Securing Electronic Mail on the National Research and Academic Network of Italy

Roberto Cecchini – INFN, Florence; *Fulvia Costa* – INFN, Padua;
Alberto D'Ambrosio – INFN, Turin; *Domenico Diacono* – INFN, Bari;
Giacomo Fazio – INAF, Palermo; *Antonio Forte* – INFN, Rome;
Matteo Genghini – IASF, Bologna; *Michele Michelotto* – INFN, Padua;
Ombretta Pinazza – INFN, Bologna; *Alfonso Sparano* – University of Salerno

ABSTRACT

Sec-mail is a group of site administrators in the GARR (Gruppo Armonizzazione Reti della Ricerca – Research Networks Harmonisation Group) network dedicated to the security of E-mail services. GARR is the National Research and Academic Network of Italy.

The main points covered are in: methodologies to improve the efficiency of spam detection (mainly tuning of SpamAssassin), definition of best practices in electronic mail administration, sender domain authentication, greylisting and spam monitoring.

Introduction

The National Research and Academic Network of Italy, or GARR, formed the working group SEC-MAIL to study IT security related problems on its network. This group was formed following a proposal by Roberto Cecchini at the V GARR Workshop held in Rome in November 2003. The group examines the following technologies:

- spam;
- viruses and worms spread up by E-Mail;
- best practices for mail-server configuration and security;
- authentication of sender mail-server;
- greylisting technologies;
- graphs and statistics.

This paper presents a summary of the group's findings to date. Since the fall of 2004, the group has its own web area and a wiki site [WSM], where both the results of the experiments and the produced documentation are made available.

Initially, the group focused its work mainly in SpamAssassin tuning in order to improve the spam detection by Bayesian filters, non-standard rules, and technologies based on mass-emailing distributed identification. Some experimental DCC (Distributed Checksum Clearinghouse) servers were set-up, and made available to the GARR community on a best-effort basis. Then, greylisting has been enabled on three pilot sites, and with an experimental technology in one of them. A huge amount of graphs and statistics has been produced which have surely helped the several experimental activities of our working group.

We will discuss our efforts to prevent spam and then discuss viruses and worms. This is followed by best practices, Sender domain authentication, greylisting technologies, and a conclusion.

Controlling Spam

Spam: A Security Problem

In the past years the number of unsolicited E-Mail messages coming in the mailboxes of users has grown from a nuisance to a real problem. Some sites report that less than 10 percent of their messages are good. Of course the problem has been transferred to the site administrator who has the control of the mail-server. The most expert user can cope with a self-made filter (procmail) and with the filter embedded in the Mail User Agent application (such as Outlook and Thunderbird).

The problem is not only in terms of time lost by users deleting spam messages, but also the resources needed in the mail-server, and the relation with security problems like viruses and worms used to send spam, scam, phishing and brute force address harvesting.

This is the background that led to the birth of a group of large site (thousand of mailboxes each) administrators in the GARR network. The group goals include the study of all security problems related to electronic mail with a clear urgency to deal with the spam explosion.

The Common Base: SpamAssassin

The group activity concentrated itself in understanding how it would be possible to improve the SpamAssassin (SA) efficiency by reducing negatives and (mainly) false positives.

SpamAssassin is based on heuristic tests using genetic algorithms, and automatic Bayesian statistical corrections. So, the spam nature of every single E-Mail message is determined in a statistical way. Thus, independently of how well tuned the configuration parameters might be, there will always be some positives (good messages erroneously tagged as spam) as well as some negatives (spam messages undetected).

When a message's "SpamAssassin score" is higher than a specific value, the message will be considered spam, and the administrator can decide its destiny: remove it, move it to a specific folder or, most commonly, just flag the E-Mail as probable spam by modifying the subject in order to leave its destiny in the users' hands.

A trivial method used to increase the number of messages detected as spam is either to decrease the spam score level, or increase the score value of some tests. This must be done carefully because, in the first case one will increase the probability of positives, while in the second case one will unbalance the score value between the hundreds of tests.

For these reasons, we decided to study independent methods to improve the filter efficiency, that is, increase the spam/ham distribution separation.

The Bayesian Classifier

The use of Bayesian filters is very effective. With this method, the filter learns the right rule from lists of "certain" spam and ham messages which have been classified by a human. The filter produces two tables of the most frequent words (actually tokens, since the message header is also analyzed) found both in spam and ham messages. After this learning phase, every incoming message is given a score, whose value is assigned depending on the spam/ham token frequency. This method is quite independent from the

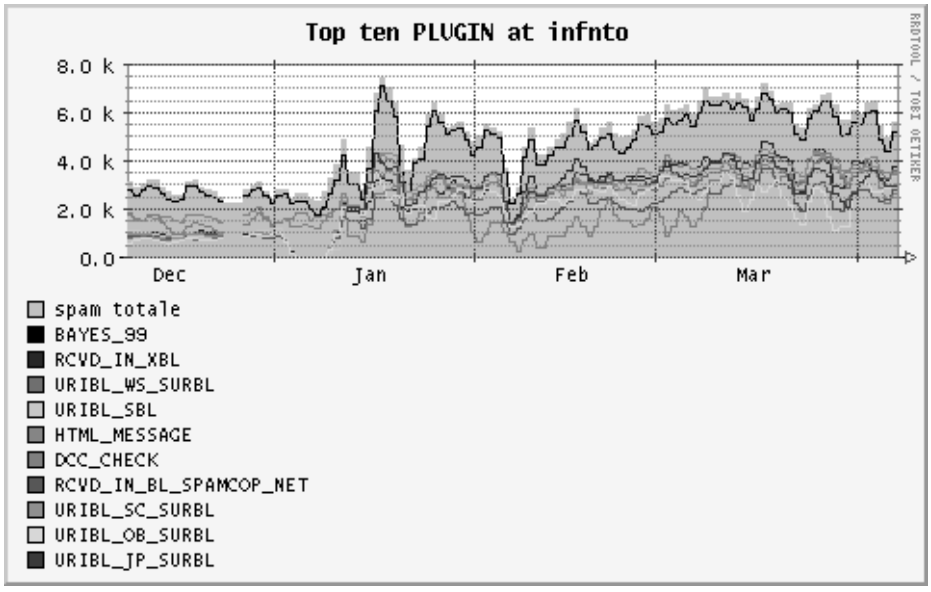

Figure 1: Usual trend for top ten SpamAssassin plug-ins at INFN-TO (Dec 2005-Apr 2006).

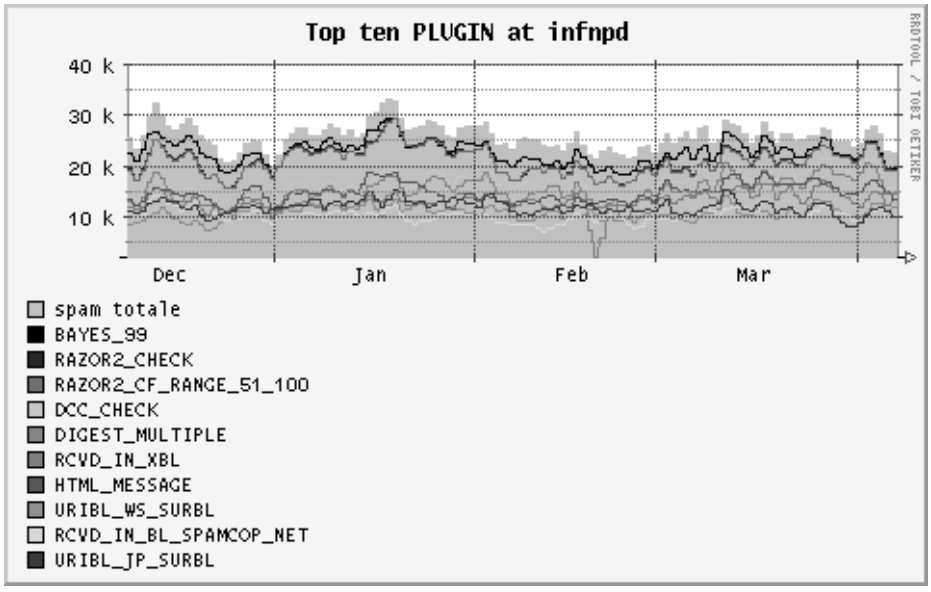

Figure 2: Usual trend for top ten SpamAssassin plug-ins at INFN-PD (Dec 2005-Apr 2006).

rule-driven one, because it also works on good messages and is customized for each mail-server target.

However, this method must be used carefully. The training of the bayesian classifier is a critical step. The efficacy of the bayesian filter is influenced by the spam/ham database size. It must be big enough to contain all relevant tokens, but not so big as to contain old (no longer relevant) tokens. Tests made in the Turin site gave the best results by lowering the parameter bayes_expiry_max_db_size from 200 to 100.

The auto-learning option can be exploited by spammers who send E-Mail messages designed to poison the DB with a huge amount of random ‘‘non spam’’ words. In this way, the Bayesian filter might even auto-poison its own DB.

The countermeasure is the DB correction, made by users’ feedback. Every day, each user can re-classify all received positives and alse negatives. Tests made in the Turin site have shown that the ‘‘right’’ DB strongly helps in this reclassification of spam/ham messages. For example, in Turin the best results have been achieved with a central DB, so that the spam is targeted on a whole site basis instead of a per-user basis.

The training of the Bayesian filter on the users feedback gave excellent results. As shown in Figure 1, the Bayesian solid test line for the BAYES_99 (99% probability of being a spam) is very close to the total spam detected line.

At the Padua site the above reclassification is made by the site administrator by manually feeding the DB with selected ham & spam. In this case the two lines are a little bit more separated (see Figure 2). Results are even worse at other sites that still haven’t applied any kind of statistical correction.

We’ve set up a monitoring system for the efficiency of the several tests used by SpamAssassin (see Figure 3 and Table 1, referred to one site only). Table 1 represents, for each plug-in:

- score: the single plug-in score assigned by SpamAssassin;
- score %: plug-in score percentage compared with the required score;
- hit: the number of E-Mail messages tagged as spam;
- hit %: percentage of E-Mail messages tagged as spam by the plug-in;

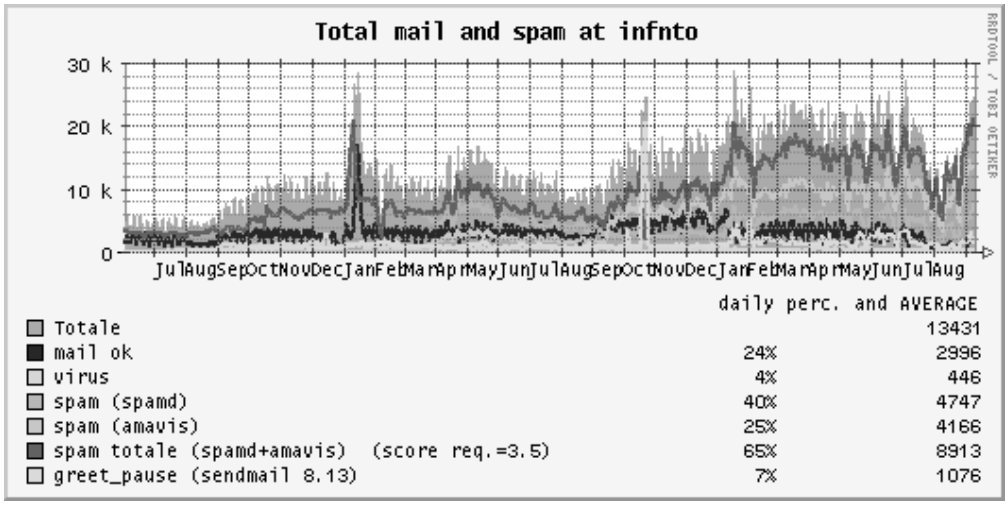

Figure 3: Number of E-Mail messages received at INFN-TO from June 2004 to September 2006.

Plug-in name	Score	Score %	Hit	Hit %
BAYES_99	4.070	116%	5640	94.0%
RCVD_IN_XBL	3.897	111%	4163	69.4%
URIBL_WS_SURBL	2.140	61%	3725	62.1%
URIBL_SBL	1.639	46%	3534	58.9%
HTML_MESSAGE	0.001	0%	3244	54.1%
DCC_CHECK	3.500	100%	3047	50.8%
RCVD_IN_BL_SPAMCOP_NET	3.500	100%	3040	50.7%
URIBL_SC_SURBL	4.498	128%	3035	50.6%
URIBL_OB_SURBL	3.008	85%	3023	50.4%
URIBL_JP_SURBL	4.087	116%	2929	48.8%

Table 1: Numeric values at INFN-TO referred to the 2006-04-07 log.

Switching SpamAssassin from release 2 to release 3 dramatically improved the spam detection ability of the filter, especially with releases 3.1.x. The Bologna site experienced a false negative reduction from 20% to 1%, and a false positive reduction from 0.8% to 0.2%.

Non-Standard Rules

In addition to the standard rules, SpamAssassin can use non-standard plug-ins to further improve its spam detection ability. The Florence site experimented with both the URIBL and SARE families of plug-ins (see Figure 4).

The URIBL family (a sort of distributed blacklist, included by default in the SpamAssassin package, starting rel. 3) is based on the harvest of sites linked from URL's internal to spam messages. Because in most cases, while message headers don't indicate where spam actually comes from, the message body must contain links useful for the spammers. We have found it very useful, especially with SpamAssassin rel.2, where it wasn't included by default.

The SARE family (still not standard) gathers different countermeasures against brand-new spamming techniques. For example, we found the gibberish plug-in (against random words inside the body of some spam messages) to be very useful.

Among non-standard plug-ins we also considered mail-scanners, which are high-performance and reliable interfaces between mail transport agents (MTA) and one or more content checkers. They perform a light anti-spam scan before the final call at SpamAssassin.

Using Mail-Scanners

Special behaviors may be obtained by using the aforementioned mail-scanners. Since the beginning of the group effort, the Turin site has been using the AMaViS [AMA] mail-scanner for both anti-virus and anti-spam filters. The Florence site, instead, developed its own mail-scanner (RJSPAM) [RJS]. However, both sites use these mail-scanners to reject spam depending on the results of SpamAssassin tests: in Turin for those users that explicitly asked for this behavior (*opt-in*), while in Florence for those users that still haven't explicitly refused it (*opt-out*).

When one receives hundreds of spam messages per day, hijacking them into special folders is completely useless: one will never check them! So, rejecting them might be a wise solution because a spam message is a sender mail-server object, not of the receiver server!

Another positive effect is in terms of performance. With mail-scanners, the scanning takes about a tenth of the time of the usual anti-virus and anti-spam software, since it interacts directly with the libraries, without calling any (slow) executables.

Finally, we suspect there is a useful side-effect: rejecting spam will cause one's (spammed) address to be removed from the spammers' mailing lists! But this behavior has yet to be confirmed (we need further collecting time).

However, the working group members didn't come to a common agreement on the policy of rejecting such messages. Some members think it is unethical, even though it respects all RFC's. Others, instead, simply don't like it.

RBL

Another independent tagging system is the one based on the RealTime Block List [RBL]. The sender E-Mail address is compared with a notorious spammers' database by a DNS query. These databases are maintained by user communities that collect the names of Internet Service Providers housing spammers, allowing open-relays, badly configured proxies or message sending from dynamic addresses. However, blocking lists must be used carefully as they are sometimes too slow in removing good sites, or not very accurate in checking that the reported site was actually guilty. For this reason, RBL scores are better used in conjunction with other rules, and not by themselves.

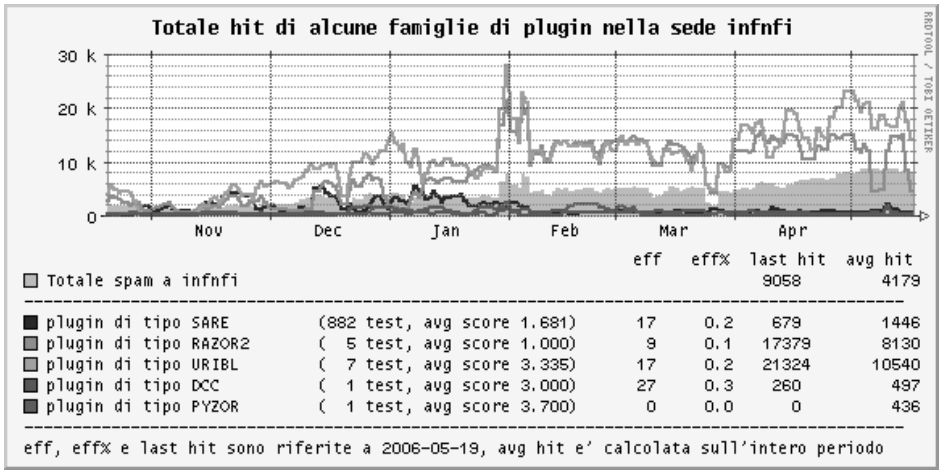

Figure 4: Usual trend for some SpamAssassin plug-in families at INFN-FI (Oct 2005-May 2006).

As stated above, among the several RBL's we have tested, there was a specific one we have found really useful: the URIBL family.

Automatic Categorizers (Community Based & Checksumming)

These systems are based on distributed servers that ''count'' the received messages in order to compute the probability they may be of type Unsolicited Bulk E-Mail (UBE).

Razor

Razor [RAZ] is based on a database of spam E-Mail messages supplied by humans. A sophisticated scoring mechanism based on correct reports or revocations of spam makes this reporting mechanism unalterable by spammers. Since Razor was based on a private protocol, it wasn't easy to enter into the razor collaborative network. However, RAZOR is partially free.

Pyzor

Pyzor [PYZ] is an open-source rewriting of Razor. Our group is trying to enter its more open collaborative network.

DCC

This product [DCC] is based on a slightly different mechanism. DCC servers automatically ''count'' bulk E-Mail messages by trying to cut away the variable elements and keeping the fixed ones, generating a checksum for each of them. After that, DCC servers exchange this information with a flooding mechanism. In this way, for every incoming message, a mail-server can ask a DCC server for the probability that this message, with such a checksum, has to actually be UBE. Any DCC server gives answers to both client types, anonymous and registered. Higher priority is usually given to the latter ones.

DCC is an open system where new servers are always welcome. All working group member sites have been using DCC clients since the group was founded, while four sites are even hosting servers. The first Italian DCC server was installed at INAF in Palermo, immediately followed by the INFN site of Turin [DCT]; then Rome and Bari. At first, we thought three servers would be enough, but new GARR site volunteer servers have always been welcome in order to better serve the increasing number of GARR clients. Our achievement is an improvement of the method efficiency, both by reducing the response time and by increasing DB data about our *domestic* spam. Even though the service is offered on a best effort basis only, the Turin server has been recently replaced with a more powerful one; due to this refurbishment, it has been included in the default DCC server alias (*dcc*.dcc-servers.net*) and is currently serving thousands of clients worldwide, checking an average of 15 M messages/day (12.5% of the total, see Figure 5).

After using this new DCC server, the Turin mail-server unexpectedly started increasing its spam detection efficiency, with false positives and false negatives nearly equal to zero. We are investigating this coincidence, but we suspect that the different new role of the Turin DCC server affected its spam detection ability (see Figure 6). In fact, because of its inclusion in the default DCC server alias, its database now contains data which is more up to date than when it was interacting with fewer clients (the *domestic* ones only) and processing fewer messages, and the DB was aware of the rest of the world by a data flooding between DCC servers only.

DCC Reputations are a distinct mechanism based on and contributing to DCC data. In part to minimize abuse by anonymous users, DCC Reputations are available only in the commercial version of the DCC software. For this reason, our working group still hasn't considered implementing it, but we cannot exclude begging it from Vernon in the future . . .

DSpam

DSpam [DSP] is a spam detection system proposed as an alternative to SpamAssassin. It's based on highly sophisticated statistical techniques only. Even though the authors achieve an efficiency better than 99.9%, not only did our testing not reach this value, but showed results worse than SpamAssassin. Perhaps our training phase quality (not accustomed to purely statistical methods) was not as good as required. For the future, we are currently considering its possible implementation as a plug-in of SpamAssassin.

Viruses and Worms

Fortunately, nowadays anti-virus filters for mail-servers are quite robust and reliable. Free software surely has good examples of well designed products (i.e., ClamAV), but commercial software is usually better due to faster virus definition updates (i.e., Sophos AV). Nevertheless, commercial software doesn't always imply more reliability.

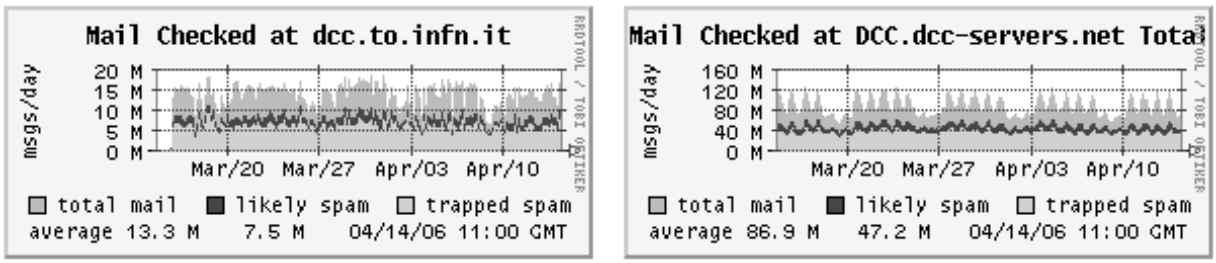

Figure 5: E-Mail checked at dcc.to.infn.it, compared with the total at dcc.dcc-servers.net.

For the whole working group the number of viruses *revealed* by E-Mail decreased very quickly by simply adopting good anti-virus filters, implementing greylisting, and following some best practices (discussed in the next section).

As shown by the green line in Figure 7, in less than one year, the total amount of viruses *revealed* in some GARR sites decreased from 30% to nearly zero.

Best Practices

An important part of the working group activity has been dedicated to "best practices," that is *suggestions* to improve the security of E-Mail services.

Among the most important points identified:

- Edge routers should allow incoming traffic through port 25 (smtp) to reach only the domain official mail-server, in order to prevent generic LAN computers from being used as mail-relay.
- Edge routers should pass only outbound traffic through port 25 (smtp) from the domain official mail-server, in order to prevent viruses and worms from sending E-Mail messages (at present, a typical behavior). Ports 587 (msa – mail message submission), and possibly 465 (formerly for Windows Outlook mail message submission, currently deprecated), must be left ***open***, so *roaming* users can use their own MTA from the exterior of the LAN.
- Roaming users must be able to use their own MTA from the exterior of the LAN by authentication only. This way one can implement sender control methods (i.e., SPF). Be careful

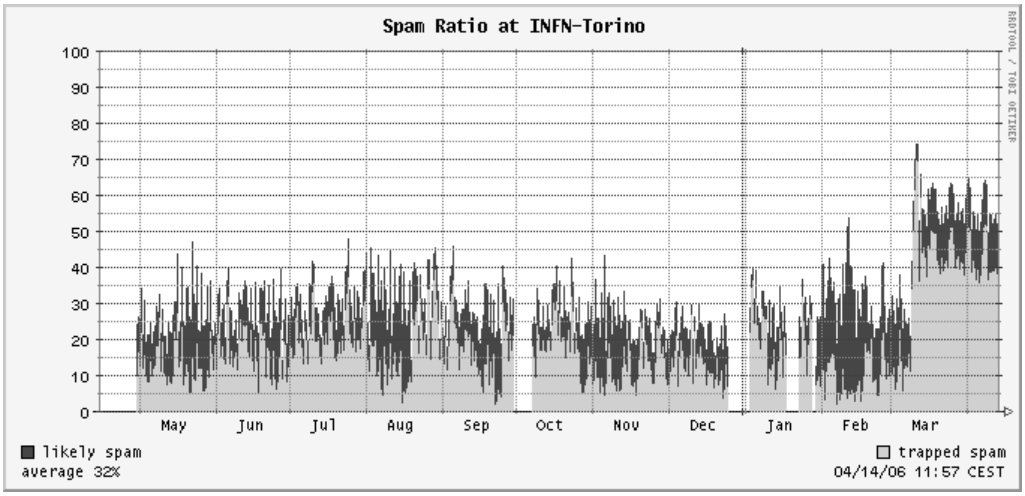

Figure 6: Spam ratio at dcc.to.infn.it from May 2005 to April 2006.

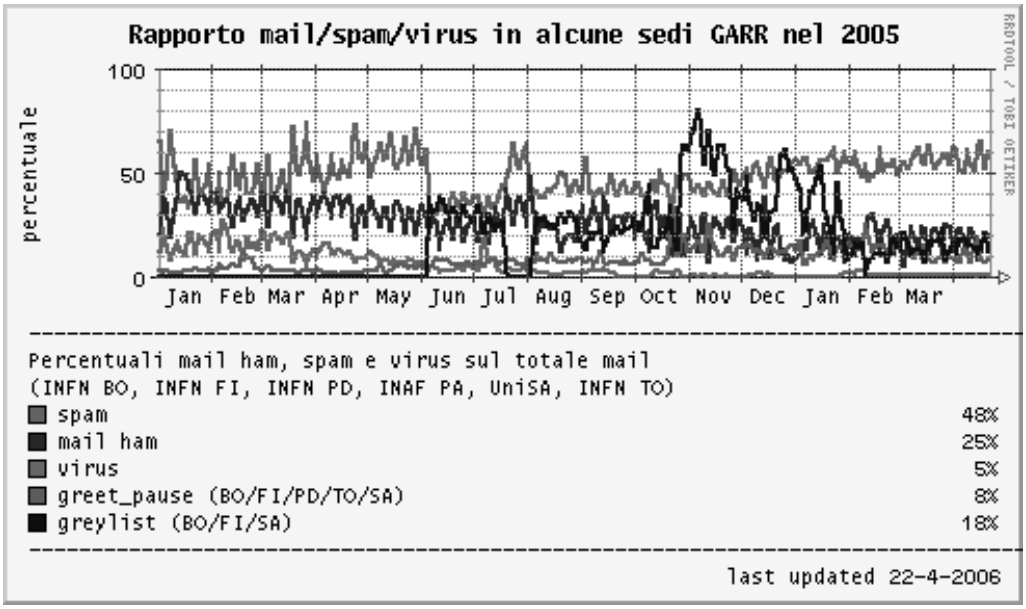

Figure 7: Ham, Spam & Virus on total E-Mail, in some GARR sites from Jan 2005 to Apr 2006.

when using authentication together with SpamAssassin! If a user's Internet Service Provider (ISP) has a bad reputation, SpamAssassin may consider his messages to be spam. In order to prevent this, the following rule should be added when using Sendmail as MTA:

```
header LOCAL_CERT_GNORRI_ON Received =~
/\HEADERSTRING\].+verify=OK\)/
score LOCAL_CERT_GNORRI_ON -15
```

where HEADERSTRING is typical of one's mail-server header.

ISP's with bad reputation are a very annoying problem. Unfortunately, in Italy it's a common issue, mainly with cheaper and faster ISP's. So, due to the latter ones, non guilty users risk being classified as spammers even when using web-mail interfaces, if they list the ISP in the message header because of some RFC (821) interpretation.

- The anti-virus software installed on the mail-server should be configured to prevent itself from sending information messages (about infected E-Mail messages) to the senders, since these ones are usually *spoofed* (false).
- The Sendmail (v. 8.13) Greet Pause feature should be kept active. Through this mechanism, the SMTP connection is rejected if the sender mail-server doesn't wait for the "220" greeting answer. This is the typical behavior of spammer software and viruses, which are usually not able either to hold-on or, after a time-out, to retry the connection as *regular* MTA's do.

For (not only) GARR users, an *Installation manual for an electronic mail service with anti-virus and anti-spam filters* [MAN] has been published (presently available in Italian only).

Sender Domain Authentication

Another important argument is the sender server authentication, because partially related to the spam problem (many spam messages come with the sender spoofed). Moreover, roaming users must be able to use their own MTA from the exterior of the LAN by authentication only.

Several methodologies have been proposed, even though not yet merged into a RFC. Among the several proposals, two different technologies have emerged. The first one is known as Sender Policy Framework [SPF], the second as Sender-ID [SID], and both aren't directly used for fighting spam, but only for authenticating the mail-server that is sending an E-Mail message. The basic concept of these systems is that the user of a generic domain can send messages only by those mail-servers explicitly authorized by their own domain. This mechanism prevents E-Mail messages with a spoofed sender address from being sent, but mainly prevents infected computers from sending spam or viruses.

Our working group chose to test SPF, verifying all possible benefits. A site can become SPF compliant by publishing the mail-server names authorized to send E-Mail messages with this site sender addresses. The best achievements would be reached whenever the majority of the sites publish the list of the authorized servers as well as enter them into their own DNS records. However, we still are far from this status. However, since some big ISP's already publish SPF records, it's possible to use this information in order to modify the SpamAssassin scoring. It shouldn't be forgotten that ISP's implementing SPF must inform their roaming users that they cannot sign their outgoing E-Mail messages with their own domain name when using *external* IP addresses, because the SPF check would fail. Thus, before publishing one's SPF record, one's users should be aware they can send E-Mail messages by using authorized mail-servers only. For this reason, relaying must be allowed but only by using specific authorization mechanisms (i.e., either via password or X.509 Certificate).

Our tests, made on an actual domain (University of Salerno), showed that 12% of messages received in one month (650K) came from senders whose mail-servers were already SPF compliant. To this value may be added another 20% of messages belonging to the examined domain, reaching therefore a considerable 32% of E-Mail messages carrying SPF information.

Starting June 2006, we decided to enable SPF on all mail-servers managed from our working group members. At present, only in a *soft* way, just to test it by the related SpamAssassin rules.

Greylisting Technologies

Nowadays over 60% of spam and viruses come from infected computers (hijacked computers) and not from real servers. Machines infected by these viruses try to emulate the behavior of a full-fledged mail-server, but this imitation fails to implement some functionalities. This lack of functionality can be used to differentiate a real mail-server from an infected machine.

The functionality used to differentiate between real servers and infected machines is the retransmission capability, that is the capability of a real server to retransmit a message if a receiver server couldn't (or wouldn't) receive E-Mail from another server (e.g., the receiver server is overloaded, or the receiver server uses Greylisting).

The Greylisting [GRE] method is very simple. It examines only three pieces of information (which we will henceforth refer to as a "triplet") in any particular E-Mail message delivery attempt:

1. The IP address of the host attempting the delivery
2. The envelope sender address
3. The envelope recipient address

From these, we now have a unique triplet for identifying an E-Mail "relationship." With this data, we simply follow a basic rule, which is:

- If we have never seen this triplet before, then refuse this delivery and any others that may come within a certain period of time with a temporary failure.

Since SMTP is considered an unreliable transport, the possibility of temporary failures is built into the core spec (see RFC 2821). As such, any well behaved message transfer agent (MTA) should attempt retries if given an appropriate temporary failure code for a delivery attempt.

The main two limits of this approach are:

- the delay time due to the temporary failure (from 30 minutes to some days);
- some ISP, like hotmail, use an entire subnet for SMTP retransmission and not a single IP thus implying more difficulty to identify the triplet.

Although the first one of these problems could be the less important (since SMTP protocol does not guarantee the delivery time for an E-Mail message), it becomes very annoying to the community of users when the E-Mail Service is used more and more like an Instant Message application rather than a postal service.

The second problem can be easily solved using the full sub range (“C” class Network) of an ISP as the IP address. This way every server in the network that will retry to forward the E-Mail message previously interrupted will be recognized as being the same server.

There is also a second approach to the subnet problem: SPF. Domains like Hotmail and AOL, which are frequently faked and abused by spammers, have introduced an SPF record into their own DNS domains. That record can be used to improve the behavior of greylisting. If a receiving server is with greylisting enabled and receives an E-Mail message from Hotmail or AOL, it could look up the SPF record of the sender domain and compare the sender IP with the ones allowed for the sender domain. If the sender address is allowed then the greylisting should accept that message as it is a real message. Spam and viruses which are sent from hijacked computers don’t come from the authorized servers but rather from the infected computer itself.

The Advantages

Sharp Reduction of Received Viruses and Spam

Greylisting reduces the number of accepted E-Mail messages processed by the receiver server and this greatly reduces the amount of spam and virus to check. The price is a rise of the average delivery time and an increase in E-Mail traffic (due to the retransmissions).

In Figure 8 one can notice that the traffic shape changed enormously in the month of June 2005, in concurrence with activation of Greylisting filter at INFN-FI. The above picture represents:

- Red line: messages identified as spam by SpamAssassin (with a score threshold of 3.5);
- Cyan line: messages rejected by greet pause of Sendmail 13.x;
- Black line: messages rejected by (classic) Greylisting;
- Yellow line: messages containing viruses;
- Blue line: “clean” messages;
- The last, the grey contour, is the sum of all the contributions.

Two important aspects to underline are the trends of the blue and grey lines: the first shows that real E-Mail preserves the same behavior, the second represents the total number of messages processed by the servers. The jump from 15K messages to over 30K daily is due to retransmissions.

Sensitive Reduction of the CPU Load

Some measurements taken at University of Salerno, prove that only 20% of the incoming E-Mail is actually re-forwarded. About 80% of E-Mail is immediately blocked by the Greylist algorithm. Considering moreover that Greylist’s algorithm is activated only during the first step of the E-Mail protocol

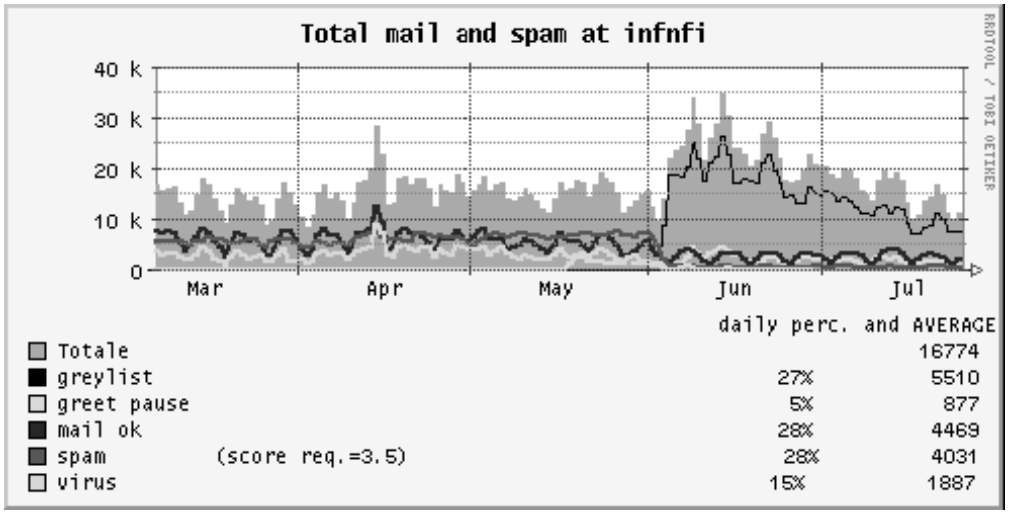

Figure 8: Number of E-Mail messages received at INFN-FI from March to July 2005.

(header transmission), it means that 80% the incoming E-Mail is zero-cost rejected as can be seen from Figures 9 & 10.

In the first graph (Figure 9) we exhibit the CPU load of a mail-server (mx3.unisa.it) equipped with greylisting technology while into the second graph (Figure 10) we are reporting the CPU load of another server (mx2.unisa.it) without greylisting functionalities. As can be seen the load in the first graph is only a small portion of the CPU utilized in the second situation.

The Disadvantages

The Delivery Time of the First E-Mail Message

The request for comment 2821 [RFC 2821], paragraph 4.5.4.1 Sending Strategy, asserts that a sender must resend a rejected E-Mail message after at least 30 minutes:

> "The sender MUST delay retrying to particular destination after one attempt has failed. In general, the retry interval SHOULD be at least 30 minutes; however, more sophisticated and variable strategies will be beneficial when the SMTP client can determine the reason for non-delivery."

Moreover it would have subsequently executed two connection attempts in the first hour and one every two or three hours thereafter:

> "Experience suggests that failures are typically transient (the target system or its connection has crashed), favoring a policy of two connection attempts in the first hour the message is in the queue, and then backing off to one every two or three hours."

Other measures (Figure 11) performed on our servers at the University of Salerno, show that approximately 30% of E-Mail is resent within approximately 10 minutes, 60-70% of incoming E-Mail is instead delivered after 30 minutes and finally, 80-90% of E-Mail is delivered within 60 minutes from the first attempt.

There is also another disadvantage that can be felt as particularly annoying: websites that require you to create an account and confirm your E-Mail address before you can begin using them. Due to the fact that greylisting will delay the initial E-Mail message containing your signup confirmation link (maybe for some minutes or perhaps some hours), it will introduce a waiting period even though the actual website may send out your E-Mail confirmation code immediately.

A Better Greylisting

A Home-made Experimental Technology

The new approach to Greylisting is the union of classic Greylisting with a spam filter. While Greylisting tells us *if a server is RFC compliant*, spam filter tells us at *which level of confidence a message can be considered ham or spam*. Joining the two algorithms allows some optimization like bypassing the Greylisting algorithm if the E-Mail message has a very low score or, in the case of multiple recipients, accept an E-Mail message if only a single well-know triplet that "introduce" the sender for all the others exists.

These optimizations aim to reduce the delivery time for the first E-Mail message.

The Achieved Result

As can be seen from Figure 12, more than 40% of E-Mail messages are delivered instantaneously. Approximately 80% of them are instead delivered

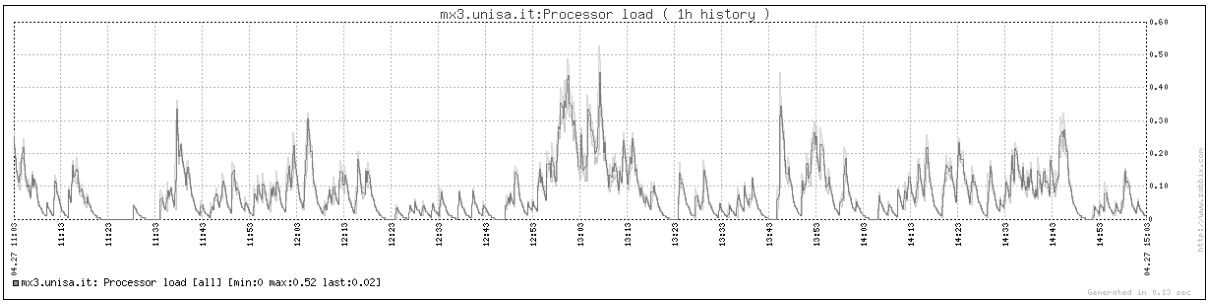

Figure 9: CPU load of a mail-server with classic greylisting at UNISA.

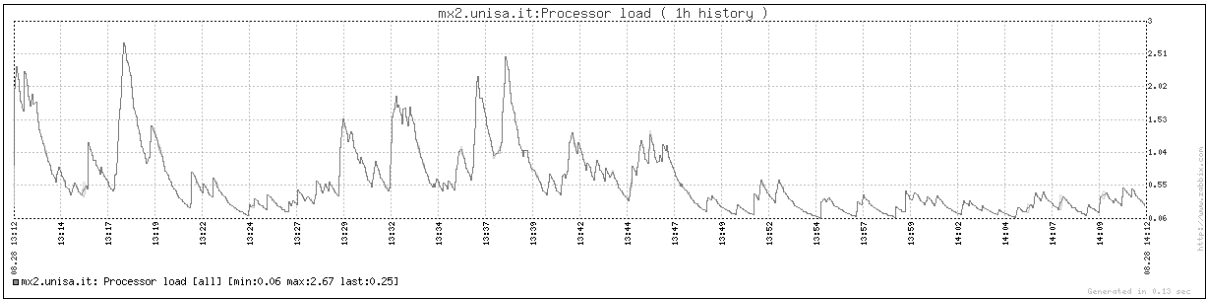

Figure 10: CPU load of a mail-server without any greylisting at UNISA.

within 30 minutes and 90%, with peaks of 100%, within 60 minutes. Essentially, from a direct comparison with the previous diagram, one can notice a sharp improvement on the accepted times.

As shown in Figure 13, in one hour the mail-servers of Università degli Studi di Salerno received 1197 E-Mail messages. Only 12% of them are accepted as ham, while 3% are instead classified as SPAM and 0.8% of E-mail is infected. The remaining part, 84%, is rejected. Practically, 16.5% is rejected thanks to Greeting Pause, 21% is rejected due to non-existent recipients and 45% is rejected by greylisting where only 9.8% of these messages are retransmitted and, therefore, accepted.

The Cost

The consequence of this result is a sensible growth in the load on the server and in the used network band. This is due to the fact that all the E-Mail messages are, at least once, processed from the anti-spam & anti-virus filter (see Figure 14).

In Figure 14, classic greylisting has been tested from October 2005 until January 2006, while experimental greylisting has been running since February 2006.

Conclusions (The Lesson Learned . . .)

A huge amount of graphs and statistics have been produced and are available in (almost) real-time on the working group web site and wiki [WSM]. They significantly helped us in highlighting many of the achieved results, or just to quickly understand *why the devil that stupid mail-server stopped working . . .*

Note that all the produced software is freely downloadable from the sites listed either in the above paragraphs, or in the ''Bibliography'' section. This is the real-life story of 10 guys that started ''playing'' within a new working group, and finished working seriously, giving valuable results to the Italian Academic (and non) network. From Figures 15 & 16 we see that, in spite of a more or less constant increase of spam messages, we still have our mailboxes clean thanks to a constant fighting activity of the whole working group. Unfortunately, new/original tools are not always useful to get good results and with our work we have demonstrated that one can achieve excellent results even by using just standard tools. We've seen that the more SpamAssassin plug-ins one uses, the more efficiency one will obtain.

The statistical (Bayesian) approach resulted very effective (> 90% hits), and spammers seem still unable to go around it.

Provided you consider it ethical, rejecting spam let's the users save time in checking the specific folders where spam messages are hijacked.

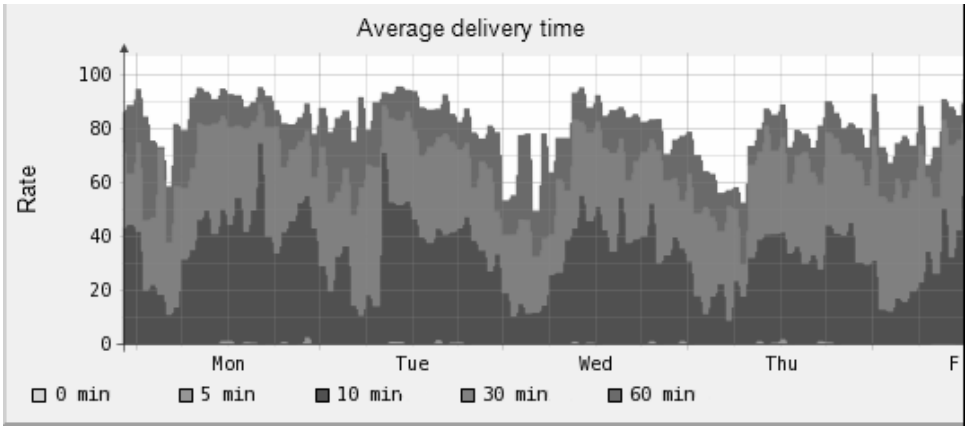

Figure 11: Average delivery time with classic greylisting at UNISA.

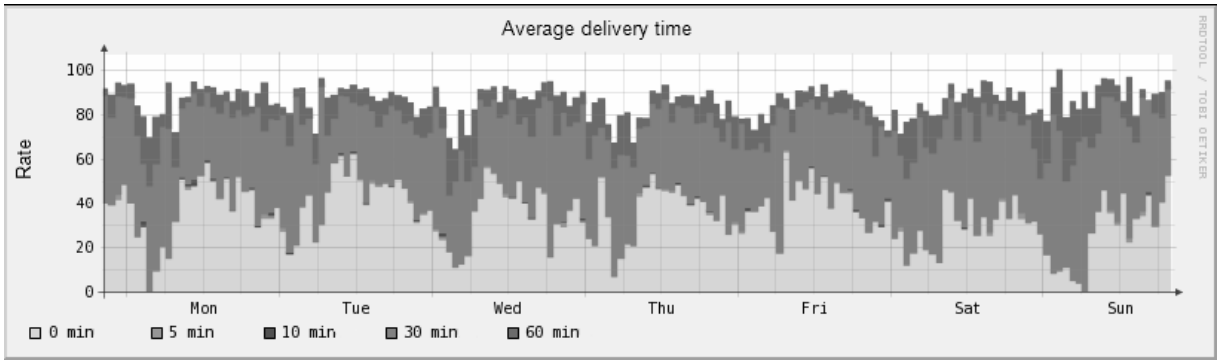

Figure 12: Average delivery time with experimental greylisting at UNISA.

Our very recent Italian DCC Servers Network has surely improved the spam detection efficacy of the Italian (and non) clients installed on mail-servers.

We found that greylisting provided a significant reduction in the amount of spam received and processed and that greylisting combined with SPF checking provided the greatest benefit.

Last, but not least, without a minimum of best practices, the best anti-spam packages might result completely useless.

Thus, without this constant battle, is the war lost? Will I ever be able to run a software that lets me forget my mail-server?

Author Biographies

Roberto Cecchini: Degree in Physics; Computing Centre Coordinator at INFN in Florence; since 1999 GARR IT security service (GARR-CERT) Coordinator; since 1998 INFN Certification Authority (INFN-CA) Coordinator.

Fulvia Costa: LAN Manager & APM at INFN in Padua.

Alberto D'Ambrosio: Electronic Technician; Computing Centre System Administrator at INFN (from 2000 at Turin Section, from 1992 to 2000 at Gran Sasso National Laboratories); previously analyst/programmer in the area of industrial automation .

Domenico Diacono: Computing Centre System Administrator at INFN in Bari.

Giacomo Fazio: IT Engineer; System and Network Administrator at INAF/CNR in Palermo: E-Mail system manager, Computing Centre Coordinator (from 1991 at IFCAI, then IASF, now INAF); previously programmer for research groups working in the area of astrophysics.

Antonio Forte: IT Technician; Computing Centre System Administrator at INFN in Rome1 (from 1/1/2004), Turin (1998-2003) & Rome2 (1996-1998).

Matteo Genghini: Electronic Engineer; since 2002 Computing Centre IT Coordinator at IASF/CNR in Bologna; previously multimedia programmer.

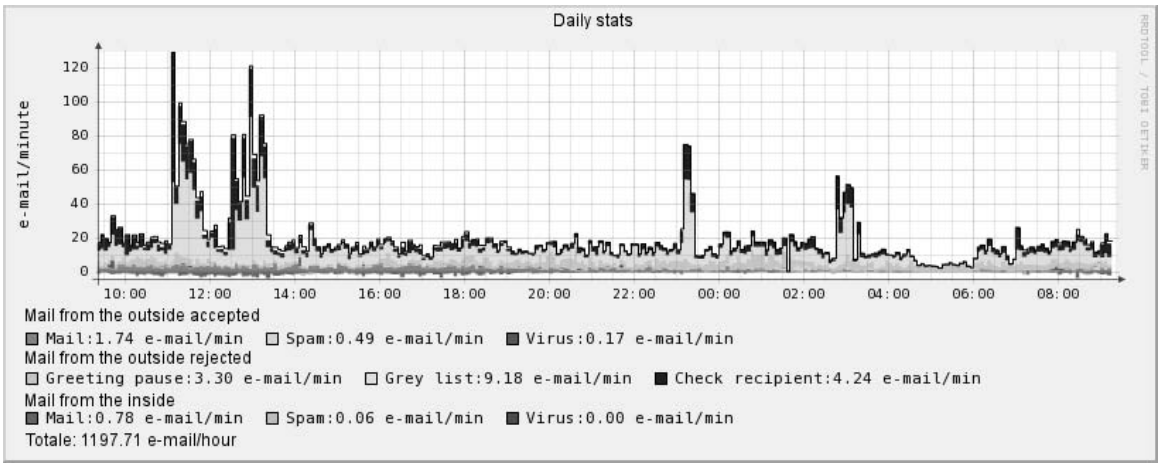

Figure 13: Daily E-Mail stats at UNISA.

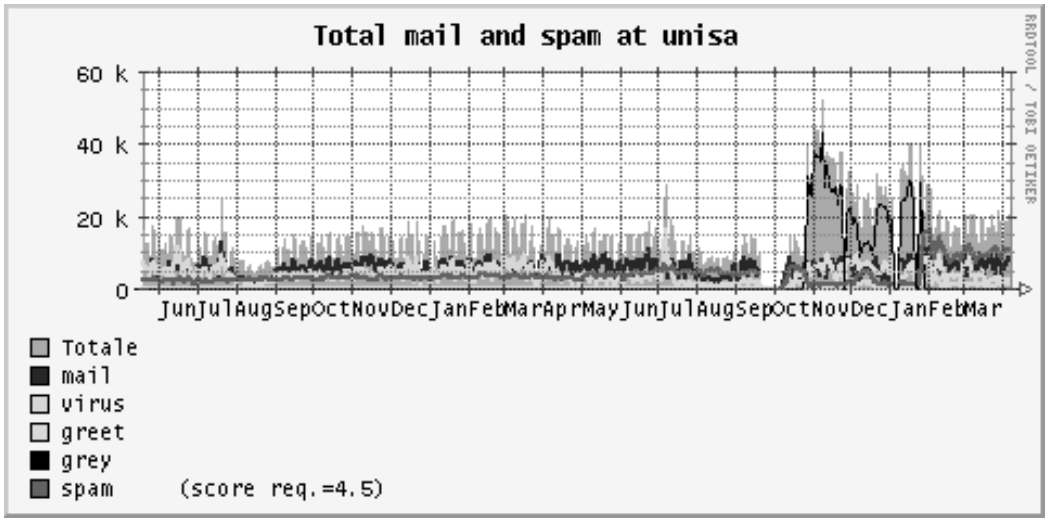

Figure 14: Number of E-Mail messages received at UNISA from May 2004 to April 2006.

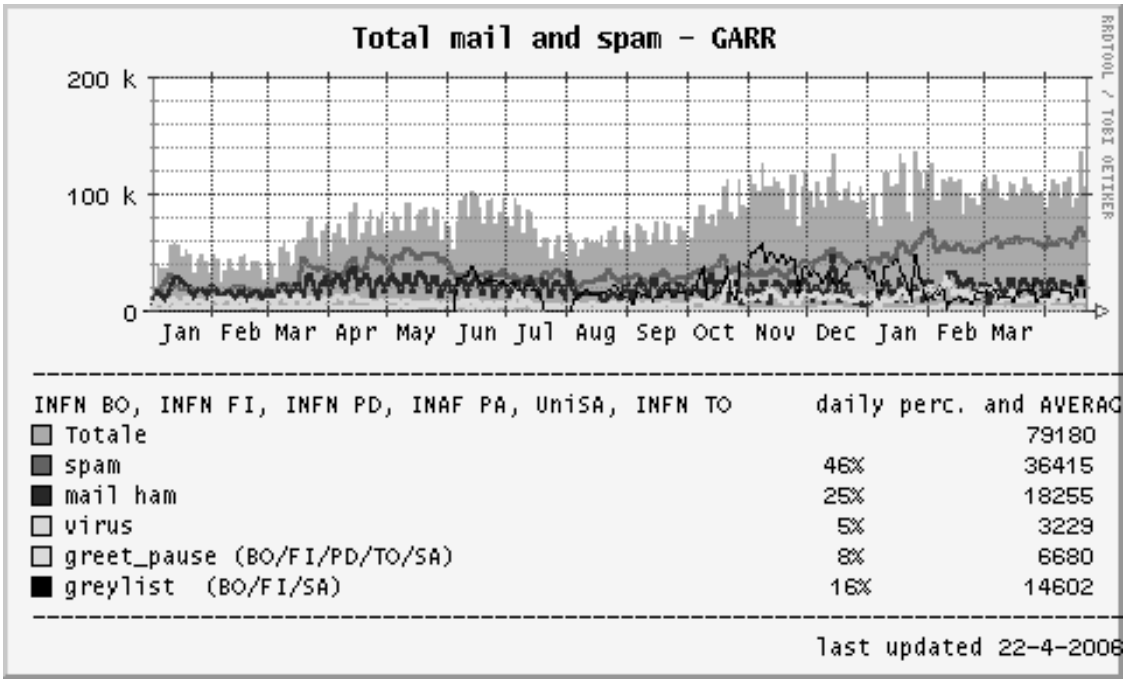

Figure 15: Number of E-Mail messages received at some GARR sites from Jan 2005 to Apr 2006.

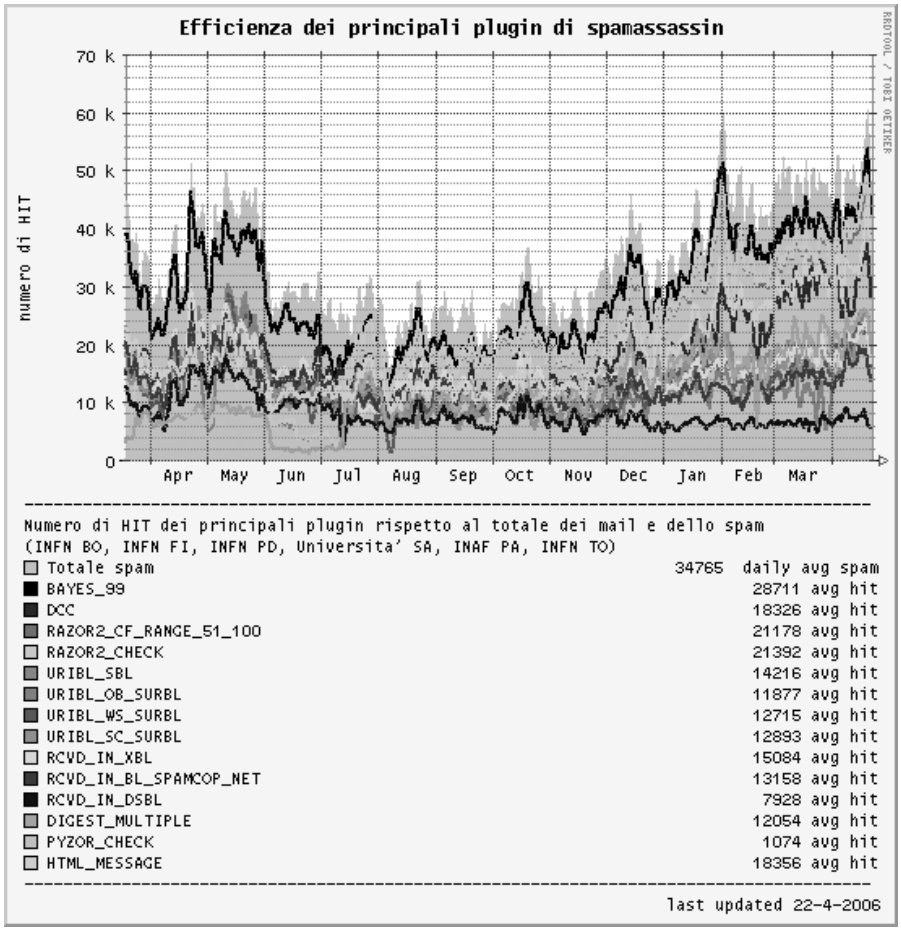

Figure 16: SpamAssassin plug-ins at some GARR sites from March 2005 to April 2006.

Michele Michelotto: Physicist; since 1997 Computing Centre Coordinator at INFN in Padua; previously, offline analysis in High Energy Physics experiments at CERN (Geneva) and INFN (Padua & Legnaro).

Ombretta Pinazza: Degree in Physics; since 1998 IT expert at INFN in Bologna; previously, Researcher at the Engineering Faculty in Bologna and Researcher at TESRE-CNR in Bologna.

Alfonso Sparano: Degree in IT Engineering; Data Processing Centre IT services administrator at the University of Salerno.

Bibliography

[AMA] http://www.ijs.si/software/amavisd/ .

[DCC] http://www.rhyolite.com/anti-spam/dcc/ .

[DCT] http://www.to.infn.it/dcc/ .

[DSP] http://www.nuclearelephant.com/projects/dspam/ .

[GRE] http://www.greylisting.org/ .

[MAN] Bar, Giorgio, Alberto D'Ambrosio, Franca De Giovanni, ''Manuale di installazione di un servizio di posta elettronica completo di filtri anti-virus e anti-spam,'' *Installation manual for an electronic mail service with anti-virus and anti-spam filters*, INFN/TC-05/09, SIS-Pubblicazioni, Frascati, Rome, Italy, http://www.lnf.infn.it/sis/preprint/pdf/INFN-TC-05-9.pdf .

[PYZ] http://pyzor.sourceforge.net/ .

[RBL] http://www.webopedia.com/TERM/R/RBL.html .

[RAZ] http://razor.sourceforge.net/ .

[RJS] http://mips.df.unibo.it/sw/rjspam0.1.tar.gz .

[SID] http://www.senderid.org/ , http://www.microsoft.com/mscorp/twc/privacy/spam/senderid/default.mspx .

[SPF] http://spf.pobox.com/ .

[WSM] http://www.garr.it/WG/sec-mail/ , http://secmail.unisa.it/ .

Glossary

Spam: E-Mail messages we wouldn't like to receive

Ham: Good (non Spam) E-Mail messages.

UCE: Unsolicited Commercial E-Mail. Commercial E-Mail messages not requested, thus not welcome, sent to one recipient or thousands.

UBE: Unsolicited Bulk E-Mail. E-Mail messages sent to thousands of recipients. Not necessarily commercials; might even be sent to check the actual existence of the recipients.

False Negatives: Spam messages erroneously detected as Ham.

False Positives: Ham messages erroneously detected as Spam.