E-mail Content Scanning with Exim 4

Tim Jackson

Overview

- Introduction to content scanning
- Content scanning at the MTA – issues
- Methods of implementing content scanning
  - ➔ Accept-and-scan
  - ➔ SMTP-time
  - ➔ Software required
- Brief overview of common software
- A look at Exiscan
- Other considerations
- Conclusions

Basic Rationale & Considerations

- Defend against viruses, spam or other unwanted messages
- Final check for messages that have passed other checks
- Not a substitute for DNSBLs and other non-content checks!
  - ➔ False positives based on content typically occur more frequently than with checks not based on content (e.g. DNSBL lookups)
- Need to consider a variety of policy and “good practice” issues, not just technical ones

Methods & Software

Inbuilt methods for content scanning

- SMTP time: Rules making decisions based on content (e.g. `$message_body`) in the DATA ACL
- Post-SMTP: Exim/Sieve filters (per-user or system filters)

Some common external content scanning software

- SpamAssassin– Free software spam checker
- Clam Antivirus – Free software virus scanner
- Sophos/sophie – commercial virus scanner/Free software daemon

Content Scanning at the MTA

Advantages

- Centralised – easy to apply consistently across large user populations
- Transparent to end users
- Easy to maintain – all in one place

Disadvantages

- Confuses the role of an MTA
- Can be a blunt instrument, takes control away from users
- Centralises burden on mail servers – resource usage!

Two primary ways to implement content scanning at the MTA

- Accept-and-scan
- SMTP-time scanning

Accept-and-scan

- Accept the message as normal, and process content later
- Internal – filters make decisions on message content
- External software - routers used to pass the message to external software - typically either fails the message and generates a bounce, or re-injects into Exim

Accept-and-scan: advantages

- Easy, does not require any extensions to Exim
- Easy to allow complete per-user configuration

Accept-and-scan: disadvantages

- After accepting, what to do if it's detected as virus/spam?
  - ➔ Drop silently – makes mail system unreliable.
  - ➔ Tag/deliver to a separate destination – OK, but recipient still ultimately gets the spam/virus
  - ➔ Create a bounce – bad practice in current climate - most senders are faked
    - ⓘ You “collateral spam” innocent third parties – adding to the problem!
    - ⓘ Even if the (faked) sender doesn't exist, you will add load to some innocent party's systems, and your queues will fill up with frozen bounces
- Typically involves scanning a message multiple times for multiple recipients

SMTP-time scanning

- Scan during the SMTP DATA phase

Advantages

- Elegant: if you're not going to accept a message, better to reject outright
- *Reduces collateral spam!* (major consideration – best practice)
- No more queues filled with bounces

SMTP-time scanning: disadvantages

- Requires sufficient resources to scan quickly & return back to SMTP session – risk of duplicates
- Stretches strict RFC compliance slightly – though shouldn't cause interoperability problems
- Per-user configuration options limited – content scanning only takes place once per message, not per recipient.

Software required

- Exim ☺
- Content-scanning patch to Exim – “glue” to pass the mail from Exim to the external software and return a result
- Scanning software (virus/spam scanners etc.)
- Should be daemonised if possible for performance

- Before diving in, consider the *policies* to be implemented as well as the *tools* to do it.

Content-scanning patches

- Exiscan
  - ➔ “Swiss army knife” - support for lots of external anti-spam/anti-virus tools. Operates in the ACL system.
- SA-Exim
  - ➔ Single-purpose spam-scanning patch for SpamAssassin. Includes 'greylisting', 'tarpitting' and more. Operates using the local_scan system and separate configuration file.
- FFPA
  - ➔ Extension to Exiscan; allows detection of attachments based on their actual file type, not just their file extensions.

Anti-virus scanning software

- ClamAV
  - ➔ Free software; very good. Regular signature updates.
  - ➔ Daemonised; includes separate daemon to monitor for signature updates.
  - ➔ Scalability and stability issues? Many use it successfully.
    - ⓘ Variation called 'nclamd', supposedly more stable, will probably be merged eventually
- Sophos
  - ➔ Commercial software, with support.
  - ➔ Doesn't include daemon, but free software 'sophie' daemon stable and works well
- Others
  - ➔ Kaspersky, ScannerDaemon etc.

SpamAssassin

- Written in Perl. Primarily pattern-matching; includes some other checks - DNSBL lookups, Razor etc.
- Works on 'points' system – each pattern (or rule) matched scores points (positive or negative)
- Points scores allocated used to flag or reject mails (e.g flag at score=5, reject at score=10)
- Can modify message content for detected spam
    - ➔ Works with SA-Exim, not with Exiscan.
- Includes Bayesian learning and analysis system
- Spammers tailor their messages to avoid SA hits
    - ➔ Use custom rulesets/add-ons

Other anti-spam software

- Spamprobe
- Bogofilter
- CRM114
- Not currently supported directly by any Exim patches mentioned

Exiscan overview

- Source code patch to Exim, maintained by Tom Kistner
  - ➔ Normally released together with or shortly after Exim releases.
- Many Exim binary distributions/packages are pre-patched
- Elegant integration. Hooks into the Exim ACL system
  - ➔ New options in DATA ACL to call external scanning software
  - ➔ Inbuilt MIME decoder
  - ➔ MIME checking to detect serious MIME errors
  - ➔ File extension matching (e.g. block all .pif files)
  - ➔ Regular expression matching of decoded or raw MIME parts
  - ➔ New ACL: acl_smtp_mime – called once per MIME part

Some brief Exiscan examples

- Too many possibilities to cover everything
- Comprehensive documentation & examples on Exiscan site

Reject spam

```
deny message   = This message was classed as spam
     condition = ${if <{$message_size}{80k}{1}{0}}
     spam      = nobody
     condition = ${if >{$spam_score_int}{99}{1}{0}}
```

Reject viruses

```
deny message   = Message contains a virus \
                 ($malware_name)
     malware   = *
```

Exiscan examples continued

MIME checking

```
deny  message = Serious MIME defect detected ($demime_reason)
      demime = *
      condition = ${if >{$demime_errorlevel}{2}{1}{0}}
```

The multiple-MX problem

- Consider whether you really need multiple MXes
- If you have more than one server, all need identical protection!
  - ➔ Avoid spam 'backdoors'
  - ➔ Avoid collateral spam

The multi-recipient problem

- Affects scenarios where not all users have the same scanning preferences
- No easy way round it due to limitations of SMTP, but reasonable workarounds available
- If you have to offer options, try to keep the choices simple. Even if “one size doesn't fit all”, maybe “two sizes” do?
- Consider SMTP defers (multiple-scan-profile method)
  - ➔ http://www.exim.org/pipermail/exim-users/Week-of-Mon-20031006/061151.html

Exim as a transparent front end

- Drop in front of existing mail system
- Route messages to the “real” MX using a manualroute router
- Simple, static example:

```
route_scanned_mail:
  driver = manualroute
  domains = somecompany.example.com
  route_data = 192.168.0.1
  transport = remote_smtp
  no_more
```

Conclusions

- Useful tool as part of a wider policy framework
- Needs responsible planning and implementation
- The Exiscan patch is widely used, stable and powerful for
  - ➔ Anti-virus/anti-spam
  - ➔ File extension blocking
  - ➔ Regular expression blocking
- At least *some* basic content scanning highly recommended!
- These slides, a copy of the summary notes, a detailed transcript, and Content Scanning HOWTO available at:
  - ➔ http://www.timj.co.uk/computing/software/exim/