RulesProjSecrecy

Rules Project: Secrecy

(part of RulesProjectPlan)

LorenWilton: 'There is a second thing here that gives me even greater concern. We have discovered that rules can be discussed openly on the users or dev list just fine, even going into some detail on what they do and how they work, and it will not have a noticible effect on how well a rule catches spam.

We have also found that the instant an actual rule is posted on the user's list, it will lose about 80% of its effectiveness, usually within about 16 hours. Within a week it will be virtually useless. Sometimes the rule will regain some effectiveness a few months later, and in rare cases posting a rule will not affect the hit rate. But in general, public posting in a readable forum of a rule body will negate the usefulness of the rule almost instantly.

One can speculate on why this happens, since the rules are there to read on any SA system, and can be trivially downloaded from SA and SARE for casual examination. Evidence shows though that this doesn't have an effect on the effectiveness of the rules. But posting the body of the rule on a mailing list does. Moderately strange, but of moderate concern, also.

I have some concern that a rules project *might* open up new rules to ineffectiveness, similar to posting them in a forum. However, the difficulties (for the average spam tool writer, at least) in using svn may prevent this from being a real problem. But it is worth devoting a few moments thought to the possibility.'

JustinMason: it *is* a problem, but in my opinion there's really nothing that can be done about this – we're an open source project, and the code is visible. while there's downsides, it also brings big benefits as well (as I said, the alternative is working for Brightmail 😉. Open development is a requirement of being an ASF project, btw.

(To cut and paste some paragraphs from a mail I sent a few months ago...) I've thought about this for several years, since I work on a project that

- (a) can use spam-forensics tidbits to great effect
- (b) is open source and therefore relies on public, open discourse
- (c) is in the antispam field, where secrecy can go a long way

I still haven't made any kind of hard-and-fast cut-and-dried decision. But my tendency is towards the following:

In cases where info must remain secret to secure convictions; keep it secret. A conviction, esp of a major spammer or ratware vendor, can make a much bigger difference to the state of antispam than anything else.

In cases where info may provide better filtering for a few months at a few sites, if kept secret; I don't think it's entirely useful to keep this secret. Here's a few reasons:

1. The ratware developer may change their tactics anyway, due to filtering

pressure from other sides. Hash-buster development in 2002 and 2003 is a great example. We thought they were changing due to Razor and Pyzor, but then realised a year later that it was AOL's similar systems that drove the changes – it had *nothing* to do with our open-source projects.

2. the secret may be rediscovered elsewhere (e.g. our realisation a few months back in SpamAssassin that we were matching about 5 different aspects of one ratware's behaviour without connecting them);

3. forcing the ratware developer to update their software and keep changing makes hard work for them, annoys their customers, and leaves the users who are using pirate copies vulnerable anyway. *Lots* of spammers are running pirated versions of their ratware, so cannot update if their spam starts hitting rules that hit structural features of the message or SMTP delivery.

4. better openness allows people to *enter* the field of antispam forensics. I found a lot of interest out there, and quite a few people already looking into the field quietly on their own, after presenting a talk about spam forensics at a hacker conference in San Diego last year. People are already doing this for fun, and for their own use, but they don't talk about it and we ourselves are missing out a lot of the details.

5. think about #4 for a sec and you'll realise that it's another form of open-source v closed-source thinking, and also analogous to the full-disclosure argument in network security.

The key factor to fix this problem, we think, is to have fast, fast turnaround on rule publishing – that way when the spammer mutates, if they do, we can keep up. we know we need to get things turning around faster – Theo's "sa-update" script (SaUpdatePlan) is the key to this.

There are other techniques, also, but let's not talk about them here... 😉

doh, I'd already written about this on the wiki – see also PublicRules.