>BUSINESS MADE **SIMPLE**

Effective Open-Source Spam Filtering For Enterprise

Chris Lewis

Thomas Choi

October 2008

NORTEL

VB2008, Ottawa

Agenda

- Introduction
- Background
- Something New - Rationale
- The Open-Source Project
    - Basic Requirements
    - Components
    - Integration
    - Test/Performance
- Advanced Techniques

Introduction/Authors

Chris Lewis

Senior Security Analyst/Anti-Spam, Nortel

Senior Technical Advisor, MAAWG

Member, Canadian Federal Anti-Spam Task Force

Thomas Choi

Nortel

Ph.D Student, Carleton University

Background

- Spam became a problem in 1994/1995
- Initially in Usenet
- Clearly would transition to Email
- Commenced Email Anti-Spam program in 1997
- Extremely customized Lyris Mailshield implementation
- VB2004 “Corporate Spam Fighting: 5 years of success and lessons Learned”: by Chris Lewis and John Morris – don't forget those lessons!

Something New - Rationale

- Lyris Mailshield has stood us in good stead
- But, getting a little elderly, higher volumes, difficult to extend with newer techniques
- Review of many other vendor offerings:
  - All missing one or more of critical features
  - Integrated poorly with existing infrastructure
  - Not, or poorly extensible/configurable
  - Not as effective as current solution

Rationale ... Continued

- Needed open architecture/modular/easy extension
- Low capital/license cost (free obviously best!)
- Use standard components to minimize development costs
- Use existing basic low-medium size server class hardware
- Focus on 3rd party/popular filtering methodologies, simple ad-hoc filtering capabilities, plus with our own “secret sauce”.
- Avoid training (software OR people) requirements

The Open Source Project

- Basic Requirements – Functional Specification
- Component Selection
- Integration
- Back end
- Testing

Basic Requirements - Filter

- Support multiple recipient domains
  - Configurable per-domain handling
  - Per-domain filter enable
  - Configurable archiving/quarantine/disposition (pass,filter, trap)
  - Output routing
- Full logging
- NEVER bounce or silent blackhole (except trap)
- Plugin architecture – each technique an independent module

Basic Filter Requirements ... Continued

- Fault tolerant (eg: failover)
- Support 3rd party facilities, eg:
  - DNSBL (IP blacklists)
  - SURBL/URIBL (URI blacklists)
  - “informational” lookups (eg: ASN)
- Content Scoring filter
- Anti-virus
- Arbitrary ad-hoc string filters anywhere/on anything
- Direct/real-time feedback to filtering

Basic “Not filter” Requirements

- Full end-user quarantine view/forward
- End-user (recipient) notification (if desired)
- Full logs in database/arbitrary queries
- (Almost) fully automated false positive handling (forward, filter tune, notification/explanation)
- Operational and Management metrics
- Postfacto analysis and automated filter tuning

Components, Filter, Open-Source

- Core SMTP listening engine/agent: Qpsmtpd (Hansen, Sergeant et. al.). 100% Perl implementation (really!)
  - Async (event driven) mode
  - Very high performance – 20M+/day small servers
  - Entirely flexible by plugin interface
  - Actively supported & robust
  - Has many sample plugins
- SpamAssassin (popular scoring addon filter). (Perl)
- ClamAV (*ix-based) anti-virus signature-based engine
- Nearly two dozen ad-hoc filtering plugins, few more than a dozen lines.
- The libraries and utilities to make the above work (eg: ParaDNS)

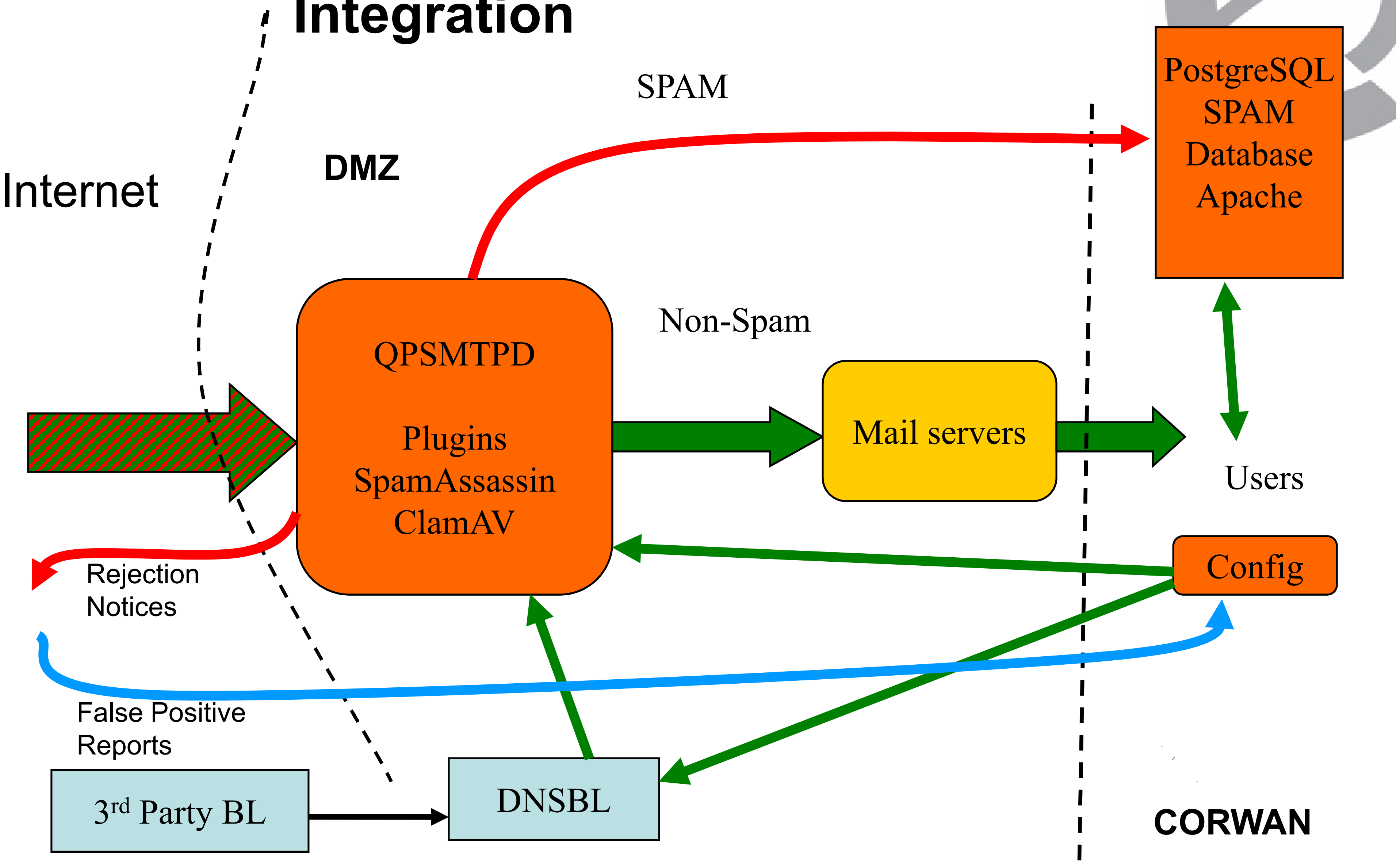
Components, Filter, Glue

- A spam filter is more than just a filter, needs:
- Start/stop/reboot/monitoring
- Log & quarantine handling and transfer
- Extended filtering heuristic processes (for things that take too long for real-time)
- Install/deployment and filtering control

Components, Backend

- PostgreSQL database
- Apache (admin and user interface)
- Interface to corporate user databases (push to filters)
- Admin (research, false positive, configuration, deployment) interface CGIs
- User interfaces (configuration and quarantine)
- Quarantine management
- Rsync – log, quarantine, configuration, software transfer

Integration



Test/Performance

- Spamtrap operating 9 months
- Performance heavily depends on “early pruning”
  - “Cheap” tests first
  - Prune filtering subsequent to block decision
  - “Expensive” (body scans, SpamAssassin, ClamAV) tests last
- Volumes: typical 7m/server (50-100/sec), mostly spamtrap
- Deployment in progress

Advanced Techniques

- State of Affairs
- Hide!
- Banner delays
- Bot fingerprinting
- DNSBLs (local and/or otherwise)
- DNSBL infrastructure
- Bounces & BATV
- Ones we've omitted and why

State of Affairs

- Underground economy (spam, phish, spyware, CC, mules) increasing
- Some LE believe larger than International Drug trade
- BOTS responsible for 80%+ of all spam.
- Most getting good at stopping BOTs (<1% deliverability)
- => BOTs shifting to reputation theft (relay through legit MTAs)
- State of Anti-Virus: disaster. (new BOT caught by AV 23% of the time by battery of 35 AV tools, only increases to 50% by 30 days)
- Inadequate AV => can’t find BOT, let alone remediate proven infections.

Hide!

- Make it difficult for BOTs to email you.
- BOTs not full MTAs, high volume/throughput requirements.
- Primary MX – “refuse connections” (Google for “nolisting”)
- Tertiary MX – “always retry”
- Dumb bots try once (primary or tertiary), get refusal or retry, and give up. Real MTAs do right thing.
- As much as 50% of BOT spam simply vanishes.
- Loss of metrics.

Banner Delays

- Most BOTs impatient, and won’t retry
- 20-40 second banner delays =>
- BOTs give up in disgust
- Some legit MTAs equally impatient, may need to whitelist some.

BOT Fingerprinting

- Most BOTs have fingerprints in the headers and SMTP protocol that can be caught by pattern matching.
- Some mutate, some don’t.
- Srizbi > 50% of all spam.
- Feed source IP of detections back into local DNSBL.

DNSBL (DNS Blacklist)

- Hundreds of 3rd party DNSBLs (IP based, domain based, URIBL filtering etc)
- A handful are both reliable and effective.
- There are DNSBLs effective to 70-80%+ of all spam & virus propagation attempts.

DNSBL Merge

- High volume receivers may impose undue loading on 3rd party DNSBL infrastructure.
- Occasional erratic delays (including DDOS on DNSBL)
- => Host them locally
- We use rbldnsd – very high performance DNS server designed for high-performance serving of DNSBL zones.
- We combine multiple 3rd party zones (plus ones we create ourselves) into a single zone.
- Each DNSBL source distinguishable by return code, multiple DNSBL results “scored”. But most hits at threshold.

Filtering/Bounces & BATV

- Accepting then bouncing email with forged from => bounce storms (aka backscatter/blowback) => evil
- Simple blackholing also evil
- Aim is inline reject, with remediation information.
- Support costs of receiving end of blowback often exceed spam
- BATV (Bounce Address Tag Validation) **see http://mipassoc.org/batv/**
- **When sending email, encode bounce address (MAIL FROM)**
- **When receiving bounce, reject email not encoded)**

Omitted Techniques & Why

- Greylisting – (force retry of “new senders”).
  - Increasing reports of BOTs doing retry.
  - Doesn’t prevent spam-by-reputation-hijacking
- Bayesian – needs training, in many cases defeated
- Checksumming (Razor/DCC et. al.) –
  - Detects bulk, not spam per-se
  - Problemmatic when outsourcing user-contact (eg: HR)
  - Needs whitelisting
  - BOT hash busting getting better