Mail filtering on huge mail servers with j-chkmail

José-Marcio Martins da Cruz
Ecole des Mines de Paris

TERENA Networking Conference 2005 – Poznan, Poland

Plan

- Huge servers
- Content and behaviour filtering
- Filter real life in huge servers
- Filter scaling – j-chkmail's content and behaviour filtering
- Cooperation between behaviour and content filtering
  - Scalable Adaptive Delay Greylisting
- Server protection
- Results
- Conclusions
Medium / huge mail servers...

- Many thousand users
- Handling many hundreds of thousands messages each day
- Handling many gigabytes each day
- Typically – university campus gateway
Medium / huge mail servers

- Heterogeneous population: computer scientists, physicians, sociologists, purchaser... -> hard to define what a typical mailbox looks like!
- Users are unknown to mail server administrator -> filters are hard to tune!
- It's usual to have filtering being done on gateways (a place where user mailbox information isn't available) to protect storage servers
- Security issues – ideal target for attacks
- Reliability issues (availability, downtime, ...)
- Low level hardware/OS constraints – limited resources (file descriptors, processes, disk I/O and network bandwidth, ...)
(Parenthesis: SMTP dialog)

martins@calloway:~> telnet paris smtp
  Trying 194.214.158.200...
  Connected to paris.
  Escape character is '^]'.
-> helo calloway.ensmp.fr
<- 250 paris.ensmp.fr Hello calloway [194.214.158.171], pleased to meet you
-> mail from:joe@ensmp.fr
<- 250 2.1.0 joe@ensmp.fr... Sender ok
-> rcpt to:martins
<- 250 2.1.5 martins... Recipient ok
-> rcpt to:tontonpion
<- 550 5.1.1 tontonpion... User unknown

-> data
<- 354 Enter mail, end with "." on a line by itself
-> From: Antoine
-> To: Sebastien
-> Subject: test telnet
->
-> C'est un test, je dis !
-> .
<- 250 2.0.0 h2QBmFBx017626 Message accepted for delivery
-> quit
<- 221 2.0.0 paris.ensmp.fr closing connection
  Connection to paris closed by foreign host.
martins@calloway:~>
Behaviour filters

• Checks how some “parameter” evolves with time (not completely true)
• Learns with the past -> spend memory to save CPU cycles
• Reject connections (or stop tests) before checking message body (SMTP DATA command)
• Some examples :
  – RBLs – known spam sources, open relay servers, ...
  – Spamtraps – email addresses “distributed” only to spammers...
  – Connection rate : bursts versus connections exponentially distributed over time
  – Greylisting : does SMTP client tries sending again after temporary failure DSNs ?
  – ...
• Also, some RFC 2821 conformity checks : EHLO, greet_pause, ... (not really behaviour)
Content filtering

- Filtering is done based mainly on the content of SMTP DATA command: headers and message body
- Many different techniques ranging from pattern matching to natural language processing
- Each new connection or message is a new event – no history
- Content filtering consumes much more resources than behaviour filtering
Content filtering

- **Pattern Matching**
  - Checks if any of defined regular expressions can be found in incoming message.
  - Hard to maintain – maintainer shall check all received *SPAMs* to find pertinent patterns.
  - Resource consuming – each expression is matched against entire message.
  - Low efficiency

- **URL filtering**
  - Checks if URLs found in incoming message are present on URL blacklist.
  - Easier to maintain – semi automatic (scripts + validation) extraction of URLs from a bunch of received *SPAMs*.
  - Very fast – only URLs found in message are checked against a large blacklist database.
  - Very efficient: results from SURBL database are better than 80 % for detection rate and less than 0.5% for false positive rate.
  - Independent of SPAM/HAM corpus – Listed URLs never appears in HAM (surbl strategy).
Content filtering

- Bayesian filters – *Bogofilter*, ...
  - Probability of being a SPAM – combines the probability of each word being a SPAM word
- Heuristic filters – *SpamAssassin*, ...
  - Many very diversified tests – message score is the sum of scores for succeeded tests
  - Tests with positive and negative weights – score evaluation isn't monotonic
- Bayesian and heuristic filters are based on statistical data from user mailbox: they learn how your mailbox looks like. Classification is optimal if incoming traffic matches your mailbox.
Life at huge servers – top hits

# 1 127535 URIBL_WS_SURBL
# 2 127101 URIBL_SBL
# 3 125917 URIBL_JP_SURBL
# 4 120728 URIBL_OB_SURBL
# 5  96849 BAYES_99
# 6  95827 RCVD_IN_BL_SPAMCOP_NET
# 7  90406 HTML_MESSAGE
# 8  71017 URIBL_SC_SURBL
# 9  46927 MIME_HTML_ONLY
#10  36806 URIBL_AB_SURBL
#11  33822 RCVD_IN_XBL
#12  30930 MIME_BOUND_DD_DIGITS
#13  30649 MPART_ALT_DIFF
#14  28472 URIBL_AH_DNSBL
#15  26638 RCVD_IN_SORBS_DUL
#16  26621 DRUGS_ERECTILE
#17  26394 MSGID_FROM_MTA_HEADER
#18  24615 RCVD_IN_DSBL
#19  23977 MSGID_FROM_MTA_ID
#20  23690 RCVD_IN_SORBS_SPAM
#21  22457 RCVD_IN_NJABL_DUL
#22  21115 RCVD_IN_NJABL_PROXY
#23  21013 RCVD_IN_SBL
#24  20262 X_MESSAGE_INFO
#25  18044 HTML_FONT_BIG

IP RBL
URL RBL
Bayesian filter
Heuristic filter
Pattern Matching

Data from prolocation.net
6 hours – 440K messages
January 2005
Thanks to Raymond Dijkxhoorn
We can see that ...

- SURBL check is the most effective criteria
- While hundreds of checks are executed, very few are enough to detect most of the incoming spam
- External criteria (URL and IP blacklists) and bayesian checks are more effective than others
- Some unreliable heuristic checks appear with high frequency (HTML_MESSAGE and MIME_HTML_ONLY)
A word about RBLs

mail-abuse.org : 175396
-- 127.1.0.1 : 1180
-- 127.1.0.2 : 168244
-- 127.1.0.3 : 855
-- 127.1.0.4 : 302
-- 127.1.0.6 : 35
-- 127.1.0.8 : 3820
-- 127.1.0.9 : 22
-- 127.1.0.10 : 875
-- 127.1.0.12 : 60
-- 127.1.0.14 : 3

• Using mail-abuse is equivalent to the policy: “I don't accept connections from ISP end users”
Filter scaling

- Resource consumption (CPU, memory, ...) shall grow slower than traffic level, or at most at the same rate.
- Remove all external dependencies (DNS, network checks) – faster and securer.
- Use only reliable criteria - avoid methods depending from typical user mailbox
- Compromise between doing well and doing fast.
- Do, whenever possible, behaviour checking – faster than content checking.
- Don't loose time – Little's Law says: *mean number of processes grows with connection rate and stay time.*
- **And the must**: the filter shall learn while it works – use memory to save CPU cycles.
**j-chkmail behaviour filtering**

- A set of very fast checks – connection rate, bounce rate, spamtraps, harvesting, RFCs compliance, greylisting, handling time (CPU usage), volume, ...

- Three levels of persistent history:
  - Recent: 20 minutes – activity of all SMTP clients
  - Medium: 5 hours – bad or dubious behaviours
  - Long: some days (on disk) – confirmed bad behaviour (usually from log files)

- A kind of Real Time Blacklist! – connection is rejected if bad behaviour over some hours or too resource consuming over some minutes.

- Behaviour filtering doesn't block too much spam, but protects the server against surges (greylisting is an exception to both – will talk about later).
j-chkmail content filtering

- **URL filtering**
  - surbl.org database – BerkeleyDB and DNS versions. BerkeleyDB is faster when running with fast SCSI disks and enough cache memory.

- **Pattern matching**
  - Very few expressions (~150)
  - Complement other checks (immediate needs or really stable/reliable expressions)

- **Heuristics**
  - Very few criteria (31 in the last snapshot and number being reduced)
  - Prefer effective criteria (high detection and low false positive rates).
  - Only positive weight checks – monotonic score evaluation
    - False positive rate is higher than other filters: message whitening is let to final user (address book, known message sources, ...)
**j-chkmail content filtering – seem by user**

- The goal is qualitative: help message classification by final user
- False positives resulting from the lack of negative weight criteria are compensated by “user address book”.
- What happens to messages?
  - Message is rejected by the server if score exceeds some threshold
  - If accepted, message score is presented at some header (X-j-chkmail-score)
  - User configures his MUA to:
    - Put messages coming from known users in normal Inbox
    - Put messages with high scores in SPAM mailbox
    - Otherwise put message in normal Inbox
Behaviour and content filtering cooperation

● Content -> Behaviour
  - Behaviour thresholds are lowered for clients sending SPAM or virus detected by content check.
    - Ex: connection rate limit for this client is divided by two if mean content score for its messages, evaluated over past 10 minutes, is too high, or if virus detected

● Behaviour -> Content
  - Some behaviour checks don't block connections but contribute to heuristic score

● WARNING – avoid closed loops, otherwise the filter may become unstable
Nice example : Adaptive Delay Greylisting
Under validation

• Greylisting, by itself, isn't scalable: number of database records grows with recipient rate (faster than message/connection rate)
• Security vulnerability – easy to poison database
• Basic idea – eliminate redundancy
  – DB records lifetime is reduced for clients with confirmed bad behaviour or some non priority client (null sender/bounces, DNS resolution, ...)
    • 192.87.30.2:joe@terena.nl:joe@ensmp.fr vs 205.158.62.177:joe@terena.nl:joe@ensmp.fr
  – Recent pending records are removed for clients with surges of dubious behaviours on short history (virus, spams, harvest, ...)
  – Limit the number of pending records per IP address
  – Greylisting database content is periodically scanned to detect very bad and very good behaviours.
Adaptive Delay Greylisting

IP/From/To

Pending.db

IP/From/To

Valid.db

Filter results

Purge

Trash

Black.db

Whiten

White.db

IP/From (domain part)
Adaptive Delay Greylisting

- Limited simulation on data from domain jussieu.fr (~ 500 000 connections a day)
- Pending entries database – 430614 records over last 5 days
  - Age limiting: bounces (3971), DNS resolution (111619), domain-name/email matching (250535), max entries exceeded (1761)
  - Number of records removed: 367886
  - Pending triplets DB size reduced to 62728 records (~ 15 %)
- Algorithms and prototype under validation at ensmp.fr – improving reliability on black and white lists generation

![Daily distribution of pending entries](chart.png)
j-chkmail – server protection

- Auto restart
  - Filter status is periodically checked by the supervisor – if it detects a problem, cleaning up is done and a new filter instance is started over.
- Connection rate control (per client SMTP).
- Simultaneous open connections control (per SMTP client).
- Global load measurement.
- When load is high, access is granted in a priority basis...
j-chkmail – connection rate surges

Incoming

Filtered

Volume data shows:
no impact on normal traffic
Connection rate control in action

- 10536 connections in 8 minutes
- 238 clients from network 66.216.119.0/24
- Each client made [28 – 67] connections
- Peak: 86 connections in the same second
- 15 messages rejected by content filtering www.rapiddealsbyemail.com
- 8156 connections rejected by connection rate – maximum allowed: 10 connections / 10 min
- No HAM lost: “Poor's man QoS”

Data from paris.ensmp.fr – April 2003
Virus filtering

- File extension based filtering (.exe, .pif, ...) - much faster than virus scanner, as check is done only on file name, not file content
- External antivirus (ClamAV, ...)

![Graphs showing network activity over time](image)
martins@paris:~> j-printstats -q -l 6h | more
Version : Joe's j-chkmail v1.7
*** Summary
First Connection : Sun Jun  6 17:33:11 2004
Last Connection  : Sun Jun  6 23:33:09 2004
Connections      : 9393
Gateways         : 4258
Throttle Max     : 445 / 10 min (for the server)
Throttle Max     : 100 / 10 min (for a single gateway)
Duration (sec)   : 0.005 16.931 7226.787 206.110 (min mean max std-dev)
Work (sec)       : 0.001 0.028 1.803 0.150 (min mean max std-dev)
...

![Charge (2 month)](image1)

![Spam - content (1 month)](image2)
j-chkmail behaviour on servers

- ensmp.fr
  - 2000 users, 60000 connections / day
  - 1 Sun E280R, Solaris 9, 2 x Sparc III 900 Mhz

<table>
<thead>
<tr>
<th>PID</th>
<th>%CPU</th>
<th>%MEM</th>
<th>VSZ</th>
<th>RSS</th>
<th>SZ</th>
<th>CLS</th>
<th>LWP</th>
<th>NLWP</th>
<th>PSR</th>
<th>S</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>28872</td>
<td>0.9</td>
<td>0.5</td>
<td>19072</td>
<td>18352</td>
<td>2384</td>
<td>TS</td>
<td>1</td>
<td>9</td>
<td>-</td>
<td>S /usr/sbin/j-chkmail</td>
<td></td>
</tr>
</tbody>
</table>

- jussieu.fr
  - 50000 users, 400000 connections / day
  - 4 mail servers under FreeBSD - j-chkmail + milter-greylist + Sophos

<table>
<thead>
<tr>
<th>PID</th>
<th>USERNAME</th>
<th>PRI</th>
<th>NICE</th>
<th>SIZE</th>
<th>RES</th>
<th>STATE</th>
<th>C</th>
<th>TIME</th>
<th>WCPU</th>
<th>CPU</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>1827</td>
<td>smmsp</td>
<td>96</td>
<td>0</td>
<td>29172K</td>
<td>26772K</td>
<td>select</td>
<td>0</td>
<td>40:45</td>
<td>3.52%</td>
<td>3.52%</td>
<td>j-chkmail</td>
</tr>
</tbody>
</table>

- pobox.sk
  - 15000 messages / hour - 1 Sun V65Z, 2.8 GHZ, under Linux
  - j-chkmail + clamd

<table>
<thead>
<tr>
<th>USER</th>
<th>PID</th>
<th>PR</th>
<th>NI</th>
<th>VIRT</th>
<th>RES</th>
<th>SHR</th>
<th>S</th>
<th>%CPU</th>
<th>%MEM</th>
<th>TIME+</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>smmsp</td>
<td>1727</td>
<td>16</td>
<td>0</td>
<td>6072</td>
<td>1692</td>
<td>3172</td>
<td>S</td>
<td>0.0</td>
<td>0.2</td>
<td>0:00.10</td>
<td>j-chkmail</td>
</tr>
<tr>
<td>smmsp</td>
<td>21448</td>
<td>16</td>
<td>0</td>
<td>145m</td>
<td>36m</td>
<td>11m</td>
<td>S</td>
<td>0.0</td>
<td>3.6</td>
<td>0:07.82</td>
<td>j-chkmail</td>
</tr>
</tbody>
</table>
j-chkmail typical filtering results

- Typical SPAM filtering figures with j-chkmail:
  - Mean connection handling time: ~ 30 ms on a Sun E280R (2 x Sparc III 900 Mhz)
  - Behaviour filtering – blocks 15-20% of incoming spam
    • Main interest is server protection
  - Greylisting – rejects 50-80% of remaining spam
  - Content filtering – rejects/tag 70 – 80% of remaining spam
  - Heuristic filtering – tags some more spam, but gives some false positives
Conclusions

● On huge servers, users satisfaction is the better filter efficiency measure
● There are more available data about spam on your mail servers than you may imagine.
● Do you want to improve your filter? “Learn while work” - this means: do real-time analysis on filtering results.
● If you can use only three filtering criteria, the good choices are:
  − connection rate control
  − greylisting,
  − URL filtering (surbl.org)
● j-chkmail: for the author, a test bench for ideas on mail filtering
Thanks to...

- Tibor Weis : pobox.sk and tuzvo.sk
- Sebastien Vautherot : jussieu.fr
- Dennis Peterson
- Raymond Dijkxhoorn : prolocation.net / surbl.org
- Jeff Chan : surbl.org