and compare the feature in terms of spammers vs. legitimate senders.

No single feature needs to be perfectly discriminative between ham and spam. The analysis below shows that it is unrealistic to have a single perfect feature to make optimal resolution. As we describe in Section 2.3, *SNARE*'s classification algorithm uses a *combination* of these features to build the best classifier. We do, however, evaluate *SNARE*'s classifier using these three different classes of features to see how well it can perform using these different classes. Specifically, we evaluate how well *SNARE*'s classification works using only single-packet features to determine how well such a lightweight classifier would perform; we then see whether using additional features improves classification.

3.1 Single-Packet Features

In this section, we discuss some properties for identifying a spammer that rely only on a single packet from the sender IP address. In some cases, we also rely on auxiliary information, such as routing table information, sending history from neighboring IP addresses, etc., not solely information in the packet itself. We first discuss the features that can be extracted from just a single IP packet: the geodesic distance between the sender and receiver, sender neighborhood density, probability ratio of spam to ham at the time-of-day the IP packet arrives, AS number of the sender and the status of open ports on the machine that sent the email. The analysis is based on the McAfee's data from October 22–28, 2007 inclusive (7 days).[2]

3.1.1 Sender-receiver geodesic distance: Spam travels further

Recent studies suggest that social structure between communicating parties could be used to effectively isolate spammers [13, 20]. Based on the findings in these studies, we hypothesized that legitimate emails tend to travel shorter geographic distances, whereas the distance traveled by spam will be closer to random. In other words, a spam message may be just as likely to travel a short distance as across the world.

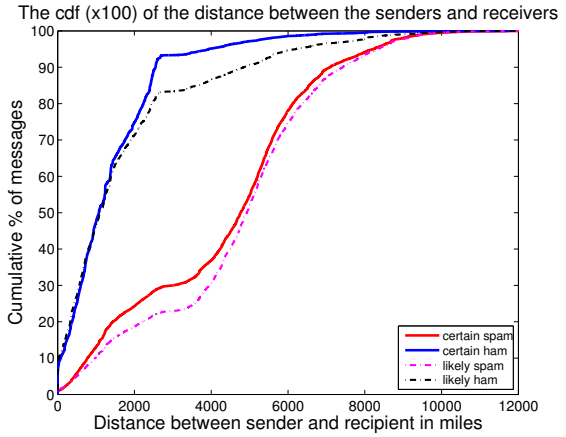
Figure 3(a) shows that our intuition is roughly correct: the distribution of the distance between the sender and the target IP addresses for each of the four categories of messages. The distance used in these plots is the geodesic distance, that is, the distance along the surface of the earth. It is computed by first finding the physical latitude and longitude of the source and target IP using the MaxMind's GeoIP database [8] and then computing the distance between these two points. These distance calculations assume that the earth is a perfect sphere. For *certain ham*, 90% of the messages travel about 2,500 miles or less. On the other hand, for *certain spam*, only 28% of messages stay within this range. In fact, about 10% of spam travels more than 7,000 miles, which is a quarter of the earth's circumference at the equator. These results indicate that geodesic distance is a promising metric for distinguishing spam from ham, which is also encouraging, since it can be computed quickly using just a single IP packet.

3.1.2 Sender IP neighborhood density: Spammers are surrounded by other spammers

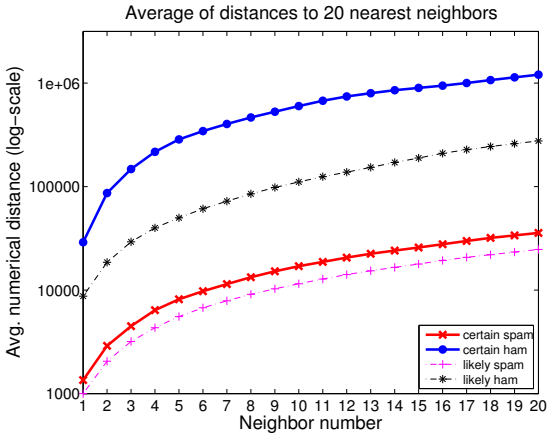
Most spam messages today are generated by botnets [33, 37]. For messages originating from the same botnet, the infected IP addresses may all lie close to one another in numerical space, often even within the same subnet. One way to detect whether an IP address belongs to a botnet is to look at the past history and determine if messages have been received from other IPs in the same subnet as the current sender, where the subnet size can be determined experimentally. If many different IPs from the same subnet are sending email, the likelihood that the whole subnet is infested with bots is high.

The problem with simply using subnet density is that the frame of reference does not transcend the subnet

[2]The evaluation in Section 4 uses the data from October 22–November 4, 2007 (14 days), some of which are not included in the data trace used for measurement study.



(a) Geodesic distance between the sender and recipient's geographic location.



(b) Average of numerical distances to the 20 nearest neighbors in the IP space.

Figure 3: Spatial differences between spammers and legitimate senders.

boundaries. A more flexible measure of *email sender density* in an IP's neighborhood is the distances to its k nearest neighbors. The distance to the k nearest neighbors can be computed by treating the IPs as set of numbers from 0 to $2^{32} - 1$ (for IPv4) and finding the nearest neighbors in this single dimensional space. We can expect these distances to exhibit different patterns for spam and ham. If the neighborhood is *crowded*, these neighbor distances will be small, indicating the possible presence of a botnet. In normal circumstances, it would be unusual to see a large number of IP addresses sending email in a small IP address space range (one exception might be a cluster of outbound mail servers, so choosing a proper threshold is important, and an operator may need to evaluate which threshold works best on the specific network where *SNARE* is running).

The average distances to the 20 nearest neighbors of the senders are shown in Figure 3(b). The x-axis indicates how many nearest neighbors we consider in IP space, and the y-axis shows the average distance in the sample to that many neighbors. The figure reflects the fact that a large majority of spam originates from hosts have high email sender density in a given IP region. The distance to the k^{th} nearest neighbor for spam tends to be much shorter on average than it is for legitimate senders, indicating that spammers generally reside in areas with higher densities of email senders (in terms of IP address space).

3.1.3 Time-of-day: Spammers send messages according to machine off/on patterns

Another feature that can be extracted using information from a single packet is the time of day when the message was sent. We use the *local* time of day at the sender's physical location, as opposed to Coordinated Universal Time (UTC). The intuition behind this feature is that local legitimate email sending patterns may more closely track "conventional" diurnal patterns, as opposed to spam sending patterns.

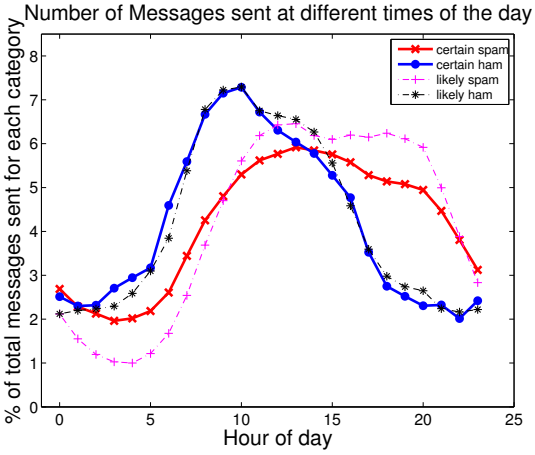


Figure 4: Differences in diurnal sending patterns of spammers and legitimate senders.

Figure 4 shows the relative percentage of messages of each type at different times of the day. The legitimate senders and the spam senders show different diurnal patterns. Two times of day are particularly striking: the relative amount of ham tends to ramp up quickly at the start of the workday and peaks in the early morning. Volumes decrease relatively quickly as well at the end of the workday. On the other hand spam increases at a slower, steadier pace, probably as machines are switched on in the morning. The spam volume stays steady throughout the day and starts dropping around 9:00 p.m., probably when machines are switched off again. In summary, legitimate

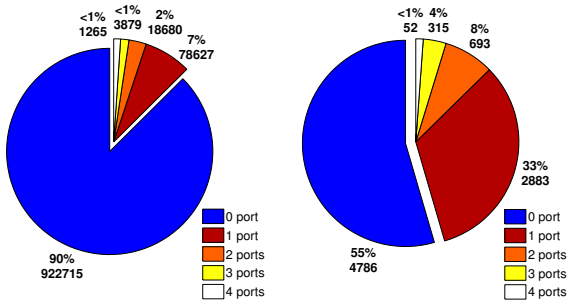
senders tend to follow workday cycles, and spammers tend to follow machine power cycles.

To use the timestamp as a feature, we compute the probability ratio of spam to ham at the time of the day when the message is received. First, we compute the *a priori* spam probability $p_{s,t}$ during some hour of the day t, as $p_{s,t} = n_{s,t}/n_s$, where $n_{s,t}$ is the number of spam messages received in hour t, and n_s is the number of spam messages received over the entire day. We can compute the *a priori* ham probability for some hour t, $p_{h,t}$ in a similar fashion. The probability ratio, r_t is then simply $p_{s,t}/p_{h,t}$. When a new message is received, the precomputed spam to ham probability ratio for the corresponding hour of the day at the senders timezone, r_t can be used as a feature; this ratio can be recomputed on a daily basis.

3.1.4 AS number of sender: A small number of ASes send a large fraction of spam

As previously mentioned, using IP addresses to identify spammers has become less effective for several reasons. First, IP addresses of senders are often transient. The compromised machines could be from dial-up users, which depend on dynamic IP assignment. If spam comes form mobile devices (like laptops), the IP addresses will be changed once the people carry the devices to a different place. In addition, spammers have been known to adopt stealthy spamming strategies where each bot only sends several spam to a single target domain, but overall the botnets can launch a huge amount of spam to many domains [33]. The low emission-rate and distributed attack requires to share information across domains for detection.

On the other hand, our previous study revealed that a significant portion of spammers come from a relatively small collection of ASes [33]. More importantly, the ASes responsible for spam differ from those that send legitimate email. As a result, the AS numbers of email senders could be a promising feature for evaluating the senders' reputation. Over the course of the seven days in our trace, more than 10% of unique spamming IPs (those sending certain spam) originated from only 3 ASes; the top 20 ASes host 42% of spamming IPs. Although our previous work noticed that a small number of ASes originated a large fraction of spam [33], we believe that this is the first work to suggest using the AS number of the email sender as input to an automated classifier for sender reputation.



(a) spam (b) legitimate mail

Figure 5: Distribution of number of open ports on hosts sending spam and legitimate mail.

3.1.5 Status of service ports: Legitimate mail tends to originate from machines with open ports

We hypothesized that legitimate mail senders may also listen on other ports besides the SMTP port, while bots might not; our intuition is that the bots usually send spam directly to the victim domain's mail servers, while the legitimate email is handed over from other domains' MSA (Mail Submission Agent). The techniques of reverse DNS (rDNS) and Forward Confirmed Reverse DNS (FCrDNS) have been widely used to check whether the email is from dial-up users or dynamically assigned addresses, and mail servers will refuse email from such sources [1].

We propose an additional feature that is orthogonal to DNSBL or rDNS checking. Outgoing mail servers open specific ports to accept users' connections, while the bots are compromised hosts, where the well-known service ports are closed (require root privilege to open). When packets reach the mail server, the server issues an active probe sent to the source host to scan the following four ports that are commonly used for outgoing mail service: 25 (SMTP), 465 (SSL SMTP), 80 (HTTP) and 443 (HTTPS), which are associated with outgoing mail services. Because neither the current mail servers nor the McAfee's data offer email senders' port information, we need to probe back sender's IP to check out what service ports might be open. The probe process was performed during both October 2008 and January 2009, well after the time when the email was received. Despite this delay, the status of open ports still exposes a striking difference between legitimate senders and spammers. Figure 5 shows the percentages and the numbers of opening ports for spam and ham categories respectively. The statistics are calculated on the senders' IPs from the evaluation dataset we used in Section 4 (October 22–28, 2007). In the spam case, 90% of spamming IP addresses have *none* of the standard mail service ports open; in contrast,

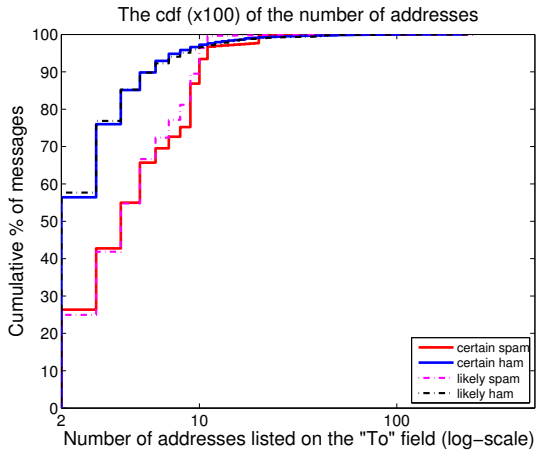


Figure 6: Distribution of number of addresses listed on the "To" field for each category (ignoring single-recipient messages).

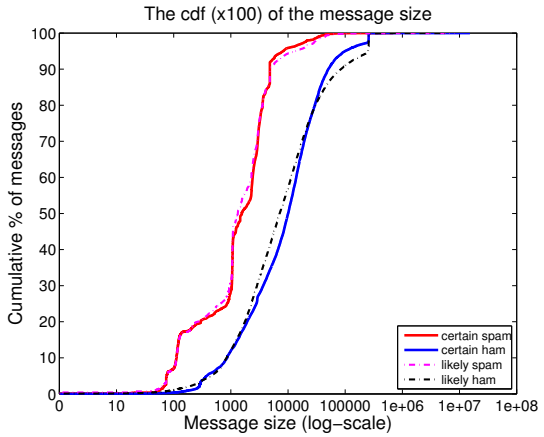


Figure 7: Distribution of message size (in bytes) for the different categories of messages.

half of the legitimate email comes from machines listening on at least one mail service port. Although firewalls might block the probing attempts (which causes the legitimate mail servers show no port listening), the status of the email-related ports still appears highly correlated with the distinction of the senders. When providing this feature as input to a classifier, we represent it as a bitmap (4 bits), where each bit indicates whether the sender IP is listening on a particular port.

3.2 Single-Header and Single-Message Features

In this section, we discuss other features that can be extracted from a single SMTP header or message: the number of recipients in the message, and the length of the message. We distinguish these features from those in the previous section, since extracting these features actually requires opening an SMTP connection, accepting the message, or both. Once a connection is accepted, and the SMTP header and subsequently, the compete message are received. At this point, a spam filter could extract additional non-content features.

3.2.1 Number of recipients: Spam tends to have more recipients

The features discussed so far can be extracted from a single IP packet from any given specific IP address combined with some historical knowledge of messages from other IPs. Another feature available without looking into the content is the number of address in "To" field of the header. This feature can be extracted after receiving the entire SMTP header but before accepting the message body. However, the majority of messages only have one address listed. Over 94% of spam and 96% of legitimate email is sent to a single recipient. Figure 6 shows the distribution of number of addresses in the "To" field for each category of messages for all emails that are sent to more than one recipient. The x-axis is on a log-scale to focus the plot on the smaller values. Based on this plot and looking at the actual values, it appears that if there are very large number of recipients on the "To" field (100 or more), there does not seem to be a significant difference between the different types of senders for this measure. The noticeable differences around 2 to 10 addresses show that, generally, ham has fewer recipients (close to 2) while spam is sent to multiple addresses (close to 10). (We acknowledge that this feature is probably evadable and discuss this in more detail in Section 6.1).

3.2.2 Message size: Legitimate mail has variable message size; spam tends to be small

Once an entire message has been received, the email body size in bytes is also known. Because a given spam sender will mostly send the same or similar content in all the messages, it can be expected that the variance in the size of messages sent by a spammer will be lower than among the messages sent by a legitimate sender. To stay effective, the spam bots also need to keep the message size small so that they can maximize the number of messages they can send out. As such the spam messages can be expected to be biased towards the smaller size. Figure 7 shows the distribution of messages for each category. The spam messages are all clustered in the 1–10KB range, whereas the distribution of message size for legitimate senders is more evenly distributed. Thus, the mes-

sage body size is another property of messages that may help differentiate spammers from legitimate senders.

3.3 Aggregate Features

The behavioral properties discussed so far can all be constructed using a single message (with auxiliary or neighborhood information). If some history from an IP is available, some *aggregate IP-level features* can also be constructed. Given information about multiple messages from a single IP address, the overall *distribution* of the following measures can be captured by using a combination of *mean and variance of*: (1) geodesic distance between the sender and recipient, (2) number of recipients in the "To" field of the SMTP header, and (3) message body length in bytes. By summarizing behavior over multiple messages and over time, these aggregate features may yield a more reliable prediction. On the flip side, computing these features comes at the cost of increased latency as we need to collect a number of messages before we compute these. Sometimes gathering aggregate information even requires cross-domain collaboration. By averaging over multiple messages, these features may also smooth the structure of the feature space, making marginal cases more difficult to classify.

4 Evaluating the Reputation Engine

In this section, we evaluate the performance of *SNARE*'s *RuleFit* classification algorithm using different sets of features: those just from a single packet, those from a single header or message, and aggregate features.

4.1 Setup

For this evaluation, we used fourteen days of data from the traces, from October 22, 2007 to November 4, 2007, part of which are different from the analysis data in Section 3. In other words, the entire data trace is divided into two parts: the first half is used for measurement study, and the latter half is used to evaluate *SNARE*'s performance. The purpose of this setup is both to verify the hypothesis that the feature statistics we discovered would stick to the same distribution over time and to ensure that feature extraction would not interfere with our evaluation of prediction.

Training We first collected the features for each message for a subset of the trace. We then randomly sampled 1 million messages from each day on average, where the volume ratio of spam to ham is the same as the original data (i.e., 5% ham and 95% spam; for now, we consider only messages in the "certain ham" and "certain spam" categories to obtain more accurate ground truth). Only our evaluation is based on this sampled dataset, *not* the feature analysis from Section 3, so the selection of those features should not have been affected by sampling. We then intentionally sampled equal amounts of spam as the ham data (30,000 messages in each categories for each day) to train the classifier because training requires that each class have an equal number of samples. In practice, spam volume is huge, and much spam might be discarded before entering the *SNARE* engine, so sampling on spam for training is reasonable.

Validation We evaluated the classifier using temporal cross-validation, which is done by splitting the dataset into subsets along the time sequence, training on the subset of the data in a time window, testing using the next subset, and moving the time window forward. This process is repeated ten times (testing on October 26, 2007 to November 4, 2007), with each subset accounting for one-day data and the time window set as 3 days (which indicates that long-period history is not required). For each round, we compute the detection rate and false positive rate respectively, where the detection rate (the "true positive" rate) is the ratio of spotted spam to the whole spam corpus, and false positive rate reflects the proportion of misclassified ham to all ham instances. The final evaluation reflects the average computed over all trials.

Summary Due to the high sampling rate that we used for this experiment, we repeated the above experiment for several trials to ensure that the results were consistent across trials. As the results in this section show, detection rates are approximately 70% and false positive rates are approximately 0.4%, even when the classifier is based only on single-packet features. The false positive drops to less 0.2% with the same 70% detection as the classifier incorporates additional features. Although this false positive rate is likely still too high for *SNARE* to subsume all other spam filtering techniques, we believe that the performance may be good enough to be used in conjunction with other methods, perhaps as an early-stage classifier, or as a substitute for conventional IP reputation systems (e.g., SpamHaus).

4.2 Accuracy of Reputation Engine

In this section, we evaluate *SNARE*'s accuracy on three different groups of features. Surprisingly, we find that, even relying on only single-packet features, *SNARE* can automatically distinguish spammers from legitimate senders. Adding additional features based on single-header or single-message, or aggregates of these features based on 24 hours of history, improves the accuracy further.

(a) Single Packet

	Classified as	
	Spam	Ham
Spam	**70%**	30%
Ham	**0.44%**	99.56%

(b) Single Header/Message

	Classified as	
	Spam	Ham
Spam	**70%**	30%
Ham	**0.29%**	99.71%

(c) 24+ Hour History

	Classified as	
	Spam	Ham
Spam	**70%**	30%
Ham	**0.20%**	99.80%

Table 1: *SNARE* performance using *RuleFit* on different sets of features using covariant shift. Detection and false positive rates are shown in bold. (The detection is fixed at 70% for comparison, in accordance with today's DNSBLs [10]).

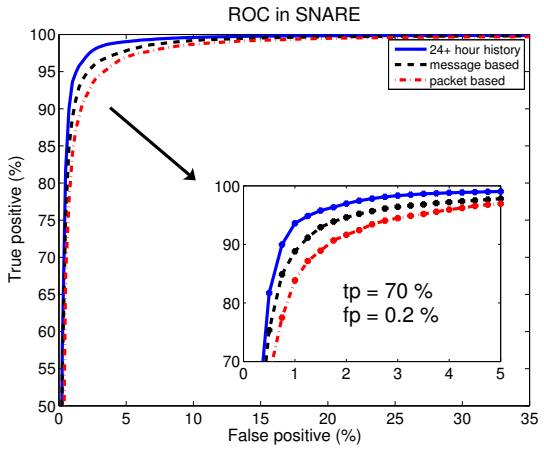


Figure 8: ROC in *SNARE*.

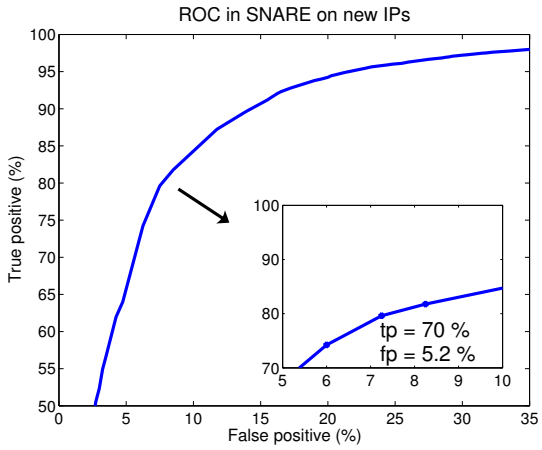


Figure 9: ROC on fresh IPs in *SNARE*.

4.2.1 Single-Packet Features

When a mail server receives a new connection request, the server can provide *SNARE* with the IP addresses of the sender and the recipient and the time-stamp based on the TCP SYN packet alone. Recall from Section 3 even if *SNARE* has never seen this IP address before, it can still combine this information with recent history of behavior of other email servers and construct the following features: (1) geodesic distance between the sender and the recipient, (2) average distance to the 20 nearest neighbors of the sender in the log, (3) probability ratio of spam to ham at the time the connection is requested (4) AS number of the sender's IP, and (5) status of the email-service ports on the sender.

To evaluate the effectiveness of these features, we trained *RuleFit* on these features. The dash-dot curve in Figure 8 demonstrate the ROC curve of *SNARE*'s reputation engine. The fp $= 0.2\%$ and tp $= 70\%$ statistics refer to the curve with 24-hour history (solid line), which will be addresses later. We check the false positive given a fixed true positive, 70%. The confusion matrix is shown in Table 1(a). Just over 0.44% of legitimate email gets labelled as spam. This result is significant because it relies on features constructed from a limited amount of data and just a single IP packet from the candidate IP. Sender reputation system will be deployed in conjunction with a combination of other techniques including content based filtering. As such, as a first line of defense, this system will be very effective in eliminating a lot of undesired senders. In fact, once a sender is identified as a spammer, the mail server does not even need to accept the connection request, saving network bandwidth and computational resources. The features we describe below improve accuracy further.

4.2.2 Single-Header and Single-Message Features

Single-packet features allow *SNARE* to rapidly identify and drop connections from spammers even before looking at the message header. Once a mail server has accepted the connection and examined the entire message, *SNARE* can determine sender reputation with increased confidence by looking at an additional set of features. As described in Section 3.2, these features include the number of recipients and message body length. Table 1(b) shows the prediction accuracy when we combine the single-packet features (i.e., those from the previous section) with these additional features. As the results from Section 3 suggest, adding the *message body length* and

number of recipients to the set of features further improves *SNARE*'s detection rate and false positive rate.

It is worth mentioning that the number of recipients listed on the "To" field is perhaps somewhat evadable: a sender could list the target email addresses on "Cc" and "Bcc" fields. Besides, if the spammers always place a single recipient address in the "To" field, this value will be the same as the large majority of legitimate messages. Because we did not have logs of additional fields in the SMTP header beyond the count of email addresses on the "To" field, we could not evaluate whether considering number of recipients listed under "Cc" and "Bcc" headers is worthwhile.

4.2.3 Aggregate Features

If multiple messages from a sender are available, the following features can be computed: the mean and variance of geodesic distances, message body lengths and number of recipients. We evaluate a classifier that is trained on *aggregate statistics* from the past 24 hours together with the features from previous sections.

Table 1(c) shows the performance of *RuleFit* with these aggregate features, and the ROC curve is plotted as the solid one in Figure 8. Applying the aggregate features decreases the error rate further: 70% of spam is identified correctly, while the false positive rate is merely 0.20%. The content-based filtering is very efficient to identify spam, but can not satisfy the requirement of processing a huge amount of messages for big mail servers. The prediction phase of *RuleFit* is faster, where the query is traversed from the root of the decision tree to a bottom label. Given the low false positive rate, *SNARE* would be a perfect first line of defense, where suspicious messages are dropped or re-routed to a farm for further analysis.

4.3 Other Considerations

Detection of "fresh" spammers We examined data trace, extracted the IP addresses not showing up in the previous training window, and further investigated the detection accuracy for those 'fresh' spammers with all *SNARE*'s features. If fixing the true positive as 70%, the false positive will increase to 5.2%, as shown in Figure 9. Compared with Figure 8, the decision on the new legitimate users becomes worse, but most of the new spammers can still be identified, which validates that *SNARE* is capable of *automatically* classifying "fresh" spammers.

Relative importance of individual features We use the fact that *RuleFit* can evaluate the *relative importance* of the features we have examined in Sections 3. Table 2 ranks all spatio-temporal features (with the most important feature at top). The top three features—*AS num*, *avg length* and *neig density*—play an important role in separating out spammers from good senders. This result is quite promising, since most of these features are lightweight: Better yet, two of these three can be computed having received only a single packet from the sender. As we will discuss in Section 6, they are also relatively resistant to evasion.

rank	*Feature Description*
1	AS number of the sender's IP
2	average of message length in previous 24 hours
3	average distance to the 20 nearest IP neighbors of the sender in the log
4	standard deviation of message length in previous 24 hours
5	status of email-service ports on the sender
6	geodesic distance between the sender and the recipient
7	number of recipient
8	average geodesic distance in previous 24 hours
9	average recipient number in previous 24 hours
10	probability ratio of spam to ham when getting the message
11	standard deviation of recipient number in previous 24 hours
12	length of message body
13	standard deviation of geodesic distance in previous 24 hours

Table 2: Ranking of feature importance in *SNARE*.

Correlation analysis among features We use mutual information to investigate how tightly the features are coupled, and to what extent they might contain redundant information. Given two random variables, mutual information measures how much uncertainty of one variable is reduced after knowing the other (i.e., the information they share). For discrete variables, the mutual information of X and Y is calculated as: $I(X,Y) = \sum_{x,y} p(x,y) \log(\frac{p(x,y)}{p(x)p(y)})$. When logarithm base-two is used, the quantity reflects how many bits can be removed to encode one variable given the other one. Table 3 shows the mutual information between pairs of features for one day of training data (October 23, 2007). We do not show statistics from other days, but features on those days reflect similar quantities for mutual information. The features with continuous values (e.g., geodesic distance between the sender and the recipient) are transformed into discrete variables by dividing the value range into 4,000 bins (which yields good discrete approximation); we calculate mutual information over the discrete probabilities. The indexes of the features in the table are the same as the ranks in Table 2; the packet-based features are marked with black circles. We also calculate the entropy of every feature and show them next to the indices in Table 3.

The interpretation of mutual information is consistent only within a single column or row, since comparison of mutual information without any common variable is meaningless. The table, of course, begs additional analysis but shows some interesting observations. The top-ranked feature, AS number, shares high mutual information (shown in bold) with several other features, especially with feature 6, geodesic distance between sender and recipient. The aggregate features of first-order statis-

	❶ (8.68)	2 (7.29)	❸ (2.42)	4 (6.92)	❺ (1.20)	❻ (10.5)	7 (0.46)	8 (9.29)	9 (2.98)	❿ (4.45)	11 (3.00)	12 (6.20)
2 (7.29)	**4.04**											
❸ (2.42)	**1.64**	1.18										
4 (6.92)	**3.87**	4.79	1.23									
❺ (1.20)	**0.65**	0.40	0.11	0.43								
❻ (10.5)	**5.20**	3.42	0.88	3.20	0.35							
7 (0.46)	**0.11**	0.08	0.02	0.08	0.004	0.15						
8 (9.29)	**5.27**	5.06	1.20	4.79	0.46	5.16	0.13					
9 (2.98)	**1.54**	1.95	0.53	2.03	0.09	1.17	0.10	2.08				
❿ (4.45)	**0.66**	0.46	0.07	0.49	0.02	0.87	0.006	0.85	0.13			
11 (3.00)	**1.87**	1.87	0.75	2.04	0.16	1.55	0.09	2.06	1.87	0.20		
12 (6.20)	**2.34**	2.53	0.49	2.12	0.20	2.34	0.07	2.30	0.52	0.31	0.73	
13 (8.89)	**4.84**	4.78	1.15	4.69	0.41	4.77	0.11	6.47	1.98	0.69	2.04	2.13

Table 3: Mutual information among features in *SNARE*; packet-based features are shown with numbers in dark circles. (The indices are the feature ranking in Table 2.)

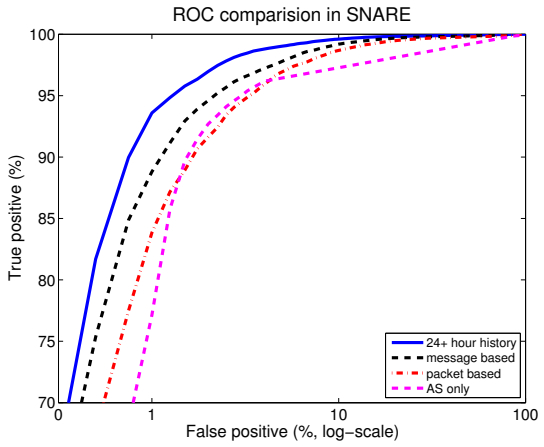


Figure 10: ROC comparison with AS-only case.

tics (e.g., feature 2, 4, 8) also have high values with each other. Because spammers may exhibit one or more of these features across each message, aggregating the features across multiple message over time indicates that, observing a spammer over time will reveal many of these features, though not necessarily on any message or single group of message. For this reason, aggregate features are likely to share high mutual information with other features that are common to spammers.

One possible reason that aggregate features have high mutual information with each other is that aggregating the features across multiple messages over time incorporates history of an IP address that may exhibit many of these characteristics over time.

Performance based on AS number only Since AS number is the most influential feature according to *RuleFit* and shares high mutual information with many other features, we investigated how well this feature alone can distinguish spammers from legitimate senders. We feed the AS feature into the predictive model and plot the ROC as the lower dashed curve in Figure 10. To make a close comparison, the "packet-based", "message-based", and "history-based" ROCs (the same as those in Figure 8) are shown as well, and the false positive is displayed on a log scale. The classifier gets false positive 0.76% under a 70% detection rate. Recall from Table 1 the false positive rate with "packet-based" features is almost a half, 0.44%, and that with "history-based" features will further reduce to 0.20%, which demonstrates that other features help to improve the performance. We also note that using the AS number alone as a distinguishing feature may cause large amounts of legitimate email to be misclassified, and could be evaded if an spammer decides to announce routes with a forged origin AS (which is an easy attack to mount and a somewhat common occurrence) [2, 26, 39].

5 A Spam-Filtering System

This section describes how *SNARE*'s reputation engine could be integrated into an overall spam-filtering system that includes a whitelist and an opportunity to continually retrain the classifier on labeled data (e.g., from spam traps, user inboxes, etc.). Because *SNARE*'s reputation engine still has a non-zero false positive rate, we show how it might be incorporated with mechanisms that could help further improve its accuracy, and also prevent discarding legitimate mail even in the case of some false positives. We propose an overview of the system and evaluate the benefits of these two functions on overall system accuracy.

5.1 System Overview

Figure 11 shows the overall system framework. The system needs not reside on a single server. Large public email providers might run their own instance of *SNARE*, since they have plenty of email data and processing resources. Smaller mail servers might query a remote

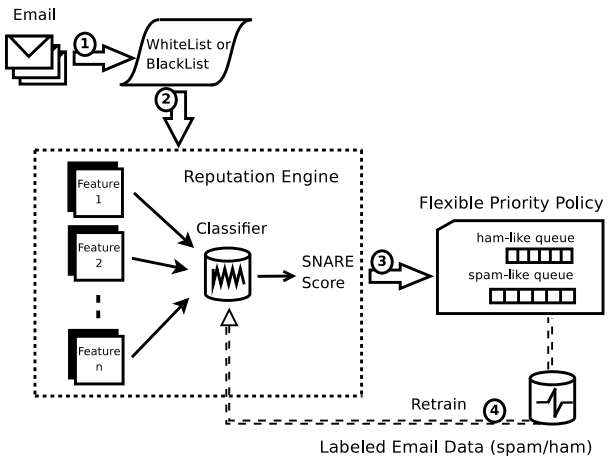


Figure 11: *SNARE* framework.

SNARE server. We envision that *SNARE* might be integrated into the workflow in the following way:

1. **Email arrival.** After getting the first packet, the mail server submits a query to the *SNARE* server (only the source and destination IP). Mail servers can choose to send more information to *SNARE* after getting the SMTP header or the whole message. Sending queries on a single packet or on a message is a tradeoff between detection accuracy and processing time for the email (i.e., sending the request early will make mail server get the response early). The statistics of messages in the received queries will be used to build up the *SNARE* classifier.
2. **Whitelisting.** The queries not listed in the whitelist will be passed to *SNARE*'s reputation engine (presented in Section 2.3) *before* any spam-filtering checks or content-based analysis. The output is a score, where, by default, positive value means likely spam and negative value means likely ham; and the absolute values represent the confidence of the classification. Administrators can set a different score threshold to make tradeoff between the false positive and the detection rate. We evaluate the benefits of whitelisting in Section 5.2.1.
3. **Greylisting and content-based detection.** Once the reputation engine calculates a score, the email will be delivered into different queues. More resource-sensitive and time-consuming detection methods (e.g., content-based detection) can be applied at this point. When the mail server has the capability to receive email, the messages in ham-like queue have higher priority to be processed, whereas the messages in spam-like queue will be offered less resources. This policy allows the server to speed up processing the messages that *SNARE* classifies as spam. The advantage of this hierarchical detecting scheme is that the legitimate email will be delivered to users' inbox sooner. Messages in the spam-like queue could be shunted to more resource-intensive spam filters before they are ultimately dropped.[3]
4. **Retraining** Whether the IP address sends spam or legitimate mail in that connection is not known at the time of the request, but is known after mail is processed by the spam filter. *SNARE* depends on accurately labelled training data. The email will eventually receive more careful checks (shown as "Retrain" in Figure 11). The results from those filters are considered as ground truth and can be used as feedback to dynamically adjust the *SNARE* threshold. For example, when the mail server has spare resource or much email in the spam-like queue is considered as legitimate later, *SNARE* system will be asked to act more generous to score email as likely ham; on the other hand, if the mail server is overwhelmed or the ham-like queue has too many incorrect labels, *SNARE* will be less likely to put email into ham-like queue. Section 5.2.2 evaluates the benefits of retraining for different intervals.

5.2 Evaluation

In this section, we evaluate how the two additional functions (whitelisting and retraining) improve *SNARE*'s overall accuracy.

5.2.1 Benefits of Whitelisting

We believe that a whitelist can help reduce *SNARE*'s overall false positive rate. To evaluate the effects of such a whitelist, we examined the features associated with the false positives, and determine that, 43% of all of *SNARE*'s false positives for a single day originate from just 10 ASes. We examined this characteristic for different days and found that 30% to 40% of false positives from any given day originate from the top 10 ASes. Unfortunately, however, these top 10 ASes do not remain the same from day-to-day, so the whitelist may need to be retrained periodically. It may also be the case that other features besides AS number of the source provide an even better opportunity for whitelisting. We leave the details of refining the whitelist for future work.

Figure 12 shows the average ROC curve when we whitelist the top 50 ASes responsible for most misclassified ham in each day. This whitelisting reduces the best

[3] Although *SNARE*'s false positive rates are quite low, some operators may feel that any non-zero chance that legitimate mail or sender might be misclassified warrants at least a second-pass through a more rigorous filter.

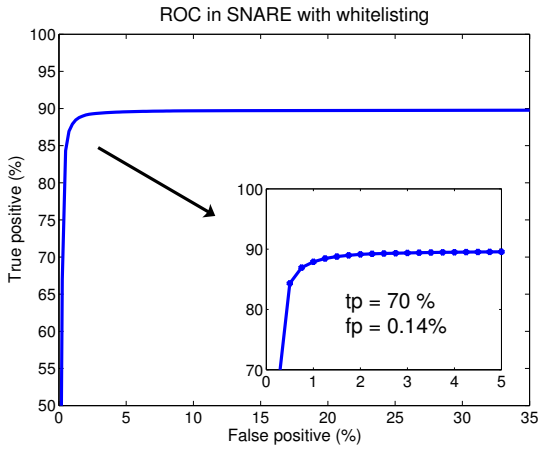


Figure 12: ROC in *SNARE* with whitelisting on ASes.

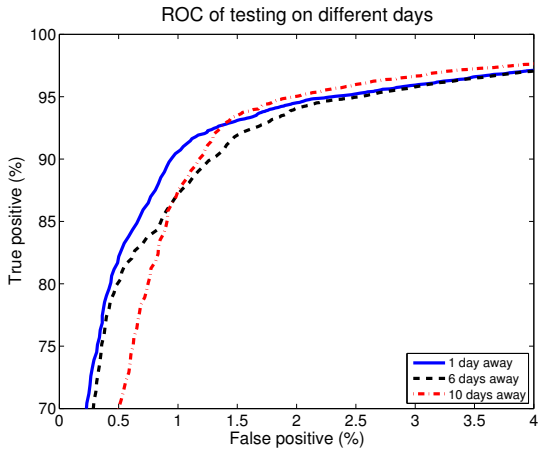


Figure 13: ROC using previous training rules to classify future messages.

possible detection rate considerably (effectively because about 11% of spam originates from those ASes). However, this whitelisting also reduces the false positive rate to about 0.14% for a 70% detection rate. More aggressive whitelisting, or whitelisting of other features, could result in even lower false positives.

5.2.2 Benefits of Retraining

Setup Because email sender behavior is dynamic, training *SNARE* on data from an earlier time period may eventually grow stale. To examine the requirements for periodically retraining the classifier, we train *SNARE* based on the first 3 days' data (through October 23–25, 2007) and test on the following 10 days. As before, we use 1 million randomly sampled spam and ham messages to test the classifier for each day.

Results Figure 13 shows the false positive and true positive on 3 future days, October 26, October 31, and November 4, 2007, respectively. The prediction on future days will become more inaccurate with time passage. For example, on November 4 (ten days after training), the false positive rate has dropped given the same true positive on the ROC curve. This result suggests that, for the spammer behavior in this trace, retraining *SNARE*'s classification algorithms daily should be sufficient to maintain accuracy. (We expect that the need to retrain may vary across different datasets.)

6 Discussion and Limitations

In this section, we address various aspects of *SNARE* that may present practical concerns. We first discuss the extent to which an attacker might be able to evade various features, as well as the extent to which these features might vary across time and datasets. We then discuss scalability concerns that a production deployment of *SNARE* may present, as well as various possible workarounds.

6.1 Evasion-Resistance and Robustness

In this section, we discuss the evasion resistance of the various network-level features that form the inputs to *SNARE*'s classification algorithm. Each of these features is, to some degree, evadable. Nevertheless, *SNARE* raises the bar by making it more difficult for spammers to evade detection without altering the techniques that they use to send spam. Although spammers might adapt to evade some of the features below, we believe that it will be difficult for a spammer to adjust all features to pass through *SNARE*, particularly without somewhat reducing the effectiveness of the spamming botnet. We survey each of the features from Table 2 in turn.

AS number AS numbers are more persistently associated with a sender's identity than the IP address, for two reasons: (1) The spamming mail server might be set up within specific ASes without the network administrator shutting it down. (2) Bots tend to aggregate within ASes, since the machines in the same ASes are likely to have the same vulnerability. It is not easy for spammers to move mail servers or the bot armies to a different AS; therefore, AS numbers are robust to indicate malicious hosts.

Message length In our analysis, we discovered that the size of legitimate email messages tends to be much more variable than that of spam (perhaps because spammers often use templates to sent out large quantities of mail [25]). With knowledge of this feature, a spammer might start to randomize the lengths of their email mes-

sages; this attack would not be difficult to mount, but it might restrict the types of messages that a spammer could send or make it slightly more difficult to coordinate a massive spam campaign with similar messages.

Nearest neighbor distances Nearest neighbor distance is another feature that will be hard to modify. Distances to k nearest neighbors effectively isolate existence of unusually large number of email servers within a small sequence of IP addresses. If the spammers try to alter their neighborhood density, they will not be able to use too many machines within a compromised subnet to send spam to the same set of destinations. Although it is possible for a botnet controller to direct bots on the same subnet to target different sets of destinations, such evasion does require more coordination and, in some cases, may restrict the agility that each spamming bot has in selecting its target destinations.

Status of email service ports Some limitation might fail the active probes, e.g., the outgoing mail servers use own protocol to mitigate messages (such as Google mail) or a firewall blocks the connections from out of the domain. But the bots do not open such ports with high probability, and the attackers need to get root privilege to enable those ports (which requires more sophisticated methods and resources). The basic idea is to find out whether the sender is a legitimate mail server. Although we used active probes in *SNARE*, other methods could facilitate the test, such as domain name checking or mail server authentication.

Sender-receiver geodesic distance The distribution of geodesic distances between the spammers' physical location and their target IP's location is a result of the spammers' requirement to reach as many target mail boxes as possible and in the shortest possible time. Even in a large, geographically distributed botnet, requiring each bot to bias recipient domains to evade this feature may limit the flexibility of how the botnet is used to send spam. Although this feature can also be evaded by tuning the recipient domains for each bot, if bots only sent spam to nearby recipients, the flexibility of the botnet is also somewhat restricted: it would be impossible, for example, to mount a coordinate spam campaign against a particular region from a fully distributed spamming botnet.

Number of recipients We found that spam messages tend to have more recipients than legitimate messages; a spammer could likely evade this feature by reducing the number of recipients on each message, but this might make sending the messages less efficient, and it might alter the sender behavior in other ways that might make a spammer more conspicuous (e.g., forcing the spammer to open up more connections).

Time of day This feature may be less resistant to evasion than others. Having said that, spamming botnets' diurnal pattern results from when the infected machines are switched on. For botnets to modify their diurnal message volumes over the day to match the legitimate message patterns, they will have to lower their spam volume in the evenings, especially between 3:00 p.m. and 9:00 p.m. and also reduce email volumes in the afternoon. This will again reduce the ability of botnets to send large amounts of email.

6.2 Other Limitations

We briefly discuss other current limitations of *SNARE*, including its ability to scale to a large number of recipients and its ability to classify IP addresses that send both spam and legitimate mail.

Scale *SNARE* must ultimately scale to thousands of domains and process hundreds of millions of email addresses per day. Unfortunately, even state-of-the-art machine learning algorithms are not well equipped to process datasets this large; additionally, sending data to a central coordinator for training could potentially consume considerably bandwidth. Although our evaluation suggests that *SNARE*'s classification is relatively robust to sampling of training data, we intend to study further the best ways to sample the training data, or perhaps even perform in-network classification.

Dual-purpose IP addresses Our conversations with large mail providers suggest that one of the biggest emerging threats are "web bots" that send spam from Web-based email accounts [35]. As these types of attacks develop, an increasing fraction of spam may be sent from IP addresses that also send significant amounts of legitimate mail. These cases, where an IP address is neither good nor bad, will need more sophisticated classifiers and features, perhaps involving timeseries-based features.

7 Related Work

We survey previous work on characterizing the network-level properties and behavior of email senders, email sender reputation systems, and other email filtering systems that are not based on content.

Characterization studies Recent characterization studies have provided increasing evidence that spammers have distinct network-level behavioral patterns. Ramachandran *et al.* [34] showed that spammers utilize transient botnets to spam at low rate from any specific IP to any domain. Xie *et al.* [38] discovered that a vast

majority of mail servers running on dynamic IP address were used solely to send spam. In their recently published study [37], they demonstrate a technique to identify bots by using signatures constructed from URLs in spam messages. Unlike *SNARE*, their signature-based botnet identification differs heavily on analyzing message content. Others have also examined correlated behavior of botnets, primarily for characterization as opposed to detection [25, 31]. Pathak *et al.* [29] deployed a relay sinkhole to gather data from multiple spam senders destined for multiple domains. They used this data to demonstrate how spammers utilize compromised relay servers to evade detection; this study looked at spammers from multiple vantage points, but focused mostly on characterizing spammers rather than developing new detection mechanisms. Niu *et al.* analyzed network-level behavior of Web spammers (e.g., URL redirections and "doorway" pages) and proposed using context-based analysis to defend against Web spam [28].

Sender reputation based on network-level behavior SpamTracker [34] is most closely related to *SNARE*; it uses network-level behavioral features from data aggregated across multiple domains to infer sender reputation. While that work initiated the idea of *behavioral blacklisting*, we have discovered many other features that are more lightweight and more evasion-resistant than the single feature used in that paper. Beverly and Sollins built a similar classifier based on transport-level characteristics (e.g., round-trip times, congestion windows) [12], but their classifier is both heavyweight, as it relies on SVM, and it also requires accepting the messages to gather the features. Tang *et al.* explored the detection of spam senders by analyzing the behavior of IP addresses as observed by query patterns [36]. Their work focuses on the breadth and the periodicity of message volumes in relation to sources of queries. Various previous work has also attempted to cluster email senders according to groups of recipients, often with an eye towards spam filtering [21, 24, 27], which is similar in spirit to *SNARE*'s geodesic distance feature; however, these previous techniques typically require analysis of message contents, across a large number of recipients, or both, whereas *SNARE* can operate on more lightweight features. McAfee's TrustedSource [4] and Cisco IronPort [3] deploy spam filtering appliances to hundreds or thousands of domains which then query the central server for sender reputation and also provide meta-data about messages they receive; we are working with McAfee to deploy *SNARE* as part of TrustedSource.

Non-content spam filtering Trinity [14] is a distributed, content-free spam detection system for messages originating from botnets that relies on message volumes. The SpamHINTS project [9] also has the stated goal of building a spam filter using analysis of network traffic patterns instead of the message content. Clayton's earlier work on extrusion detection involves monitoring of server logs at both the local ISP [16] as well as the remote ISP [17] to detect spammers. This work has similar objectives as ours, but the proposed methods focus more on properties related to SMTP sessions from only a single sender.

8 Conclusion

Although there has been much progress in content-based spam filtering, state-of-the-art systems for *sender reputation* (e.g., DNSBLs) are relatively unresponsive, incomplete, and coarse-grained. Towards improving this state of affairs, this paper has presented *SNARE*, a sender reputation system that can accurately and automatically classify email senders based on features that can be determined early in a sender's history—sometimes after seeing only a single IP packet.

Several areas of future work remain. Perhaps the most uncharted territory is that of using temporal features to improve accuracy. All of *SNARE*'s features are essentially discrete variables, but we know from experience that spammers and legitimate senders also exhibit different temporal patterns. In a future version of *SNARE*, we aim to incorporate such temporal features into the classification engine. Another area for improvement is making *SNARE* more evasion-resistant. Although we believe that it will be difficult for a spammer to evade *SNARE*'s features and still remain effective, designing classifiers that are more robust in the face of active attempts to evade and mis-train the classifier may be a promising area for future work.

Acknowledgments

We thank our shepherd, Vern Paxson, for many helpful suggestions, including the suggestions to look at mutual information between features and several other improvements to the analysis and presentation. We also thank Wenke Lee, Anirudh Ramachandran, and Mukarram bin Tariq for helpful comments on the paper. This work was funded by NSF CAREER Award CNS-0643974 and NSF Awards CNS-0716278 and CNS-0721581.

References

[1] FCrDNS Lookup Testing. http://ipadmin.junkemailfilter.com/rdns.php.

[2] Internet Alert Registry. http://iar.cs.unm.edu/.

[3] IronPort. http://www.ironport.com.

[4] McAfee Secure Computing. http://www.securecomputing.com.

[5] SORBS: Spam and Open Relay Blocking System. http://www.au.sorbs.net/.

[6] SpamCop. http://www.spamcop.net/bl.shtml.

[7] SpamHaus IP Blocklist. http://www.spamhaus.org.

[8] GeoIP API. MaxMind, LLC. http://www.maxmind.com/app/api, 2007.

[9] spamHINTS: Happily It's Not The Same. http://www.spamhints.org/, 2007.

[10] DNSBL Resource: Statistics Center. http://stats.dnsbl.com/, 2008.

[11] ALPEROVITCH, D., JUDGE, P., AND KRASSER, S. Taxonomy of email reputation systems. In *Proc. of the First International Workshop on Trust and Reputation Management in Massively Distributed Computing Systems (TRAM)* (2007).

[12] BEVERLY, R., AND SOLLINS, K. Exploiting the transport-level characteristics of spam. In *5th Conference on Email and Anti-Spam (CEAS)* (2008).

[13] BOYKIN, P., AND ROYCHOWDHURY, V. Personal email networks: An effective anti-spam tool. *IEEE Computer 38*, 4 (2005), 61–68.

[14] BRODSKY, A., AND BRODSKY, D. A distributed content independent method for spam detection. In *First Workshop on Hot Topics in Understanding Botnets (HotBots)* (2007).

[15] BURGES, C. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*, 2 (1998), 121–167.

[16] CLAYTON, R. Stopping spam by extrusion detection. In *First Conference of Email and Anti-Spam (CEAS)* (2004).

[17] CLAYTON, R. Stopping outgoing spam by examining incoming server logs. In *Second Conference on Email and Anti-Spam (CEAS)* (2005).

[18] FRIEDMAN, J., AND POPESCU, B. Gradient directed regularization. *Stanford University, Technical Report* (2003).

[19] FRIEDMAN, J., AND POPESCU, B. Predictive learning via rule ensembles. *Annals of Applied Statistics (to appear)* (2008).

[20] GOLBECK, J., AND HENDLER, J. Reputation network analysis for email filtering. In *First Conference on Email and Anti-Spam (CEAS)* (2004).

[21] GOMES, L. H., CASTRO, F. D. O., ALMEIDA, R. B., BETTENCOURT, L. M. A., ALMEIDA, V. A. F., AND ALMEIDA, J. M. Improving spam detection based on structural similarity. In *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI)* (2005).

[22] GOODMAN, J., CORMACK, G., AND HECKERMAN, D. Spam and the ongoing battle for the inbox. *Communications of the ACM 50*, 2 (2007), 24–33.

[23] HULTON, E., AND GOODMAN, J. Tutorial on junk email filtering. *Tutorial in the 21st International Conference on Machine Learning (ICML)* (2004).

[24] JOHANSEN, L., ROWELL, M., BUTLER, K., AND MCDANIEL, P. Email communities of interest. In *4th Conference on Email and Anti-Spam (CEAS)* (2007).

[25] KANICH, C., KREIBICH, C., LEVCHENKO, K., ENRIGHT, B., PAXSON, V., VOELKER, G. M., AND SAVAGE, S. Spamalytics: An empirical analysis of spam marketing conversion. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS)* (2008).

[26] KARLIN, J., FORREST, S., AND REXFORD, J. Autonomous security for autonomous systems. *Computer Networks 52*, 15 (2008), 2908–2923.

[27] LAM, H., AND YEUNG, D. A learning approach to spam detection based on social networks. In *4th Conference on Email and Anti-Spam (CEAS)* (2007).

[28] NIU, Y., WANG, Y.-M., CHEN, H., MA, M., AND HSU, F. A quantitative study of forum spamming using context-based analysis. In *Proceedings of the 14th Annual Network and Distributed System Security Symposium (NDSS)* (2007).

[29] PATHAK, A., HU, C., Y., AND MAO, Z., M. Peeking into spammer behavior from a unique vantage point. In *First USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)* (2008).

[30] QUINLAN, J. Induction of decision trees. *Machine Learning 1*, 1 (1986), 81–106.

[31] RAJAB, M., ZARFOSS, J., MONROSE, F., AND TERZIS, A. A multifaceted approach to understanding the botnet phenomenon. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement (IMC)* (2006).

[32] RAMACHANDRAN, A., DAGON, D., AND FEAMSTER, N. Can DNSBLs keep up with bots? In *3rd Conference on Email and Anti-Spam (CEAS)* (2006).

[33] RAMACHANDRAN, A., AND FEAMSTER, N. Understanding the network-level behavior of spammers. In *Proceedings of the ACM SIGCOMM* (2006).

[34] RAMACHANDRAN, A., FEAMSTER, N., AND VEMPALA, S. Filtering spam with behavioral blacklisting. In *ACM Conference on Computer and Communications Security (CCS)* (2007).

[35] Private conversation with Mark Risher, Yahoo Mail., 2008.

[36] TANG, Y. C., KRASSER, S., JUDGE, P., AND ZHANG, Y.-Q. Fast and effective spam IP detection with granular SVM for spam filtering on highly imbalanced spectral mail server behavior data. In *2nd International Conference on Collaborative Computing (CollaborateCom)* (2006).

[37] XIE, Y., YU, F., , ACHAN, K., PANIGRAHY, R., HULTEN, G., AND OSIPKOV, I. Spamming bots: Signatures and characteristics. In *Proceedings of ACM SIGCOMM* (2008).

[38] XIE, Y., YU, F., ACHAN, K., GILUM, E., GOLDSZMIDT, M., AND WOBBER, T. How dynamic are IP addresses. In *Proceedings of ACM SIGCOMM* (2007).

[39] ZHAO, X., PEI, D., WANG, L., MASSEY, D., MANKIN, A., WU, S. F., AND ZHANG, L. An analysis of BGP multiple origin AS (MOAS) conflicts. In *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement (IMW)* (2001).