Connection Rate Control with j-chkmail

Some results from domain ensmp.fr

work in progress...

Jose-Marcio.Martins@ensmp.fr
Summary

- Our mail server environment
- Connection rate control – why and how?
- Events and results
- Conclusions
Our environment
Our mailserver environnement

- Internet connection – 100 Mbps
- 1500 users – 2500 e-mail addresses
- 30-50000 connections / day
- 0.8 – 1 Go / day (in + out)
- Sun E280R – 2x900 Mhz proc
- Sendmail 8.12.10 Beta2
- RBLs mail-abuse.org, osirusoft.com (non-official DNS slaves for this zones) and dnsbl.ensmp.fr

- A medium server, where we can validate our ideas...
Milter Filter – j-chkmail

- Detecting dangerous attachments (.exe, .pif, ...)
- Dumb content filtering – almost 1100 regular expressions (URLs most of the time) of the kind:
  
  ```
  http://[^/]*herbalpills[^/]*(biz|com|net)
  http://[^/]*[0-9]{1,3}\.\.[0-9]{1,3}\.\.[0-9]{1,3}
  ```

- Connection rate / open connections per gateway

- Other behaviour checking: gateways doing empty connections, gateways with bad DNS resolution (connections quota), ...

- Real time monitoring
Monitoring with the filter

- Milter is a very nice tool to monitor sendmail
- To know what the filter is doing is very important!
  - Filter may reject/replace messages (know why)
  - Evaluate parameters settings
  - Evaluate filter efficiency
  - Experiment new ideas (development)
Monitoring

● Which data
  ✓ Log (syslog) : connections and all rejections
  ✓ Context/results of connections saved on file (circular buffer)
  ✓ Filter status/counters periodically dumped to files
  ✓ Special events dumped to files (x-files, content reject, ...)

● Tools
  ✓ Command line : j-printstats
  ✓ rrdtool

● Currently moving from “periodical dumps” to “shared memory” (mmap, shm...)
Doing connection rate control - why ?
Doing connection rate control – why?

1) Detect and block SPAM
2) Detect Attacks
3) Protect server against resources exhaustion
Detecting and blocking SPAM

- Spammers goal:
  - Distribute as much messages as possible
  - As quick as possible – don’t loose time, as time is money!
  - Don’t damage mailserver – at least, not before ending spam distribution

- How to detect?
  - Content checking
  - Client behaviour checking
    - Connection rate control
    - Gateways doing dictionary attacks
    - RBLs
    - ...

Detecting attacks

- Attacker goal:
  - DoS – deny of service
  - Misuse of server resources

- Detection of “clients” abnormal behaviour
  - High connection rate
  - Too much open connections
  - Connections lasting open too long time
  - Resources being exhausted
  - Too much system errors
  - ...

Protecting server resources

● SPAM may not be a problem
● High connection rate isn’t a problem if the server is able to handle it.
● GOAL: No matter what’s going on, keep your server up and running!
● How to:
  ✔ continuously evaluate server load (CPU, number of processes, network, ...)
  ✔ If high,
    • selectively reject connections, based on how each client contributes to the server load
  ✔ If too high
    • stop accepting connections.
Protecting server resources – how to

How clients contribute to server load (idea to test):

- Compute, for each client, his contribution to:
  - CPU load
    - Connection rate * mean service time
  - Blocked resources (processes, threads, file descriptors)
    - Connection rate * mean connection duration (stay time)
    - Open connections – (files descriptors, processes) – instantaneous values
  - Others: mean message size, ...
- Sort contributions by values in decreasing order.
- Rejection rate, for each IP, shall be proportional to its contribution.

Question: how to evaluate optimal global rejection ratio?

N.B. - Compute values at two scales (1 and 10 min) and compare them.
Protecting server resources

- What's in being done on j-chkmail
  - For each connection:
    - Evaluate system load (or system idleness)
      - CPU load (vmstat, getloadavg), file descriptors, ...
    - Reject if Load > L2 (say 90 %)
    - Accept if Load < L1 (say 60 %)
    - If (L1 < Load < L2)
      - Accept if Trusted client
      - Evaluate how this client contributes to server load
      - Reject if too high
      - Else accept

"reject" means: "reject connection" or "don't do heavy processing"
So, what to do for what?

<table>
<thead>
<tr>
<th></th>
<th>Connection Rate</th>
<th>Open Connections</th>
<th>Behaviour</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fighting Spam</td>
<td>++</td>
<td>-</td>
<td>++</td>
</tr>
<tr>
<td>Detecting Attacks</td>
<td>++</td>
<td>++</td>
<td>++</td>
</tr>
<tr>
<td>Protecting Server Resources</td>
<td>++</td>
<td>++</td>
<td>+</td>
</tr>
</tbody>
</table>

- Compute global and per IP values
- Connection rate isn't enough – better adding other parameters (length of connections, service time, message sizes, ...)
Events and results...
**1st event**

- Jun 2002 – dictionary attack lasting for two months
  - ✔ Usual connection rate value increased from 1200/h to 3-5000/h with some 10000/h peaks
  - ✔ Connections coming from many varying gateways...
- What we've done:
  - ✔ First of all, use sendmail options (BadRcptThrottle, access database, local RBL, ...)
  - ✔ Limit, on the filter, connection rate to 10 connections / 10 minutes per IP address
2\textsuperscript{nd} event

- Nov 2002 –
  - 2000 connections in 2 minutes – file descriptors exhaustion
  - Filter died but relaunched by supervisor

- What we've done:
  - Count the number of available file descriptors and stop accepting connections if less than 50 (unknown clients) or 20 (everyone)
  - Problem: resource consuming (getrusage doesn't give this information)

  - Alternative: replace counting file descriptors by counting open connections and assume
    - \#file descriptors in use equals 2 * \#open connections (valid for j-chkmail)
2\textsuperscript{nd} event – typical exploit

- Filter will die just after when all the file descriptors will be in use (16 consecutive errors)
- Sendmail will stop accepting connections when MaxDaemonChildren will be reached

<table>
<thead>
<tr>
<th>forever</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Launch connection &amp;</td>
</tr>
<tr>
<td>● Launch connection &amp;</td>
</tr>
<tr>
<td>● Sleep 1</td>
</tr>
<tr>
<td>end forever</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>forever</th>
</tr>
</thead>
<tbody>
<tr>
<td>TELNET victim 25</td>
</tr>
<tr>
<td>WAIT OK</td>
</tr>
<tr>
<td>SEND HELO</td>
</tr>
<tr>
<td>WAIT OK</td>
</tr>
<tr>
<td>SEND MAIL FROM</td>
</tr>
<tr>
<td>WAIT OK</td>
</tr>
<tr>
<td>SEND RCPT TO</td>
</tr>
<tr>
<td>WAIT OK</td>
</tr>
<tr>
<td>SEND DATA</td>
</tr>
<tr>
<td>WAIT OK</td>
</tr>
<tr>
<td>forever</td>
</tr>
<tr>
<td>SEND DUMMY LINE</td>
</tr>
<tr>
<td>SLEEP some time</td>
</tr>
<tr>
<td>end forever</td>
</tr>
</tbody>
</table>
3rd event – fast distributed spam

April 2003

- 10536 connections in 8 minutes
- 238 gateways from network 66.216.119.0/24 (rapiddealsbyemail.com)
- Connections per gateway: [28 – 67]
- Peak: 86 connections in the same second (21:48:29)
- Connections rejected by contents: 15 (on the first 3 minutes)
  In this case: http://[^/]*greatedeals.com
- Connections rejected by connection rate: 8156
- All other connections are empty (User unknown...)
- No legitimate message was blocked during the attack – Goooooood!
3rd event – Histogram of connections

Connection / hour - Apr 10, 2003

Sendmail
Filter
### 3rd event – Histogram of connections

#### 12h – 23h

<table>
<thead>
<tr>
<th>Hour</th>
<th>sm</th>
<th>filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1292</td>
<td>969</td>
</tr>
<tr>
<td>13</td>
<td>1241</td>
<td>991</td>
</tr>
<tr>
<td>14</td>
<td>1191</td>
<td>1096</td>
</tr>
<tr>
<td>15</td>
<td>1168</td>
<td>1092</td>
</tr>
<tr>
<td>16</td>
<td>1429</td>
<td>1369</td>
</tr>
<tr>
<td>17</td>
<td>1474</td>
<td>1388</td>
</tr>
<tr>
<td>18</td>
<td>1168</td>
<td>1110</td>
</tr>
<tr>
<td>19</td>
<td>1023</td>
<td>915</td>
</tr>
<tr>
<td>20</td>
<td>897</td>
<td>768</td>
</tr>
<tr>
<td>21</td>
<td>11214</td>
<td>11136</td>
</tr>
<tr>
<td>22</td>
<td>913</td>
<td>813</td>
</tr>
<tr>
<td>23</td>
<td>737</td>
<td>674</td>
</tr>
</tbody>
</table>

#### 21h - 22h

<table>
<thead>
<tr>
<th>Hour</th>
<th>sm</th>
<th>filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>21:00</td>
<td>115</td>
<td>105</td>
</tr>
<tr>
<td>21:10</td>
<td>110</td>
<td>102</td>
</tr>
<tr>
<td>21:20</td>
<td>103</td>
<td>83</td>
</tr>
<tr>
<td>21:30</td>
<td>127</td>
<td>112</td>
</tr>
<tr>
<td>21:40</td>
<td>10650</td>
<td>10637</td>
</tr>
<tr>
<td>21:50</td>
<td>109</td>
<td>97</td>
</tr>
</tbody>
</table>

#### 21h40 - 21h50

<table>
<thead>
<tr>
<th>Hour</th>
<th>sm</th>
<th>filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>21:40</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>21:41</td>
<td>159</td>
<td>159</td>
</tr>
<tr>
<td>21:42</td>
<td>1081</td>
<td>1079</td>
</tr>
<tr>
<td>21:43</td>
<td>1160</td>
<td>1156</td>
</tr>
<tr>
<td>21:44</td>
<td>870</td>
<td>872</td>
</tr>
<tr>
<td>21:45</td>
<td>1061</td>
<td>1058</td>
</tr>
<tr>
<td>21:46</td>
<td>1815</td>
<td>1801</td>
</tr>
<tr>
<td>21:47</td>
<td>2481</td>
<td>2462</td>
</tr>
<tr>
<td>21:49</td>
<td>35</td>
<td>33</td>
</tr>
</tbody>
</table>

Who’s suffering?  
Sendmail or filter?  
Or client?

---

Jose-Marcio.Martins@ensmp.fr  
MotM – Sendmail – 31/07/2003
**3rd event – Connection rate control results**

Connect/reject from 66.216.119.0/24  
(measured by the filter)

<table>
<thead>
<tr>
<th></th>
<th>Connect</th>
<th>Reject</th>
</tr>
</thead>
<tbody>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>41</td>
<td>149</td>
<td>0</td>
</tr>
<tr>
<td>42</td>
<td>1072</td>
<td>3</td>
</tr>
<tr>
<td>43</td>
<td>1145</td>
<td>355</td>
</tr>
<tr>
<td>44</td>
<td>860</td>
<td>587</td>
</tr>
<tr>
<td>45</td>
<td>1047</td>
<td>959</td>
</tr>
<tr>
<td>46</td>
<td>1790</td>
<td>1764</td>
</tr>
<tr>
<td>47</td>
<td>2452</td>
<td>2464</td>
</tr>
<tr>
<td>48</td>
<td>1998</td>
<td>2001</td>
</tr>
<tr>
<td>49</td>
<td>23</td>
<td>23</td>
</tr>
</tbody>
</table>

![Graph showing Connect/Reject data](image-url)
3rd event – Detection of attack

An indicator to detect attacks

- Compute, for each minute, the number of connections on the past 16 minutes
- Compute the mean and standard deviation of the data set
- Compare the number of connections on the last minute with mean and std dev
- If the number of connections in some time interval is assumed to be a random variable with poisson distribution law,
  
  $P[X > (\text{mean} + 3.5 \times \text{stddev})] < 0.001$  
  - stddev = sqrt(\text{mean})

This seems to be true, if rate > 20 connections/minute and bucket size near 1 minute

Apr 10 21:42:00 paris j-chkmail[4444]: *** DoS - THROTTLE : 10.933 3.127 161.000
3rd event – What if not?

**System point of view**

- Assume mean connection time equals 10 s (usually between 5s and 20s)
- Mean connection rate around 20/s
  - 200 sendmail processes – 200 milter threads (400 LWPs)
- Peak connection rate around 80/s
  - 800 sendmail processes – 800 milter threads (1600 LWPs)

OBS: Little's law says: \[ N = (\text{arrival rate}) \times (\text{stay time}) \]
3rd event – What if not?

Filter point of view...

- If the filter was alone on the system...
- Stability condition:
  - arrival rate * (service time / number of processors) < 1
    - Mean 20/s * (0.050 s / 2) = 0.5  OK!
    - Max 80/s * (0.050 s / 2) = 2     KO!
- But the filter isn't alone: sendmail (handling the same load) and bind...
3rd event – What can we learn from this...

- Connection rate control stops connection at early states of protocol avoiding “heavier” processing (checking contents, ...)
- Connection rate control
  - detected the spam
  - detected the attack
  - protected server resources, but if the attack continued some minutes more, the system could probably be down, as this is a blind “Open loop control”
- Closed loop control shall include global server load (feedback) and selectively reject TCP connections when the load becomes too high (no nullserver)
- Connection rate control isn't enough, but is better than nothing!
Conclusions
Connection rate control to stop spam

- Usually 500 – 1000 spam connections rejected each day – usually much more during weekends
- Very effective to stop attacks and spam coming from robots
- Currently, very few false positives
- Close unwanted connections at early states of connection, avoiding heavy processing (content check), and allowing server to handle higher connection rates
- Block some external “friends” (list servers, friend domains)
- Solution – class IP networks by trust level: “local”, “friends” and “unknown” and set different values to each class
- List all known friends – not a real problem for most sites
Connection rate control for sendmail 8

● Connection Rate Control
  ✓ Patch available at http://j-chkmail.ensmp.fr/sm
  ✓ Reject by sendmail (C code) with configured max values
  ✓ Reject by LOCAL_RULES (contrib by Stephane Lentz)

● Open Connections control
  ✓ Thinking about
  ✓ Shall track sendmail processes (fork and die)
    ● Probable best solution by integrating the client address to struct PROCS_T (util.c), and the client address as parameter to proc_list_add call
    ● The number of open connections for a particular client can be obtained by counting is the number of components with his address on the array ProcListVec
**Connection rate control and RFC2821**

- sendmail/filter knows at connection time what will happen but shall wait till “EHLO/MAIL” to send a reject code and close the connection (if the client sends a quit)
- Danger! Client drives the connection but might be naughty (no QUIT command to close the connection) – connection will be closed only after a long timeout.
- No problem if client ask server to close the connection
- Solutions:
  - ✔ Send the reject code and close connection without waiting for client QUIT command – violates RFC 2821
  - ✔ Two connection rate levels
    - ✔ If L1 < rate < L2 - respect RFC2821 and send smtp reject code (nullserver)
    - ✔ If rate > L2 reject TCP connection (client already received L2 – L1 rejects)
      - don't launch nullserver
• Closing unwanted connections at early states of SMTP protocol avoid heavy message processing and allow server to handle higher level traffic

• Connection rate control and other “behavioral” criterias are interesting ways to explore