ROBERT HASKINS

ISPadmin

INTERVIEW WITH VIPUL VED PRAKASH

Robert Haskins has been a UNIX system administrator since graduating from the University of Maine with a B.A. in computer science. Robert is employed by Renesys Corporation, a leader in real-time Internet connectivity monitoring and reporting. He is lead author of *Slamming Spam: A Guide for System Administrators* (Addison-Wesley, 2005).

- *rhaskins@usenix.org*

I'd like to thank Vipul Ved Prakash for taking time out of his busy schedule to answer my questions. I interviewed Vipul via email during January 2005.

RH: You are well known within the anti-spam community as the author of Vipul's Razor, as well as a founder of Cloudmark. For those readers who don't know, and to provide context to get started, I'd like you to give me some background regarding your past accomplishments. How did you get to where you are today?

VVP: During the mid-nineties, I closely followed the Cypherpunks list and was fascinated by the funkier of the cryptographic protocols that cypherpunks wrote, cited, and talked about. The notions of trust and reputation were particularly interesting to me, and I thought a lot about the mechanics of reputation networks and the problems that could be solved with them. Following a brief stint on USENET, I was suddenly inundated with spam. There wasn't much in terms of anti-spam on the Net back then, and anti-spam seemed like the perfect test application for a reputation network. That was the genesis of Vipul's Razor. At its heart, Razor is a system for assigning reputations to people who submit spam reports and, in turn, to the reports themselves.

Perl is my favorite programming language, and I've written a bunch of Perl modules that are published through CPAN. By way of merging my interests in Perl and cryptography, I've written implementations of some of the more popular cryptographic algorithms as Perl modules. I also wrote an implementation of RSA in 512 characters that was formatted to look like a dolphin. Thinkgeek carried it on a t-shirt for a while.

Another interesting project I worked on was CODD—which measures contributions to open source projects by doing source analysis to find authorship attributions. CODD is now developed at the University of Madrid, and the good folks there are doing some remarkable things with it.

RH: Can you describe how the Razor system works, and how it is similar to and different from other related anti-spam systems, namely the Distributed Checksum Clearinghouse?

VVP: Vipul's Razor is a network for sharing information about spam in propagation. The system builds a continually updating model of known spam messages, which is used by mail delivery applications to filter out subsequent deliveries of known spam.

A set of signature (fingerprinting) schemes is used to reduce spam messages to a set of signatures. Reporting and checking of messages are done through these one-way signatures. Reporters have an identity in the Razor system and have to authenticate themselves prior to nominating a message as spam. The back end is composed of a set of nomination servers that accept reports and a set of catalogue servers that serve the database of signatures for known spam messages.

Reports gather on the nomination side of the back end, where they are evaluated by TeS (the Truth Evaluation System), Razor's trust system. TeS examines the reports as they come in to determine the degree of agreement on whether a signature is considered "spammy" by the community. TeS also identifies the users who have reported spam in the past and whose historical decisions were mostly "unchallenged" by the community. These users accrue trust points and are eventually considered to be trusted users. It is the reports of trusted users that determine the "spaminess" of a signature, which is reflected in its confidence value. Signatures that cross the confidence threshold to become spam are replicated over to the catalogue side of the back end, from where they are "fed back" to the community.

Vipul's Razor is a collaborative classifier driven by content samples reported by its users. It predicts whether a message is spam or legit. DCC, on the other hand, determines the "bulkiness" of a particular mailing. DCC works by collating signature sightings from participating MTAs to see how many copies of a particular mailing were sent out. The task of classifying bulk into spam and desired bulk is left to out-of-bound methods in DCC.

RH: You mention trust points, scores, and confidence levels. I'd like some idea of the make-up and threshold values for some of the more common trust/scoring/confidence mechanisms in the Razor system. For example, what does a message score consist of? What score causes a message to be considered spam? What do trust and confidence scores consist of, and how are they generated?

VVP: All signatures have a confidence level associated with them; it ranges from -100 (legitimate) to 100 (spam). The confidence of a signature is a function of the number of trusted reports as well as the level of trust of the reporters who submit the reports. Razor Agents also come with a razor-revoke tool, which is used to make negative assertions—"this message is not spam." Revokes factor into the confidence as well, by pushing the confidence toward -100. TeS attempts to determine whether there is statistical consensus among trusted reporters on a particular signature. When statistical consensus is discovered, TeS selects, through a rather complex set of heuristics, a few reporters to award for participating in the reporting process. Those that disagree with the consensus are selected for penalty. An award is a positive trust point added to a reporter's trust counter, and a penalty is one or more trust points subtracted from the trust counter.

TeS is an inductive trust system. It starts out with a few trusted reporters (myself, some of the early Razor users, and folks at Cloudmark) and assigns trust to new users who tend to agree with the existing trusted users. Once the new users have accrued enough trust points to be considered trusted, they participate in selection of still newer trusted reporters. As you can imagine, over time the system becomes progressively harder to game, as the spammer has to subvert an increasing number of trusted users in order to change the disposition of their spam to legitimate. In fact, the spammer must first become trusted by agreeing with trusted reporters, i.e., by reporting messages as correctly as spam.

RH: I want to talk a little bit about the relationship between Cloudmark and Razor, but before I do that I'd like readers to have some background on Cloudmark. Can you describe how Cloudmark came to be, and what the Cloudmark products and services are?

VVP: Cloudmark was born in September 2001. I was working on the design of Razor 2, as a sudden boost in usage had put the prototypical first version at its limits. I bumped into Jordan Ritter, my co-founder at Cloudmark, on IRC. Heo was very interested in Razor and was also working on text classification algorithms for anti-spam, and he proposed that we start a company. After many whiteboard sessions to merge our visions, we founded Cloudmark with the goal of building widely deployable and highly accurate spam filters.

Today, Cloudmark provides anti-spam products to more than a million consumers and several thousand corporations. We build high-performance anti-spam engines for ISPs and large enterprises. These engines are licensed by leading mail infrastructure companies like Sendmail and Openwave. Cloudmark's SafetyBar and CEE products are Windows siblings of Razor and integrate into Outlook, Outlook Express, and Microsoft Exchange. Partner companies have integrated the SafetyBar/Razor technology in other mail products. We recently launched Cloudmark Immunity, which incorporates a proximity-based classifier for real-time online learning based on feedback from users. Immunity is designed for use in large enterprises.

RH: Can you go into some detail about exactly what anti-spam checks are in the various Cloudmark products? Does it use Razor data or a separate data set? Also, please talk more about Cloudmark Immunity and the "proximity-based classifier," as I am not sure what that is.

VVP: Cloudmark SafetyBar and Cloudmark Exchange Edition products plug into the same network as Razor and use the same data. The difference is that they support more signature schemes and perform with better accuracy and precision. Razor Agents provide accuracy to the order of 90%, whereas SafetyBar and CEE record 98% or higher with very few false positives. In fact, in the last four tests conducted by *PC Magazine*, SafetyBar was the only product to record zero false positives. Cloudmark's enterprise and ISP products are based on homegrown classifier technologies, but are trained from the data set created by Razor. Over a million people report spam to Razor, and the reports are verified by TeS, which results in a high-quality, comprehensive database with which to train our classifiers.

Immunity was designed to solve some of the problems inherent in statistical classification systems such as naive Bayesian, which, while providing compact hypotheses, were not designed to work in adversarial and rapidly evolving environments like anti-spam. Immunity maps its training set in a hyper-dimensional space (such that a spam is a point in this space) and classifies incoming documents based on the disposition of points that fall in its neighborhood. Unlike Bayesian classifiers, Immunity's classifier can be trained on one-off samples, can modify its model based on individual pieces of feedback (since accepting feedback is a simple matter of making an entry into the hyperspace), and can provide filtration that is specific to individual users. It's also more robust in that it is not vulnerable to common poisoning attacks.

RH: Some people have criticized you/Cloudmark for using the Razor spam signature data for commercial purposes (i.e., in Cloudmark's products). Is this a valid complaint? How do you respond to this criticism? What have you done (if anything) to help mitigate the issue?

VVP: We pour the data submitted by Razor users and by SafetyBar users into the same funnel and it all goes through the same trust system. Both communities benefit from the collective reports, so they actually do better than they would have without each other.

When we founded Cloudmark, there was concern about Razor Agents disappearing or moving entirely into the commercial realm. Such concerns have now been alleviated, as we have done a major release of Razor Agents almost every quarter since and continue to add hundreds of new Razor users every day.

Merging open source and commercial software worlds is hard, especially in the face of extreme ideological commitments that people make about developing software. It has been an interesting experience for me in that regard, and I think we've done a decent job of it.

RH: I'd like to get your ideas specifically about the area of reputations in combating spam. [For background on anti-spam reputations, please see my article in the February 2005 issue of *LinuxWorld* titled "The Rise of Reputations in the Fight Against Spam."] Can you give readers some idea of how reputations have changed the fight against spam? How have spammers changed their tactics in response to reputation-based systems? I'd like to hear your views not only as they apply to Razor/Cloudmark, but the other reputation services as well—Verisign, Ironport, Kelkea, etc.

VVP: I believe reputation systems are central to the fight against spam; I like to think of a reputation system as a predictive model that uses a combination of historical performance and opinions of trusted sources to make a good/bad prediction about an unknown object. The objects evaluated in the context of email are usually IP addresses, sender domains, individual or institutional senders, email content, and newsletters/mailings.

Reputations require identities. Since email, as a protocol, is mostly identity-free, identification mechanisms have to be overlaid. SPF, Sender-ID, and DomainKeys, three schemes that are garnering a lot of support, attach identity to sender hosts and domains via reverse DNS lookups. A few reputation systems are cropping up to assign reputations to domains and hosts as identified by these schemes. Cloudmark Rating is one of them. As I mentioned earlier, we also use reputations in the context of Razor. From that perspective, Razor's signature schemes can be thought of as a way to assign identity to spam content.

In general, reputation systems are good news for good guys. Once you can identify yourself, recipients can ensure delivery. Legitimate communications can pass through anti-spam mechanisms without evaluation if identity and reputation can be established.

Reputations are changing the playing field, i.e., the anti-spam problem is moving to a slightly different place. There are two kinds of attacks spammers can mount against reputation-based systems. The first is to avoid identification. As long as identification systems are not pervasive, spammers will try to blend in with "unknown" mail. To fight SPF/DomainKeys, spammers will register a lot of domains and cycle through them as soon as a domain is considered spammy. In response, reputation systems need to be quick, allowing only a little spam to get through per domain. If the system is fast enough to make the cost of domain registration prohibitive, then we have a good defense against this attack.

The second, more ominous, attack is to hijack the mail infrastructure of senders with good reputations. Spammers are doing this today through zombie networks. The current zombies don't exploit SPF/DomainKeys, but it is the obvious next step to infect machines inside a good network, look up advertised MX servers, and inject spam through them. This is a hard attack to defend against, especially with identity-based methods, which assume perfect internal security. To provide meaningful filtration, reputation systems would have to learn the patterns of abuse associated with hijacking attacks. Personally, I believe the solution to this attack would come from content-based methods, something along the lines of raising a "content-exception" so that reputation systems don't persecute abused domains, while still disabling the propagation of spam.

I am not very familiar with the internals of other reputation systems. I believe Verisign and Habeas are providing accreditation (as opposed to reputation) services, where they do out-of-band research and accreditation of good senders.

RH: Each anti-spam vendor seems to have its own idea of what a reputation is. Obviously, this poses a major stumbling block for both anti-spam vendors and spam-fighting system administrators. Anti-spam reputations are often compared to credit bureaus, except that the risk is the "spamminess" of the sender rather than someone not paying their bill. Are we going to have a common definition of a reputation the way credit bureaus have? Failing that, will there be a standard API so that there can be one programming interface to multiple "reputation bureaus"?

VVP: There is considerable value in a standardized reputation API, but we need to be careful about representing the semantics intended by individual reputation systems. Reputations are measurements of behavior against a defined policy. In a space where different vendors are measuring very different aspects of email, it becomes necessary to understand exactly what they are measuring and how these measurements map to a filtration policy.

Many ISPs have their own reputation databases based purely on the quantity of email received from senders,\; others measure conformance to their AUP; yet others assign reputation by co-relating senders with the nature of their emails' content. Cloudmark's reputation system is based on reports from the trusted members of its community of users, whereas Habeas is based on real-world identity and mail-stream permission levels. Almost every reputation system out there is different.

One option for standardization is to allow the user to ask contextual questions of a reputation service so it can offer a binary decision (accept/drop) to the user. The other is to find a way to encapsulate the diversity of semantics so that the burden of finding the appropriate policy is shifted to the user of the reputation system.

RH: You mentioned SPF and DomainKeys; recently in the open source arena we have seen lots of attention on the area of reputations as well as domain authentication (GOSSiP and Sender Policy Framework/Aspen, among others). What impact will open source anti-spam reputation applications such as GOSSiP and SPF have on fighting spam?

VVP: GOSSiP combined with SPF and DomainKeys has the potential to enable private, quasi-disjoint, peer-reviewed email networks. It will be very exciting, especially when GOSSiP or a similar distributed reputation scheme achieves critical mass. The cost of joining the email network would increase selectively for spammers, but good guys would be able to get "introductions" to the network. I also think augmenting protocols is best done as open-source projects. Since adoption is the biggest hurdle these schemes face, publication under liberal licenses helps immensely.