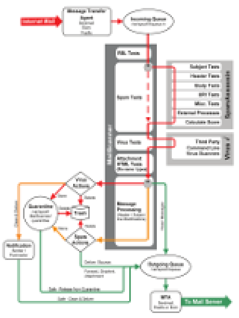
Setup of a clustered antispam and antivirus service based on Mailcleaner®suite[†].

Andrea Lora,[a] Luca Ianniello,[a] Augusto Pifferi.[a]



In order to provide an effective antispam and antivirus service to support the email servers delivered at C.N.R. - Area della Ricerca RM1 we setup a clustered installation of Mailcleaner, an open source suite based on Mailscanner. Here we will describe the setup activity and the configuration of all components. For more information about Mailcleaner and Mailscanner refer to the websites of both projects[1,2].

Keywords: Antispam, antivirus, spamassassin, mailcleaner.

1 Introduction

On late summer 2012 the research campus "Area della Ricerca RM1" of C.N.R. didn't yet had its own email server, all the research institute inside the Area were using an external email server hosted at "Area della Ricerca di Tor Vergata". Because of the growth of the number of used mailboxes, we started to deploy a new email server. The first step was to evaluate different open source antispam and antivirus services available on the web.

We choose Mailcleaner, for its detailed ajax admin interface, its clustering capabilities, the availability of a quarantine area and because it's open source!

2 Architecture

We planned to setup a redundant architecture dual-frontend dual-backend, in order to implement HA (High Availability) and load balancing. The redundant design allows us to operate maintenance of a single machine without putting the entire system offline. At frontend layer we simply exploit the capability of DNS to define more than one MX records for each domain, in case of system outage of the first server, email delivery is granted via the second one. At backend layer we used load balancing techniques issued by our firewall. The overall architecture is shown in Fig. 1

The balancer is not a single point of failure (SPoF) because it's implemented with two nodes, ensuring maximum availability, on the other hand the balancing is not configured per session but per source ip address, so often a mailhub keeps many more sessions than the other one, as shown in Tab. 1.

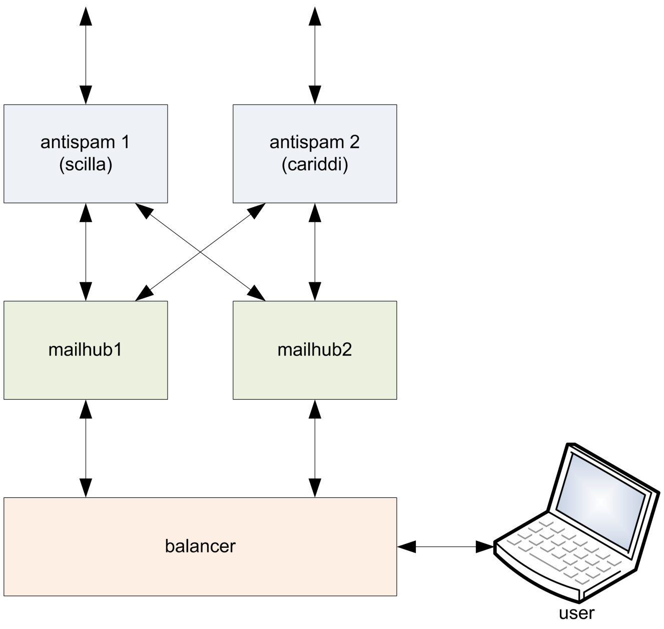


Fig. 1 Overall architecture

3 Installation

Mailcleaner is distributed as a self-installable distribution, based on Debian with latest cutting-edge kernel re-

[a] *CNR - Istituto di Cristallografia, Strada Provinciale 35/d, Montelibretti, Italia*

† rapporto tecnico IC 13/10 registrato con numero di protocollo IC/1880 del 20/12/2013

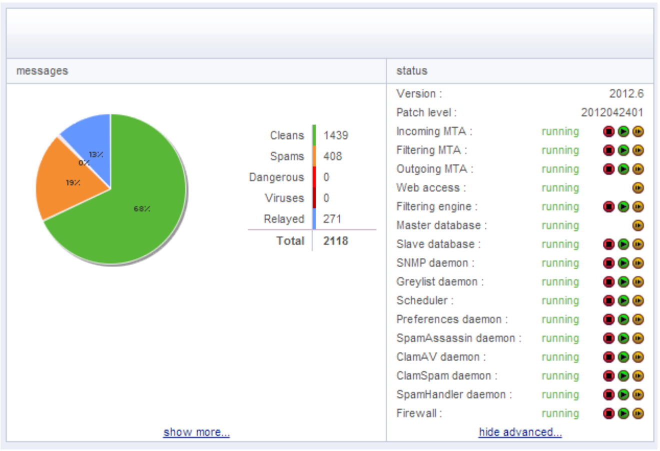
`root@mailhub1:~# vnstat`	`root@mailhub2:~# vnstat`
`Database updated: Mon Oct 28`	`Database updated: Mon Oct 28`
`10:47:48 2013`	`10:47:44 2013`
`eth0 since 10/01/13`	`eth0 since 10/01/13`
`rx: 86.68 GiB`	`rx: 152.85 GiB`
`tx: 45.48 GiB`	`tx: 96.05 GiB`
`total: 132.16 GiB}`	`total: 248.90 GiB`

Table 1 Network usage.

lease. The setup process is quite easy. Once installed on the first machine we installed it again on a second machine in order to setup a clustered deployment.

Clustering feature is really powerful and well described on the official wiki.

Once finished, the system is up and running, the only package we had to install as separate process is ClamSpam. We installed it using the unofficial signatures released by Sanesecurity.[3] A status window of the overall system is immediately available, see Fig. 2. It is useful to get an overview on the status of processes, messages handled, number of spam or dangerous messages received and resources used.



Cleans	1439
Spams	408
Dangerous	0
Viruses	0
Relayed	271
Total	2118

Version :	2012.6	
Patch level :	2012042401	
Incoming MTA :	running	● ▶ ⏯
Filtering MTA :	running	● ▶ ⏯
Outgoing MTA :	running	● ▶ ⏯
Web access :	running	⏯
Filtering engine :	running	● ▶ ⏯
Master database :	running	⏯
Slave database :	running	● ▶ ⏯
SNMP daemon :	running	● ▶ ⏯
Greylist daemon :	running	● ▶ ⏯
Scheduler :	running	● ▶ ⏯
Preferences daemon :	running	● ▶ ⏯
SpamAssassin daemon :	running	● ▶ ⏯
ClamAV daemon :	running	● ▶ ⏯
ClamSpam daemon :	running	● ▶ ⏯
SpamHandler daemon :	running	● ▶ ⏯
Firewall :	running	● ▶ ⏯

Fig. 2 Status windows.

4 Initial Configuration

The first configurations include IP addressing, DNS and time server configuration. Then we started to configure all those domains whose antispam and content filtering system should be on our machines. We called our servers Scilla and Cariddi, as the two epic monsters which generated storms between Sicily and Italy destroying passing ships.

Mailcleaner sets a value for spam score of an email, and when it's greater than the spam threshold, it is possible to take different actions: tag, quarantine or delete the message. We opt to simply tag email and deliver it, in order to avoid that false positives can cause excessive delay in email delivery or loss of messages. Quarantine action will be taken into account in case of a dangerous content inside the message.

This action will isolate dangerous messages inside the Mailcleaner system, avoiding to deliver them to user's mailbox. In the meanwhile the user receives a notification that messages are retained in his quarantine area, so he can decide to review and optionally relay them.

Another feature we choose to not use is greylisting. This is a method to defend email users by spam. A Mail Transfer Agent (MTA) using greylisting will "temporarily reject" any email from a sender it does not recognize. If the email is legitimate the originating server will try again o deliver it after a delay and, if sufficient time has elapsed, the email will be accepted. However this introduce an unacceptable delay that we choose to avoid.

In order to secure all connections to web interface for administrators and users while they browse their quarantine, we also installed a SSL certificate.

5 Description of Components.

To understand the architecture of mailcleaner we need to describe the activities to be done by a spam server to do its job.

The first task is to receive the email from the external sources. This is the reason why the spam servers are often configured as MX records for a domain. During this task the spam server stores the incoming email into a queue. The second task is to check the queue of inbound messages and process them. At this stage further actions are evaluated for the spam retention. The third task is to delivery the processed message to the last server.

The first task is accomplished by the incoming queue, managed from an instance of the Exim daemon named Stage1, it is responsible for accepting messages received from an external internet MTA and verifying some basic rules, such as consistency of sender domain and recipient mailbox. Stage1 can also operate some network checks, for example it can check if the sender MTA is publically blacklisted.

The second task is carried out by the Filtering queue, another instance of Exim named Stage2. It's probably the most important process because it must check the content of every received email in order to mark spam messages. This decision is taken through different algorithms that take into account lots of parameters such as known signatures, blacklists checks or content of the body message of the email. We'll look into that deeply in chap. VI.

The last task is the final delivery. Another instance of Exim, named Stage4, is responsible of delivering messages to the recipient email servers according to the result of the filtering action.

6 The Filtering Stage

It is possible to apply up to six different filters to distinguish spam from ham messages, which can be ordered according to settings applied by administrators.

1. Trusted sources: filter that accept email originated by specific trusted SMTP servers or SPF capable domains. It can also use information from DNS white list services.
2. Bayesian filters: the one used by mailcleaner is Bogofilter. It is an email filter that classifies messages as spam or ham (non-spam) by a statistical analysis of the message's header and content (body). The program is able to learn from the user's classifications and corrections.
3. Clamspam: it is a signature-based filter that can identify phishing attempts, dangerous links or contents, malware, fakes and other generic spam. We used signatures from Sanesecurity.[3]
4. PreRBL: it is a Real-time Blackhole Lists system based on a network of databases containing IP addresses of servers used to originate or forward spam messages. Lists are kept up to date in order to avoid false positives, however our implementation only add a hit (positive value on the overall score) when a RBL match is found.
5. UriRBL: it's another Real-time Blackhole Lists system, based on databases of web sites uri that have appeared in unsolicited messages. Unlike most lists, UriRBLs are not lists of message senders, because web sites seen in unsolicited messages tend to be more stable than the rapidly changing botnet IP addresses used to send the vast majority of them.
6. Spamassassin: it's probably the most strong engine for spam filtering. It applies lots of checks on incoming messages including RBLs previously discussed and:
    - Distributed Checksum Clearinghouse (DCC), Razor and Pyzor, three similar Hash-Sharing Systems. Every time a spam message is detected, a checksum of the message is calculated and reported to an online database. When an email server queries that online database about the hash of a message, it compares the hash with all strings contained and returns a result.
    - SPF: Sender Policy Framework (SPF) is an email validation system designed to prevent spam by detecting email spoofing, a common vulnerability, by verifying sender IP addresses. SPF allows administrators to specify which hosts are allowed to send email from a given domain by creating a specific SPF record (or TXT record) in the Domain Name System (DNS). Email exchangers use the DNS to check that email from a given domain is being sent by a host sanctioned by that domain's administrators. Sender Policy Framework is defined in IETF publication RFC 4408.[4]
    - DKIM: DomainKeys Identified Mail (DKIM) is a method for associating a domain name with an email message, thereby allowing a person, role, or organization to claim some responsibility for the message. The association is set up by means of a digital signature which can be validated by recipients. Responsibility is claimed by a signer, independently of the message's actual authors or recipients, by adding a DKIM-Signature: field to the message's header. The verifier recovers the signer's public key using the DNS, and then verifies that the signature matches the actual message's content. DomainKeys Identified Mail is defined in IETF publication RFC 4871.[5]

6.1 Antivirus and Content Protection.

Mailcleaner can also process every inbound and outbound email message for virus identification and emoval. Many antivirus software can be configured inside Mailcleaner, we choose to use the default one, Clam AV. Configuration is quite simple, automatic update is executed by default every hour in order to ensure maximum protection against virus threats.

Mailcleaner can also check for IFrame objects, web bugs and other malformed HTML inside the body of the emails and also scan the content of a compressed archive. In case of a positive the infected email will be cleaned (attachment removed and html sanitized) and a notification will be delivered to the recipient. The infected but cleaned message will be kept in quarantine for 15 days, so that recipients can check it and, optionally, release from quarantine to their mailboxes.

A flow chart of every single email that cross our servers is reported in Figure 3.

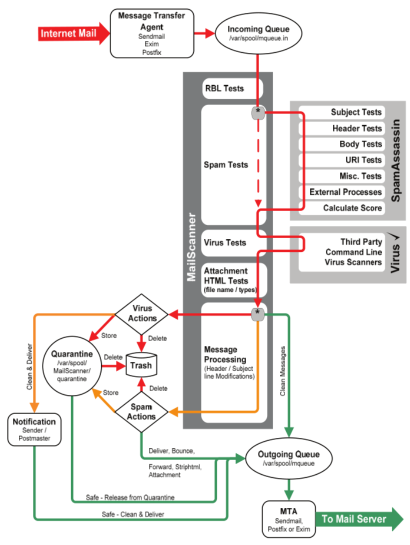


Fig. 3 Mail flow inside mailcleaner.

7 Custom Filters Setup

It is possible to fully customize all the settings of the system, included which checks operate and in which stage apply them. In order to avoid a too aggressive behavior against external servers not well configured or not well maintained, we choose to keep a more conservative, permissive profile.

RBL checks can be applied both at incoming or filtering stage, the main difference is that at incoming stage a message delivered from a blacklisted server is immediately rejected, while if operating the same check at filtering stage will result as a score assigned to every single message received. If the overall score of the message is greater than the system threshold then the message will be marked as spam, but delivered to the recipient.

We applied all the dynamic checks at spamassassin level, while, at incoming stage, we only check for some basic tests, such as the existence of the sender domain and the delivery mailbox.

8 Interaction with the Email Server

There are many interactions between Scilla and Cariddi and the email servers they deliver to.

First of all the way to balance all messages passing from a system to the other: mailcleaner allows to set many delivery server for any domain, so we added both mailhub1 and mailhub2 in load-balancing mode; inside the two hubs we installed HA-proxy, a load balancing daemon, in order to balance also the outbound traffic to the content filtering system.

The content filtering system is also used to scan outgoing messages, protecting our server from being blacklisted worldwide; users access their quarantine areas and their personal set of antispam rules authenticating themselves via an IMAP callout to the email servers; messages whose recipient does not exists are immediately rejected, this is possible via a SMTP query to the servers.

Another interaction between those two systems is about the notification of false positives and false negatives. Every user can notify the administrator that a received message has been incorrectly rated, so that administrators can instruct the bayesian filter (bogofilter) to correctly detect messages similar to the reported one.

This system has been also used to manually adjust some filters in order to avoid false positives, i.e. a spamassassin score is assigned every time the “soma” word or some variations of the word occurs in the body of an email. Unfortunately the italian word “somma” means “sum”, so we had lots of false positive matching of that rule. We identified that mistake and correct it manually lowering the score of the FRT_SOMA rule.

9 The Learning System

As described before in this document Mailcleaner includes Bogofilter as a statistical analysis filter against spam. The statistical technique used by Bogofilter is known as the bayesian technique and its use for spam detection was described by Graham and Robinson.[6–8]

Configuring Bogofilter requires continuous training with huge amount of both spam and ham messages. At the present we didn’t configure an effective fully-automated learning system, it will be argument for further enhancements of the system.

We would like to implement a system that automatically learn as ham any messages reported by user as false positives, and learn as spam messages reported asfalse negatives. Furthermore it should learn as spam (ham) messages whose spamassassin score is greater (less) than a threshold.

10 Monitoring

Mailcleaner provides itself a built-in console to overview the status of the whole system, including cpu load, disk and memory usage, number of messages in spools, status of all vital processes and a pie diagram that represents the number of messages processed by the system.

All elements of the architecture are automatically monitored via SNMP by our Zabbix platform,[9] in order to identify an issue as soon as possible helping troubleshooting and limiting the downtime and unavailability of a service.

At the present we monitor the number of messages in all three queues, the reachability of both servers and the usage of resources.

The most important OID to analyze that way is 1.3.6.1.4.1.2021.8.1.101.6. It returns an integer list containing the number of queued messages for the three stages (e.g .:|190|4|26). Since the information about queues is relative to the local server we should remind to monitor both servers in the Mailcleaner cluster.

Choosing alert thresholds isn’t immediate due to the different job every queue is in charge of. Incoming queue can grow larger after massive attacks or when distribution list messages are sent. Filtering spool contains messages processed by the antispam and the antivirus engines. These jobs are heavily cpu and io related so this queue can grow up easily if the system is in resources starvation. Outgoing spool is in charge to deliver to the final destination the email. If the messages cannot be delivered immediately (e.g. temporary failure of destination host) they are kept here.

11 Conclusion

An effective antispam and content filtering system is the base of a well working email server architecture, because on one hand it keeps users safe from virus and free from spam, on the other hand it makes possible to save storage space and internet bandwidth, limiting the amount of resources wasted to handle useless messages.

References

1 Mailcleaner official online documentation, available at http://www.mailcleaner.org/doku.php.

2 Mailscanner official online documentation, available at http://www.mailscanner.info/documentation.html.

3 Sanesecurity project website http://sanesecurity.com.

4 M. Wong, W. Schlitt, Sender Policy Framework (SPF) for Authorizing Use of Domains in E-Mail, Version 1, Request for Comments: 4408.
URL https://tools.ietf.org/html/rfc4408

5 E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, M. Thomas, DomainKeys Identified Mail (DKIM) Signatures, Request for Comments: 4871.
URL https://tools.ietf.org/html/rfc4871

6 P. Graham, "A plan for Spam", August 2002 .
URL http://www.paulgraham.com/spam.html

7 P. Graham, "Better Bayesian Filtering", January 2003.
URL http://www.paulgraham.com/better.html

8 G. Robinson, "Spam Detection", September 2002.
URL http://radio-weblogs.com/0101454/stories/2002/09/16/spamDetection.html

9 A. Pifferi, G. Nantista, L. Ianniello, A. Lora, M. Simonetti, Analisi e Implementazione di Sistemi per il Monitoraggio della Rete Wireless Relativa al Progetto ADD (Anti Digital Divide) e delle Infrastrutture di Campus AdR RM1., SMART eLAB 2 (2013) 1–9. doi:10.30441/smart-elab.v2i0.46.