White Paper : Using Bayesian filtering against Spam

Anyone who uses email will be familiar with spam – the unwanted messages that clog up inboxes and which have recently changed from being merely being a time-consuming and bandwidth-gobbling nuisance to becoming a real danger. With V-spam, for example, a virus can be downloaded if the recipient of the email clicks on a link. And with phishing the recipient is tricked into giving away confidential information through an email that purports to come from a legitimate organization such as a bank. Recent studies show that more than half of all emails can be deemed as falling into this category and that the problem is continuing to grow.

Before discussing some of the ways in which it can be dealt with, however, it is valuable to consider exactly what is meant by the term ‘spam’. An email about special offers on Viagra, for instance, can’t be called spam if it is received by someone who has requested it, but the same email could well be designated as spam if it arrives in the inbox of someone who doesn’t want it. Perhaps the simplest definition of spam is email that is unsolicited and transmitted in bulk – and both these criteria need to apply. An unsolicited email in itself need not necessarily be spam – it may be an initial sales or job enquiry, for instance. And an email that has been sent in bulk may also be genuine: perhaps a newsletter to which recipients have subscribed.

Although legislative measures to counter spam are being adopted in many countries, and ideally the issue should be tackled by ISPs and telcos who can track where the spam is coming from, in the short-term at least many businesses will need to take steps themselves to minimize the problem. It is important, however, that this is done in a measured way since an over-zealous approach could result in an unacceptably high level of false-positives: genuine emails that are misclassified as spam. A range of anti-spam measures is available – one of the most convenient being internet security appliances which are typically all-in-one boxes that combine anti-spam software with additional features such as anti-virus and URL filtering.

SpamAssassin, for example - the anti-spam software that is incorporated in Equiinet's NetPilot and SentryPilot appliances – uses a variety of techniques to identify spam, including the analysis of individual words or patterns (characters broken up with a repeated punctuation mark, say, or with embedded numerals) as well as the recognition of stylistic features such as font sizes and colours. These are then combined with external tests, such as the spamcop.net blacklist of known spam-senders. Independently, each of these factors only contributes a small increase or decrease to the likelihood that any given message is spam, but used in combination - by allocating scores to various elements – the end result can be very accurate.

However, with the features and their scores fixed and assigned in advance by the SpamAssassin development team, it is possible for spammers to test their messages using SpamAssassin and then adjust the message contents accordingly in order to reduce the score. Each iteration of SpamAssassin's rules, therefore, had to catch up with the spammers' latest tricks; the spammers would then find out about the new rules; and so the arms race continued.

In order to break out of this vicious circle SpamAssassin now includes Bayesian filtering. This is a mathematical approach that, unlike many other anti-spam technologies, adapts over time and takes the changing strategies of spammers into account. It therefore offers moving goal posts which obviously make it far harder for the spammers to score.

Central to Bayesian filtering is the principle that the likelihood of events happening in the future can be inferred by analyzing past events. Spam emails are therefore likely to be made up of similar elements, while valid emails (sometimes referred to as ‘nonspam’ or ‘ham’) will have their own determining characteristics. Bayes classifiers learn as they go, updating both the rules and the scores so that when a new evasion trick comes along, the message may still have enough other bad features that the filter will recognize it as spam and if so, the system will learn the new trick automatically. If it should happen that the message is sufficiently different from any previous spam that it gets through, it is up to the recipient to tell the system that it made a mistake.

The characteristics of both spam and nonspam, however, will vary between organizations. A pharmaceutical company, for instance, may have the word ‘Viagra’ appearing quite frequently in the valid emails it receives, while for other companies the same word often denotes spam. Unless there is an acknowledgement of these different sorts of characteristics in companies’ email traffic, it is very likely that genuine emails will be misinterpreted as spam by spam filters, resulting in false positives.

An effective spam filter will of course minimize the occurrence of false positives, whilst recognising as much spam as possible. Since Bayesian filtering works adaptively this method offers the greatest potential to get nearest to zero false positives and 100 per cent spam detection.

It is essential, though, for the Bayesian filter to be given a period of time to analyze a company's incoming and outbound mail in order to make tailored Bayesian word databases for both spam and valid mail. By comparing outbound mail and known spam, all words and tokens can be given a probability value as to whether an individual word is an indicator that the email is spam or nonspam.

The fact that these databases are tailored to the company is vital. To take our earlier example further – if the word 'Viagra' often appears in the company's outbound mail and incoming legitimate messages, the system will recognize that for this particular company the word does not necessarily increase the likelihood of a message being spam, and the danger of the word generating false positives will be lessened.

Whilst a period of at least two weeks' initial learning is required before the filter should be considered usable, the system will of course keep improving over time as it learns more about the characteristics of the company's legitimate email and spam. In addition to this tailored learning process, a good Bayesian filter will also include a pre-loaded file of known spam which should be updated regularly to take account of recent activities by spammers.

After the learning period, when an email arrives the words that are most likely to appear in a spam message are identified by the Bayesian filter, as well as those which most commonly appear in nonspam messages. These two elements are then balanced against each other and the probability of the whole message being spam is calculated: a much more sophisticated technique than merely using keywords, where an email from a known contact could be mistakenly identified as spam because it contained the word 'free!'.

The positive or negative score contributed by the Bayesian system is then used to score the message according to good and bad factors. If the Bayes system is really sure the message is spam it adds a big score (+5), while if it is certain it is ham it subtracts a similar value. The values in the scoring system have been carefully adjusted by SpamAssassin's developers so that a total score of 0 means the email is almost certainly ham while a score of 10 means it is almost certainly spam.

The system administrator can then optionally set thresholds: a lower threshold, so that email with a score below this is delivered automatically, or an upper one, so that email is discarded automatically. The default setting of Equiinet's appliances allows for both thresholds to be set, so that any message with a score in between may be placed in quarantine, diverted to someone for human inspection, or passed on to the recipient with the spam-score marked in the message header. This approach works much better than using a single threshold, since SpamAssassin gets the opportunity to ask for help on any messages where the validity is unclear.

The relative importance of Bayesian filtering used in combination with other anti-spam techniques, therefore, changes with experience. The Bayes system has an idea of how much it has learned, so as well as saying whether a message is spam or not, it can also say how sure it is. Initially, before the training process, the Bayes system isn't confident and contributes nothing to the spam score. But it learns quickly and soon becomes so confident that it can out-vote the old fixed-rule method. Even when the Bayes classifier is well-trained, however, the fixed-rule score still makes a useful contribution. It is particularly helpful, for instance, when the Bayes system is faced with a really unusual message unlike any it has seen before.

One of the best ways of combining Bayesian filtering with other anti-spam techniques can be seen in the default email filter policy of Equiinet's NetPilot and SentryPilot internet security appliances. This policy, uniquely, allows for all messages to be quarantined where the fixed-rule system – and, later on, the Bayes system – has the least bit of uncertainty and the initial decision as to which of these messages is spam is passed to the administrator. Although this may seem an onerous task, it is only a relatively short one and the results are very worthwhile.

Once the administrator has classified sufficient messages – perhaps two weeks' worth, or a couple of hundred – the training starts to pay off and the number of messages being quarantined will be found to decrease markedly. The administrator may then choose to keep the quarantine system in operation, in case a valid email is accidentally misclassified and in acknowledgement that 100 per cent accuracy, whilst an ideal, is unlikely ever to be achievable. Alternatively, the quarantine system may be switched off and end user recipients be relied on for the reporting of any leaked spam. Each time the system makes a mistake and lets through a spam message, it is important that the recipient feeds that information back to the system using the 'X-Spam-ReClassify:' link in the message header. It will then learn new keywords in the spam it missed and update its idea of how best to tell the difference between good and bad messages.