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}\.[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 it's 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 ?

	Connection Rate	Open Connections	Behaviour	
			Client	Server
Fighting Spam	++	-	++	
Detecting Attacks	++	++	++	+
Protecting Server Resources	++	++	+	++

- ✔ 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

2nd 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)

2nd event – typical exploit

forever

- Launch connection &
- Launch connection &
- Sleep 1

end forever

- 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

```
TELNET victim 25
WAIT OK
SEND HELO
WAIT OK
SEND MAIL FROM
WAIT OK
SEND RCPT TO
WAIT OK
SEND DATA
WAIT OK
forever
   SEND DUMMY LINE
   SLEEP some time
end forever
```

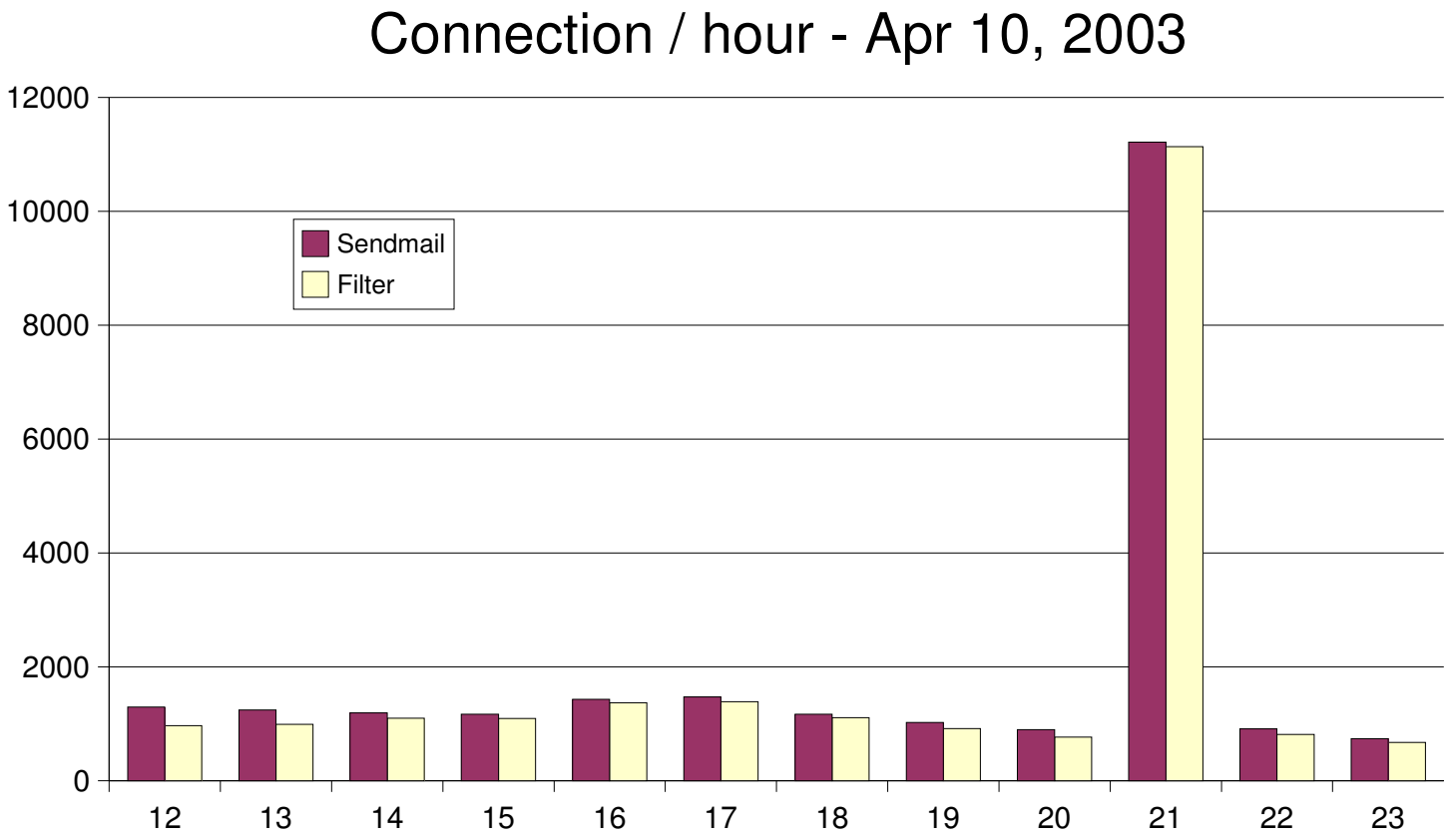
3rd event – fast distributed spam

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 – Gooood !**

3rd event – Histogram of connections



3rd event – Histogram of connections

12h – 23h

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

21h - 22h

Hour	sm	filter
21:00	115	105
21:10	110	102
21:20	103	83
21:30	127	112
21:40	10650	10637
21:50	109	97

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

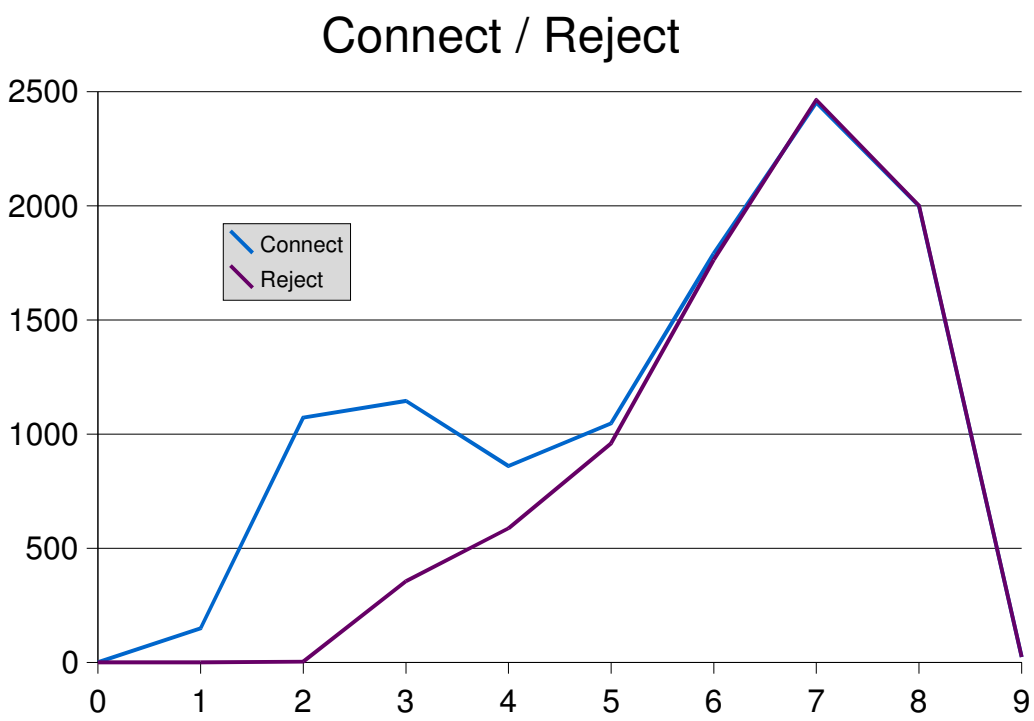
21h40 - 21h50

Hour	sm	filter
21:40	10	9
21:41	159	159
21:42	1081	1079
21:43	1160	1156
21:44	870	872
21:45	1061	1058
21:46	1815	1801
21:47	2481	2462
21:48	**1978**	**2008**
21:49	35	33

3rd event – Connection rate control results

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

21h	Connect	Reject
40	0	0
41	149	0
42	1072	3
43	1145	355
44	860	587
45	1047	959
46	1790	1764
47	**2452**	**2464**
48	**1998**	**2001**
49	23	23



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 > (mean + 3.5 * stddev)] < 0.001 - stddev = sqrt(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
Apr 10 21:49:39 paris j-chkmail[4444]: *** DoS - THROTTLE : END

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 = (arrival rate) * (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**