Weeding Spammers at the Root: A Precise Approach to Spam Reduction

Tu Ouyang and Michael Rabinovich
EECS Department
Case Western Reserve University

***Abstract* – Email spam has become one of the most critical threats affecting Internet user experiences. Existing anti-spam techniques, such as spam filters and reputation systems, face growing difficulties due to spammers' use of multimedia content (which is difficult to filter) and computer zombies (which mask true spammer identity). We propose SpamWeeder, a simple idea for spam prevention, which targets the root of the spam problem - the email address trafficking. SpamWeeder can track the email address trafficking channels, expose the parties at the source of these channels, and precisely block all email from all and only parties belonging to a given distribution channel. SpamWeeder can further generate warnings to users against submitting their email addresses to parties likely to engage in address trafficking. Finally, our approach is transparent to users and can be installed autonomously by an organization intranet, without need for any changes to the rest of the email infrastructure.**

I. Introduction

As access bandwidth increases and becomes less expensive, the bar for spammers to generate disruptive volumes of spam gets lower. Higher Indeed, spammers need fewer resources, or smaller-size botnets, to accomplish their goals. The currently prevalent approach to combat spam relies on spam filters. Unfortunately, being a heuristics-driven technology, filters inevitably miss some spam and block legitimate email. As email messages become richer in content, and spammers use more sophisticated disguising, the job of filters will only get harder.

We propose a precise approach to reduce spam, by targeting the root of the problem, which is unscrupulous sharing of user email addresses. Entities involved in such sharing form distribution networks rooted at the parties who obtain our email through *harvesting*. Harvesters employ a number of techniques, such as crawling people's home pages and other documents containing email addresses, or probing mail servers for random user names. Some of these methods have known counter-measures, for example, disguising email appearance in documents[1] or configuring mail servers not to bounce emails to non-existing users (with useful feedback to legitimate senders becoming a casualty of the spam war).

A more challenging type of harvesting involves *email leaking*, when users *volunteer* their email addresses to unscrupulous businesses while making online purchases, registering for Web services, posting on blogs, subscribing to mailing lists, or even providing their emails *out-of-band*, e.g., when making a purchase by phone. Users today protect themselves from email leaking through maintenance of separate email accounts, which they close as spam levels become intolerable. Unfortunately, replacing email addresses is a labor-intensive and error-prone process as the user needs to notify and re-subscribe with legitimate correspondents that they would like to retain.

This paper outlines an automated and user-transparent approach to counter harvesting that results from email leaking. Our approach integrates two key ideas. First, we propose a technique that allows the user to easily *expose* the sources of the email-trafficking networks and reliably *block* emails from all the entities rooted at a given source without affecting other correspondents. This technique involves cooperation of user's Web browser, email client and client's mail server. In other words, it is fully confined to the *client's site*, by which we refer to an organization's intranet for corporate users, the combination of a home network and ISP-provided mail server for residential users, or, in the case of web-hosted mail service providers such as gmail, the combination of a user's computer and the provider's mail servers. Thus, client sites can use our approach *today*, without depending on any changes to the rest of email infrastructure.

Second, we utilize users' actions as they react to spam to automatically generate input to a reputation system, which would rate web sites that require users to enter their email addresses. Reputation systems hold great promise to combat spam in that they leverage community effort to identify and neutralize likely abusers. However, a reputation system is only as good as the input data that drives it, and providing reliable input is difficult: spammers may intentionally pollute the reputation database, legitimate users might be too busy to supply input, and spammers can easily change their identities once their reputation degrades. In our approach, the input to the reputation system occurs as a side effect of *self-serving* user behavior, and reputation scores are bound to web sites providing Web forms for entering email addresses. Unlike forged email identities, any such hostname would have to be a properly registered DNS name, which complicates minting spurious identities. The input to the reputation system can also include other helpful hints such as the referrer's hostname, whether the form's URL uses a raw IP address instead of hostname, etc.

Our approach is based on special hard-to-guess email address aliases. Manipulating email addresses, and utilizing

The presentation was made possible in part through financial support from School of Graduate Studies at Case Western Reserve University

[1]While this involves an ingenuity contest between users and harvesters, generally, users have an upper hand in this race because they mask emails by hand while harvesters employ automatic parsers.

unique hard-to-guess addresses in particular, has been the basis for a number of anti-spam techniques (see Section II) and is not a novel contribution here. Our contribution is in the way we use these aliases to identify and block spam perpetrators and, most importantly, in making these aliases trivial for users to employ. We named our approach SpamWeeder to reflect its targeting the roots of the email-trafficking networks. We refer to these networks as *"spammer trees"* due to their hierarchical nature: the root in each network sells the email address to some number of perpetrators who in turn sell it to further traffickers, etc.

II. Related Work

A prevalent current approach to control spam is through spam filters, which apply various rules and heuristics to incoming messages. Recently, filters were proposed that go beyond textual content and attempt to extract text embedded in images or videos [3]. However, while successfully removing large amounts of spam, spam filters still miss some amounts of spam and block some legitimate email due to their fundamentally approximate nature.

An important component in spam filter decisions is input from various reputation systems. Reputation systems target email senders and have been reported to achieve a high spam block percentage and low false positive rate [11]. However, obtaining reliable input to reputation systems involves significant effort. Further, the emergence of netbots as conduit of spam hinders these systems effectiveness by blurring true sender identity among numerous zombie computers even when reputation is bound to sender IP addresses (which are harder to forge than sender's email).

A number of promising approaches deal with forged sender identities (e.g., Microsoft's Sender ID and Yahoo's Domain Keys, among others) and introduce economic mechanisms to combat spam [8], [7]. Unlike SpamWeeder, these approaches, while often backward-compatible, require changes to the email infrastructure to become fully effective.

Closer related to SpamWeeder are several anti-spam approaches involving email address manipulation. Hall introduced the notion of email channels, which the user can allocate, assign to correspondents, or discard to block access [4]. SpamWeeder in essence assigns a new channel to each correspondent, but hides channel management from the user.

Ioannidis proposes giving out encrypted email addresses which encapsulate policy [5]. A modified email server would check the incoming email's destination address and follow the policy embedded in the address to decided what to do with this email. A variant of this idea (minus encryption) is implemented by Spamgourmet [16], which maintains a mail server where users can register their true email address and obtain a stem email addresses in the spamgourmet.com domain, e.g., *user@spamgourmet.com*. The user can then give out email addresses in the form of *anystring.X.user@spamgourmet.com*, which would remain valid only for X incoming messages and be forwarded to the user's true account. Unfortunately, nothing prevents a spammer from simply using new strings for *anystring* above to defeat this approach. Furthermore, neither of these approaches is suitable for long-term subscriptions or other permanent services. Many mail servers allow users to make up arbitrary addresses on the fly in the form of *user+anystring@domain.com*, and then enact filters on individual strings. This feature is a simplified version of the Spamgourmet approach and exhibits similar limitations.

Spamarrest [14] allows users to create ad-hoc addresses in the *@spamarrest* domain; users then access their mail directly on Spamarrest's mail server. In addition, Spamarrest employs a challenge/response approach to block non-human correspondents, where senders must respond to a challenge before their mail would be delivered. Our approach differs from Spamarrest in that we hide address manipulation from the end user, who does not have to manage or be concerned with multiple email addresses. The challenge/response approach is orthogonal to, and could in principle be used in SpamWeeder, although it breaks a number of legitimate email usage scenarios (e.g., sending to a mailing list, or legitimate automated email such as calendar alerts).

Some commercial web sites, such as ebay and craiglist.com, supply users with substitute addresses. Mail sent to this substitute address is ultimately delivered to user's registered true email address. However, without software support, once the user replies to an email, their real email address would appear in the reply message and thus be exposed. Also, the substitute address is specific to the particular Web site that issued it.

Like SpamWeeder, PwdHash[13] also employs a browser extension to prevent leaking of user information. PwdHash targets password phishing, while SpamWeeder targets email address trafficking. Besides browser, our approach requires a coordinated email client extension.

III. The SpamWeeder Approach

The SpamWeeder approach integrates two subsystems, a *track & kill* subsystem that allows a user to identify roots of the spammer trees and block mail from entire spammer trees, and an *early warning* subsystem that receives input from the track & kill subsystem and warns users of the web sites involved in email address trafficking. The early warning system is in essence a reputation system except it is used to *preempt* spam rather than block it.

A. The Track & Kill System

Many web sites require users to input their email, as part of registration or subscription process, to send a confirmation to an on-line purchase, to submit an information request, or to post on a discussion board or a blog. This has become so pervasive that not complying entails severe limitations in what

one can do on the Internet. At the same time, providing one's email address may start a life cycle of spam: some web sites will send unsolicited emails to users, or worse, distribute users' email to other parties which could generate their own spam or distribute the address further. In this scenario, the original web site becomes the root of an email address-trafficking tree, which we refer to as a "spammer tree". The track & kill subsystem of SpamWeeder provides a precise and user-transparent way to identify the roots of the spammer trees and selectively block email from any parties that belong to these trees.

1) Site-Specific Email Aliases: Our basic idea stems from the observation of what users do today to confront email address trafficking. Many users utilize free public mail services and create special email accounts to provide to untrusted web sites. As the spam volumes inevitably grow with time, users switch to another account, and the cycle repeats. Unfortunately, this process is too coarse-grained and cumbersome. It requires users to manage several email accounts. Worse, switching to a new email address blocks the user from *all* services and emails bound to the old address, even legitimate ones. Updating all the services and notifying all correspondents of the new address is no easy task.

In SpamWeeder, the web browser automatically generates a new hard-to-guess mail alias every time the user attempts to submit their email address to a new Web site, and then substitutes user's real email with this alias. We currently use an MD-5 hash over a string comprising user name and clock time to generate the alias, with a safety check for uniqueness. In this way, SpamWeeder associates a unique alias with every web site. Specifically, the Web browser parses every Web form the user is submitting; any time it detects the user's email address in a form field, the browser executes the following conceptual steps:

- Check the *alias mapping database* maintained by the SpamWeeder to see if the user previously sent its email address to this Web site.[2] If this is a new Web site, generate a new hard-to-guess alias, interact with user's mail server to link this alias to user's real email mailbox, and record this new alias and its corresponding Web site in the alias mapping database.
- Depending on the outcome of the check above, replace the true email address submitted by the user with the new alias or the alias previously generated for this Web site. Then, send the modified form to the Web site.

Note that the user will continue using his real email address for future interactions with this Web site, e.g., if the site uses this address for login. The browser will consistently replace the real address with the same alias, and everything will work transparently as long as these interactions are confined to the same Web site.

To keep site-specific aliases transparent to the user, the mail client manipulates the mail message headers: on viewing a message, the mail client replaces the "to" address from the alias to the real user's address, storing the alias in a newly introduced header. On replying or forwarding the message, the mail client moves the alias from the special header back to the "from" address, preventing the exposure of the real address.[3] Obviously, a reply to mail sent to the real address would retain the real address in its "from" header. Note that the newly minted email aliases are kept on the mail server and are hence as secure and loss-resilient as the main address.

Some client sites attempt to prevent *outbound* spam from their network by checking that the "from" field in the outgoing mail contain the real user email address. These client sites would have to access SpamWeeder mapping database to obtain the real address for the alias before peforming the check above.

2) Precise Tracking of Spammer Trees: Site-specific aliases enable precise tracking of spammer tree roots. When an alias is trafficked along the spammer tree, any spammer on the tree will obviously use this alias in its spam. Using the alias-to-site mapping database, SpamWeeder can identify *precisely* which web site is responsible for starting the trafficking chain, no matter how many intermediaries were between the offending party that triggered the inquiry and the original trafficking source. Consequently, the mail client in SpamWeeder provides an "expose" button to the user to display the root source of the currently viewed message.

Not only does this capability allow spam victims to expose the roots of email address trafficking and take corrective actions with the appropriate ISPs and spam filters, but the possibility of irrefutable exposure would also (hopefully) serve as a deterrent to potential roots against the distribution of the email addresses entrusted to them. Currently, the anonymity entails the impunity, which in turn encourages the disruptive behavior.

In principle, with more bookkeeping, SpamWeeder could identify not just the root but all members of the spammer tree. To this end, SpamWeeder only needs to record all senders from whom it has ever received mail to a given alias. Unfortunately, the value of this information is limited because spammers typically forge sender information. Further, SpamWeeder can only identify the members of the spammer tree but not its topology (i.e., who passed the alias to whom). Fortunately, the irrefutable information about the roots of spammer trees that SpamWeeder does provide has the most value in spam prevention since without the roots there would be no spammer trees. Note that the roots cannot be forged because they represent the web sites to which the mail addresses are submitted.

A potential objection to our spammer tree tracking approach is that it fails to identify the true root in the case when a

[2]We currently assume that web sites are identified by the two top components in the hostname of the corresponding URL, e.g., the Web site for "ipl.eecs.case.edu/spamweeder" would be "case.edu". Adding support for international web sites is trivial by considering the three top hostname components.

[3]Note that this special header is never exposed outside user's computer.

user explicitly permits a site to share user's address with the site's partners, and one of these partners subsequently leaks the address. We counter that sites must screen their partners before providing them with our email addresses and assume responsibility for misbehaving partners.

3) Precise Spam Blocking: In addition to tracking spammer trees as described above, SpamWeeder allows the user to selectively block email originated by all parties belonging to a given spammmer tree, without affecting the rest of the mail. Note that it can do so without making the user explicitly handle multiple email addresses as existing approaches do. Instead, when a user identifies a particular email as spam, (s)he can simply direct SpamWeeder to block any further email from this spammer along with all other parties belonging to the same spammer tree, by clicking on a "block" button provided by the mail client. We believe the spammer tree is the right granularity for spam blocking because, barring a security breach (see Sec. III-B.4), the root of the tree is always directly or indirectly responsible for the address leak to all the tree members.

SpamWeeder fulfills this request by simply invalidating, at the user's mail server, the alias to which the current mail message was addressed. Any further mail to this alias will be naturally dropped, and this is precisely the mail originated by the members of the corresponding spammer tree: any member of the spammer tree will be blocked (because the email address it received from the root is the blocked alias), and any correspondent not on this tree will not be affected (because aliases are unique to their corresponding roots; hence the correspondent that has not obtained the user address from the blocked root would have to have a different alias to reach the user).

B. The Early Warning System

The track & kill system described above provides the user with two actions when viewing a message: the "expose" action that identifies the root of the corresponding spammer tree and the "block" action that refuses future messages from any members of the spammer tree. But these actions also provide valuable hints regarding web sites engaged in mail address trafficking. SpamWeeder utilizes this information to generate input to an *early warning system*, which can warn users against submitting their email addresses to certain web sites.

1) Web Sites Rating: The "expose" action by the user reflects the fact that the user might have considered a particular mail message as *possible* spam and was annoyed enough to want to know what web site was the root cause of it. It is a milder indication of a potentially abusive web site than the "block" action that positively indicates a "guilty verdict" by some user: indeed, the latter reflects the fact that the user finds mail ultimately traced to this web site as undesirable. Consequently, the "block" action should reduce the rating of the affected web site more severely than the "expose" action. Still, if enough users find a particular web site questionable as indicated in their "expose" action on this site, their aggregate opinion can add up to a low rating of the site.

In essence, the SpamWeeder early warning system is just a reputation system, but with the following twists:

- While existing reputation systems, in the context of spam-fighting, typically rate subnets and email addresses of mail senders, SpamWeeder rates the web sites to which users can potentially submit their email addresses. We believe binding reputation to these web sites is more effective because the web sites' identities is more difficult to change than the sender's.
- While existing reputation systems are typically used to filter spam that already occurred, the primary goal of SpamWeeder's early warning system is to *prevent* spam by alerting users against submitting their email addresses to questionable web sites.
- While existing reputation systems often rely on manual input from users and network administrators, SpamWeeder generates input automatically as a side effect of serf-serving user behavior.

2) Mail Sources Rating: Although the primary goal of the early warning system is to rate web sites and warn users against submitting their email addresses to them, it can also be used in a more traditional way, as a blacklist provider to spam filters. Similar to web sites, an "expose" action casts a doubt, and "block" action delivers a verdict, on all senders from the corresponding spammer tree. Thus, these actions can help to blacklist mail sources. Like existing anti-spam blacklists supported by SpamAssassin [15] and IronPort [6], the SpamWeeder blacklist can be bound to senders' IP or email addresses. In particular, SpamWeeder can extract sender IP from the "Received" fields in email message header. [1] pointed out that the sender IP address can also be forged and proposed to use the IP address prior the first trusted relay server (as indicated in the relay path from the message header) for blacklisting. We can adopt this approach as well.

One advantages of our approach remains that the input for blacklisting is generated automatically. Another advantage is that the actions that provide input to blacklisting apply to whole classes of mail sources (all those that belong to corresponding spammer trees) rather than individual sources. This makes more data available to rate the sources and makes mail source forging less effective.

3) Peer-to-Peer Early Warning System: While the track & kill system is confined to the client's intranet, the early warning system can benefit from cooperation across intranets. We envision a peer-to-peer network of SpamWeeder early warning systems, where each system aggregates web site ratings contributed by its intranet users and shares these ratings with its peer systems representing other intranets. We believe such a network can be designed around a set of distributed hash tables and built on top of an existing DHT substrate, for

example, OpenDHT [12]. We have left the architecture of this system to future work.

4) Security: Two issues posing a perennial challenge to reputation systems are data pollution (where an attacker distorts the reputation system by submitting large amounts of false input) and Sybil attacks (in which an attacker exacerbates the pollution effect by mimicking multiple user identities) [2]. While security issues will require a careful separate study, it appears that the SpamWeeder early warning system could be in a better position than typical reputation systems in this regard.

Within a client site, the SpamWeeder early warning system is relatively immune to Sybil attacks. Indeed, because it collects input from local mail clients, the system can ensure that input comes only from entities with valid email accounts, by having the extended mail client submit account credentials along with reputation input. While data pollution is impossible to prevent, its scope is now limited: since pollution must originate from legitimate clients (even if unbeknown to them in case their computers are hacked), low-volume pollution can be outweighed by good quality data (unless a large number of client machines are compromised) and high-volume pollution can often be isolated with appropriate data mining.

Across client sites, when early warning systems join in a P2P network, a standard web-of-trust approach can provide a level of confidence in the integrity of each peer system. In this approach, the sysadmin operating an intranet must be vouched for by some number of already trusted colleagues before her peer system, which aggregates ratings from her intranet clients, gets accepted into the network.

Unfortunately, our early warning system is not immune to data pollution in the aftermath of an intrusion. If an intruder gains a capability to snoop on network packets, he can harvest email aliases from passing messages and then traffic them with no fault of the original correspondent. Similarly, malware on an infected computer may harvest email aliases stored in the local address book. The subsequent spam will result in negative input to the early warning system, and the user may invalidate the offending alias blocking the innocent correspondent. The computer and network intrusion protection is an important issue and needs to be addressed separately. In the meantime, the reputation system heuristics will need to be tuned to minimize the possibility of these false positives.

IV. Out-of-band Address Sharing

The approach we outlined in the previous section provides precise spam protection when users submit their email addresses via a Web form, which we conjecture is the prevalent way of providing email to untrusted parties. Still, users occasionally provide their email addresses *out-of-band*, e.g., over the phone, to obtain a purchase confirmation, or through a business card. Out-of-band address sharing presents a difficult challenge: no system can hide the address if the user gives it to the other party directly. We can only offer some tools and recommendations, discussed below, to ameliorate this problem.

First, SpamWeeder provides a Web interface where the user can input the name of the party being given the email address and receive a SpamWeeder-generated alias. In this way, SpamWeeder can track this alias and block any possible spam originated from this transaction in the normal way. Obviously, the alias in this case is no longer fully transparent to the user. However, this is still a significant improvement over ad-hoc email accounts because the user can forget about the alias after giving it to the other party. The user will only see its true address in any subsequent communication.

Second, in other situations, e.g., when the user is away from a computer and cannot generate the alias above, or in the business card scenario, the user can maintain a separate mail account just for these occasions. This is similar to what most of us do today when we have a separate highly exposed account, but with an important difference. Because the scope of this account's usage is limited to mostly one-time interactions, switching to a different account no longer entails manual profile update with multiple web sites. In the business card scenario, the user may still need to inform correspondents of the address change, but the scope of the problem is now limited.

V. Prototype Implementation

We have implemented a proof-of-concept prototype of SpamWeeder and are working on its evaluation plan. The prototype is publicly available at [17]. It was implemented to run on a variety of operating system; we tested it on Windows XP, Mac OS X (Tiger and Leopard versions), and Linux 2.6 (the mail client needs to run with root privileges on Linux).

Figure 1 presents the high-level architecture of the prototype (the reputation system has not been implemented and is included in the figure for completeness). Our prototype uses an unmodified Xmail mail server [18] and is built in the form of extensions to Mozilla Firefox browser [9] and Mozilla Thunderbird mail client [10] as well as an new component referred to as the *coordinator*.

The extended browser examines every HTML form submitted by the user and replaces the user's email address with an appropriate alias as described in Section III-A.1. For example, if user's real email address is "John@mail.com", the extension would substitute it with an alias that might look like "J-6476829908cdc259a24604ce263a6a16@mail.com".

The extended Thunderbird mail client keeps the aliases hidden from the user by replacing them with the real address on message viewing, and keeps the real address hidden from the correspondents by replacing them back with the aliases on message replying and or forwarding. The extension further adds two extra tools into Thunderbird's "Tools" menu, which invoke the "expose" and "block" actions on a currently viewed message as described in Sections III-A.2 and III-A.3.

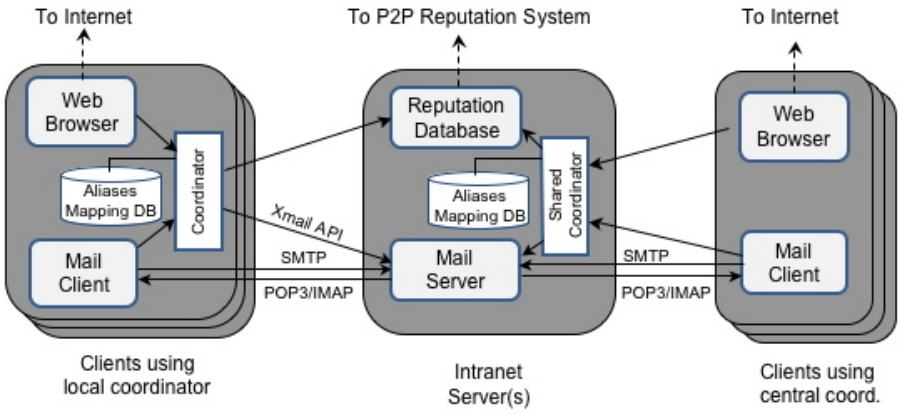

Fig. 1. A conceptual architecture of SpamWeeder

The coordinator maintains the alias mapping database and encapsulates the interactions of the browser and mail client with the mail server and, in the future, the reputation system. Having a separate coordinator leads to flexible deployment and interoperability of the SpamWeeder components. In one deployment option, shown in the right-hand side of the figure, clients use a shared coordinator, perhaps co-located with the mail server. This is the simplest option to the users, but it might raise possible privacy concerns (it involves a shared alias mapping database that stores the web sites to which the user has submitted its mail address). Alternatively, a user can deploy the dedicated coordinator on their local machine (as illustrated on the left-hand side of the figure). This keeps the alias database private but requires somewhat more configuration effort. Note that a client site can use a combination of both deployments for different users.

The distinct coordinator also supports interoperability with different mail servers. Mail servers have no standard administration interface for adding and removing aliases. The coordinator acts as a wrapper that encapsulates the mail server specifics. In the future, we plan to publish the protocol for the interactions between the coordinator and the browser and mail client. This would further decouple various components of SpamWeeder - Web browsers, mail clients, coordinators, mail servers, and the reputation system.

VI. Conclusion and Outlook

This paper presents a simple idea of fighting spam email, which allows users to precisely expose parties engaged in email address trafficking and block all email from all parties belonging to a given trafficking chain. We achieve this by dynamically assigning unique email aliases to web sites contacted by the users, and by making these aliases transparent to the users. Our approach can be autonomously adopted by a client intranet *today*, without any changes to the rest of the email infrastructure. The output from the system can be used for a new reputation system that would warn against untrustworthy web sites and enhance traditional spam filters. We implemented a prototype of our approach and plan to use it estimate the impact of our approach on the performance of client installations (in particular, the effect of a large number of mail aliases on mail server performance) as well as its effectiveness and usability.

Acknowledgements

We would like to thank Mark Allman of ICSI and Shubho Sen of AT&T Labs for discussions of the ideas discussed in this paper. We also thank Mark Allman for insightful comments on early drafts of this paper, which significantly improved its current version. We are grateful to Lann Martin of CWRU for help with manipulating message headers in Thunderbird. Finally, we thank the anonymous referees for their helpful comments.

References

[1] Alex Brodsky and Dmitry Brodsky. A distributed content independent method for spam detection. In *First workshp on Hot Topics in Understanding Botnets*, 2007.
[2] J. R. Douceur. The sybil attack. In *Int. Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.
[3] G. Fumera, I. Pillai, and F. Roli. Spam filtering based on the analysis of text information embedded into images. *Journal of Machine Learning Research*, 7, 12 2006.
[4] Robert J. Hall. How to avoid unwanted email. *Comm. of the ACM*, 41(3), March 1998.
[5] J. Ioannidis. Fighting spam by encapsulating policy in email addresses. In *The Network and Distr. Sys. Security Symp.*, 2003.
[6] http://www.ironport.com.
[7] B. Krishnamurthy. SHRED: Spam harassment reduction via economic disincentives. IETF-56 talk. www.research.att.com/ bala/papers/shred-ietf56-talk.pdf.
[8] T. Loder, M. Van Alstyne, and R. Wash. An economic solution to the spam problem. In *ACM Conf. on E-Commerce*, 2004.
[9] Mozilla Firefox. http://www.mozilla.com/firefox/.
[10] Mozilla Thunderbird. http://www.mozilla.com/thunderbird/.
[11] Results of Anti-Spam Solution Testing: http://www.opus1.com/www/whitepapers/antispamfeb2007.pdf.
[12] S. Rhea, B. Godfrey, B. Karp, J. Kubiatowicz, S. Ratnasamy, S. Shenker, I. Stoica, and H. Yu. OpenDHT: a public DHT service and its uses. In *SIGCOMM*, pages 73–84, 2005.
[13] B Ross, C Jackson, N Miyake, D Boneh, and JC Mitchell. Stronger password authentication using browser extensions. In *the 14th Usenix Security Symposium*, 2005.
[14] http://spamarrest.com/.
[15] The Apache SpamAssassin Project. spamassassin.apache.org.
[16] www.spamgourmet.com.
[17] http://ipl.eecs.case.edu/spamweeder/.
[18] Xmail server. http://www.xmailserver.org/.