CLOUDMARK®

A Reputation-Based Approach for Efficient Filtration of Spam

ABSTRACT

Cloudmark's collaborative approach to fighting spam—with its additional attribute of "reputation" added to a community of real users who identify, report, and corroborate suspect messages in real time—has proven more effective and faster than traditional blocking or filtering methods. At the core of the Cloudmark Global Threat Network is the Trust Evaluation System. The Trust Evaluation System ensures the "reputation" of reporters by tracking how often the larger recipient community agrees with their assessment of a message. In addition, Cloudmark uses an automated system of highly-proficient, fingerprinting algorithms. Advanced Message Fingerprinting maintains the privacy of the content and reduce the amount of data to be analyzed. Once a message fingerprint is cataloged as spam, all future messages matching that fingerprint are automatically filtered. Because a reputation-based collaborative system does not draw blanket conclusions about terms, hosts, or people, it has proven to increase accuracy, particularly as it relates to false positives and critical false positives, while simultaneously decreasing administration costs.

INTRODUCTION

Spam is everywhere. Not only is it an annoyance, it erodes the very productivity gains brought about by the advent of information technology. Workers spending hours plowing through email everyday have to contend with a significant amount of illegitimate email. Although automated spam filters have dramatically reduced spam, the amount of time and training required to use these filters often equals or exceeds the time required to simply delete unfiltered spam.

Considering that spam essentially consists of a single message seen by a large number of individuals, there is no reason why the training load associated with an automated spam filter can't be distributed across a large community of individuals who all receive the same unwanted messages. In this scenario, a self-organizing community collectively classifies new messages as "spam" or "not spam."

The collaborative decision making of the community not only reduces the training cost and learning curve for individuals and the administrative costs for companies, it also reduces the cost of accuracy, such as that associated with misclassification of a message. For our discussion, misclassification of messages fall into three categories:

1) Misclassification of legitimate email as spam—termed a "false positive"
2) Misclassification of legitimate, and business-critical, email as spam—termed a "critical false positive"
3) Misclassification of spam as legitimate email—termed a "false negative"

Of potential misclassification, false criticals and false positives are of greatest concern. False positives and critical false positives have a more immediate and significant cost to organizations, while false negatives erode overall productivity. (Coincidentally, in our research of anti-spam solutions, we've noted that false positives and critical false positives are the most underreported of all spam misclassifications.) Therefore, if a community— by properly training a spam filter—reduces the false positive and critical false positive rate, the cost incurred to the sender of a legitimate message misclassified as spam is reduced, as well as the cost incurred to the recipient who needs the message.

Reducing training and accuracy costs, particularly accuracy as it pertains to false positives, is the primary motivation for creating collaborative spam filtering architectures, such as Vipul's Razor™ [5] and its progeny. These systems, described below, enable users to identify and submit fingerprints of messages. When the fingerprinted messages are substantiated as spam by a community of users, they are placed in a catalog of known spam messages.

THE FUNDAMENTALS OF CLOUDMARK TECHNOLOGY

The primary author's involvement with spam began in 1996 when spam started trickling in through his 300-baud modem as a result of USENET posts. At the time, he was researching anonymous remailers7 and onion routing8, and he wanted to consider anti-spam in a broader context that included messaging systems where the sender was anonymous and the network topology was unknown. The fundamental idea that met these design criteria was to provide a way for the first few recipients to identify a message as spam, and then to have an automated way of informing the rest of the user community that the message is spam so the message could be filtered out before others read it. In other words, collaborative human intelligence "identifies" a message as spam and an automated technology verifies and prevents its proliferation.

The first prototype of this system, dubbed Vipul's Razor, was released as an Open Source project in 1998. In 2001, along with a major update to Razor (Razor2), Vipul co-founded a company called Cloudmark to work on messaging security technology in a dedicated setting. Today, the collaborative classifier that underlies Razor2, and all of Cloudmark's products, is known as the Cloudmark Global Threat Network service. The service operates at a massive scale—filtering spam globally for some 180 million people. The goal of the service is to accurately determine if a message is spam or legitimate email based on the first few reports so that only a few reporters are necessary to train the classifier for a new spam attack. At the core is a reputation metric analyzer that ensures the integrity of user-submitted feedback by modeling historical consensus and disagreement in the recipient community. This automated and real-time approach significantly reduces the individual training and corporate administrative cost.

RELATED DIRECTIONS

The alarming increase of spam traffic in recent years has fostered considerable research and development of anti-spam technologies. Many novel methods have been discovered, evaluated, and deployed. Some of the more popular approaches, described below, are address whitelisting, IP blacklisting, and Bayesian classifiers.

Address Whitelisting

A simplistic, yet popular, classifier known as "address whitelisting" permits delivery of mail only from people known to, and approved by, the recipient. The premise is that exclusion of spammers from the "allowed sender" list keeps spam out of the mailbox. This classifier is very easy to implement and works well for recipients who communicate only with a well-defined and static set of correspondents. However, its performance is suboptimal in the general case, where recipients must continually update and train their own version of the classifier as their correspondence network expands. The accuracy cost is also high because address-based classification is susceptible to address forgery, and a well defined *allow list* hampers first-contact communications. Many of the shortcomings associated with address whitelisting can be alleviated with sender authentication schemes and reputation-based training, as we discuss later in this paper.

IP Blacklisting

Another popular method for stopping spam is blocking all incoming traffic from mail servers that are known to send spam or that have the potential to send spam due to misconfiguration. Servers that wish to reject mail from known spam mail servers can fetch a list of known spam-generating mailservers by IP address (commonly referred to as RBLs, or Real-time Blackhole Lists) and block all inbound connections from these systems. While this is a powerful tool for stopping the most abusive mail servers on the internet, blocking every incoming message from a specific mail server, even one predominately used for spam, may increase the false positive and critical false positive rate of the spam filter. Without a method for rapidly providing feedback about servers on the block lists that are likely also generating legitimate traffic, organizations striving for zero or near-zero false positives are forced to look for technologies that provide greater granularity regarding the disposition of a message.

Bayesian Classifiers

Statistical text classification systems, like Naive Bayesian (NB) classifiers[6], classify based on the semantic similarity of incoming mail to a corpus of prior messages. NB classifiers tokenize mail content into words and phrases (or other linguistic units) and register the probability of the appearance of various words and phrases in spam and legitimate messages. The learned set of linguistic units and their corresponding probabilities constitute the "hypothesis" used to classify incoming mail. While statistical text classifiers must be trained incrementally by the recipient, the training events are rare, compared to the frequency of incoming mail. Most implementations come with a built-in hypothesis that serves as a starting point to offset the training requirements of the tool. Once trained, statistical text classifiers are quite accurate at identifying legitimate communications and reasonably good at identifying spam. They are known to perform best in single-user environments where the training corpus accurately reflects specific user preferences. Most real-world deployments of statistical text classification are augmented with orthogonal classifiers, such as blacklisting, to derive acceptable spam detection performance.

ARCHITECTURE

The Cloudmark Global Threat Network service is a community-based, rather than an individual-based, filter training system. It does not rely upon any one semantic analysis scheme; rather, it uses a large set of orthogonal fingerprinting schemes that are trained by the community.

The service consists of four important architectural components: *Agents, Nomination servers, Catalog servers*, and a reputation system known as the *Trust Evaluation System*. The Agent software suite comprises a variety of software packages used by email recipients to report on messages, such as "Message is spam" or "Message is not spam". This feedback is routed to the Nomination servers, and small fingerprints are generated on the message. The fingerprint size is on the order of 14 to 20 bytes. Not only does fingerprinting a message, rather than transmitting its entire content, protect the privacy of the email recipient, it also dramatically reduces the cost associated with transmitting, storing, and processing feedback. We discuss spam fingerprinting in more detail later in this paper.

The submitted feedback is passed to the Nomination servers, which collect all fingerprints nominated by the recipients as either potentially new spam or as false positives. If all users were equally and consistently capable of determining the disposition of a message, and a single user could nominate a fingerprint as "spam" or "not spam," then the fingerprint would be redistributed to the community. The Cloudmark Global Threat Network service, however, requires submitted feedback to be corroborated by multiple, trusted members of the community. The logic that determines the community's faith in the validity of a fingerprint is embodied in the Trust Evaluation System. It alone determines which new reports are valid or invalid. An overview of how the Trust Evaluation System works is presented in a following section.

Once the Trust Evaluation System determines that a fingerprint is "spammy," the fingerprint is added to the Catalog server. All messages received by a user are fingerprinted, and the fingerprints queried against the Catalog server. If the queried fingerprint exists in the Catalog server, the agent filters the message as spam. If the fingerprint is not in the Catalog server, and the recipient feels that the message is spam, then the recipient submits a fingerprint to the Nomination server and the process begins again.

An agent residing on a user's desktop computes a fingerprint of a new email and submits this fingerprint (1) to the Spam Fingerprint Catalog server. If the Catalog server has the fingerprint in its database, the server tells the agent (2) that the message has been flagged by the community as spam. If the fingerprint is not in the Catalog server, and the recipient feels that the message is spam, the recipient instructs the agent to transmit the fingerprint (3) to the Nomination server, which in turn, inserts the fingerprint into the Nomination database (4). The Trust Evaluation System continually watches (5) the Nomination database to see if there are any new fingerprints –that have been submitted by multiple, trusted email recipients. If enough trusted recipients submit the same fingerprint, the fingerprint is promoted to the Catalog server, and the process continues.

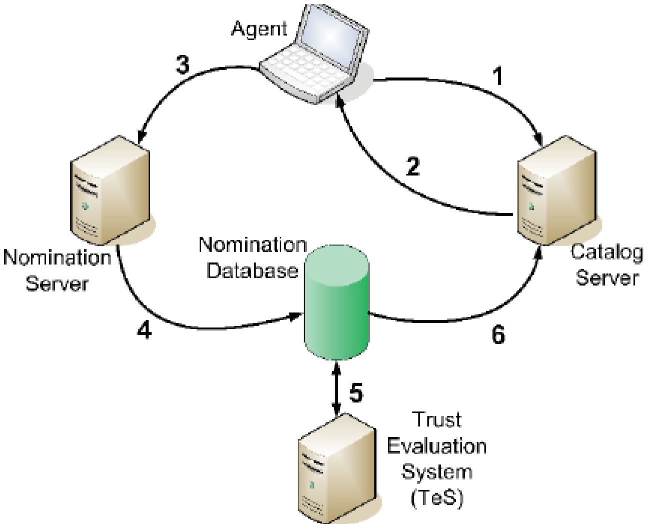

Figure 1: The process flow of the Cloudmark Global Threat Network service.

Trust Evaluation System

The Trust Evaluation System ES is the reputation metric, or trust system, of GTN service that evaluates every new piece of feedback submitted to the Nomination servers. The primary function of TeS is to assign a "confidence" to fingerprints—a value between C_{mn} (legitimate) and C_{mx} (spam), based on the "reputation" or "trust level" of the individual reporting the fingerprint. The trust level, *t*, is a finite numeric value attached to every community reporter. The value *t* is, in turn, computed from the corroborated historical confidence of the fingerprints nominated by the reporter. The circular assignment effectively turns the classifier into a stable closed-loop control system.

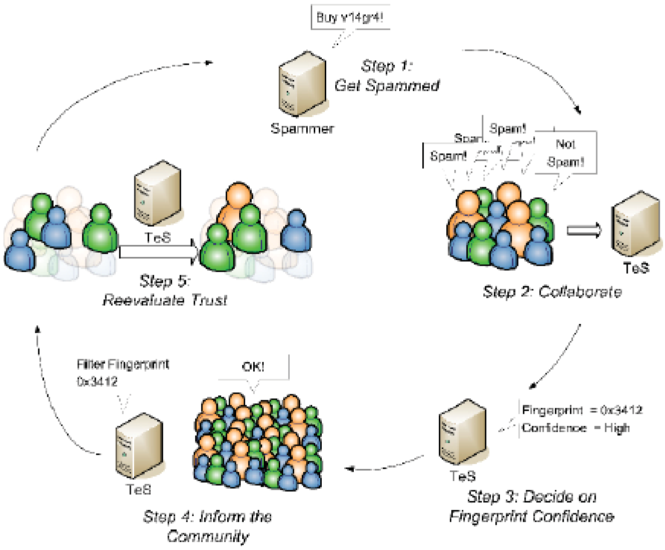

Figure 2: The process flow of the Trust Evaluation System.

The heart of the Cloudmark Global Threat Network service is the Trust Evaluation System. This is the component that determines both the confidence the community has in the disposition of a fingerprint and the trust that the system places in the decisions made by members of the community. In a continuous process, members of the community receive new spam (1) and report their feelings ("spam" or "not spam") about the message to the Nomination server, which in turn reports it to the sytstem (2). Based upon the trust associated with each individual reporter, it assigns confidence to the fingerprint (3) and reports it to the GTN service for distribution to the community (4). The system then reevaluates the community trust values to determine who should gain and lose trust as a result of their individual assessments of the message.

Just as in the real world, trust is earned slowly and is difficult to attain. New recipients start with a trust level of zero. In the very beginning (at the launch of the classifier), there were only a few hand-picked recipients with a high trust level. As zero-valued, untrusted community members provide feedback, TES rewards reporters whose feedback agrees with those of highly-trusted, highly-reputable members of the community. In other words, TES assigns trust points to recipients when their reports are corroborated by other highly trusted recipients. In practice, for every fingerprint that achieves a high confidence meaning that the fingerprint was reported and corroborated by highly trusted recipients—TES gives one of first reporters of the fingerprint a small trust reward.

Untrusted recipients who report often and report correctly eventually accrue enough trust rewards to become trusted recipients themselves. Once trusted, they implicitly begin to participate in the process of selecting newer trusted recipients. In this manner, TES selectively inducts a community of "highly-reputable," "highly-trusted" members—reporters who routinely make decisions that are honored by the rest of the community. TES also penalizes recipients who disagree with the trusted majority. Penalties are harsher than rewards, so while gaining trust is hard, losing it is rather easy.

The second aspect of TES's responsibility is to assign confidence to fingerprints. Fingerprint confidence is a function of the reporter's trust level and the disposition (block/unblock) of their reports.

TES updates confidence in real-time with every report. Once the confidence reaches a threshold, known as *average spam confidence*, it is promoted to Catalog servers. If a promoted fingerprint is unblocked by trusted recipients, its confidence can drop below the average spam confidence, which results in its immediate removal from the catalog servers. The real-time nature of confidence assignments results in an extremely responsive system that can self correct within seconds.

In more formal terms, a given set of fingerprint reporters, *R*, who each have a trust level t_r , send in reports that have a *disposition* d_r , where d_r = -1 if the fingerprint misclassifies a legitimate message as spam and t_r = 1 if the message is spam. After a number of fingerprints are collected, it is possible to compute a fingerprint confidence using the following equation:
However, it is important to note that TeS uses a variation of the above algorithm to reduce its attack vulnerability.

Emergent Properties of TES

TES has several desirable, even surprising, emergent properties when deployed on a large scale. These properties are critical to the effectiveness of the system and typical of well-designed reputation metrics. We discuss some of these properties in this section and contrast them with related properties of other anti-spam approaches.

$$confidence = \min\left\{c_{max}, \max\left\{c_{min}, \sum_{r \in R} d_r \max\{t_r, 0\}\right\}\right\}$$

Responsiveness

TES's reward selection metric prefers those recipients who report correctly and early. This means that over time TES can identify all such reporters whose initial reports have a high likelihood of being accepted as spam by the rest of the community. As the group of trusted recipients becomes larger, the first few reports are extremely reliable predictors of a fingerprint's final disposition. As a result, GTN can respond extremely quickly to new spam attacks.

Anti-spam methods that either require expert supervision, or that are inherently unable to train on individual samples, have significantly longer response latencies. These systems are unable to stop short-lived attacks that are not already addressable by their existing filtering hypotheses.

Self Correction

The ability to make negative assertions ("Message is not spam"), combined with the dynamic nature of the confidence assignment algorithm, permits speedy self-correction when the initial prediction is incompatible with the consensus view. Since confidence and trust assignments are intertwined, community disagreement results in immediate correction of the confidence of fingerprints, as well as a trust reduction for reporting the fingerprints as spam. This results in a historical trend toward accuracy because only the reporters who consistently make decisions aligned with the consensus retain their trusted status. From a learning perspective, the reporter's reputation or trust values represent the entire history of good decisions and mistakes made by the classifier.

Modeling Disagreement

One of things we learned almost immediately after the launch of TES was that certain fingerprints would wildly flip-flop across the average spam confidence level. These fingerprints usually represented newsletters and mass mailings that were considered desirable by some and undesirable by others. The community of trusted recipients disagreed on the disposition of these fingerprints because there was no "real" community consensus on whether or not the message was spam. By modeling the pattern of disagreement, we taught TES to identify this kind of disagreement and flag such fingerprints as *contested*. When agents query contested fingerprints, they are informed of the contention status so they can classify the source emails based on out-of-band criteria, which can be defined subjectively for all recipients.

Contention modeling is extremely important for a collaborative classifier like the Cloudmark Global Threat Network service because it scopes the precision of the system. If the limitations of the classifier are known, other classification methods can be invoked as required. In the Cloudmark Global Threat Network service, contention logic is also a catch-all defense against fingerprint collision. If a set of spam and legitimate email happen to generate the same fingerprint, the fingerprint is flagged as contested, which excludes its disposition from the classification decision. Historically aggregated contention rates in the service are an indicator of the level of disagreement in the trusted community. The level of disagreement in the service is very low, which implies that the trust model can successfully represent the collective wisdom of the community.

Most machine learning systems, including statistical text classifiers like Naive Bayes, are unable to automatically identify contested documents. This is why statistical classifiers tend to work better in single-user environments where recipient preferences are consistent over time.

Resistance to Attack

An open, user feedback-driven system like the Cloudmark Global Threat Network service is an attractive attack target for spammers. There are essentially two ways of attacking the service. One is through a technique called *hash busting*, which attacks fingerprinting algorithms by forcing them to generate different fingerprints for each mutation of a spam message. Fingerprinting algorithms are designed to be resistant to hash busting, as discussed later in this paper. The second vector for attack is through incorrect feedback. Most commonly, attackers attempt to unblock their mailings before broadcasting them to the general population. However, an attacker must first be considered trusted in order to affect the disposition of a fingerprint. In order to gain trust, the attacker must provide useful feedback over a long period of time, which requires blocking spam that others consider to be spam. In other words, spammers must behave like good recipients for an extended period of time to get even a single identity to be considered trusted. If they do spend the effort building up a trusted identity, the amount of damage they can do with one or few trusted identities is negligible because the disagreement from the majority of the trust community will result in harsh trust penalties for the spammer identities. As the pool of trusted users grows, it gets harder to gain trust and easier to lose it. Participation is proportional to the strength of attack resistance.

Expert-supervised systems are resistant to such attacks by definition but are unable to scale to a large number of experts. Similarly, statistical text classification systems must go through supervised training to avoid corpus pollution. Supervision limits the amount of training data that can be considered. One real-world limitation, for example, is in a supervised classification system's inability to adequately detect "foreign language" spam—that is, spam in languages not understood by supervisors.

A Collective Definition of Spam

Perhaps the most passionately debated question in anti-spam circles—what constitutes spam—still lacks a universal answer. Even though discussions tend toward philosophy, it is an important question to answer for creators and users of anti-spam classifiers. Luckily for us, one of the side effects of the Global Threat Network service is a generative definition of spam. The emails that achieve a high confidence are "spam" and the ones that don't are "not spam." The emails that get contested are spam for some and not others. These metrics are a direct representation of the beliefs of the community as to what spam is.

Many anti-spam systems—like IP blacklists operated by a small group of otherwise extremely informed experts—often train their system based on personal experience, strict policies, or minority perspective resulting in wide-scale misclassification. The cause of poor training is often limited visibility, referred to as *availability bias*[1], and has led to many lawsuits and general distrust of anti-spam techniques. A community-based classifier is resistant to such biases.

Fingerprinting and Its Impact on System Performance

The Global Threat Network service relies on advanced message fingerprinting algorithms for identifying messages classified by the community as being spam. All of the fingerprinting algorithms employed by the service take the same general form: They are a many-to-one mapping between messages and the field of 14 to 20 byte numbers. A good fingerprinting algorithm would map many similar messages, namely mutations of one another, to the same fingerprint while not mapping any additional messages to the fingerprint.

We have formalized these two properties of fingerprinting algorithms by creating two metrics: *multiplicity* and *cross-class collision*. Multiplicity encapsulates the versatility of a single fingerprint to classify mutations of a single spam species. Cross-class collision is the extent of intersection between fingerprints generated from a corpus of spam and legitimate messages. It measures the *potential* rate at which the fingerprint could cause false positives in the system. While the creation of fingerprinting schemes is a creative process, these metrics work as a general framework to evaluate the efficacy of new fingerprint schemes.

These metrics were developed in-house because literature on fingerprint-based spam filters is sparse. This paper constitutes the first time our internal fingerprint evaluation system has been publicly documented.

We begin by denoting the set of spam messages by S and the set of non-spam messages as S' . It is well known that spammers will mutate a single message to escape naive signature schemes. We define a *mutation set* as being an agreed-upon set of messages to be derived from a single source message, or to share a common message ancestor. We assume that mutation sets do not overlap.

Classifying spam perfectly into mutation classes is, for all practical purposes, an impossible task. However, for the evaluation of new fingerprinting algorithms, a reasonable approximation can be achieved by classifying corpora manually.

We denote a class of fingerprinting algorithms by f . f_c is a perfect cryptographic hash, which generates a unique value for every unique message, regardless of how small a mutation may exist between any two messages. f_o is a fingerprint generated by a limited oracle, where all spam messages in a single mutation class generate the same fingerprint. f_e is the fingerprinting algorithm that we develop. The fingerprinting algorithm should operate exactly like the oracle and generate a single fingerprint for all messages in the same mutation class.

$$S = \{s_1, \ldots, s_i\}$$

$$S' = \{s'_1, \ldots, s'_j\}$$

$$\mathbf{M} = \{M_1, \ldots, M_k\}$$

$$1 \le x \le k : M_x \subseteq S$$

$$1 \le x < y \le k : M_x \cap M_y = \emptyset$$

The set of fingerprints previously generated by system users is denoted by F_{cat} , which is an abstraction of the catalog server. Additionally, S_{cat} is denoted as the set of spam messages in the catalog server.

On the back end, two factors drive the accuracy and the false positive rate. We know that any new spam that comes in is either something we have seen before, a mutation of an old spam campaign, or a completely new campaign. If it is an old spam campaign, then it should be in the content filtering system. If it is a new campaign, the data feedback system will report the message fingerprints into the system, and the fingerprints that identify the attack will be rapidly propagated throughout the system and community. Regarding mutations of old campaigns, we don't want to allow a spammer to apply a simple mutation, such as reformatting the message or changing a URL, to avoid a previously-generated signature. To prevent mutations of old campaigns from slipping past our system to the users, fingerprinting algorithms with high multiplicity must be employed in order to resist mutations of old campaigns.

A cryptographic hashing algorithm, for example, would not be an appropriate fingerprint because of its sensitivity to mutations, but would fare well when it comes to generating fingerprints that would unintentionally also apply to legitimate messages.

If a large number of messages that do not have any mutations are received by our agents, the multiplicity numbers will look artificially low. Therefore, we use an additional metric, known as *unbiased multiplicity*, for evaluating fingerprinting algorithms during the design stage. This metric quantifies how close an experimental fingerprinting algorithm comes to generating only a single metric per mutation class.

$$multiplicity = \frac{|f_c(S)|}{|f_e(S)|}$$

A fingerprint with high multiplicity is capable of covering multiple mutations of the same spam from a single campaign. From the standpoint of the community, a high-multiplicity fingerprint means that a single spam campaign consisting of multiple mutations will be eliminated far earlier than if multiple fingerprints are required.

It is possible to generate a fingerprint with extremely high multiplicity, where volumes of messages are covered by the same fingerprint. The danger, however, is that a high-multiplicity fingerprint would also cover messages that are not spam, or cause collisions with messages contained in the *legitimate* class. Since TES independently protects against false positives by contesting fingerprints that cover both spam and legit messages, a high cross-collision rate renders colliding fingerprints ineffective in stopping spam.

$$unbiased = 1 - \frac{|f_e(S)| - |f_o(S)|}{|f_c(S)| - |f_o(S)|}$$

During metric design, we quantified a fingerprint algorithm's risk for a high contention rate by examining the *cross-mutation collision*, or *cmc* rate, or the rate at which a single fingerprint covers multiple mutation classes. Note that the inverse mapping of the fingerprint function is a subset of the message set, that is,

$$f^{-1}(f(M)) = S \cup S'$$

The cross-mutation collision rate is really a proxy metric for the *cross-class collision*, or *ccc* rate, or the rate at which a legitimate message and a spam message maps to the same fingerprint.

In summary, a perfect fingerprinting algorithm will generate zero cross-class collisions and will have an unbiased multiplicity of 1. Currently we employ eight different fingerprinting algorithms, each of which has a method of operation that is mutually orthogonal. The generation of additional fingerprinting algorithms is left as an exercise to the reader.

$$cmc = \frac{\left| S \cap \left[\bigcup_{1 \le x \le k} \bigcup_{x < y \le k} f_c(M_x) \cap f_c(M_y) \right]^{-1} \right|}{|S|}$$

For those readers who are familiar with Vipul’s Razor and Cloudmark systems, we would like to point out that the open-source Vipul’s Razor agents implement two of the six fingerprinting algorithms in Global Threat Network service. The independence of fingerprinting schemes makes it

$$ccc = \frac{\left| S' \cap \left[f_c(S') \cap f_c(S) \right]^{-1} \right|}{|S'|}$$

possible to deploy versions of the service agents with differing QoS and performance characteristics as required by environmental and business factors.

Overall Fitness

From a user’s standpoint, the most important metrics are the *accuracy* and the *false positive* rate, denoted by *a* and *fp* in the equations below. We define accuracy as the percentage of spam messages sent to a user’s mailbox that are correctly classified as spam without user intervention. Also, we define the false positive rate as the number of non-spam messages that have to be unblocked by user feedback divided by the number of messages that are checked for their spam disposition.

In practice, the Global Threat Network allows precise quantification of accuracy and training costs based on the cumulative number of blocks, unblocks, and checks. In fact, the GTN is one of the few anti-spam systems whose production performance can be actively measured by its developers in real time.

Sender Reputation

Several initiatives are being pursued that attempt to compute reputation of senders rather than of email. Both Sender Policy Framework (SPF)[4] and Domain Keys Identified Mail (DKIM)[2] attempt

$$a = \frac{filteredspam}{filteredspam + unfilteredspam}$$

$$fp = \frac{unblocked}{checked}$$

to identify a sender by the set of email servers they use to send out mail. SPF is the more widely deployed of the two, but DKIM is gaining ground. The basic idea behind the SPF scheme is to allow senders to publish a list of servers they use for sending mail through a DNS record. For example, examplesender.com can publish that they send mail from mx1.examplesender.com or mx2.examplesender.com. Before accepting mail, the recipient mail server can then ensure that a sender claiming to be examplesender.com is actually coming from one of the mail servers in example sender.com's DNS records. DKIM signs all outgoing messages with an asymmetric key whose public counterpart is published through the sender's DNS. SPF and DKIM essentially make it very hard to forge the identity of the sender, making spam filtering based upon sender information more feasible.

Once the sender's identity is established with SPF or DKIM, reputation systems like TES can be employed to measure "the goodness of senders" over time. There are quite a few sender reputation projects underway[3] that aim to track and compute sender reputation and use it to filter mail from bad senders. Sender reputation would also enable more robust versions of the "address whitelist" classifier described earlier.

How accurate would such classifiers be? While we believe sender authentication is a useful and healthy augmentation to email, our perspective on the usefulness of sender authentication in the context of anti-spam differs from the prevalent optimism seen in the industry today. The problem with sender authentication schemes is that they do not identify individual senders. They currently associate identities only with a collection of senders behind a host. To be more exact, sender authentication schemes identify the software and network infrastructure used by a sender to send out email. There are two problems with this. First, a sender's reputation is affected by the behavior of all senders with whom the sender shares network resources. Second, the sender's reputation could be affected by malicious code that hides on the sender network to send out spam or malicious code via email. The first is a problem of *granularity* and the second is a problem of *impersonation*.
We contend that host-level sender authentication and reputation will be good at identifying immaculately good senders, or sender collections, such as small organizations and phishing-afflicted institutions that can successfully enforce a conscientious sender policy and safeguard their networks from zombies. Sender authentication will also be effective at identifying the absolutely bad senders and networks created solely to send spam. However, networks with a large number of users, and networks where security can be breached by spam-spewing zombies, will end up with sullied reputations.

If we were to design a fingerprinting algorithm that simply used the authenticated sender host as the fingerprint, the algorithm would have a high multiplicity and also a high cross-collision rate. As we described above, a high cross-collision rate results in the classifier's inability to use the contested fingerprints—sender hosts in this case—to make filtration decisions. Thus, a sender host-based classifier would have to rely on out-of-band methods to classify a large amount of email.

DKIM does allow for weak authentication of individual senders. The reliability of individual sender authentication is a function of a domain's ability to authenticate senders within the domain. Although internal authentication of senders through methods such as SMTP-AUTH is not widely deployed today, it's a very tractable solution.

Pushing authentication, and eventually reputation, to individual senders will alleviate the problem of granularity and will make for better classifiers. It might also be possible to solve the problem of impersonation by modeling sender patterns that differentiate zombie activity from that of the legitimate user.

The collaborative filtering system we describe does not need to establish trust for a global pool of mail senders. The system we describe needs to establish a weaker form of trust that states that email recipients will correctly or incorrectly classify spam regardless of who they are, among a pool of users whose number is relatively small compared to the number of users of email worldwide.

CONCLUSION

We described the architecture and operation of the Cloudmark Global Threat Network service and illustrated the emergent properties of the reputation system underlying the classifier. We also presented a framework for evaluating the efficacy of spam fingerprinting algorithms.

Finally, we also contrasted the GTN approach with other popular methods for classifying spam. The actual architecture and algorithms currently used in the Global Threat Network service are quite complex. The descriptions above have been simplified to highlight the central themes. We hope that we have conveyed the importance of reputation-based methods in the fight against spam.

REFERENCES

1. Availability heuristic. http://en.wikipedia.org/wiki/Availability_heuristic. Accessed on September 28, 2005.
2. DomainKeys Identified Mail. http://mipassoc.org/dkim/. Accessed on September 28, 2005.
3. The rise of reputations in the fight against spam. http://linuxworld.sys-con.com/read/48128.htm. Accessed on September 28, 2005.
4. Sender Policy Framework. http://spf.pobox.com/. Accessed on September 28, 2005.
5. Vipul's Razor. http://razor.sf.net/. Accessed on September 28, 2005.
6. A Plan for Spam. http://www.paulgraham.com/spam.html, August 2002. Accessed on September 28, 2005.
7. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM, February 1981.
8. R. Dingledine, N. Mathewson, and P. Syverson. Tor: The second-generation onion router. In Proceedings of the 13th Usenix Security Symposium, 2004.

For more information
visit us at www.cloudmark.com

Authors:
Vipul Ved Prakash
Adam J. O'Donnell

CLOUDMARK®

Americas Headquarters
Cloudmark, Inc.
San Francisco, USA

Asia Pacific Headquarters
Cloudmark, Inc.
Singapore

Europe Headquarters
Cloudmark Europe Ltd.
London, UK