# PublicRules

## Wouldn't a secret ruleset work better?

A common question regarding SpamAssassin's rules is, why aren't they kept secret? Doesn't publishing the rules alert the 'bad guys', causing them to change their spam patterns to evade the rule?

This is true, but only to a degree. In a way, this is an example of the 'Security through obscurity' fallacy.

### Spammers aren't all that smart

Firstly, not all bad guys follow SpamAssassin's ruleset development. We know that some do – especially the more technically adept, such as the phishing gangs. However, many spammers don't.

For example, the 'JODY' rule (matching the text 'My Wife, Jody') was part of SpamAssassin for many years, spam hitting that text has been observed in mail going back into the 1990s, and modern spam mail still hits that pattern. Many chain-letter scams rely on word-for-word reproduction of a piece of text like this.

In addition, some more complex rules that match mail delivery patterns are effectively unevadable; for example, spamware applications that delivery through broadband-connected end-user machines running an open proxy will always run the risk of being listed in the Spamhaus XBL blocklist.

Other rules match on side-effects or behaviours of the spamware applications that would require expensive changes in those applications. Since most spammers now do not develop the spamware software themselves, this introduces a 'window of opportunity' between the rule being developed, the spamware being fixed, released, and eventually upgraded by the spammers. This can take several months, during which the rule is highly effective – and many spammers carry on using old and vulnerable versions of spamware apps for years.

### Open Source development

By publishing the rule in SpamAssassin, it's exposed to the open-source development process; in other words, it's open to being improved by a far wider range of people, and exposed to far wider testing against real-world conditions, than in a small environment.

Even what appears to be a fantastic rule on a small scale, is frequently revealed to have weaknesses once other eyes review it. By keeping the rule secret, it will not be exposed to this scrutiny, review, and collaborative development.

### Spammers can work out how to get around secret rulesets

Several years ago, after the release of Vipul's Razor, we started to see HashBuster text appearing in spams. We assumed these were added to defeat Razor, DCC, Pyzor and other (open source) hash-sharing collaborative filtering schemes.

We later had the opportunity to review some documentation for a spamware application which was very specific about the real reason they had added this feature; it was to evade AOL's secret implementation of a similar system.

It seems that the spammers had figured out that AOL was using hash-sharing as an anti-spam system, and, through trial and error, possibly using 'test' accounts on AOL and manually changing the spam being sent, had figured out how to circumvent this.

In other words, even though they had no access to the source code or published details of AOL's system, they could reverse-engineer enough details to get around it.

### A Test Case

Here's an analysis of the change in effectiveness of the MIME_BOUND_DD_DIGITS rule over time, during 2004 and 2005.