Spam Mitigation using Spatio-Temporal Reputations from Blacklist History *

Andrew G. West, Adam J. Aviv, Jian Chang, and Insup Lee
Dept. of Computer and Information Science - University of Pennsylvania - Philadelphia, PA
{westand, aviv, jianchan, lee}@cis.upenn.edu

ABSTRACT

IP blacklists are a spam filtering tool employed by a large number of email providers. Centrally maintained and well regarded, blacklists can filter 80+% of spam without having to perform computationally expensive content-based filtering. However, spammers can vary which hosts send spam (often in intelligent ways), and as a result, some percentage of spamming IPs are not actively listed on any blacklist. Blacklists also provide a previously untapped resource of rich historical information. Leveraging this history in combination with spatial reasoning, this paper presents a novel reputation model (PreSTA), designed to aid in spam classification. In simulation on arriving email at a large university mail system, PreSTA is capable of classifying up to 50% of spam not identified by blacklists alone, and 93% of spam on average (when used in combination with blacklists). Further, the system is consistent in maintaining this blockage-rate even during periods of decreased blacklist performance. PreSTA is scalable and can classify over 500,000 emails an hour. Such a system can be implemented as a complementary blacklist service or used as a first-level filter or prioritization mechanism on an email server.

1. INTRODUCTION

Roughly 90% of the total volume of email on the Internet is considered spam [5], and IP-based blacklisting has become a standard tool in fighting such influxes. Spammers often control large collections of compromised machines, *botnets*, and vary which hosts act as the spamming mail servers. As a result, some 20% of spam emails received at a large spam trap in 2006 were not listed on any blacklist [21].

Blacklists provide only a static view of the current (or recently active) spamming IP addresses. However, when viewed over time, blacklists provide dense historical (temporal) information. Upon inspection, interesting properties emerge; for example, more than 25% of the IPs once listed on the blacklist were re-listed within 10 days, and overall, 45% were re-listed during the observation period.

It is known that spamming IP addresses exhibit interesting spatial properties. Previous studies have shown that spamming IPs are distributed non-uniformly throughout the address space [19, 21, 28], and they can often be clustered into spatial groups indicative of spamming behavior. For example, AS-membership has been shown to be a strong predictor of spamming likelihood [11], as well as BGP prefixes, and the host-names of reverse DNS look-ups [19].

In this paper we propose a novel method to combine blacklist histories with spatial context to produce predictive reputation values capable of classifying spam. Our model, **Pre**ventive **S**patio-**T**emporal **A**ggregation (PreSTA), monitors blacklist dynamics, interpreting listings as a record of negative feedback. An entity (*i.e.*, an IP address) is then evaluated based on its own history of negative feedback and the histories of spatially related entities. Spatial adjacency is multi-tiered and defined based on multiple grouping functions (*e.g.*, AS-membership, subnet, *etc.*). A reputation value is computed for each grouping, and these are combined using a standard machine learning technique to produce ham/spam classifications.

We implemented PreSTA and analyzed incoming email traces at a large university mail server. We found that PreSTA can classify an additional 50% of spam not identified by blacklists alone while maintaining similar false-positive rates. Moreover, when PreSTA is used in combination with traditional blacklists, on average 93% of spam is *consistently* identified without the need for content-based analysis. This result was found to be stable: As the underlying blacklist suffers large deviations in detection accuracy, PreSTA maintains steady-state performance. Further, PreSTA is highly *scalable*: Over 500,000 emails an hour can be scored using a single-threaded implementation on a commodity server.

We do not propose that PreSTA can (or should) replace context-based filtering. Instead, PreSTA can be leveraged just as blacklists are today – as a preliminary filter to avoid more computationally expensive analysis. Use-cases could include a complimentary service to blacklists (perhaps implemented by the blacklist provider) or an email prioritization mechanism for overloaded mail servers.

PreSTA's applicability is not confined to email spam detection. Related work has already shown PreSTA reputations helpful in prioritizing edits and detecting vandalism on Wikipedia [30], and PreSTA may be further applicable to an entire class of dynamic trust management problems [9,

*This research was supported in part by ONR MURI N00014-07-1-0907. POC: Insup Lee, lee@cis.upenn.edu

29] that are characterized by the need for decision-making in the presence of uncertainty and partial-information.

2. RELATED WORK

Spam filtering based on network-level properties of the source IP address is a popular choice for mitigating spam. Unlike content-based filters (*e.g.*, those based on Bayesian quantifiers [24]), these techniques tend to be computationally inexpensive while achieving relatively good performance.

IP blacklists [3, 7] are one such network-level filtering strategy. Blacklists are collections of known spamming IP addresses collated from various institutions (*e.g.*, large email providers). They tend to be well-regarded because they are maintained by reputable providers and incorporated into many email server's. Blacklists are only a static snapshot of spamming hosts, but over time, IP addresses are listed, delisted, and re-listed. It is precisely this history that PreSTA leverages in generating IP reputation.

Filtering based on blacklists alone is imperfect [25]. Listing latency is a commonly cited weakness [20], as is incompleteness. One study reported that 10% of spamming IPs observed at a spam-trap were not blacklisted [23]. Such situations motivate PreSTA; in these partial knowledge scenarios, an unlisted IP address can be viewed in terms of its previous listings (if any) and its spatial relation to other known spamming IPs.

The non-uniform distribution of spamming IPs on the Internet is a well-studied phenomenon. Spamming IPs tend to be found near other spamming IPs [23] and in small regions of the address space [21]. Most such IPs tend to be short-lived [28]; further supporting the use of spatial relationships. Although PreSTA employs basic spatial measures in its preliminary implementation, more advanced relationships could be exploited, such as those suggested in [11, 19]. Additionally, dynamically shaped groups could be used [27].

A key difference between PreSTA and similar work is its combination of temporal history provided by blacklists and the spatial dynamics of spamming IPs. Perhaps the closest related system is SNARE by Hao *et al.* [11]. In addition to demonstrating interesting spatial measures (including geographic distance), SNARE utilizes *simple* temporal metrics to perform spam filtering (*e.g.*, the time-of-day an email was sent) and applies a lightweight form of aggregation (*e.g.*, mean and variance) to detect abnormal patterns. In contrast, PreSTA's temporal computation has more depth, aggregating time-decayed compounding evidence that encodes *months* of *detailed* blacklisting events. Indeed, [11] identifies many valid measures of spamming behavior, but is incapable of Internet-wide scalability due to a reliance on high-dimensional learning. PreSTA spam detection computes over a single feature, IP address (and groups thereof), and is extremely scalable with high accuracy.

Similar techniques are claimed by two commercial services: Symantec [26] uses "IP reputation" in its security software, and SenderBase [12] by Ironport uses spatial data to build IP reputations. The procedures are proprietary, so a detailed comparison is not possible. However, the binary output of the public-facing query mechanisms correlate well with PreSTA's classifications.

PreSTA can also be examined in the context of general-purpose reputation systems/logics, such as EigenTrust [16] or TNA-SL [14]. A key difference involves the nature of feedback; namely, PreSTA considers only negative feedback. Conventional algorithms aggregate over both positive and negative feedback, and feedback is indefinitely retained and associated with a single *discrete* event. PreSTA utilizes *expiring feedback*, where a negative observation (*e.g.*, sending spam) is valid for some finite duration (the blacklist period), after which, it is discarded.

3. REPUTATION MODEL

Although our presentation of PreSTA is focused on the domain of spam detection, it is important to note that PreSTA defines a general reputation model. There are two requirements for potential applications: (1) Access to a history of negative feedback (as achieved via IP blacklists); and (2) the ability to define spatial partitions over entities (as achieved via the IP address hierarchy). The reputation values computed consider both the history of negative feedback for an individual entity and those of related entities.

In the temporal dimension, a history of negative feedback, stored in a *feedback database*, is required. An entity is considered *active* in the database when an associated negative feedback has been recently received (*i.e.*, the entity is listed on the blacklist). After some interval, the feedback expires, and the entity is considered *inactive* (*i.e.*, the entity is delisted from the blacklist). A query to the database returns an entire history of active and inactive events, to which a decay function is applied. The function weighs distant and recent events appropriately and permits compounding evidence to accumulate against entities.

A set of *grouping functions* define spatial relevance. A grouping function maps an entity to other entities that share behavioral properties. More than one grouping function can (and should) be defined, and they may be singular in nature (*i.e.*, an entity is in a group by itself). The temporal history of each spatial grouping is considered, resulting in multiple reputation values. These component reputations are then combined so that a single entity is evaluated based on multiple contexts of negative feedback.

In the remainder of this section the model is formalized. First, the computation and its normalization are discussed, and following that, the feedback database is presented.

3.1 Reputation Computation

The goal of the reputation computation is to produce a quantified value that captures both the spatial and temporal properties of the entity being evaluated. Spatially, the size of the grouping must be considered, and temporally, the history of negative feedback must be weighted in proportion to its spatial relevance.

To capture these properties, three functions are required – two temporal and one spatial:

- $\boldsymbol{hist(\alpha, G, H)}$ is a temporal function returning a list of pairs, (t_{in}, t_{out}), representing listings from the feedback history, H, according to the grouping of entity α by grouping function G. The values t_{in} and t_{out} are time-stamps bounding the active duration of the listing. Active listings return $(t_{in}, \perp)$.

- $\boldsymbol{decay(t_{out}, h)}$ is a temporal function that exponentially decays input times using a half-life h, and it takes the form $2^{-\Delta t/h}$ where $\Delta t = t_{now} - t_{out}$ is of the same unit as h. It returns a value in the range $[0, 1]$, and for consistency, $decay(\perp, h) = 1$.

- $\boldsymbol{size(\alpha, G, t)}$ is a spatial function returning the magnitude, at time t, of the grouping defined by G, of which α is/was a member. If G defines multiple groupings for α, only the magnitude of one grouping is returned. The choice of group is application specific.

Raw reputation can be defined as follows:

$$\boldsymbol{raw\_rep(\alpha, G, H)} = \sum_{\substack{(t_{in},t_{out})\in \\ hist(\alpha,G,H)}} \frac{decay(t_{out}, h)}{size(\alpha, G, t_{in})} \quad (1)$$

This computation captures precisely the spatio-temporal properties required by PRESTA. Temporally, the listing history of an entity/group is captured at each summation via the *hist*() function, and events occurring recently are more strongly weighted via the *decay*() function. Spatially, grouping function G defines the group membership, and each summation is normalized by the group size.

When two or more grouping functions are defined over the entities, multiple computations of *raw_rep*() are performed. Each value encodes the reputation of an entity when considered in a different spatial context. How to best combine reputation is application specific, and for the spam application, machine learning techniques are used (see Sec. 5.7).

The values returned by *raw_rep*() are strictly comparable for all spatial groupings defined by G and the history H. High values correspond to less reputable entities and vice-versa. However, it is more typical for reputation systems [14, 16] to normalize values onto the interval $[0, 1]$ where lower values correspond to low reputation and vice-versa. Ultimately, machine learning does not require normalized values. Such values do, however, enable the model to be consistent with other reputation systems and provide an absolute interpretation that permits manually-authored policies (*e.g.*, allow access where $reputation > 0.8$).

Normalization requires knowledge of an upper bound on the values returned by *raw_rep*(). This cannot be generally defined when the de-listing policy is non-regular. However, if listings expire after a fixed duration d (or a greatest lower-bound for d can be computed), then it is possible to compute an upper bound. Such a bound is found by considering an entity who is as bad as possible; one that is re-listed immediately after every de-listing, and thus, is always active in the feedback database. Considering a grouping of size 1, the *raw_rep*() computation reduces to a geometric sequence:

$$\texttt{MAX\_REP} = 1 + \frac{1}{1 - 2^{-d/h}} \quad (2)$$

Similarly, the same worst case reputation occurs for groups of larger size, however, instead of a single entity acting as a bad as possible, the entire group is simultaneously re-listed immediately following each de-listing. Normalized reputation is now defined as:

$$\boldsymbol{rep(\alpha, G, H)} = 1 - \left(\frac{raw\_rep(\alpha, G, H)}{\texttt{MAX\_REP}}\right) \quad (3)$$

This reputation computation can be modified depending on the entities being evaluated or the nature of the negative feedback database. For example, one can eliminate spatial relevance by using grouping functions that define groups of size 1. Or, one can eliminate all temporal aspects by defining the return of *decay*() as a constant (C). Both such usages are later employed in spam detection; the former due to dynamism in IP address assignment, and the latter due to properties of the blacklist in question. Note that when $decay(t_{out}, h) = C$, $\texttt{MAX\_REP} = decay(\perp, h) + C$.

3.2 Feedback Database

The feedback database, H, depends on the nature of feedback available. PRESTA is most adept at handling *expiring* feedback like that present in IP blacklists. By definition, an expiring feedback occurs when an entity is active (listed) in the database before removal (de-listed) after a finite duration. In this case, H is a record of the entries/exits of listings such that the active database can be reproduced at any point in time.

Feedback can also be *discrete*, where negative feedbacks are associated with a single time-stamp. This is the model most often seen in general-purpose reputation management systems [14, 16]. In such cases, *hist*() always returns pairs of the form $(t_{in}, \perp)$, and thus the associated listings do not decay. A discrete database can be transformed into a compatible H by setting an artificial timeout x, (*e.g.,* $(t_{in}, t_{in} + x)$). Further, listings should not *overlap* (*i.e.,* an entity having multiple active listings). Spam blacklists are inherently non-overlapping, and pre-processing can be applied over feedbacks when this is not the case.

4. SPAM DETECTION SETUP

As presented, PRESTA defines a general model for reputation. Here, we apply PRESTA for the purpose of spam detection. Two properties of spam and IP blacklists are well leveraged by PRESTA. First, spammers are generally found "near" other spammers, and their identifiers, IP addresses, can be spatially grouped based on the IP address hierarchy. Second, blacklists are a rich source of temporal data.

It should be noted that other sources of negative feedback besides IP blacklists could be employed by PRESTA. Any manner of negative feedback associating spamming and IP addresses is sufficient. IP blacklists, however, are a well-regarded and generally trusted source of negative feedback. They are centrally maintained and reputation computed over them can be seen as a good global quantifier. IP blacklists do have weaknesses, and readers should take care not to associate these flaws to the PRESTA model.

4.1 Data Sources

***Blacklists*:** To collect blacklist data, we subscribed to a popular blacklist-provider, Spamhaus [7]. The arrival and exit of IP addresses listed on three Spamhaus blacklists (updated at thirty-minute intervals) were recorded for the duration of the experiment:

- POLICY BLOCK LIST (PBL): Listing of dynamic IP addresses (*e.g.*, those provided by large ISPs such as Comcast or Verizon).
- SPAMHAUS BLOCK LIST (SBL): Manually-maintained listing of IPs of known spammers/organizations. Typically these are IPs mapping to dedicated spam servers.
- EXPLOITS BLOCK LIST (XBL): Automated listing of IPs caught spamming; usually open proxies or machines that have been compromised by a botnet.

As the latter two blacklists contain IP addresses known to have participated in spamming, only these are used to build

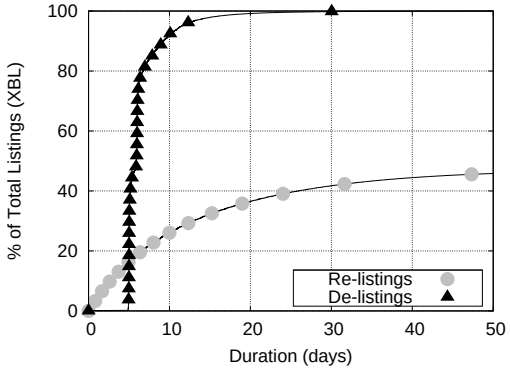

Figure 1: XBL Durations & Re-listing Rates

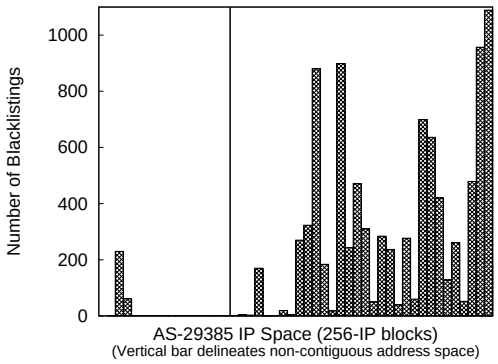

Figure 2: Behavioral Variance within an AS

reputation. The PBL is a preventative measure (however, it is used when examining blacklist performance) which lists hosts that should never be sending email, on principle.

The mechanism by which a blacklist entry occurs, be it accurate or otherwise, is beyond the scope of this work. Removal from the blacklist takes two forms: manual de-listing and timed-expiration. Given its rigorous human maintenance, the SBL follows the former format. The XBL, on the other hand, defaults to a more automated time-to-live de-listing policy. Empirical evidence shows the bulk of such listings expire 5-days after their appearance (see Fig. 1). However, in the case a blacklisted party can demonstrate its innocence or show the spam-generating exploit has been patched, manual removal is also an option for the XBL. Manual de-listings can complicate the calculation of `MAX_REP`, but as we will show, worst case spamming behaviors are rarely realized, permitting strong normalization.

***AS Mappings*:** For the purpose of mapping an IP address to the Autonomous System(s) (AS(es)) that *homes* or *originates* it, CAIDA [2] reports are used. These are compiled from Route Views [8] data and are essentially a snapshot of the BGP routing table.

***Email Set*:** The timestamp and connecting IP address of approximately 31 million email headers were collected at the University of Pennsylvania's engineering email servers between 8/1/2009 and 12/31/2009. The servers host approximately 6,100 accounts, of which roughly 5,500 serve human-users, while the remaining are for various administrative and school uses (*e.g.*, aliases, lists, *etc.*).

A considerable number of emails (2.8 million) in the dataset were both sent and received within the university network. Such emails are not considered in the analysis. Many intra-network messages are the result of list-serves/aliasing, and by excluding them, only externally arriving emails are considered. Our working set is further reduced to 6.1 million emails when analysis is conducted "above the blacklist," or those mails not currently listed on a blacklist (see Sec. 5.1).

A Proofpoint [6] score was provided with each email to categorize it as either spam or ham (not spam). Proofpoint is a commercial spam detection service employed by the University whose detection methods are known to include proprietary filtering and Bayesian content analysis [24] similar to that employed by SpamAssassin [1]. Proofpoint claims extremely high accuracy with a low false-positive rate. Given no other consistent scoring metric and a lack of access to the original email bodies, the Proofpoint score is considered the ground truth in forthcoming analysis.

4.2 Temporal Properties of Spamming IPs

PreSTA leverages the temporal properties of IP blacklists by aggregating the de-listings and re-listings of blacklist entries. Fig. 1 displays the analysis of those two statistics. Of IP-addresses de-listed during the experiment period, 26% were re-listed within 10 days. Overall, 47% of such IPs were re-listed within 10 weeks, and it is precisely such statistics that motivate PreSTA's use of temporal data.

Given that IP addresses are frequently re-listed, we examined the rate at which de-listing occurs; 80% of XBL entries were de-listed at, or very close to, 5 days after their entry (Fig. 1). Even so, this 5-day interval is not fixed. Despite a non-exact expiration, `MAX_REP` is well computed using $d = 5$ (days). Raw reputation values rarely exceeded the calculated `MAX_REP` (less than 0.01% of the time).

The SBL requires a manual confirmation of innocence before de-listing can occur and has no consistent listing length. Thus, `MAX_REP` computation cannot proceed as with the XBL. Instead, the strong assurance provided by de-listing events can be leveraged in reputation calculation. A de-listed IP was verified to be non-spamming, and so there is no reason to decay entries as they exit the list. Formally, $\forall t_{out},\ decay(t_{out}) = 0$, but as previously, $decay(\perp) = 1$. In such circumstances, the `MAX_REP` value for such IPs is computed as 1 (*i.e.*, the IP address is currently listed).

Adjusting the $decay()$ function in this way permits the reputations' of SBL IPs to be based solely on spatial properties. This is a feature of the reputation model, as it allows for flexibility in weighing context when it comes to spatial and temporal information. In a similar way, one can focus solely on temporal properties by defining singular groups, and both produce useful spam classifications (see Sec. 5.7).

4.3 Spatial Properties of Spamming IPs

The hierarchical nature of IP address assignment provides natural spatial groupings for use by PreSTA. Starting at the lowest level, a local router or DHCP service assigns IP addresses to individual machines. The selection pool is likely well-bounded to a subnet (*i.e.,* a /24 or /16). In turn, these routers operate within an ISP/AS, which get their allocations from Regional Internet Registries (RIRs), whose space is delegated from the Internet Assigned Number Authority [4] (IANA). A clear hierarchy exists, and at each level, a

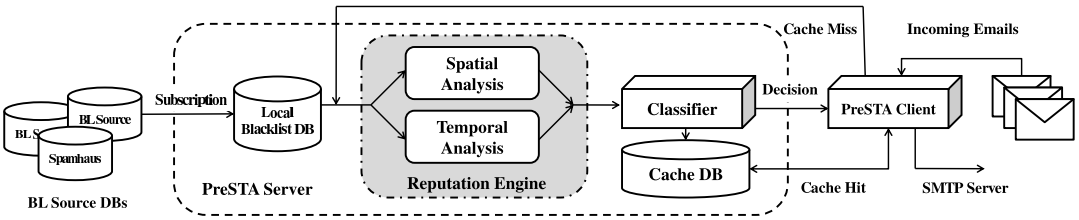

Figure 3: PreSTA **Spam Detection Architecture**

unique reputation can be applied. We focus our groupings at the following three levels: (1) the AS(es) that home(s) the IP, (2) the 768-IP block membership (a rough approximation of a subnet), and (3) the IP address itself.

Despite its easily partitioned nature, it remains to be shown that the IP assignment hierarchy provides relevant groupings. Previous work and anecdotal evidence suggest that AS-number is one of the strongest identifiers of spammers. Indeed, entire AS/ISPs, such as McColo [17] and 3FN [18], have been shut down as a result of their malicious nature. Moreover, in [11], AS-level identifiers were used as a reliable indicator of spamming hosts – indicating that 20 ASes host nearly 42% of spamming IPs.

At the subnet level, it was found that groupings of 768 IP-addresses (*i.e.*, three adjacent /24s) well contain malicious activity (see Sec. 5.5 for details). Fig. 2 visualizes the quantity of of XBL listings in /24 blocks of the address space for an ISP in Uzbekistan. Clearly, there is strong variance across the address space – some regions are highly listed while others are not. The AS-level reputation of this ISP is comparatively poor due to the quantity of listings, but within the address space, certain block-level reputations are ideal. This suggests that AS-level reputation alone may be too broad a metric.

Finally, using a grouping function that singularly groups entities effectively removes spatial relevance from reputation computation. Intuitively, the reputation of a single IP address should be considered because many mail servers use static addresses. However, the often dynamic nature of address assignment implies that unique IP addresses are not singular groupings, but rather, could represent many different machines over time. A recent study reported that the percentage of dynamically assigned IP addresses[1] on the Internet is substantial and that 96% of mail servers using dynamic IPs send spam almost exclusively [31].

5. SPAM IMPLEMENTATION

In this section the implementation of PreSTA for spam detection is described. It is designed with three primary goals: It should produce a classifier that is (1) lightweight; (2) capable of detecting a large quantity of spam; and (3) do so with a low false-positive rate. Design decisions are justified with respect to these goals. Further, the practical concerns of such an implementation are discussed.

The work-flow begins when an email is received and the connecting IP address and timestamp are recorded. Assuming the IP is not actively blacklisted, PreSTA is brought to bear. The IP is mapped to its respective spatial groupings: itself, its subnet, and its originating AS(es). Reputations are calculated at each granularity and these component reputations are supplied as input to a machine-learning classifier trained over previous email. The output is a binary ham/spam label along with each of the three component reputations – all of which may be used by a client application. This procedure is now described in detail, and a visual reference of the PreSTA work-flow is presented in Fig. 3.

5.1 Traditional Blacklists

In Sec. 4.1 the Spamhaus blacklists were introduced. They not only provide the basis on which reputations are built, but in an implementation of PreSTA, it is natural to apply them as intended – to label emails originating from *currently active* IPs as spam. When applied to the email data-set, the blacklists (PBL included) captured 91.0% of spam with a 0.74% false-positive rate. This detection rate is somewhat higher than previous published statistics[2] [15].

Had the intra-network emails not been excluded from analysis, the blacklists would have captured a similar 90.9% of spam emails with a much-reduced 0.46% false-positive rate. The exclusion of such emails, while inflating false-positive rates, permits concentration only on the more interesting set of externally-received emails and does not bias results. The usage of blacklists (independent of spatio-temporal properties), enables fast detection of a large portion of spam emails with minimal time and space requirements – the active listing requires roughly 100MB of storage.

Given the temporal statistics presented in Sec. 4.2, we also experimented with increasing the blacklists' listing period to determine if simple policy changes could greatly affect blacklist performance. This was not the case; increasing the active duration of expired listings (but not those suspected of being manually de-listed) by 5 days increased the detection rate less than 0.05%, and longer listing durations show minimal accuracy improvements at the expense of significant increases in false-positive rates.

5.2 Historical Database

Before reputation can be calculated, a historical feedback database must be in place. As described, Spamhaus blacklists are retrieved at 30-minute intervals. The `diff` is calculated between consecutive copies and time-stamped entries/exits are written to a database. When a new listing appears, the spatial groups (IP, subnet, and AS(es)) that IP is a member of are *permanently* recorded. For example, if IP i was blacklisted as a member of AS a, that entry will always be a part of a's blacklist history.

Roughly 1GB of space is sufficient to store one month's blacklist history (the XBL has 1.0–1.5 million IPs turn over on a daily basis). Fortunately, an extensive history is not

[1]Recall that Spamhaus' PBL blacklist is essentially a listing of dynamic IP addresses. It is constructed mainly using ISP-provided data, and as such, is far from a complete listing.

[2]Our analysis of blacklist performance is from a single-perspective and may not speak to global effectiveness.

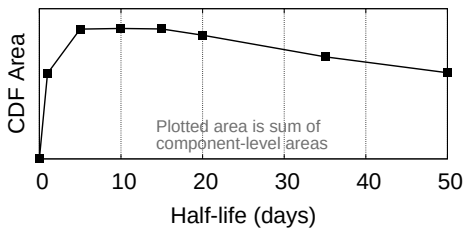

Figure 4: Affect of Half-Life on CDF Area

required given the exponential *decay*() function[3]. For example, given a 10-day half-life, a 3-month old XBL entry contributes 0.6% the weight of an active listing. Lengthy histories offer diminishing returns. To save space, one should discard records incapable of contributing statistical significance. Further, such removal saves computation time because the smaller the set *hist*() returns, the fewer values which must be processed by *raw_rep*().

5.3 Grouping Functions

Given an entity (IP address) for which to calculate reputation, three grouping functions are applied:

- IP Function: An IP is a group in and of itself, so such a grouping function mirrors its input.
- Subnet Function: IP subnet boundaries are not publicly available. Instead, an estimate considers blocks of IP addresses (we use the terms "subnet-level" and "block-level" interchangeably). IP space is partitioned into /24s (256 IP segments), and an IP's block grouping consists of the segment in which it resides as well as the segment on either side; 768 addresses per block. Thus, block groupings overlap in the address space, and a single IP input returns one block of IPs (three /24s). Although such estimations may overflow known AS boundaries, these naïve blocks prove effective.
- AS Function: Mapping an IP to its parent AS(es) requires CAIDA [2] and RouteViews [8] data. Note that some AS boundaries overlap in address space and some portions of that space (*i.e.,* unallocated portions) have no resident AS whatsoever. An IP can be homed by any number of ASes, including none at all, the technical considerations of which are addressed in Sec. 5.5. The function's output is all the IPs homed by an AS(es) in which the input IP is a member. Each returned IP is tagged with the parent AS(es), so a well-defined subset of the output can be chosen.

5.4 Decay Function

The decay function (Sec. 3.1) controls the extent to which temporal proximity factors into reputation. It is configured via its half-life, h. If h is too small, reputations will decay rapidly and provide little benefit over using blacklists alone. Too large an h will cause an increase in false positives due to stale information.

[3]This minimal history requirement was of benefit to this study. Reputations must *warm-up* before their use is appropriate. Indeed, collection of blacklist data began in 5/2009, three months before the first classifications.

A good half-life will maximize the difference between the reputations of spam and ham email. Analyzing email predating the evaluation period, the reputation-CDFs for both spam and ham emails (as in Fig. 6) were plotted using different h, seeking to maximize the area between the curves. In Fig. 4 the calculations from these experiments are presented. A value of $h = 10$ (days) was found optimal and this value is used in the spam application[4]. With the half-life established and having chosen $d = 5$ (days), `MAX_REP`$= 4.14$.

As described previously, two separate *decay*() functions are employed depending on whether a listing appeared on the SBL or the XBL. Manually maintained, de-listing from the SBL is not decayed, but the XBL is decayed using the aforementioned 10-day half-life. A special flag attached to each time pair returned by *hist*() allows both listings to be used in combination.

5.5 Reputation Calculation

Given the feedback database (Sec. 5.2), output (sets of IP addresses) of the three grouping functions (Sec. 5.3), and the decay function (Sec. 5.4), reputation may now be calculated at each granularity, returning three reputation values. Calculation closely follows as described in Sec. 3.1.

Calculation of IP-level and subnet-level reputation is straightforward per the reputation model with $size() = 1$ and $size() = 768$, respectively. The particulars of AS-level calculation are more interesting. An IP may be a member of any quantity of ASes, including none at all. If an IP is multi-homed, the conservative choice is made by selecting the most reputable AS-level reputation. Those IPs mapping to no AS form their own group, and the reputation for this group is designated as 0 because, in general, unallocated space is only used for malicious activity (see Sec. 7). In this spatial grouping, *size*() is not constant over time. Instead, magnitudes are pre-computed for all AS using CAIDA data and updated as BGP routes change.

5.6 Calculation Optimizations

PreSTA must calculate reputation efficiently to achieve the desired scalability. It should not significantly slow email delivery (latency), and it should be capable of handling heavy email loads (bandwidth). Caching strategies and other techniques that support these goals are described below:

- AS Value Caching: Reputations for *all* ASes are periodically recalculated off-line. Calculation is (relatively) slow given that *hist*() calls return large sets.
- Block/IP Value Caching: Similarly, block and IP reputations can be cached after the first cache miss. Cache hit rates are expected to be high because (1) an email with multiple recipients (*i.e.,* a carbon copy) is received multiple times but with the same source IP address, and (2) source IP addresses are non-uniformly distributed. For the 6.1 million (non-intra-network, non-blacklisted) emails in the working data-set, there are 364k unique IP senders and 176k unique 'blocks.'
- Cache Consistency: Caches at all levels need to be flushed when the blacklists are updated (every 30 minutes), to avoid inconsistencies involving the arrival of

[4]Although it was found unnecessary, h could be optimized on an interval basis, much like re-training a classifier. However, experiments showed minor variations of the parameter to be inconsequential.

new listings. As far as time-decay is concerned, a discrepancy of up to 30 minutes is inconsequential when considering a 10-day half-life.

- WHITELISTING: There is no reason to calculate reputation in trusted IP addresses, such as one's own server. Of course, whitelists could also be utilized in a feedback loop to alleviate false-positives stemming from those entities whose emails are misclassified.

Using these optimizations, the PRESTA implementation is capable of scoring 500k emails an hour, with average email latency on the order of milliseconds[5]. Latency and bandwidth are minimal concerns. Instead, it is the off-line processing supporting this scoring which is the biggest resource consumer. Even so, the implementation is comfortably handled by a commodity machine and could easily run adjacent to an email server. Pertinent implementation statistics, such as cache performance, are available in Sec. 6.4.

5.7 Reputation Classification

Extraction of a binary classification (*i.e.,* spam or ham) is based on a *threshold* strategy. Emails evaluated above the threshold are considered ham, and those below are considered spam. Finding an appropriate threshold can be difficult, especially as dimensionality grows, as is the case when classifying multiple reputation values. Further, a fixed threshold is insufficient due to temporal fluctuations; as large groups (botnets) of spamming IPs arise and fall over time, the distinction between good and bad may shift.

A *support vector machine* (SVM) [13] is employed to determine thresholds. SVM is a form of supervised learning that provides a simple and effective means to classify multiple reputation values. The algorithm maps reputation triples (a feature for each spatial dimension) from an email training set into 3-dimensional space. It then determines the surface (threshold) that best divides spam and ham datapoints based on the training labels. This same threshold is then applied during classification. The SVM routine is tuned via a *cost* metric that is correlated to the eventual false-positive rate of the classifier.

The classifier is adjusted (re-trained) every 4 days to handle dynamism. A subset of emails received in the previous 4 days are trained upon, and the resulting classifier is used for the next 4 day interval. The affect of different training periods has not been extensively studied. Clearly, large periods are not desired; the reputation of distant emails may not speak to the classification of current ones. Too short a period is poor because it requires extensive resources to retrain so frequently. Analysis found 4-day re-training to be a good compromise. However, the re-training period need not be fixed, and future work will explore re-training rates that adjust based on various environmental factors.

At each re-training, 10,000 emails (5% of the non-intra-network, non-blacklisted email received every 4 days) were used, and emails were labeled as spam/ham based on the Proofpoint score. In a more general use case, there would be some form of client feedback correlated across many accounts that can classify spam post-delivery and train various spam detectors. Since we do not have access to such user behavior, correlation statistics, or any external spam filters, the provided Proofpoint values are assumed.

[5]Statistics are based on a single-threaded implementation. Concurrency and other programming optimizations would likely improve PRESTA's performance and scalability.

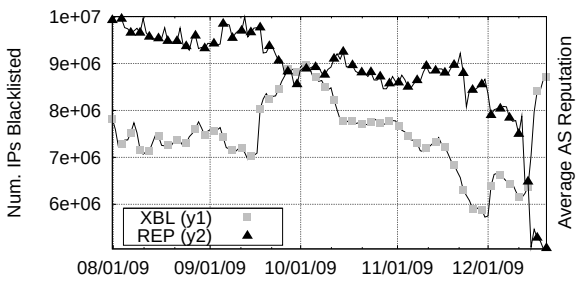

Figure 5: XBL Size Relative to Global Rep.

Post-training, the false-positive (FP) rate of the classifier is estimated by measuring the error over the training set (assuming one does not over-fit the training data). The estimated FP-rate is a good indicator of the true FP-rate, and the SVM cost parameter is adjusted to tune the expected FP-rate. All classifier statistics and graphs hereafter were produced with a 0.5% tolerance for false-positives (over the classification set), as this simplifies presentation. This FP-rate (0.5%) is a reasonable setting given that blacklists are widely accepted and achieved a 0.74% FP-rate over the same dataset. Additionally, these rates are somewhat inflated given the decision to exclude intra-network emails, which are unlikely to contribute false-positives (the blacklist FP-rate was reduced one-third to 0.46% with their inclusion). In Sec. 6.5, the trade-off between the FP-rate and spam blockage is examined in greater depth.

6. EXPERIMENTAL ANALYSIS

Experimental analysis begins by examining component reputations individually. From there, two case studies are presented which exemplify how PRESTA produces metrics outperforming traditional blacklists in both spatial and temporal dimensions. Finally, the detection results of the PRESTA spam filter are presented.

To best simulate a real email server load, it is assumed emails arrive in the order of their timestamps and are evaluated relative to this ordering. Additionally, cache population/flushing and classification re-training are performed at the relative time-intervals outlined in the previous section.

6.1 Blacklist Relationship

In examining how reputations quantify behavior, we apply a simple intuition: One would expect to see a clear push-pull relationship between an entity's reputation and the number of corresponding entries on the blacklist. To confirm this hypothesis, the size of the XBL blacklist[6] was graphed over time and compared to the average reputation of *all* ASes. Results are presented in Fig. 5. An inverse relationship is observed, confirming the hypothesis. When the number of listings decreases, reputation increases – and vice versa.

6.2 Component Reputation Analysis

In order for component reputations (IP, block, and AS) to be useful in spam detection they must be *behavior predictive*. That is, the reputations of ham emails should exceed those

[6]The XBL is the driving force behind reputation. The SBL is also a contributor, but is orders of magnitude smaller.

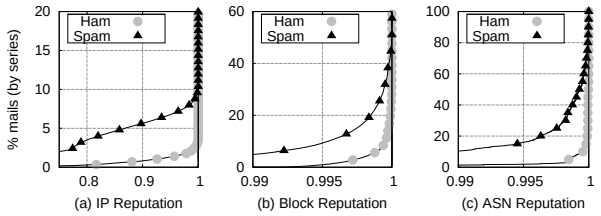

Figure 6: CDFs of Component Reputations

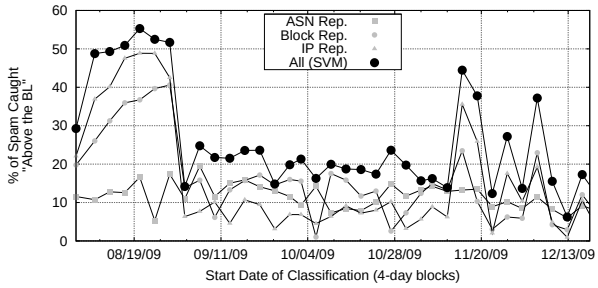

Figure 7: Component Reputation Performance

of spam emails. This relationship is visualized in the CDFs of Fig. 6. All component reputations behave as expected. Fig. 6 also displays the benefit of multiple spatial groupings. While 90% of spam emails come from IPs that had ideal reputation (*i.e.,* a reputation of 1) at the time of receipt, this is true for just 46% of blocks, and only 3% of AS.

The CDFs of Fig. 6 imply that each component reputation is, in and of itself, a metric capable of classifying *some* quantity of spam. However, it is desirable to show that each granularity captures *unique* spam, so that the combination of multiple reputations will produce a higher-order classifier of greater accuracy. In Fig. 7, the effectiveness of each component reputation is presented. The percentage of spam caught is "above the blacklist," or more precisely, the percentage of spam well-classified by the reputation value that was not identified by the blacklist alone[7]. Crucially, the combined performance (the top line of Fig. 7), exceeds that of any component, so each spatial grouping catches spam the others do not. On the average, PreSTA is able to capture 25.7% of spam emails not caught by traditional blacklists.

We are also interested in determining which grouping provides the best classification. AS-level reputation is the most stable of the components, individually capable of classifying an additional 10-15% of spam above the blacklist. However, during periods of increased PreSTA performance, it is often the block and IP levels that make significant contributions. This is intuitive; AS-level thresholding must be conservative. Given their large size, the mis-classification of an AS could result in an unacceptable increase in the FP-rate. Meanwhile, the cost associated with a mis-prediction is far less for block and IP groupings.

These results suggest that considering more spatial dimensions should increase performance, that is, when there are non-overlapping classifications. However, there are diminishing returns. Each additional component reputation requires increased resources in evaluation and classification.

[7]Given the inclusion of traditional blacklist filtering, the primary concern is those emails that are not actively listed.

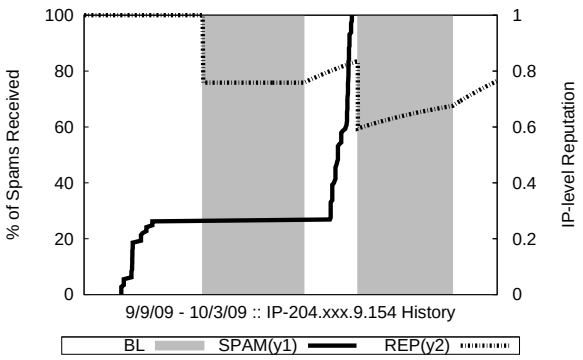

Figure 8: Single IP Behavior w.r.t. Blacklisting

An application should seek a minimal set of dimensions to best represent and classify its data.

6.3 Case Studies

Two case studies exemplify the types of spam behavior able to evade blacklists, yet captured via PreSTA. First, Fig. 8 shows the *temporal* sending patterns of a single spamming IP address. This IP was blacklisted twice during the course of the study (as indicated by shaded regions), and the time between listings was small (≈ 2 days). The controller of this IP address likely used blacklist counter-intelligence [22] to increase the likelihood that spam would be delivered: Notice that no spam was observed when the IP was actively listed, but 150 spam emails were received at other times.

Traditional blacklist are reactive, binary measures that do not take history into account. During the intermittent period between listings, as far as the blacklist is concerned, the spamming IP is an innocent one. However, as shown in Fig. 8, the IP-level reputation metric compounds prior evidence. Thus, if PreSTA had been in use, the intermittent influx of email likely would have been identified as spam.

Secondly, Fig. 9 visualizes a case study at the AS-level utilizing both *spatial* and *temporal* dimensions. In the early stages of data collection anomalous activity was noticed at a particular AS (AS#12743)[8]. Even when compared to the other four worst performing ASes during the time block, ASN-12743's drop in reputation is astounding. Nearly its entire address space, some 4,500 addresses, were blacklisted over the course of several days – likely indicative of an aggressive botnet-based spam campaign. After this, the reputation increases exponentially (per the half-life), eventually returning to innocent levels.

With traditional blacklists, an IP must actually send spam before it can be blacklisted. In the ASN-12743 case, this means all 4,500 IPs had some window in which to freely send spam. However, as the IPs were listed in mass, the *reputation* of the AS drops at an alarming rate, losing more than 50% of its value. Had PreSTA been implemented, the reputation of the AS (and the blocks within) would have been low enough to classify mails sourced from the remainder of the space as spam, mitigating the brunt of the attack.

6.4 Implementation Performance

An important aspect of PreSTA is its scalability, and to best evaluate this our PreSTA simulation mimicked the

[8]PTK-Centertel, a major Polish mobile service provider.

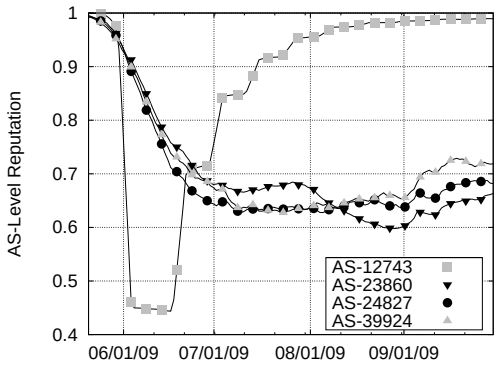

Figure 9: Temporal Shift within Grouping (AS)

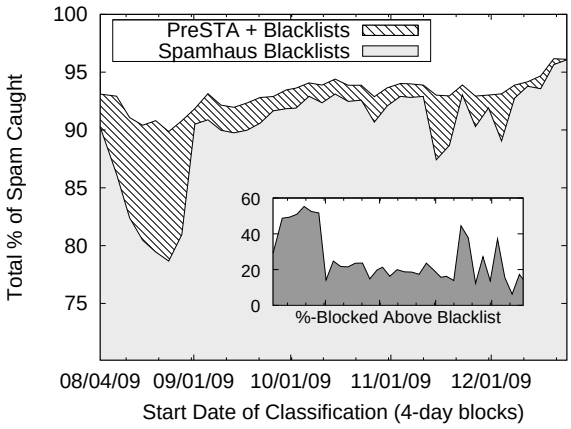

Figure 10: Blacklist and PreSTA Blockage

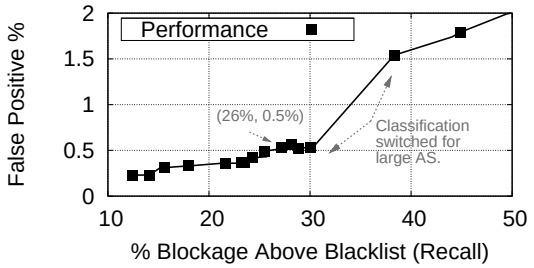

Figure 11: Characteristic ROC Trade-Off

normal processing of a mail server. The blacklist history and cached reputation scores were regulated so that only the knowledge available at the time of arrival is used to evaluate an email. PreSTA requires a warm-up period to gather enough temporal knowledge to process correctly; hence, historical blacklist storage began three months prior to the first email being scored.

The effectiveness of the cache and the latency of the system is also of interest. Caching is highly effective: 56.8% of block-level calculations are avoided, and 43.1% are avoided at the IP-level (recall that *all* AS-level calculations are performed off-line and then cached). The reputation of an incoming email is calculated in nearly real time, with the average email being processed in fractions of a second. Under typical conditions, over 500,000 emails can be scored in an hour, using commodity hardware.

Re-training the classifiers and rebuilding the AS-cache are the most time consumptive activities. Fortunately, training new classifiers takes only minutes of work for a 10,000 email training set, and only needs to be performed every 4 days. Re-training is also done off-line and does not affect current scoring. Rebuilding the AS reputation cache must be done every 30 minutes, once new blacklist data is available, but it need not delay current scoring as the previous AS-level reputations are still relevant (at most 30 minutes old).

6.5 Spam Mitigation Performance

The spam detection capabilities of PreSTA are summarized in Fig. 10. On average, 93% of spam emails are identified when used in conjunction with traditional blacklists. This may seem to be a nominal increase over blacklists alone; however, the inset of Fig. 10 is more intuitive – PreSTA blocks between 20% and 50% of those mails passing the Spamhaus blacklists, with a 25.7% average (identical to the top line of Fig. 7). Had PreSTA been implemented on our university mail server during the study, it would have caught 650,000 spam emails that evaded the Spamhaus blacklists.

Most interestingly, PreSTA provides consistent and steady state detection. For example, consider the significant variations in blacklist performance seen throughout the study (for example, in late August 2009 and in mid-November 2009). PreSTA is nearly unaffected during these periods and may be a useful stop-gap during variations in blacklist accuracy. While the blockage-rates of the blacklists fluctuate 18% over the course of the study, PreSTA is far more consistent, exhibiting just 5% of variance. Further, it is likely that continued analysis will show similar variations in blacklist performance. Periods of high de-listing are likely followed by periods of high re-listing as spammers try to maximize the utility of available IPs.

Ultimately, the performance attainable by the classifier is dependent on the number of false-positives (FPs) tolerated. To this point, the FP-rate has been fixed at 0.5%; however, as exemplified in Fig. 11, the FP-rate is tune-able and strongly correlates with the blockage rate. The plot is generated over a characteristic interval of email from mid-October 2009, and is akin to the precision/recall graphs common in machine-learning. We remind readers that the decision to exclude intra-network emails from the dataset (see Sec. 5.1) significantly inflates the presented FP-rates.

7. EVASION AND GAMESMANSHIP

To be effective, PreSTA must be robust to evasion and gamesmanship – an entity should be unable to surreptitiously influence its own reputation. Given the use of IP blacklists as a feedback source, the most effective way to avoid PreSTA is to avoid getting blacklisted. However, such a technique is not fail-safe; a single evasive entity may still have poor reputation at broader granularity. When negative feedback exists, and an IP has been blacklisted, the best recourse is patience. Over time, the weight of the listing decays. As such, there is no way to evade PreSTA in the temporal dimension.

However, spammers are migrant and the spatial dimension affords greater opportunities. While IP and block magnitudes are fixed, an AS controls the number of IPs it broadcasts. An actively evasive AS would ensure its entire allo-

cation is broadcasted. More maliciously, a spammer may briefly hijack IP space they were *not* allocated in order to send spam from stolen IPs. Such *spectrum agility* was shown by [21] to be an emergent spamming technique. Fortunately, if the hijacked IP space was not broadcasted (*i.e.,* unallocated), emails from these IPs would map to the special grouping "no AS", whose reputation is zero (per Sec. 5.5). However, if the hijacked space was broadcasted by a reputable AS, evasion may be possible. Fortunately, [21] observes the use of unallocated space is most prevalent.

The previous scenario can be described as a *sizing attack* and is of most concern to PreSTA. The entities being evaluated should not be able to affect the size of their spatial groupings. However, this attack is only effective when the group size is non-singular, and an simple mitigation technique is to always include a grouping function defining singular groups. Further, an implementation should assign persistent identifiers to entities. When identifiers are non-persistent, PreSTA could fall victim to a Sybil attack [10] since an entity could evade negative feedback by simply changing identifiers.

8. CONCLUSIONS

In this paper, we have introduced PreSTA, a novel reputation model designed to combine the rich historical information of blacklists and the spatial relationships of spamming IPs. We have shown PreSTA reputations to be an effective measure for classifying spam, identifying up to 50% of spam not caught by blacklists alone. Our preliminary implementation, which combines PreSTA with blacklists, mitigates 93% of spam on average and is stable – reducing the effects of blacklist fluctuations. Finally, PreSTA proves scalable and is able to efficiently handle production email workloads, processing over 500k emails an hour.

Having demonstrated PreSTA's proficiency in the field of spam detection, one must consider how this capability is best utilized. Although we make no claims it can (or should) replace content-based filtering, PreSTA could be applied as an initial filter or grey-listing mechanism. Alternatively, the system could be used to prioritize the processing of incoming email in high-volume situations. Since it is based on centralized blacklist information, PreSTA could be installed as a parallel service provided by blacklist providers.

Further, PreSTA's applicability is broader than email spam. PreSTA has already proven effective in the detection of Wikipedia vandalism [30] and shows promise in other domains ranging from prioritization of BGP announcements to reputation for web-based service *mash-ups*. Any service that requires dynamic decision making and has access to records of historical feedback is a candidate. Ultimately, PreSTA reputations may be utilized as an effective means of performing dynamic access-control and mitigating malicious behavior, two extremely relevant issues as service paradigms shift to more distributed architectures.

9. ACKNOWLEDGMENTS

The authors would like to thank Wenke Lee and David Dagon, both of Georgia Tech, for their initial guidance on this project. Additional thanks go to Charles 'Chip' Buchholtz of UPenn-CETS, who performed mail dumps and aided us in obtaining permission to process those logs.

References

[1] Apache SpamAssassin. http://spamassassin.apache.org/.

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

[3] DNSBL.info: Blacklists. http://www.dnsbl.info/dnsbl-list.php.

[4] Internet Assigned Numbers Authority. http://www.iana.org/.

[5] MessageLabs Intelligence. http://www.messagelabs.com/.

[6] Proofpoint, Inc. http://www.proofpoint.com/.

[7] Spamhaus Project. http://www.spamhaus.org/.

[8] Univ. of Oregon Route Views. http://www.routeviews.org/.

[9] M. Blaze, S. Kannan, A. D. Keromytis, I. Lee, W. Lee, O. Sokolsky, and J. M. Smith. Dynamic trust management. *IEEE Computer (Special Issue on Trust Mangement)*, 2009.

[10] J. Douceur. The Sybil attack. In *1st IPTPS*, March 2002.

[11] S. Hao, N. A. Syed, N. Feamster, A. G. Gray, and S. Krasser. Detecting spammers with SNARE: Spatio-temporal network-level automated reputation engine. In *USENIX Security Sym.*, 2009.

[12] IronPort Systems Inc. Reputation-based mail flow control. White Paper, 2002. (SenderBase).

[13] T. Joachims. *Advances in Kernel Methods - Support Vector Learning*, chapter Making Large-scale SVM Learning Practical, pages 169–184. MIT Press, Cambridge, MA, 1999.

[14] A. Jøsang, R. Hayward, and S. Pope. Trust network analysis with subjective logic. In *Proceedings of the 29th Australasian Computer Science Conference*, 2006.

[15] J. Jung and E. Sit. An empirical study of spam traffic and the use of DNS black lists. In *Proc. of the 4th ACM SIGCOMM Conference on Internet Measurement*, pages 370–375, 2004.

[16] S. D. Kamvar, M. T. Schlosser, and H. Garcia-molina. The EigenTrust algorithm for reputation management in P2P networks. In *Proc. of the Twelfth WWW Conference*, May 2003.

[17] B. Krebs. Host of Internet spam groups is cut off. http://www.washingtonpost.com/wp-dyn/content/article/2008/11/12/AR2008111200658.html, November 2008. (McColo).

[18] B. Krebs. FTC sues, shuts down N. Calif. web hosting firm. http://voices.washingtonpost.com/securityfix/2009/06/ftc_sues_shuts_down_n_calif_we.html, June 2009. (3FN).

[19] Z. Qian, Z. Mao, Y. Xie, and F. Yu. On network-level clusters for spam detection. In *Proceedings of the 17th Annual Network and Distributed System Security Symposium (NDSS)*, 2010.

[20] A. Ramachandran, D. Dagon, and N. Feamster. Can DNSBLs keep up with bots? In *Proc. of the 3rd CEAS*, 2006.

[21] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. In *Proc. of SIGCOMM 2006*, 2006.

[22] A. Ramachandran, N. Feamster, and D. Dagon. Revealing botnet membership using DNSBL counter-intelligence. In *USENIX: Steps to Reducing Unwanted Traffic on the Internet*, 2006.

[23] A. Ramachandran, N. Feamster, and S. Vempala. Filtering spam with behavioral blacklisting. In *Proc. of Computer and Communications Security (CCS '07)*, pages 342–351, 2007.

[24] M. Sahami, S. Dumais, D. Heckerman, and E.Horvitz. A Bayesian approach to filtering junk e-mail. In *AAAI-98 Workshop on Learning for Text Categorization*, 1998.

[25] S. Sinha, M. Bailey, and F. Jahanian. Improving spam blacklisting through dynamic thresholding and speculative aggregation. In *Proceedings of the 17th NDSS*, 2010.

[26] Symantec Corportation. IP reputation investigation. http://ipremoval.sms.symantec.com/.

[27] S. Venkataraman, A. Blum, D. Song, S. Sen, and O. Spatscheck. Tracking dynamic sources of malicious activity at internet scale. In *Neural Information Processing Systems '09*, 2009.

[28] S. Venkataraman, S. Sen, O. Spatscheck, P. Haffner, and D. Song. Exploiting network structure for proactive spam mitigation. In *16th USENIX Security Symposium*, pages 149–166, 2007.

[29] A. G. West, A. J. Aviv, J. Chang, V. S. Prabhu, M. Blaze, S. Kannan, I. Lee, J. M. Smith, and O. Sokolsky. QuanTM: A quantitative trust management system. In *EUROSEC 2009*, pages 28–35, March 2009.

[30] A. G. West, S. Kannan, and I. Lee. Detecting Wikipedia vandalism via spatio-temporal analysis of revision metadata. In *EUROSEC '10*, pages 22–28, Paris, France, 2010.

[31] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber. How dynamic are IP addresses? In *SIGCOMM '07*, 2007.