This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE INFOCOM 2010 proceedings
This paper was presented as part of the main Technical Program at IEEE INFOCOM 2010.

A Collaboration-based Autonomous Reputation System for Email Services

Mengjun Xie and Haining Wang
Department of Computer Science
College of William and Mary
Williamsburg, VA, USA
Email: {mjxie,hnw}@cs.wm.edu

***Abstract*—This paper presents CARE, an autonomous email reputation system based on inter-domain collaboration. Within the framework of CARE, each domain independently builds its reputation database based on both the local email history and the information exchanged with other collaborating domains. CARE examines the trustworthiness of the email histories obtained from collaborators by correlating them with the local email history. To validate the efficacy of CARE, we have analyzed real email logs, conducted a DNS-based estimation experiment, and performed a series of simulations. Our experimental results show that CARE can effectively improve the reliability and performance of email systems.**

I. Introduction

An email reputation system, which provides a goodness measure (quantitative or qualitative) of email server behavior, is in great demand given the swarm of spam over the Internet. Email reputation systems can assist email servers in deciding the action for an incoming message: directly drop/accept or apply spam filters. A well-functioned reputation system can effectively improve the performance and reliability of email service by saving cost on spam filtering and ensuring rejection of unwanted messages and acceptance of wanted messages.

Existing email reputation systems can be roughly classified into two categories based on their purposes: the systems only for spam rejection and the systems for both nonspam acceptance and spam rejection. A variety of DNS-based blacklists (DNSBLs) such as [1], [2] exemplify the former, while the Gmail reputation service [3] illustrates the latter. Compared to the reputation systems for spam rejection, the reputation systems for nonspam acceptance and spam rejection are much preferred as they also provide ratings for legitimate email senders.

Incorporating the reputation of legitimate email senders can effectively improve the reliable delivery of legitimate email. Nowadays spam filters are ubiquitously deployed to fight spam. Due to its computational intensiveness, spam filtering could become a processing bottleneck when an influx of email occurs and result in loss of nonspam messages. Moreover, aggressive spam filters may cause email loss. A recent study on email loss [4] reveals that the email accounts with spam filter lost significantly more legitimate messages than the email accounts with no spam filter[1]. Many anecdotal reports including the loss of email submissions discussed in the "end2end" mail-list [5] indicate the existence of email loss due to spam filtering. The email delivery crises emerging in aforementioned situations can be much relieved by a reputation system that rates legitimate email senders.

However, existing reputation systems that rate both spam senders and nonspam senders suffer either from being isolated or from relying on a central authority. A reputation system using only local email history is constrained by its scope of communications. Its effectiveness will be degraded when new senders continuously appear, and the constant appearance of new senders has been observed in the email logs of two universities we studied. Considering the service scale, ordinary email service providers can rarely achieve the same success as the Gmail [3] if their reputation systems only use local information. RepuScore [6] is another email reputation framework that rate both nonspam senders and spam senders. However, it requires a hierarchical architecture and a central authority to maintain the reputation database, which poses a challenge to the large-scale trust management and deployment.

In this paper we present CARE, a **C**ollaboration-based **A**utonomous email **RE**putation system rating both spam domains and nonspam domains. Working at domain level, CARE enables a domain to build its reputation database, including both frequently contacted and unacquainted email sending domains, by (1) locally recording email sending behavior of remote domains and (2) exchanging the local information with other collaborating domains. CARE examines the trustworthiness of email histories obtained from collaborators by correlating them with local email history, and integrates the local and remote information to derive the reputation of remote domains. We demonstrate the effectiveness of domain collaboration on improving the coverage of CARE by comparing two email log traces from two universities, conducting a large experiment of DNS snooping to study the collaboration among multiple domains, and performing extensive simulations in a large-scale environment.

The remainder of this paper is organized as follows. Section II briefly overviews related work in email reputation. Section

[1] The messages are neither in the inbox nor in the spam folder.

III presents the motivation of this work. Section IV details the design of CARE. Section V validates the effectiveness of CARE. Finally, Section VI concludes the paper.

II. Related Work

Email reputation systems rate email sending entities based on the history of their sending behaviors. The entity can be email address, IP address, or domain name. Some reputation systems use qualitative measures (e.g., good or bad) while others use quantitative measures (e.g., spam score is 58). A brief taxonomy of email reputation systems is given in [7].

Address based reputation systems are very popular. Email address based whitelists and blacklists, e.g., DOEmail[8], are commonly used by individuals. To defeat email address spoofing, many sender authentication schemes have been proposed, in which SPF (Sender Policy Framework) [9] and DKIM (DomainKeys Identified Mail) [10] are the most noticeable. SPF and DKIM can help identify the sending party but cannot determine its legitimacy as spammers also embrace these schemes [3], [11]. As one type of IP-address-based blacklists, DNSBLs (e.g., [1], [2]) disseminate blacklists through DNS and are widely used. However, the effectiveness of DNSBLs has been questioned [12], [13], [14], [15]. Besides DNSBLs, other types of IP-address-based reputation systems such as [16] also exist. These systems usually are commercial and use proprietary techniques for reputation maintenance and dissemination.

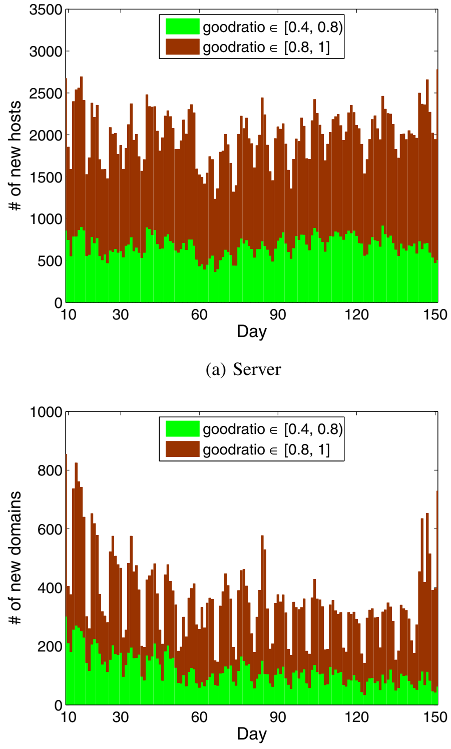
The Gmail reputation system [3] rates domains instead of IP addresses. It uses only local information and identifies the sending domain using both heuristics and SPF and DKIM. Singaraju *et al.* [6] proposed a collaborative email reputation framework called RepuScore, which also rates domains.

Leiba *et al.* [17] presented an algorithm to derive the reputation of email domains and IP addresses by analyzing the SMTP sending paths (in the message header) of known legitimate messages and spam messages. Golbeck *et al.* [18] proposed an algorithm to infer the relative reputation ratings of email contacts based on the exchange of reputation values. Chirita *et al.* [19] developed a reputation scheme called MailRank, which can compute a global reputation score as well as a personalized score for each email address. Both [18] and [19] assume the existence of global email social networks and compute reputation scores in a centralized manner.

Collaboration has been applied into the spam signature generation and email address whitelist population. Vipul's razor [20] maintains a collaborative network through which the signatures of human-identified spam are submitted and distributed. To expand whitelists in an automatic manner, LOAF [21], FOAF [22], and RE: [23] have been developed. These systems leverage the social connections to find indirect relations between senders and recipients.

III. Motivation

As demonstrated by [24] and [25], local email histories can be used to enhance the quality of email service. However, they also reveal that it is impossible to cover all incoming email messages merely based on local history. In other words, there are always messages from unseen sources. Unfortunately, the dynamics of such messages, which directly affect the performance of a local-history-based reputation system, have not yet been studied. This motivated our following study on the dynamics of incoming email.



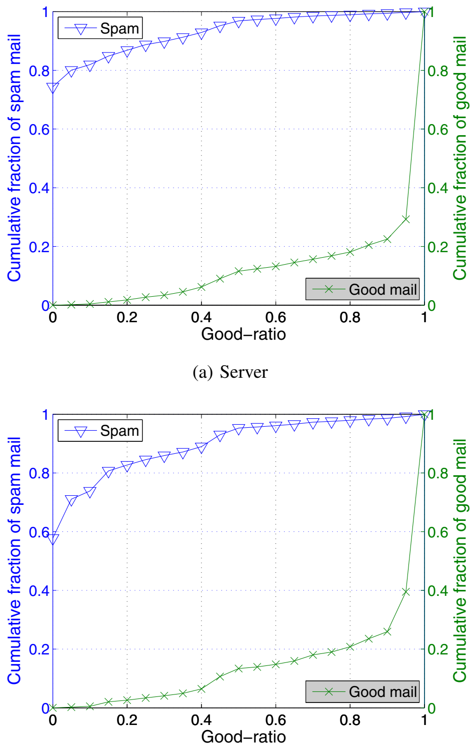
(a) Server

(b) Domain

Fig. 1. Number of newly-appeared senders per day

We collected 151-day email logs for inbound messages from our campus email servers. The logs are daily-based and span from 2007/11/01 to 2008/03/31 with only one daily log missing. For each inbound message, we logged the time of message arrival, the IP address and domain name (if any) of the sending server, and the score (between 0 and 300, the bigger, the more likely to be spam) given by the spam filter. We removed those records without valid fully qualified domain names (FQDN), since their corresponding messages are almost certainly spam. As original logs do not contain the name of the domain in which each sending server resides, we derived the domain information using `dig` and added it into the logs. We observed that a significant number of newly-appeared (never recorded by any previous logs) servers and domains consistently show up in daily logs, even after 100 days. The average numbers of newly-appeared servers and domains per day are 27,733 and 1,152, respectively. More importantly, this observation also holds for those servers and domains that mostly send legitimate email.

We use metric "good-ratio" to measure sending behavior of a server (and domain). The good-ratio of a sending server/domain is computed by dividing the number of nonspam messages over the number of total messages sent from the



(a) Server

(b) Domain

Fig. 2. CDF of good-ratio for spam and nonspam (or good) email

server/domain across all logs. Good-ratio “1” means always sending nonspam email and good-ratio “0” means always sending spam. We classify the messages with scores no greater than 10 (The default threshold is 50) as nonspam and the rest as spam. This classification results in around 1.8% false negatives (i.e., uncaught spam) and zero false positive (i.e., misclassified nonspam) in one of campus email archives, which contains 1,800 manually verified spam messages. We further reduce false negatives by applying a few well-established heuristics, such as identifying sending servers with dynamically-allocated IP addresses by domain name. Despite that false positives and false negatives may still exist after conservative classification and rectification, we believe that the misclassification is minor and should not affect our measurement results.

Figure 1 illustrates the dynamics of the numbers of the newly-appeared servers and domains whose good-ratios are no less than 0.4 in daily logs. The servers (domains) are further divided into two groups; one group with good-ratio in [0.4, 0.8) and the other with good-ratio in [0.8, 1]. In Figure 1, we can see that the number of newly-appeared servers (domains) per day is not negligible. For example, even after 100 days, newly-appeared servers with good-ratio over 0.8 per day are still counted by thousands. Compared to newly-appeared servers per day, newly-appeared domains with high good-ratios per day are counted by hundreds, still too many to be ignored.

These measurement results suggest that using only local history information may not be sufficient for building a high-quality reputation system. Intuitively, the coverage of senders can be improved through collaboration, as an email sender that is new to one receiver may be old to others. Naturally, the peers that have frequent email communications and behave consistently well become candidates of collaborator. As shown in a recent spam study [25], there exist good email servers from which most of email is nonspam for most of time. However, that analysis is based on the data from one vantage point and does not study the sender behavior at the domain level. Therefore, we use our email logs to verify if their observation holds here and up to the domain level.

Taking spam (nonspam) messages from the servers with valid domain names in all logs as a whole, we examine the proportion of spam (nonspam) contributed by the servers with a specific good-ratio. we plot the CDFs (i.e., cumulative distribution functions) of good-ratio for spam and nonspam at host level in Figure 2 (a). The curve at the left top shows the CDF for spam while the curve at the right bottom shows the CDF for nonspam. The CDFs at the domain level are shown in Figure 2 (b). We also examined the CDFs with different time windows (i.e., number of days) and time ranges (i.e., starting and ending days) and found that those CDFs are very similar to those shown in Figure 2. In general, our results conform to the findings in [25]. The servers with high good-ratios send the majority of nonspam email and the servers with low good-ratios send the most of spam, which makes the use of reputation system very helpful. For instance, the servers with good-ratios no smaller than 0.8 send over 80% of total nonspam email but less than 1% of total spam. Hence, whitelisting these servers would save a great deal of filtering resources and improve the delivery of legitimate email. Moreover, the server-level observations also apply at the domain level. The curve shapes in Figure 2 (b) are similar to those in Figure 2 (a), indicating that well-behaved domains do exist.

Based on the measurement results, we conclude that (1) email senders can be rated by their long-term behaviors; (2) local observation may not suffice for building a high-quality reputation system. These two factors are instrumental to the design of CARE.

IV. System Design

CARE is designed to be an autonomous system. Each domain equipped with CARE independently determines collaborating domains, exchanges information with collaborating domains, and derives reputation scores of remote domains. Information exchange occurs between two domains that mutually agree on collaboration. In case no collaboration is available, the system can continue functioning by using only local email history information. The autonomy eases incremental deployment of CARE.

As a reputation system, CARE operates collaboratively with other anti-spam techniques. A typical use of CARE is functioning as a preprocessor of a content-based spam filter. Messages from domains with sufficiently high reputation scores are directly accepted while messages from domains

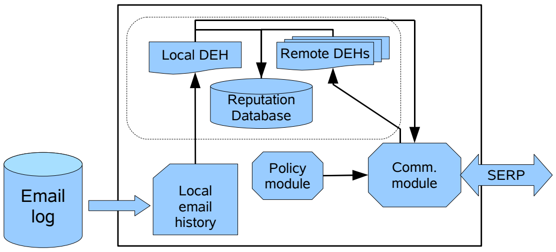


Fig. 3. Architecture of CARE system

with very low reputation scores are directly rejected. For the rest of email, those messages from domains with no reputation are directly passed to the filter and all the others are marked with their reputation scores before passing.

The architecture of CARE is shown in Figure 3. The local email history module takes the log of local email servers as input and derives the local Database of Email History (DEH). The information of local DEH is used in both information exchange and reputation derivation. In information exchange, the collaborating domains are determined based on their behaviors recorded by local DEH. In reputation derivation, local DEH involves in both calculating reputation scores and assessing the trustworthiness of remote information from collaborating domains. Through the communication module, a CARE system exchanges information with other collaborating systems via Simple Email Reputation Protocol (SERP). To provide flexibility in deciding collaborating domains, a policy module is incorporated allowing system administrators to apply their admission control policies. For example, local domain might be forbidden to exchange information with some remote domains due to certain policy, although those domains are regarded as collaborating candidates by the CARE system. Using both local DEH and remote DEHs, reputation scores of email sending domains are computed and stored into the reputation database.

In this section, we first highlight the rating issues in CARE. Then, we describe how to build the local email history and reputation database. After that, we detail the communication module and SERP protocol.

A. Rating Issues

CARE uses domain as the reputation identity, following Gmail reputation service [3] and RepuScore [6]. Rating domain is preferred to rating server (IP address), due to the following reasons. First, domain is easier to be authenticated than IP address, as email authentication schemes such as SPF and DKIM have become popular. Second, email sending policies are usually applied at domain level for nonspam domains and rating domain can provide better scalability. Third, an IP address can be used by multiple entities simultaneously, while a domain represents only one entity at any time. Last but not least, legitimate email servers are usually placed in a separate subdomain in a large ISP and can be easily distinguished from the subdomain where a spam botnet resides.

CARE does not differentiate email relaying servers from email originating servers in rating. If an email server not only sends email for its own domain, but also relays email for other domains, all relayed messages will be counted onto the relay domain. "No open-relay" has been a rule for email server administration and well followed by decent email service providers.

B. Local Email History

Local email history contains the information of both inbound and outbound email transmissions occurred locally. If multiple email servers are used in the local domain, the local email history is the integration of the information recorded by all those servers. Local email history records the basic information of email transmissions such as email transfer time, spam or not, source and destination at host and domain levels. This information usually can be directly extracted from email log. The domain of a remote host can be easily retrieved, e.g., by using DNS utility "`dig`". Local email history does not include the information of the messages whose sending domains cannot be identified. In general, legitimate email servers of an ISP can be distinguished from spam bots residing inside the same ISP by domain name, because legitimate email servers are usually placed in a separate domain for management and security reasons. For instance, the broadband host "ip70-161-245-78.hr.hr.cox.net" is in domain "hr.cox.net" while the legitimate email server "smtp.west.cox.net" is in domain "west.cox.net." Email authentication schemes such as SPF and DKIM can further enhance the accuracy of domain identification as the binding between an email server and its domain is explicitly expressed by special DNS records. The local email history can be updated either online or offline.

A CARE system also maintains a special database called Database of Email History (DEH), which is derived from the local email history and used in information sharing. Each sending domain has a record ($\mathcal{X}$, TM, GM, AD) in the database. A record profiles the email sending behavior of a domain in the past W days. W is a tunable parameter and decided by system administrator. $\mathcal{X}$ is the name of sending domain. TM and GM are the numbers of total messages and good (i.e., nonspam) messages from $\mathcal{X}$, respectively. AD is the number of the active days, in each of which, at least one message from $\mathcal{X}$ is received. The database is updated periodically, e.g., once per day, and the history information beyond W days could be removed to save disk space.

C. Reputation Database

We derive a domain's reputation by combining both local DEH and remote DEHs collected from collaborating domains. Initially, only local database is available. Under this circumstance, the reputation derivation is simplified into computing the good-ratio of each domain in the local database. After exchanging information with collaborating domains, we also use remote databases in derivation of domain reputation. However, the information from collaborating domains may not be fully trusted, because the authenticity of information is self-warranted. Therefore, we introduce a trustworthiness score for each remote database. In the absence of a central

Algorithm 1 Computing Domain Reputation

```
1: Input: $\mathsf{DEH}_\mathcal{I}$ and all collected remote $\mathsf{DEH}_\mathcal{R}$s.
2: Output: reputation score for every sender in $\mathsf{DEH}_\mathcal{I}$ and $\mathsf{DEH}_\mathcal{R}$s.
3: for each remote $\mathsf{DEH}_\mathcal{R}$ do
4:     compute trustworthiness score $\theta_\mathcal{R}$.
5: end for
6: for each sender $\mathcal{X}$ in $\mathsf{DEH}_\mathcal{I}$ and those $\mathsf{DEH}_\mathcal{R}$s that contain it do
7:     compute the reputation score of $\mathcal{X}$.
8: end for
```

authority, we rely on the local information to assess the trustworthiness of a remote database. Specifically, we examine the correlation between the local database and remote database on sending domains in common, and use the correlation result to compute the trustworthiness score of that database. The remote databases with high trustworthiness scores are deemed reliable. A domain's reputation score is the weighted average of good-ratios derived from the local and remote databases, and the weight for each remote database is set as the trustworthiness score of that database.

Algorithm 1 describes the general process of computing domain reputation. The notations used in the algorithm and the rest of the section are summarized in Table I. In general, the subscript of a symbol represents a history recording domain; it can also represent the domain's DEH when the context is clear. We use $\mathcal{R}$ for a generic collaborating domain and $\mathcal{I}$ for the local domain. The superscript of a symbol represents an email sending domain (seen by either local domain or a collaborating domain). For clarity, a collaborating domain with which we exchange information is termed as a collaborator, while a domain logged in either the local history or a remote history is termed as a sender.

TABLE I
SUMMARY OF NOTATIONS

DEH	Database of Email History
SMD	Set of Major Domains, for history correlation
W	History Window
$\mathcal{X}$	Domain $\mathcal{X}$
KU($\mathcal{X}$)	public key of domain $\mathcal{X}$
KI($\mathcal{X}$)	private key of domain $\mathcal{X}$
$GM_\mathcal{D}^\mathcal{X}$	# of good messages from $\mathcal{X}$ received by $\mathcal{D}$
$TM_\mathcal{D}^\mathcal{X}$	# of total messages from $\mathcal{X}$ received by $\mathcal{D}$
$AD_\mathcal{D}^\mathcal{X}$	# of active days $\mathcal{X}$ sending email to $\mathcal{D}$
$dg_\mathcal{D}^\mathcal{X}$	good-ratio of $\mathcal{X}$ according to $\mathsf{DEH}_\mathcal{D}$
$ds_\mathcal{D}^\mathcal{X}$	domain score of $\mathcal{X}$ according to $\mathsf{DEH}_\mathcal{D}$
$dr^\mathcal{X}$	domain reputation of $\mathcal{X}$
$\gamma_\mathcal{R}$	supporting factor of $\mathsf{DEH}_\mathcal{R}$ for computing $\theta_\mathcal{R}$
$\omega_\mathcal{R}$	correlation coefficient of $\mathsf{DEH}_\mathcal{R}$ for computing $\theta_\mathcal{R}$
$\theta_\mathcal{R}$	trustworthiness score (weight) of $\mathsf{DEH}_\mathcal{R}$
β	threshold used in constructing SMD
δ	threshold used in computing $\gamma_\mathcal{R}$

The process of computing domain reputation consists of two steps. In the first step (lines 3–5) we compute the trustworthiness score of each remote database $\mathsf{DEH}_\mathcal{R}$, and in the second step (lines 6–8) we compute the reputation score of each sender recorded by either the local database or remote databases.

To compute the trustworthiness score of remote database $\mathsf{DEH}_\mathcal{R}$, we first derive the Set of Major Domains (SMD) of that database. SMD contains those domains that behave well and stay active. Such a well-behaved domain is indicated by a high domain score. The domain score is defined as $ds = \frac{GM}{TM} \times \frac{AD}{W}$. $ds_\mathcal{R}^\mathcal{X}$ stands for the domain score of sender $\mathcal{X}$ in database $\mathsf{DEH}_\mathcal{R}$ and can be easily computed from the record of $\mathcal{X}$ in $\mathsf{DEH}_\mathcal{R}$. Senders with sufficiently high domain scores are added into SMD, that is, $\mathsf{SMD} = \{\mathcal{X} : ds^\mathcal{X} \geq \beta\}$, where β is the threshold decided locally. A high value of β implies a high good-ratio (we define the good-ratio as $dg = \frac{GM}{TM}$) and a high ratio of active days ($\frac{AD}{W}$). By setting β to an appropriate value, we can ensure that the majority of domains in SMD are legitimate.

Then, we compute the intersection set (denoted by INT) between local SMD ($\mathsf{SMD}_\mathcal{I}$) and $\mathcal{R}$'s SMD ($\mathsf{SMD}_\mathcal{R}$) for remote database $\mathsf{DEH}_\mathcal{R}$. Formally, $\mathsf{INT}_\mathcal{R} = \mathsf{SMD}_\mathcal{I} \cap \mathsf{SMD}_\mathcal{R}$. We also compute supporting factor $\gamma_\mathcal{R}$ from $\mathsf{INT}_\mathcal{R}$. By definition,

$$\gamma_\mathcal{R} = \frac{min(||\mathsf{INT}_\mathcal{R}||, \delta)}{\delta}, \tag{1}$$

where $||S||$ represents the cardinality of set S, and δ is a pre-defined system parameter. The rationale behind computing SMD and INT is to find a reliable common base for correlation computing. In other words, the sending domains in common for correlation computing (i.e., the set of domains represented by INT) are expected to manifest consistent sending behaviors to receiving domains including the local domain $\mathcal{I}$ and remote domain $\mathcal{R}$. Legitimate email service providers usually present this characteristic. Computing γ is to take the size of common base into consideration.

After that, we compute the correlation coefficient of $\mathsf{DEH}_\mathcal{R}$ (denoted by $\omega_\mathcal{R}$) from $\mathsf{INT}_\mathcal{R}$ based on city block distance (also called Manhattan distance or taxicab distance). For the domains in $\mathsf{INT}_\mathcal{R}$, we first derive their good-ratio vectors in $\mathsf{DEH}_\mathcal{I}$ and $\mathsf{DEH}_\mathcal{R}$ (denoted as $\mathbf{V}_\mathcal{I}$ and $\mathbf{V}_\mathcal{R}$ respectively). With $\mathsf{INT}_\mathcal{R} = \{\mathcal{X}1 \ldots \mathcal{X}n\}$, $\mathbf{V}_\mathcal{I} = [dg_\mathcal{I}^{\mathcal{X}1} \ldots dg_\mathcal{I}^{\mathcal{X}n}]$ and $\mathbf{V}_\mathcal{R} = [dg_\mathcal{R}^{\mathcal{X}1} \ldots dg_\mathcal{R}^{\mathcal{X}n}]$. Then, the city block distance between $\mathbf{V}_\mathcal{I}$ and $\mathbf{V}_\mathcal{R}$, denoted as $dist(\mathbf{V}_\mathcal{I}, \mathbf{V}_\mathcal{R})$, is obtained by summing up the differences of good-ratios in $\mathsf{DEH}_\mathcal{I}$ and $\mathsf{DEH}_\mathcal{R}$ for each domain in $\mathsf{INT}_\mathcal{R}$. Formally,

$$dist(\mathbf{V}_\mathcal{I}, \mathbf{V}_\mathcal{R}) = \sum_{i=1}^{n} |dg_\mathcal{R}^{\mathcal{X}i} - dg_\mathcal{I}^{\mathcal{X}i}|. \tag{2}$$

We get the correlation coefficient $\omega_\mathcal{R}$ by normalizing $dist(\mathbf{V}_\mathcal{I}, \mathbf{V}_\mathcal{R})$ into $[0, 1]$, that is,

$$\omega_\mathcal{R} = 1 - \frac{dist(\mathbf{V}_\mathcal{I}, \mathbf{V}_\mathcal{R})}{||\mathsf{INT}_\mathcal{R}||}. \tag{3}$$

We derive the trustworthiness score of $\mathsf{DEH}_\mathcal{R}$, $\theta_\mathcal{R}$, by multiplying $\mathsf{DEH}_\mathcal{R}$'s supporting factor $\gamma_\mathcal{R}$ and its correlation coefficient $\omega_\mathcal{R}$. Formally,

$$\theta_\mathcal{R} = \gamma_\mathcal{R} \cdot \omega_\mathcal{R} = \frac{min(||\mathsf{INT}_\mathcal{R}||, \delta)}{\delta} \cdot (1 - \frac{dist(\mathbf{V}_\mathcal{I}, \mathbf{V}_\mathcal{R})}{||\mathsf{INT}_\mathcal{R}||}). \tag{4}$$

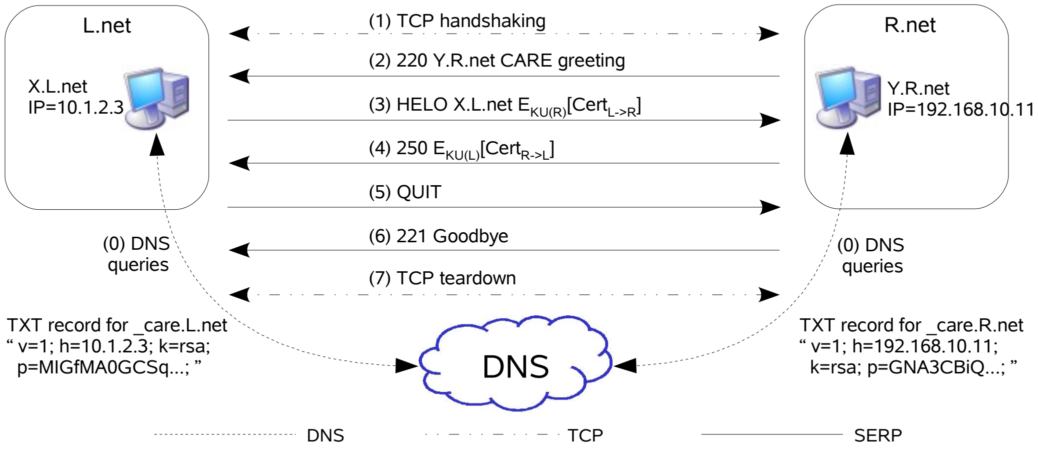

L.net
X.L.net
IP=10.1.2.3
R.net
Y.R.net
IP=192.168.10.11
(1) TCP handshaking
(2) 220 Y.R.net CARE greeting
(3) HELO X.L.net $E_{KU(R)}[Cert_{L->R}]$
(4) 250 $E_{KU(L)}[Cert_{R->L}]$
(5) QUIT
(6) 221 Goodbye
(7) TCP teardown
(0) DNS queries
(0) DNS queries
TXT record for _care.L.net
" v=1; h=10.1.2.3; k=rsa; p=MIGfMA0GCSq...; "
TXT record for _care.R.net
" v=1; h=192.168.10.11; k=rsa; p=GNA3CBiQ...; "
DNS
DNS
TCP
SERP

Fig. 4. Procedure of a successful mutual agreement establishment via SERP

We also incorporate a list of trusted collaborators into CARE. Administrators can add their trusted collaborating domains into the list or remove any domain from it. The weight of each domain in the list, that is, θ, is 1. This offsets the potential inaccuracy in computing the trustworthiness score, since it is possible that a collaborating domain's view is different from the local view on the same sending domain.

Finally, for each domain $\mathcal{X}$ we derive its reputation score $dr^{\mathcal{X}}$ by computing the weighted average of $\mathcal{X}$'s good-ratios ($dg^{\mathcal{X}}$) from the DEHs that record $\mathcal{X}$. We use $\mathcal{D}$ to represent a generic domain whose DEH contains a record for $\mathcal{X}$. The weight for remote database DEH$_{\mathcal{R}}$, $\theta_{\mathcal{R}}$, is in $[0, 1]$ and the weight for the local database DEH$_{\mathcal{I}}$, $\theta_{\mathcal{I}}$, is always 1. Formally, the reputation score of domain $\mathcal{X}$ is defined as

$$dr^{\mathcal{X}} = \frac{\sum_{\mathcal{D}\in Q}\theta_{\mathcal{D}}\cdot dg^{\mathcal{X}}_{\mathcal{D}}}{\sum_{\mathcal{D}\in Q}\theta_{\mathcal{D}}}, \tag{5}$$

where $Q = \{\mathcal{D} : \mathcal{X} \in \text{DEH}_{\mathcal{D}}\}$. Note that the local good-ratio $dg^{\mathcal{X}}_{\mathcal{I}}$ can be 0. In this case, sender $\mathcal{X}$ has not been recorded by the local domain. By using the weighted average, the sending domains recorded by both the local domain and collaborators are assessed from a broader view, while the sending domains recorded only by the local domain are not affected.

Due to the space limit, we give a brief analysis of our reputation mechanism on attack-resistance. The reputation derivation of CARE makes it difficult for a spammer to gain a high reputation score, because a high score requires consistently good behavior recorded by both local domain and collaborators. A spammer could deliberately present different behaviors to the local domain and a collaborator in order to lower the trustworthiness score of that collaborator. However, this type of pollution also requires the spammer to stay long and behave well.

D. Simple Email Reputation Protocol (SERP)

CARE systems communicate with one another via SERP. Through SERP, a CARE system can transfer DEH as well as other messages to its counterpart. SERP adopts a DNS-based authentication scheme, borrowing the idea of DNS-based email authentication schemes. The DNS-based authentication is lightweight, easy to install, and incrementally deployable. SERP requires every deployment domain (e.g., example.com) to publish a special TXT resource record[2] in its _care DNS subdomain (_care.example.com in this example). The record must specify the domain name (or IP address) of the host on which CARE is serviced and the associated public key. By doing so, a remote CARE host can be authenticated by first querying the TXT DNS record under the _care subdomain of the domain where the host resides, and then checking if the domain name (or IP address) of the host is listed in that record.

Among all domains that have direct email communication with the local domain, CARE selects those domains that behave consistently well for collaboration. These domains can be easily decided by examining the local email history. Apparently, they also must have a valid TXT DNS record for CARE. Each CARE system maintains a list of remote domains satisfying these requirements and uses it for selecting collaborating domains.

To collaborate, two domains are required to reach a mutual agreement on information exchange before sharing DEHs. With the agreement, the two domains will play dual roles of service requester and provider.

Figure 4 illustrates the procedure of establishing a successful mutual agreement via SERP. In the figure, the CARE hosts in domain L.net (L for short) and R.net are X.L.net (X for short) and Y.R.net, respectively. Both domains have published their CARE DNS records. Since every CARE system keeps a list of collaborating domains, by periodically querying the CARE DNS records of those domains, each system can readily know the positions of CARE hosts inside those domains. The activity of periodic DNS query is shown as step (0) in Figure 4. After X successfully resolves the domain name of Y, it sends a request to Y for establishing a TCP connection. When Y receives this request, by checking its list of collaborating domains, it can instantly decide how to react: accepting the

[2] In case DNS SRV resource record[26] is chosen to publish CARE service, a separate DNS TXT record is still needed for carrying public key and CARE host information.

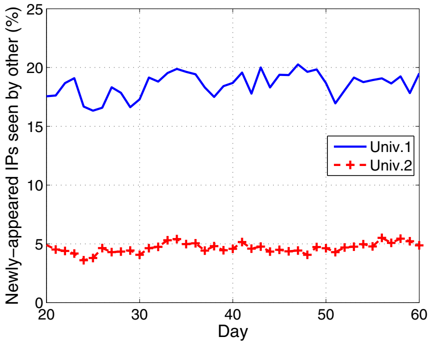


Fig. 5. Percentage of newly-appeared IP addresses that have been recorded by the other university over all newly-appeared IP addresses in daily logs

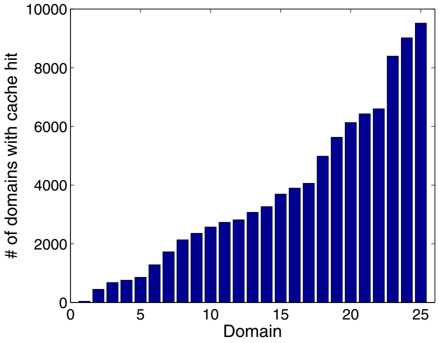


Fig. 6. Number of DNS cache hits for 25 .edu domains

request if the remote host is in the list or rejecting the request otherwise. In this example, Y accepts the request. The TCP handshaking process is marked as step (1) in Figure 4. Semantically, neither step (0) nor step (1) is part of SERP. However, SERP needs step (0) for authentication and step (1) for a reliable data connection and authorization.

After the TCP connection is established, host Y sends a greeting message to the requesting host X (step (2)), indicating its identity. After receiving the greeting message, X issues command "HELO" (step (3)), followed by the domain name of X and a certificate ($\text{Cert}_{L\rightarrow R}$) encrypted by the public key of domain R (i.e., KU(R)). The certificate $\text{Cert}_{L\rightarrow R}$ means that domain L allows domain R to access L's DEH. It is composed by concatenating message M and its signature, that is, $\text{Cert}_{L\rightarrow R} = M||E_{KI(L)}[H(M)]$, where H(M) is the hash value of M and KI(L) is the private key of domain L. Message M contains: certificate issuer L and recipient R, certificate starting and expiration times, and the updating interval of L's DEH. The communication proceeds if the certificate is accepted by Y. Echoing the offer of X, Y responds by sending its certificate $\text{Cert}_{R\rightarrow L}$ back to X (step (4)). After a successful exchange, X sends command "QUIT" (step (5)), indicating completion of the mutual agreement. The TCP connection is torn down (step (7)) as soon as Y acknowledges the "QUIT" command (step (6)). The above procedure will be repeated once either of the certificates expires.

After mutual agreement, two domains can exchange their DEHs with each other. The data exchange procedure is similar to the agreement establishing procedure. The data exchange can be optimized since DEH is usually updated gradually. We can make a snapshot of DEH as the reference base and generate a difference file for each update to DEH. Then, we just transfer the appropriate difference file(s) instead of a whole DEH, reducing the total bytes of data transmission.

V. System Evaluation

Our evaluations focus on the potential benefit of using CARE. Specifically, we are interested in the increase of coverage brought by collaboration, that is, the reduction of the number of newly-appeared sending domains thanks to collaboration. We have analyzed real email logs, conducted a DNS-based estimation experiment, and performed extensive simulations to validate the effectiveness of CARE on improving the coverage.

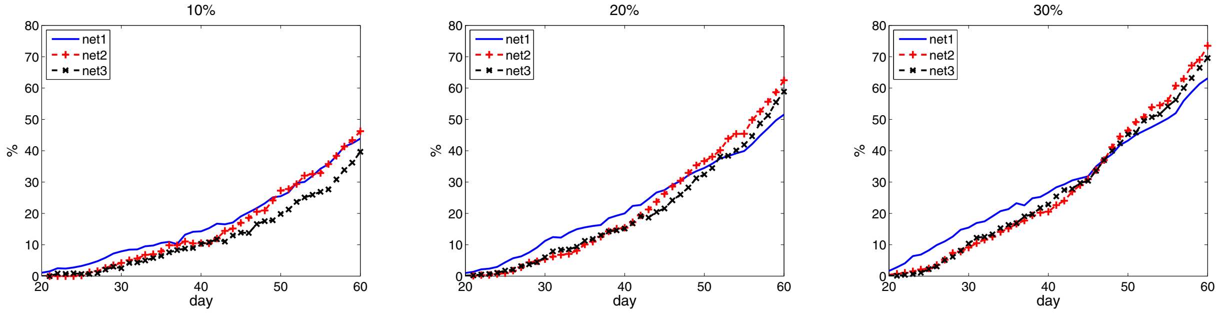
A. Log-based Experiment

We first collected two-month email logs from two universities recorded in the same time period. All email logs are daily based and record the source IP addresses (no domain information) and spam information for inbound messages.

For newly-appeared IP addresses in the daily logs of each university, we examine how many of them have already been recorded by the other university and calculate the percentage. The dynamics of the percentage in the daily logs are shown in Figure 5, which demonstrates the effectiveness of collaboration. The curve for university 1 shows that about 16% to 20% of the newly-appeared IP addresses in university 1's daily logs have already been recorded in university 2's logs. For university 2, the percentage reduces to 5% but is stable. The difference of percentage for two universities may be attributed to the difference of total IP addresses in their email logs. On average, university 1 records about 42 thousand IP addresses daily, while university 2 observes about 87 thousand IP addresses per day. However, the stability of both curves implies the stable gain from collaboration in the long run.

B. DNS-based Experiment

Results from the log comparison are encouraging. However, we cannot obtain more email logs for comprehensively evaluating CARE. We conducted a DNS-based experiment to estimate the potential benefit that could be achieved by multi-domain collaboration. As an email sending server usually sends a DNS MX query to obtain the location of receiving server before launching an SMTP transaction, we can infer whether email has been sent to a given domain by snooping (using iterative mode) the DNS cache of the sending server. If the MX record of the receiving domain can be found in the DNS cache, it is highly likely that email communication between the two domains has occurred recently. Clearly, the number of cache hits by DNS snooping may not accurately reflect real email communications. Nevertheless, DNS snooping provides an efficient way of estimating the gain of multi-domain collaboration and the derived result can serve as a lower bound.



(a) (b) (c)

Fig. 7. Percentage of newly-appeared nonspam domains that are covered after using CARE in each day

We randomly selected 25 .edu domains from the logs used in Section III as collaborating domains, and randomly selected 50,000 inactive domains as DNS snooping target domains. Each of these snooped domains sent less than five messages in total in the logs, and none of them had sent any messages in the past month before snooping. We use these inactive domains as "ongoing" and "new" sending domains to study how many of them can be covered by our collaborators. After DNS probing test, we found that 36,646 out of 50,000 domains can be snooped. Then, we probed the DNS servers of each of these domains to find out how many MX records of the selected 25 .edu domains were in the DNS cache.

We count cache hits of each .edu domain and show the numbers of cache hits for all 25 .edu domains in ascending order in Figure 6. From the figure, we can see a clear diversity on the number of probed domains with cache hit among different .edu domains. Some .edu domains can be hit in DNS caches of more than 8,000 inactive domains, while some other .edu domains have less than 1,000 hits. For a .edu domain, a hit in the DNS cache of an inactive domain means that the .edu domain received email from the inactive domain and thus had this domain in its local email history. Therefore, we can benefit more by collaborating with the .edu domains that have more cache hits. Overall, the total number of cache hits for 25 .edu domains is 12,660, indicating that the email histories from 25 collaborating .edu domains can cover at least 34.6% of newly-appeared domains. The gain from multi-domain collaboration could be bigger with more collaborators and more types (e.g., .com) of collaborators. In addition, we probed all the inactive domains within one day. This implies that all 36,646 domains appeared in the same day, which, however, is unlike to happen in practice according to our measurement results. Thus, the benefit could be even higher in reality.

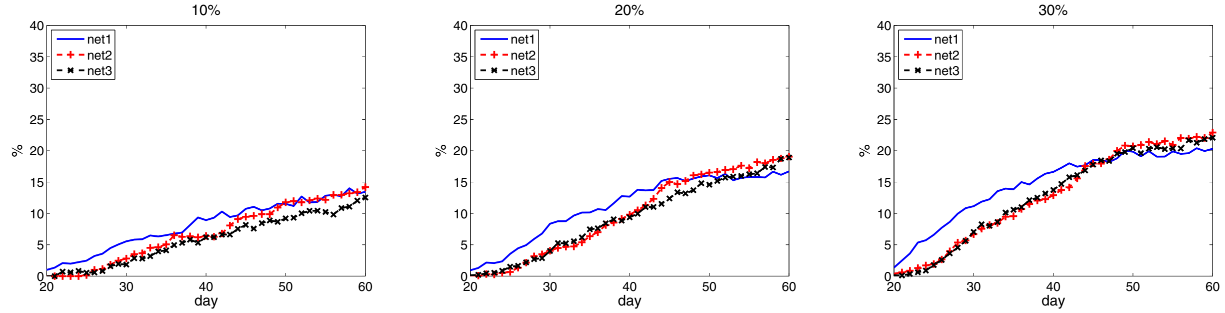
C. Simulation

We applied simulation to further study the dynamic characteristics of CARE. We implemented a CARE simulator with full CARE functionality. The simulator is driven by the configuration of email domains (1,200 nonspam domains and 10,000 spam domains) and the daily traces of email communications among those domains (60 days). Both spam and nonspam domains are dynamically born in the trace. Nonspam domains stay until the end of trace, while spam domains only stay for a short random period. Spam domains always send spam to nonspam domains, while nonspam domains send to one another both nonspam and spam messages. We use three types of network topologies (power-law, small world, and random graph) for nonspam domains. We readily acknowledge that the generated traces may not be representative, since there is no a priori knowledge on the topology and dynamics of email domains. However, the emphasis here is on the effectiveness of CARE with no assumption on network and communication patterns.

The simulator first randomly picks a given percentage of nonspam domains as CARE domains using the domain configuration, and then starts simulation using the daily traces. In each day, the simulator first does the message receiving and history recording driven by the trace records, then updates the reputation database of each CARE domain. In simulation, CARE is used as the preprocessor of a spam filter. The spam filter has fixed false positive rate and false negative rate, 0.01 and 0.05, respectively. Messages from a domain with reputation score 0.8 or higher are regarded as nonspam and messages from a domain with reputation score 0.1 or less are regarded as spam. If the reputation score cannot ensure acceptance or rejection, CARE tags the message with its reputation score and passes it to the filter. All processing results are logged into the database of email history to compute reputation. All CARE domains use the same parameter setting (history window $W = 30$, $\beta = 0.3$, and $\delta = 3$).

We first investigate how CARE improves domain coverage through collaboration. Figure 7 shows the dynamics of percentage of newly-appeared nonspam domains that are covered after using CARE in each day. The results are displayed from day 20 because of history accumulation. The "net1," "net2," and "net3" stand for power-law topology, small world topology, and random graph topology, respectively. To illustrate the effect of increasing CARE deployment ratio on the coverage, we set the percentage of the nonspam domains that use CARE as 10%, 20%, and 30% in the simulation, and display their results in Figure 7's (a), (b), and (c) respectively. For each given combination of network setting and CARE domain percentage, we run ten trials and use the average as the result.

Figure 7 clearly demonstrates the effectiveness of CARE



(a) (b) (c)

Fig. 8. Percentage of more nonspam messages being directly accepted after using CARE in each day

collaboration on improving the coverage of nonspam domains. Moreover, by comparing (a), (b), and (c) of Figure 7, we can see that the increase of percentage of CARE deployment renders the growth of percentage of domains being covered.

Increased coverage indicates that more nonspam messages can have reputation scores and be protected from being dropped by spam filter. We compare the number of the nonspam messages that are directly accepted under CARE collaboration to that using only local information and find that CARE does increase the number of directly accepted nonspam messages. Figure 8 shows the percentage of increase in terms of nonspam messages being directly accepted in each day. We can see that more nonspam messages are accepted under all three different network topologies. The percentage of increase keeps growing with time in all simulation environments. This indicates that the use of CARE can prevent considerably more nonspam messages from being lost. It is also notable from Figure 8 that more CARE deployments (10% vs. 20%) result in more nonspam messages being directly accepted.

VI. Conclusion

In this paper we have presented the motivation, design, and evaluation of CARE, a collaboration-based autonomous email reputation system. CARE is a generic email reputation system in that it rates both spam and nonspam domains in an autonomous manner. Using CARE, each domain derives the reputation scores of email sending domains by sharing its local observations with those domains that manifest consistently good behavior. In the evaluation of CARE, we used real email log traces from two universities to quantify the benefits of collaboration between two domains and conducted a large DNS snooping experiment to estimate the potential gain brought by multi-domain collaboration. Moreover, we performed extensive simulations to further investigate CARE in a large-scale environment. Our experimental results evidently demonstrate the effectiveness of CARE.

Acknowledgments

We would like to thank Zhenhai Duan and William and Mary IT Department for the help on email log traces. This work was partially supported by NSF grants CNS-0627339 and CNS-0916022.

References

[1] “SpamCop Blocking List,” http://www.spamcop.net/bl.shtml.

[2] “The Spamhaus Project,” http://www.spamhaus.org/.

[3] B. Taylor, “Sender reputation in a large webmail service,” in *CEAS 2006*.

[4] S. Agarwal, V. N. Padmanabhan, and D. A. Joseph, “Addressing email loss with suremail: Measurement, design, and evaluation,” in *Proc. USENIX Annual Technical Conference 2007*, 2007.

[5] M. Welzl, “end2end-interest: A message to authors of pfldnet papers,” http://mailman.postel.org/pipermail/end2end-interest/2008-January/, 2008.

[6] G. Singaraju and B. Kang, “Repuscore: Collaborative reputation management framework for email infrastructure,” in *Proc. the 21st Large Installation System Administration (LISA)*, 2007.

[7] D. Alperovitch, P. Judge, and S. Krasser, “Taxonomy of email reputation systems,” in *ICDCS 2007 Workshops*, 2007.

[8] D. Erickson, “DOEmail - default off email,” http://doemail.org/.

[9] M. W. Wong and W. Schlitt, “RFC 4408: Sender policy framework (SPF) for authorizing use of domains in e-mail, version 1,” 2006.

[10] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas, “RFC 4871: Domainkeys identified mail (DKIM) signatures,” 2007.

[11] G. Singaraju, J. Moss, and B. Kang, “Tracking email reputation for authenticated sender identities,” in *CEAS 2008*, 2008.

[12] A. Ramachandran and N. Feamster, “Understanding the network-level behavior of spammers,” in *Proc. ACM SIGCOMM 2006*, 2006.

[13] Z. Duan, K. Gopalan, and X. Yuan, “Behavioral characteristics of spammers and their network reachability properties,” in *Proc. IEEE ICC 2007*, 2007.

[14] A. Ramachandran, N. Feamster, and S. Vempala, “Filtering spam with behavioral blacklisting,” in *Proc. ACM CCS 2007*, 2007.

[15] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, and J. Barker, “Detecting spam zombies by monitoring outgoing messages,” in *Proc. IEEE INFOCOM 2009*, 2009.

[16] “Sender score,” http://www.senderscore.org/.

[17] B. Leiba, J. Ossher, V. Rajan, R. Segal, and M. Wegman, “Smtp path analysis,” in *CEAS 2005*, Mountain View, CA, July 2005.

[18] J. Golbeck and J. Hendler, “Reputation network analysis for email filtering,” in *CEAS 2004*, 2004.

[19] P.-A. Chirita, J. Diederich, and W. Nejdl, “MailRank: Using ranking for spam detection,” in *Proc. ACM CIKM 2005*, 2005.

[20] V. V. Prakash and A. ODonnell, “Fighting spam with reputation systems,” *ACM Queue*, vol. 3, no. 9, November 2005.

[21] M. Ceglowski and J. Schachter, “LOAF,” http://loaf.cantbedone.org/.

[22] D. Brickley and L. Miller, “FOAF vocabulary specification 0.9,” http://xmlns.com/foaf/spec/, 2007.

[23] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazieres, and H. Yu, “RE: Reliable email,” in *Proc. USENIX NSDI 2006*, San Jose, CA, May 2006.

[24] R. D. Twining, M. M. Williamson, M. Mowbray, and M. Rahmouni, “Email prioritization: Reducing delays on legitimate mail caused by junk mail,” in *Proc. USENIX Annual Technical Conference 2004*, 2004.

[25] S. Venkataraman, S. Sen, O. Spatscheck, P. Haffner, and D. Song, “Exploiting network structure for proactive spam mitigation,” in *Proc. USENIX Security 2007*, 2007.

[26] A. Gulbrandsen, P. Vixie, and L. Esibov, “RFC 2782: A DNS RR for specifying the location of services (DNS SRV),” 2000.