;login:

THE MAGAZINE OF USENIX & SAGE

June 2003 • volume 28 • number 3

inside:

CONFERENCE REPORTS

2003 MIT Spam Conference

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

conference reports

This issue's reports focus on the 2003 Spam Conference, USITS '03, and FAST '03

Our thanks to the Summarizers:

for Spam '03:
Chris Devers
for USITS '03:
Ajay Gulati
Xuxian Jiang
for FAST '03:
Scott Banachowski
Nate Edel
Preethy Vaidyanathan

2003 MIT Spam Conference

Cambridge, MA

January 17, 2003

http://spamconference.org/

Summarized by Chris Devers

[Editor's Note: The summaries by Chris Devers were condensed for publication in ;login:*.]*

Sparse Binary Polynomial Hashing and the CRM114 Discriminator

William S. Yerazunis, Mitsubishi Electric Research Laboratories

Yerazunis wrote the CRM114 filtering mini-language and then wrote MailFilter in CRM114 as an implementation that can be used with other spam-fighting programs. The basic idea is to decompose a message into a set of "features" composed of various runs of single words, consecutive words, words appearing within a certain distance of one another, etc.

He claimed that with this software he could get better than 99.9% accuracy in nailing spam, and a similar percentage in avoiding "ham" (the term everyone was using for false positives – legit mail that was falsely identified as spam). One of Yerazunis' observations is that the best way to defeat the spam problem is to disrupt the economics: if a 99.9% or better filter rate were to become the norm, then the cost of delivering spam could be pushed higher than the cost of traditional mail and the problem would naturally go away without requiring legislation.

The Spammer's Compendium

John Graham-Cumming, POPFile

Most of this very entertaining talk was about the ingenious tricks that spammers resort to to obfuscate spam against filters, including, most diabolically, one example that placed each column of monospace text in the message into an HTML column, so that the average HTML-capable mail client would render the message properly, but it would be absolute gibberish to most mail filters. The ultimate lesson was that any good filter has to focus not on "ASCII space" (the literal bytes as transmitted) but the "eye space" (the rendered text as seen by the user), which, by extension, may mean that any full-scale spam parser/filter could also have to include a full-scale HTML and JavaScript engine.

As for Graham-Cumming's software, it's a Perl application, available for all platforms (Windows, Mac, and, of course, Linux) that enables users to filter POP3 mail. Interesting stuff if you're a POP user: *http://popfile.sourceforge.net*.

ShopIP

John T. Draper

Most of Draper's work seemed to be focused on profiling spammers, as opposed to profiling spam itself, by throwing out a series of honeypot addresses and using data collected to hunt down spammers.
http://spambayes.sourceforge.net

Spam Research: Establishing a Foundation and Moving Forward

Paul Judge, CipherTrust

Judge's big argument, which no one really disagrees with, is that spam has become not just a nuisance but an actual information security issue. To that end, he is advocating much more collaborative effort to address the problem than we have seen to date: conferences like this, mailing list discussions, better tools, and public data repositories of known spam (and ham). To that last point, one of his observations (which others made as well) was that there are no universally agreed-on standards for what qualifies as spam, so repositories for spam will not be accurate for all users (e.g., spam for your programmers will be the bread-and-butter of your marketing department). Plus, there are obvious privacy issues in publishing your spam and ham

for public scrutiny. And to add another wrinkle, one danger of public spam/ham databases is that spammers can poison them with false data, screwing things up for everyone. That said, he encouraged users to help out with building *http://spamarchive.org*.

better bayesian filtering

Paul Graham, Arc Project

Graham is the man who organized the conference and kicked off everything this week with his landmark paper from last fall, "A Plan for Spam." Graham's spam-filtering technique famously makes use of Bayesian statistics, a technique popular with nearly all of the speakers. The nice thing about a statistical approach, as opposed to heuristics, simple phrase matching, RBLs, etc., is that Bayesian statistics can be very robust and accurate; the down sides are that they have to be trained against a sufficiently large "corpus" of spam (most techniques have this property, though) and they have to be continually retrained over time (again, this is common). Graham was too modest to produce numbers, but subjectively his results seemed to be even better than what Yerazunis gets with MailFilter by an order of magnitude or more.

Like other speakers, he predicted that spammers are going to make their messages appear more and more like "normal" mail, so we're always going to have to be persistent about this; as one example, he showed us an email he received IN ALL CAPS from a non-English-speaker asking for programming help, and although it was legit, the filters insisted otherwise. "That message is the one that keeps me up at night."

Everyone interested in the spam issue should go read Graham's paper immediately.

Internet Level Spam Detection and SpamAssassin 2.50

Matt Sergeant, MessageLabs

SpamAssassin is a well-known Perl application for heuristically profiling messages as spam, adding headers to the message, saying, for example, "I am 72% sure this is spam because it has X Y Z," and passing off the message to procmail, or whatever, to be handled accordingly. SA can handle a message throughput great enough that it can be deployed at the network level (whereas some of the other applications, which might have somewhat better hit rates, are still too inefficient at this point). Deployed this way, the differences in effectiveness for single vs. multiple users becomes very apparent, as 99% effective rates fall down into the 95–80% range. This happens because, again, different users define different things as spam, so mapping one fingerprint to all users can never work quite right.

For an example of a tool that your company can deploy right now and get fast, decent results, SA looks like a good choice; but for the long run it looks like a Bayesian technique is going to get better performance, and SA is adding a statistical component to its toolkit. Good talk.

Anti-Spam Techniques at Python.org

Barry Warsaw, Pythonlabs at Zope Corporation

This was another example of the "monocultures are dangerous" philosophy, as Warsaw explained how he is helping to use a variety of anti-spam techniques – from clever Exim MTA configuration to good use of SpamAssassin and procmail to fine-tuning of the Mailman mailing list engine – to work together to manage the spam problem for all things Python (Python.org, Zope, many mailing lists, a few employees, etc.).

He pointed out that some very simple filters can be surprisingly effective: run a sanity check on the message's date, look for obviously forged headers, make sure the recipients are legit, scan for missing Message-ID headers, etc. In response to the person who originally posted the article, yes, he did mention blocking outgoing SMTP as an effective element of a many-tiered spam management approach.

Among other tricks for getting the different filtering tiers to play nice together, they make heavy use of the X-Warning header so that if an alarm goes off in one tier of their mail architecture, other components can respond appropriately. Cited projects included ElSpy and SpamBayes.

Spam: Threat or Menace? An ISP's View

Barry Shein, The World

His core argument is that spam is "the rise of organized crime on the Internet," that filters are nice but that the mail architecture itself is fundamentally flawed, and that ISPs like his – in 1989, The World was the world's first dialup ISP – are being killed by the problem.

Shein was very annoyed that all these talented people are having to clean up a mess like this when they should be out working on more interesting stuff. His big hope seemed to be that legislation will someday come to the rescue, but he sounded very pessimistic. (Others in the room seemed to feel that this was a very interesting machine-learning problem and weren't really fazed by his pessimism – but, then, most of the people in the room don't run ISPs.)

He also suggested that we need to find a way to make spammers pay for the bandwidth they are consuming (rather than having users and ISPs shoulder the burden) but didn't seem to know how we might go about implementing this. At all.

Smartlook: An E-Mail Classifier Assistant for Outlook

Jean-David Ruvini, e-lab Bouygues SA

This was an interesting product. Ruvini's company is developing an extension to

Outlook 2000 and XP that will watch the way users categorize messages into folders, come up with a profile for what kinds of messages end up in which folders, and then try to offer similar categorization on an automatic basis. Think of it as procmail for Outlook, without having to mess with (or even be aware of!) all the nasty recipes.

Obviously, if you have a spam folder, then spam will be one of the categories it looks for, but, more broadly, it will try to categorize all your mail as you would ordinarily categorize it. This makes SmartLook a broader tool than "just" a spam manager.

SmartLook is another statistical filter, though it uses non-Bayesian algorithms to get results. e-labs' tests suggest that the product is able to properly categorize messages about 96% of the time, with no false positives, and (for their tests, mind you) that it performed better than Bayesian filters over three months of usage.

One nice property of this tool was that it works well with different (human) languages – some strategies fall apart and/or need retraining when you switch from English to some other language. For certain markets (e-lab is in France) this is a crucial feature, and having a tool that works with one of the biggest mail clients out there (most people don't use Mutt or Pine, sadly enough) can be very valuable. Very clever – watch for the inevitable embrace and extend three years from now.

Lessons from Bogofilter

Eric Raymond, Open Source Initiative

He didn't say anything about guns, but he did try to correct one of the other speakers for misusing the term "hacker."

Like Graham, ESR is a Lisp fan, but he knows that the vast majority of people aren't, and he also knows that the vast majority of people need to be using something like Graham's spam software. So on a lark, he came up with a clean version in C, named it bogofilter, and put it on SourceForge, where a community sprang up to, well, embrace and extend it.

As good as Graham's Bayesian algorithm is, ESR felt – as did many of the other speakers – that the nature of your spam/ham corpus is much more significant than the relative difference among any handful of reasonably good algorithms. (Back to the often-repeated point about how corpus effectiveness falls apart when used for a group of users, as opposed to individuals.)

To that end, he strongly felt that the best way to deal with the spam problem is to get good tools into the hands of as many people as possible, and to make them as easy to use as possible.

As an example, one of the first things he did was to patch the Mutt mail agent so that it had two delete keys: one for general deletion and one to "get rid of this because it's spam." That second key, and interface touches like it, seem like the way to get average people to start using filters on a regular basis.

Spam Filtering: From the Lab to the Real World

Joshua Goodman, Microsoft Research

Unlike ESR, Goodman felt that algorithm selection does make a big difference, but, this being Microsoft, he refused to disclose what algorithms his team is working with – except to say that, when delivered, they will be more accessible for average users than SpamAssassin, procmail recipes, or Mutt.

Microsoft has been working on the spam problem since 1997, but because of how big they are, they've had unique problems in bringing solutions to market. As a case in point, they tried to introduce spam filters in a 1999 Outlook Express release, but were immediately sued by email greeting card company Blue Mountain because their messages were being inaccurately categorized as spam. With that in mind, they have been very reluctant to bring new anti-spam software out since then, because they would like to see legislation protecting "good faith spam prevention efforts."

As a very large player, Microsoft faced certain difficulties in developing useful filters: It may make sense for you as an individual to filter all mail from Korea, but this doesn't work so well if you are trying to attract customers from Korea. This has forced them to put a lot of work into thoroughly testing different strategies before offering them to the public.

In spite of what millions of Webmail users might have expected, Hotmail and MSN are currently being filtered by Brightmail's service, and plans are underway to re-introduce spam-management features to client-side software again. (Just imagine how bad it would be if they weren't paying someone to filter for them!)

An interesting barrier his group has had to grapple with was what he called the "Chinese menu" or "madlibs" spam generation strategy: that it's easy to come up with a template for spam – "[a very special offer] [to make your penis bigger] [and please your special lady friend all night!" vs. "[an exclusive deal] [for genital enlargement] [that will boost your sex life!]" etc. – and have a small handful of options for each "bucket" multiplying into a huge variety of individual messages that are easy for a human to group together but almost impossible for software to identify.

Integrating Heuristics with n-grams using Bayes and LMMSE

Michael Salib, extremely funny MIT student

Unlike nearly all other filter writers of the day, Salib's approach was heuristic: find a handful of reasonable spam discriminators, throw them all against his mail, and see how much he can identify that way. "It's sketchy, but this is a class

project. I don't have to be realistic. These results may be completely wrong."

Much to his surprise, he's trapping a lot of spam. He pulls in a little bit of RBL data ("the first two or three links from Google, whatever"), looks for some patterns, and then churns it through LMMSE, an electrical engineering technique that, as far as he can tell, doesn't seem to be known in other fields. Basically, this involves running the messages through a series of scary-but-fast-to-calculate linear equations. It turns out that he can process this much faster than a Bayesian filter, to the point that customizing his approach for each user in a network would actually be feasible.

For a small spam corpus, he got results better than SpamAssassin did, though for a large corpus his results were worse; he couldn't really account for why this would be the case, or predict how things would scale as the corpus continued to grow.

Forty Years of Machine Learning for Text Classification

David D. Lewis, Independent Consultant

The core of Lewis's argument, as ESR said earlier in the day, is that for any machine-learning technique, the quality of the learning corpus is much more important than the algorithm used. Bayes is one such algorithm, but there are many other good ones in the literature. Lewis pointed out that all of this has been publicly discussed since the first machine-learning paper was published in 1961.

Observations: "Lots of task[-non-specific] stuff works badly, but task-specific stuff helps a lot." It is important to use different bodies of text for training and for general use, so that you don't train your machine to focus too much on certain types of input (this is a point that Microsoft's Goodman made as well).

As Graham did, Davis emphasized that spam is going to slowly start looking more like natural text, and we're going to have to deal with this as time goes on. *http://www.daviddlewis.com/events/*

How Lawsuits Against Spammers Can Aid Spam-Filtering Technology: A Spam Litigator's View from the Front Lines

Jon Praed, Internet Law Group

To a burst of tremendous applause, this talk began with the sentence, "My name is Jon Praed, and I sue spammers."

He brought a legal take on the "not everything is spam to everybody" angle, emphasizing that we need a precise definition of what qualifies as Unsolicited Commercial Email (UCE). In particular, it has been difficult trying to pin down whether the mail was really unsolicited, as this is where the spammers have the most wiggle room. However, if you can track down the spammer, they have, to date, rarely been able to verify that the user asked for mail, and so Praed has been able to successfully prosecute several spammers using this angle. But he doesn't expect this to work forever.

According to Praed, "Laws against spam exist in every state, and more are pending," but he doubts that a legal solution will ever be completely effective as long as spam is lucrative. By analogy, he pointed out that people still rob banks, and that has never been legal.

Praed informed the audience that there are several ways to get back at spammers, including injunctions, bankruptcy, and contempt, and all of these can be very effective. He pointed out that, to be blunt, a lot of these people are desperate low-lifes, and spam has been their biggest success in life. After these legal responses, their lives all get much worse.

It hadn't occurred to me to see spammers as pitiful before, but I can now. Most importantly, Praed stressed that these legal remedies can be very effective, and he strongly warned against taking vigilante action. This is almost always worse than the spam itself, and it only serves to get you in even deeper trouble than the spammer.

Most spam comes from offshore spam houses, abuse of free mail accounts (Hotmail and Yahoo, free signups at ISPs, etc.) and bulk software (which may apparently soon become illegal in certain areas, provided that a law can be found to ban spam software while allowing tools like Mailman and Majordomo). Interestingly, he questioned the idea that IP spoofing is a big problem and claimed that in every case he has dealt with he has been able to track down the messages to a legit source sooner or later.

Suggestion: If you get a spam citing a trademarked product (e.g., Viagra), forward it to the trademark holder and they will almost always follow up on it. Suggestion: Be fast in trying to track down spammers, as some of them have gotten in the habit of leaving sites up long enough for mail recipients to visit, but taking them down before investigators get a chance to take a look. Legal observation: Spam is almost always fraud, and can be prosecuted accordingly.

Praed wrapped up his talk by citing the encouraging precedent that the famous Verizon Online v. Ralsky case set: (1) that the court is interested in where the harm occurs, not where the person doing harm was when causing it, and (2) it is assumed that you have to be familiar with a remote ISP's acceptable usage policies, and ignorance is no defense. (Just as you can't say, "I didn't know it was illegal to shoot someone," Ralsky couldn't say that he didn't know Verizon prohibits spam. He had to have known that the AUP wouldn't allow what he was doing, so he deliberately didn't read it.)

That precedent makes the idea of future prosecution of spammers much more encouraging. While, again, legal solutions may never eliminate the spam problem, a precedent like this can be an

important supplement to filtering efforts.

Desperately Seeking: An Anti-Spam Consortium

David Berlind, ZDNet executive editor

His talk was primarily about how he receives a huge quantity of email from ZDNet readers, and he can't afford to use any spam-filtering solution strategy that would allow *any* false positives. As one of the speakers said, getting a 0% false positive rate is easy: just classify nothing as spam. Getting a 100% hit rate is also easy: just classify everything as spam. Any solution besides those two is always going to have some degree of error either way, and determining how much of what kind of error you want to accept is up to you.

Most users will tolerate a moderate false negative rate (some spam gets through) if it means that the false positive rate (legit mail is deleted) is very low. In Berlind's case, the false positive rate has to be vanishingly small, because reading all customer mail is, to him, a critical sign of respect for his readers.

Further, his business is also a legitimate mass emailer, sending out millions of free newsletters to users every day, and if Shein's proposal to bill bulk mailers were to catch on, even a very low rate would quickly put a company like Berlind's in the red. One obvious solution, which wasn't mentioned: start charging a subscription for these mailings, and make them profitable. I don't want to see this happen but if it did, then the economics would tilt back toward making things feasible again.

Though Berlind is appreciative of the anti-spam work that is being done, he is skeptical of how pragmatic most of what is being proposed can really be. He feels we need a massive effort to rework the way mail is handled and, to that end, hopes ZDNet can help promote a cooperative effort between the parties working on this. They don't want to be involved – they are journalists and publishers, not standards developers – but they are eager to get things going and want to cover the story as it progresses.

As Shein said, he feels it's a waste for all these talented people to be working on combating penis enlargement offers, and he hopes that we can find a way to get past this and work on real problems "like world peace."

Fighting Spam in Real Time

Ken Schneider, Brightmail

As mentioned earlier, Brightmail provides an ASP service for real-time filtering of both incoming and outgoing mail. As would perhaps be expected, bigger ISPs and networks attract larger amounts of spam: 50% of mail coming into big ISPs and 40% coming into big companies is now spam. Brightmail offers the Probe Network, a patented system of decoy honeypot addresses that gathers data for analysis at their logistics center, and then distributes spam-filtering rules to their clients where a plug-in for $MTA (using the open source or proprietary MTA of the client's choice) can act on the database.

An interesting property of their system is that they have a mechanism for aging out dormant rules as well as for reactivating retired ones, so that the currently active rule set can be kept as lean and efficient as possible. A big source of difficulty for them is legitimate commercial opt-in lists, because things have gotten more shady and blurry over time and it's now hard to distinguish this mail from much of the spam out there. Whitelists help here, but the problem remains difficult.