Spam Attacks: P2P to the Rescue

Ernesto Damiani
DTI - Università di Milano
26013 Crema - Italy
damiani@dti.unimi.it

S.De Capitani di Vimercati
DTI - Università di Milano
26013 Crema - Italy
decapita@dti.unimi.it

Stefano Paraboschi
DIGI - Università di Bergamo
24044 Dalmine - Italy
parabosc@unibg.it

Pierangela Samarati
DTI - Università di Milano
26013 Crema - Italy
samarati@dti.unimi.it

Andrea Tironi
DTI - Università di Milano
26013 Crema - Italy
tironi@dti.unimi.it

Luca Zaniboni
DTI - Università di Milano
26013 Crema - Italy
zaniboni@dti.unimi.it

ABSTRACT

We propose a decentralized privacy-preserving approach to spam filtering. Our solution exploits robust digests to identify messages that are a slight variation of one another and a peer-to-peer architecture between mail servers to collaboratively share knowledge about spam.

Categories and Subject Descriptors: C.2.0 [Computers-Communication Networks]: General –*Security and protection*; C.2.4 [Computers-Communication Networks]: Distributed Systems –*Security and protection*

General Terms: Security, Design

Keywords: Spam filtering, structured P2P, reputation

1. INTRODUCTION

Spam has been known as a major problem since long, but its impact on the global network infrastructure has now reached epidemic proportions (http://www.spamcop.net/spamstats.shtml). Due to customers' complaints, governments started to contemplate anti-spam legislation (EC Directive on Privacy and Electronic Communications, 2002/58/EC), while several companies began offering spam filtering products to mail server operators and ISPs. While most commercial anti-spam filters claim a much higher success rate than 95% in identifying spam, a huge amount of it still winds up in users' in-boxes, even when client-side and server-side filters are used in conjunction. It may be argued that this lack of success in the war against spam is partly due to the elusive nature of the notion, which is difficult to identify by means of a software program. Recently, some approaches based on the collaborative sharing of knowledge about spam between P2P users have been proposed [2, 4]. While these approaches represent a first step toward the design of a P2P collaborative spam filtering solution, they do not take into consideration some important aspects (e.g., the confidentiality of the messages and the robustness against attacks). We build on the idea that a P2P enabled polling mechanism can help in determining *what a community considers to be spam* and getting rid of it. Our proposal is aimed at achieving both *flexibility* and *effectiveness*. Firstly, our hierarchy-aware P2P architecture can be deployed in a variety of organizational situations, in presence of multiple mail servers of different size and reliability. Secondly, our P2P-based anti-spam filtering engine rigorously protects users' privacy, avoiding to disclose the content of the messages they receive, and is robust about countermeasures that spammers themselves may take to impair its effectiveness.

WWW2004, May 17–22, 2004, New York, New York, USA.
ACM 1-58113-912-8/04/0005.

2. ANTI-SPAM P2P ARCHITECTURE

Our anti-spam system is based on a three-tiered architecture, with users at the lower level and a P2P network connecting mail servers above them. The P2P network comprises of two families of nodes: peers and super-peers [3]. Each set of users together with their mailer form a *cluster*. Intra-cluster data communication takes place via direct links between the users and their mailer, while inter-cluster communication takes place via the P2P network. In our approach users do not participate themselves as nodes of the P2P network for performance and privacy reasons. In particular, spam reports by users are communicated by the mail server without indication of the identity of the users who originated them. Each mail server knows the identity of its users (although it does not propagate it in association with reports), so we can safely assume that each user is identified by her mailer via a unique identifier. As for mailer's identity, we rely on the fact that machines playing a specific role as mail servers are likely to have a network-wide name registered in the Internet *Domain Name System* (DNS), responsible for translating names to IP addresses.[1]

We exploit mail servers as a distributed repository of knowledge about spam, to be used by our filtering service. Each mail server, in turn, gets to know which messages are spam simply by (transparently) polling the opinions of its users.

3. PROTOCOL

We assume each mail server s is associated with a pair of keys, $\langle$public,private$\rangle$, and it uses its private key to sign outgoing communications. Furthermore, we assume that each message m can be identified by a digest that is robust against typical disguising attempts, so that we can identify two messages to be *the same message* if they map to a similar digest, even if their text is not identical.

At each tier, information is maintained about spam detected or received. Intuitively, the idea is that the super-peers in the network maintain a distributed collection of spam digests that peers have identified; peers can query this collection to obtain information about unknown emails. While our approach can be adapted to different ways of managing spam information, here we assume that each mail server maintains the following information. For each

[1]http://www.ietf.org/internet-drafts/draft-danisch-dns-rr-smtp-03.txt.

message m, the mail server records the number of copies directed to its users that it has received; the number of users who have reported the message as spam; and the number of users who have submitted a contrary report (if the message was sent to them already tagged as spam). The mail server also maintains control information, mainly in the form of thresholds it uses to determine when to enact polling or to tag a message as spam. In particular, it maintains: the list of messages classified as spam; the number of occurrences of the same message that triggers the suspicion of a bulk mailing; the number of user reports about a message needed by the mail server to classify it as spam; the number of contrary reports referred to a message that the mail server considers as a sufficient indication that the message should have not been tagged as spam; the threshold that measures whether external reports are sufficient to consider the message as spam. In addition, each mail server maintains a reputation for each other mail server s in the network, which measures the server's credibility in s's statement and which it uses to properly weight notifications coming from s. Each mail server acting as a super-peer in our P2P network maintains also track of spam reports received from the mail servers referring to it. Each spam report is stored at the super-peer in the form in which it has been received, i.e., signed by the mail server that has expressed it, so that further recipients of the report will be able to assess its authenticity.

User tier

At the user tier, users receive emails. Upon reception of a message m, a user can report the fact that m is spam to its own mailer. We assume that the decision that the message is spam has been done personally by the user and that generating the report, i.e., explicitly countersigning the message as spam, does not require any additional effort on the part of the user. If the email received by the user has already been tagged as spam by the mail server, and the user agrees with that, the user does not need to do anything else. On the contrary, if the user does not agree with the current assessment of the message she can send a *contrary report* to her mailer.

Peer tier

At the peer tier, each mail server receives emails directed toward its users as well as spam notifications or contrary reports from its users. As for emails, when the number of received occurrences of a given message reaches the suspicious threshold, the server sends a query to the super-peers inquiring whether the message has been reported as spam by other mailers. In response to such a query the mail server will receive a set of signed spam reports. It then performs an aggregation of the reports, weighting them differently depending on the reputations of the mail servers involved, to determine whether m is to be considered spam. If the aggregation produces a value greater than the specified threshold, the mail server adds the message to the spam catalog, so to be able to tag as such any other copy of the message its users will receive. As for spam notifications, the server records any spam notification from its users. When the number of spam notifications reaches the established threshold, the server adds the message to the spam catalog and sends a message to its super-peers reporting that it considers the message to be spam. Note that this notification to the super-peers comes directly from the mail server and does not forward the identity of the users that have reported the message as spam. The reporting can include, if the server so wishes, a confidence the server has in making such a statement and that intuitively relates to how stringent or loose the triggering threshold is. The reporting to the upper level is signed by the mail server, and in such a form it is stored by the super-peers for further communication.

Super-peer tier

Mail servers at the super-peer tier also serve as collectors and pollers of spam reports. The super-peers' additional workload consists in managing spam reports and spam inquiries coming from the mail servers that refer to them, or from other super-peers. Upon reception of a new spam report from a mail server, the super-peer adds a corresponding entry in its catalog. Upon reception of a query from a mail server, the super-peer will both broadcast the query toward other super-peers in the P2P network. Each of the super-peers receiving the query will respond on the network returning the reports about the message that appear in its spam report catalog. The super-peer directly inquired will then return all the reports received as well as those it has locally stored to the inquiring mail server. As for all the communications of our protocol, the query response is signed by the super-peer.

4. NOTES ON THE APPROACH

It is worth to point out some key aspects of our solution. First, super-peers provide a communication channel between mail servers and do not perform any intermediate aggregation of reports. This way no complete trust in the super-peers is required (they cannot fake reports) and each mailer can weight reports depending on the reputation it has on whomever expressed them. Another aspect worth noticing is that every communication is always signed. The reason for this is guaranteeing the authenticity of the report content as well as of its originator.

In our approach, two key security aspects have to be taken into consideration: *1)* an effective digest mechanism and *2)* a secure process for sharing these digests. For the digest mechanism, we used a slight variation of the Nilsimsa digest (http://lexx.shinn.net/cmeclax/nilsimsa.html) computed on the message after filtering out the usual noise spammers may include (e.g., random spaces or letter permutations faking spelling errors). The security of the sharing mechanism is provided by the way the protocol is designed [1]. Also, we assume a node connects to a super-peer only if it is a reliable node registered in the DNS system as the mail server for a domain and that the node will connect with several super-peers (thus providing redundancies which will permit to identify anomalies).

5. CONCLUSIONS

We presented a solution exploiting the P2P potential to make a first step toward a spam-free email system. This system would be a significant improvement to the current Internet infrastructure, as it can be testified by longtime Internet users who remember the email experience of a few years ago that is today lost for the great majority of the user population. Our spam report sharing protocol can be extended to the inclusion of other methods for classifying spam (e.g., Bayesian filters).

Acknowledgments

This work was supported in part by the European Union within the PRIME Project in the FP6/IST Programme under contract IST-2002-507591 and by the Italian MIUR within the KIWI and MAPS projects.

6. REFERENCES

[1] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. Managing and sharing servents' reputations in P2P systems. *IEEE TKDE*, 15(4):840–854, July/August 2003.

[2] J. Metzger, M. Schillo, and K. Fischer. A multiagent-based peer-to-peer network in java for distributed spam filtering. In *Proc. of the CEEMAS*, Czech Republic, June 2003.

[3] B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proc. of the ICDE*, Bangalore, India, March 2003.

[4] F. Zhou, L. Zhuang, B. Zhao, L. Huang, A. Joseph, and J. Kubiatowicz. Approximate object-location and spam filtering on peer-to-peer systems. In *Proc. of the ACM/IFIP/USENIX International Middleware Conference*, June 2003.