

Staying ahead of spammers

Developing new anti-spam measures

christian mock

<cm@quintessenz.org>

Linuxdays.lu, 2006-01-26

Agenda

- What this talk is not about
- Greylisting: magic bullet or annoyance?
 - combine Greylisting + SPF + DNSBLs + ...
 - Code & numbers
- Spamtraps: beyond just collecting junk
 - Run your own DNSBL!
 - no Code, disappointing numbers
- More ideas?

About me

- Co-founder of one of .at's first ISPs
- now working in IT Security + Antispam
 - Protecting a few 10'000s of mail accounts from spam
- Experimenting with anti-spam techniques on my home mail server
- Talks about anti-spam techniques at linuxdays.lu, Chaos Communication Camp, Linuxwochen.at

What this talk is not about

- General anti-spam measures
 - you should've attended the tutorial today :-)
 - or see <http://www.tahina.priv.at/~cm/talks/>
- I assume you know the basics
 - DNSBLs
 - SpamAssassin
 - Greylisting

Greylisting

- Basic idea: don't accept message from new sender/host at first try, make them wait for a few minutes
- Works quite well against spam, worms
- Problem: Imagine you run a big mail server with lots of traffic to lots of greylisting destinations -> huge queues
- Long usenet discussions “is good” / “is evil”

Other problematic stuff

- SPF: problems with forwarders, people sending from unlisted hosts, etc
- Strict HELO string checking: misconfigured systems
- Reverse DNS lookup: often impossible to get ISP to configure correctly
- Dialup DNSBLs: people should be able to send from dynamic addresses, only zombies shouldn't

Mix it all together...

- Combine all those partially effective, partially problematic techniques into something better: **“Selective Greylisting”**
- Do all the checks
- If everything is OK, accept message
- If “problems” turn up, use greylisting

The Checks

- Reverse DNS lookup of client IP
- HELO name must be FQDN and exist in DNS
- Is client IP on a dialup DNSBL?
- SPF “pass” or “none” (using Mail::SPF::Query's “guess” mode)
- Greylisting tuple is
`client_addr, helo_name, sender_domain`

The code

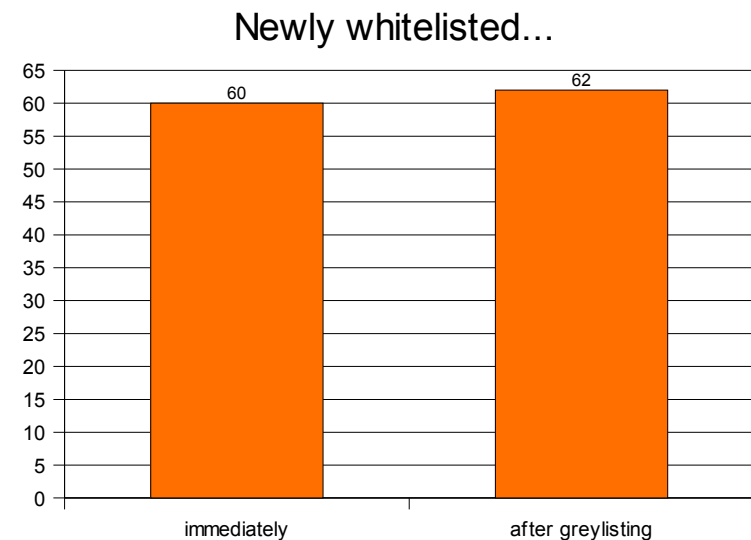
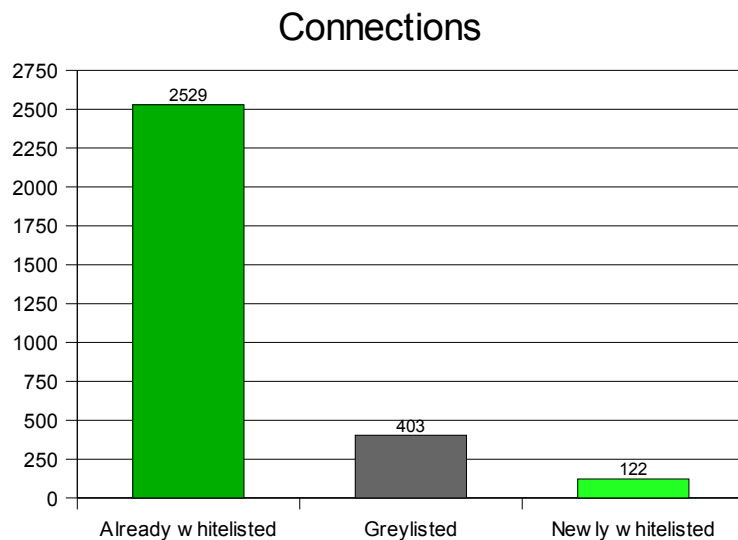
- Postfix policy delegation daemon in Perl
- <http://www.tahina.priv.at/~cm/spam/>
- Extensible
- Policy is coded because policy **is** code
- Runs fine since ~ 6 months
- Some features still missing
 - e.g. database expiration

And... does it work?

- Rejection rates decreased by 1/3-1/2
 - Not every client is greylisted
- I still receive about the same amount of spam
- It helps a lot against stupid worms
 - Sober.latest generated a new HELO for every connect -> new greylisting

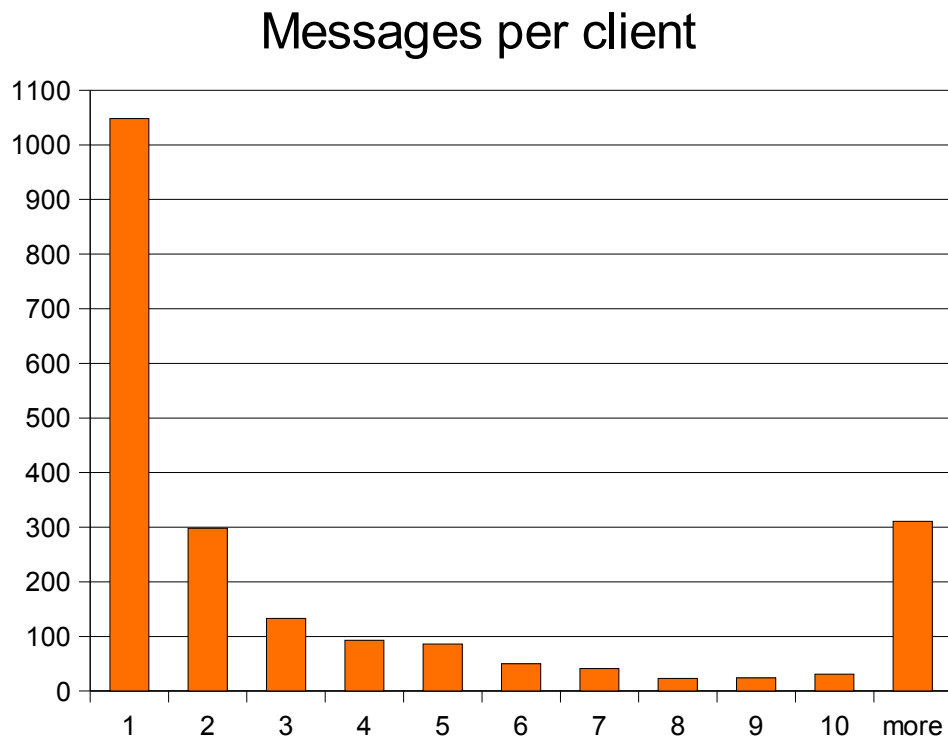
The numbers

- 4 weeks worth of logs, DB already filled with entries



More numbers

- How many messages do whitelisted hosts send?



Room for improvement
(1- and 2-message clients are probably spammers)

Make those spamtraps do real work

- Train SA's Bayes filter on the spam
 - Easy, forward to “|sa-learn –spam”
- Extract client IP and URLs, build your own DNSBL
- run DNSBLs via rblDNSd
 - great tool
 - easy to use

The code

- URL extraction via SpamAssassin itself
 - URI DNSBL will be used by it, after all
 - plus it saves work
 - use internal SA variables to get the URLs
- Bayes also trained directly
- The code is unpublished: ugly and ineffective
 - maybe someday

The numbers

- 93 rejects from the client IP DNSBL
 - in one month; total rejects: 30000
- 94 matches on the URI DNSBL
 - in two months
 - all but 1 were detected by SA anyways
- So this is just a waste of CPU cycles, the public DNSBLs work quite good as they are

That's it

Thanks for your time...

...and don't hesitate to ask any questions left

<http://www.tahina.priv.at/~cm/talks/>

<http://www.tahina.priv.at/~cm/spam/>