Una soluzione AntiVirus e AntiSpam

ILC-CED-2003-03

Alessandro Enea

Consiglio Nazionale delle Ricerche

Istituto di **L**inguistica **C**omputazionale

Alessandro.Enea@ilc.cnr.it

Novembre 2003

Indice

1. Introduzione

In queste pagine viene descritta una soluzione per il filtraggio automatico di messaggi di posta elettronica con allegati virus e/o con contenuti "SPAM".
Questa soluzione è adottata da Maggio 2003 sul mail server del nostro Istituto, ed è basata sull'utilizzo di un software Open Source che ha il compito di scomporre opportunamente il contenuto dei messaggi in arrivo (ed anche di quelli in partenza) per poi inviarli ai programmi utilizzati per l'analisi dei contenuti (un software antivirus commerciale ed un software antispam open source).
La problematica del filtraggio antivirus é molto ben conosciuta, mentre due parole vanno spese per introdurre la problematica antispam. Lo SPAM, anche indicato con la sigla "UCE" (Unsolicited Commercial Email) o "UBE" (Unsolicited Bulk Email) e con altre sigle meno conosciute, è in continua espansione (ormai oltre il 50% delle e-mail ricevute nel mondo è spam). Ormai molte organizzazioni hanno una forte esigenza di poter controllare, e ridurre al minimo, il flusso di e-mail considerate SPAM, per evitare intasamenti del server di posta e/o delle caselle degli utenti, ed inutili perdite di tempo per cancellare tali messaggi. Secondo uno studio sulle aziende americane pubblicato a novembre 2003 dagli analisti di Nucleus Research (http://www.nucleusresearch.com) ogni azienda ha perso nel 2003, in media, 874 dollari per dipendente, a causa del tempo sprecato per cancellare lo spam e distinguerlo dalle e-mail buone. Tale danno può essere ridotto almeno del 26% utilizzando un software antispam. Esistono ormai sul mercato numerose soluzioni commerciali al problema, ma esistono altrettanto valide soluzioni Open Source che consentono una notevole riduzione dei costi.
I prodotti esistenti (sia commerciali che Open Source) si dividono in due categorie in base alla filosofia adottata:

1. utilizzare un filtro antispam sul client dell'utente finale;
2. utilizzare un filtro antispam sul server di posta elettronica.

Noi abbiamo adottato un prodotto della seconda categoria che, unito al filtro antivirus, garantisce un aggiornamento centralizzato e costante sia della conoscenza dei contenuti spam che del database dei virus esistenti.

2. Introduzione a Amavisd-new

Il componente principale di questa soluzione è costituito dal software Open Source **Amavisd-new** (http://www.ijs.si/software/amavisd); si tratta di una interfaccia software che si pone fra il server di posta elettronica (nella terminologia tecnica inglese "MTA" (message transfer agent), nel nostro caso il noto Sendmail) ed uno o più content checkers: il software antivirus e/o il software antispam (nel nostro caso sono utilizzati entrambi).
Un documento tecnico utilissimo per iniziare è il seguente: http://www.amavis.org/amavis.html.
Amavisd-new é il nome della nuova versione di Amavisd, che era la versione "demone" standalone di AMaViS (http://www.amavis.org) o Amavis-perl. La nuova versione si caratterizza per le sue elevate performance di velocità di elaborazione e per l'arricchimento delle opzioni di configurazione disponibili.
Il software Amavisd-new va normalmente installato sul server principale di posta elettronica, non necessariamente dove risiedono le mailbox degli utenti sulle quali vengono memorizzati i messaggi in arrivo.
Amavisd-new utilizza il modulo Perl Net::Server che offre un veloce ambiente "pre-forked multichild". Rispetto alla versione precedente sono disponibili diverse nuove features, fra cui "SMTP-in/SMTP-out capability". Questo lo rende efficacemente utilizzabile su mail gateways molto utilizzati per il controllo anti-virus e/o anti-spam.

Prima di addentrarci nei dettagli dell'installazione e della configurazione di Amavisd-new (paragrafo 6) vedremo:

- come installare il software antispam SpamAssassin (paragrafo 3);
- come installare il software antivirus McAfee (paragrafo 4);
- come viene modificato il flusso di ricezione e trasmissione dei messaggi di posta elettronica con l'introduzione di Amavisd-new associato al mailer Sendmail (paragrafo 5).

3. SpamAssassin

SpamAssassin é un software Open Source da utilizzare come filtro della posta elettronica per identificare i messaggi spam.
Utilizzando le sue regole di base applica una vasta gamma di prove euristiche sulle intestazioni della posta e sul testo del corpo per identificare "lo Spam".
I metodi utilizzati per l'identificazione dello spam sono:

- **analisi delle intestazioni**: gli spammers usano un certo numero di trucchi per mascherare le loro identità, tentano di far credere all'utente finale che il loro messaggio è simile a qualunque altro messaggio, o di far credere che ad un certo momento è opportuno abbonarsi a qualche mailing list. SpamAssassin può scoprire questi tipi di messaggi;
- **analisi del testo**: i messaggi Spam hanno spesso uno stile caratteristico (una forma molto gentile) ed alcuni dinieghi caratteristici e testo in stile CYA. SpamAssassin può scoprire anche questi;
- **blacklists**: SpamAssassin supporta molte utili blacklists già esistenti, come quelle di mail-abuse.org, ordb.org e altre;
- **Razor**: Vipul's Razor é un database aperto alla collaborazione di chiunque ed ha come obbiettivo quello di tenere traccia dell'evoluzione dei messaggi spam. Contando sul fatto che lo spammer tipicamente opera inviando un identico messaggio a centinaia di persone, Razor cerca di intercettarlo velocemente permettendo alla prima persona che lo riceve di aggiungerlo al suo database e a questo punto tutti i server che riceveranno lo stesso messaggio saranno in grado di bloccarlo.

3.1 Installazione

Il software Spam Assassin si compone di un gruppo di script Perl e, come qualunque altro software scritto con questo linguaggio, può essere scaricato da CPAN (Comprehensive Perl Archive Network) (http://www.cpan.org) con le seguenti istruzioni (inserite da riga di comando come root):

 # perl -MCPAN -e shell
> o conf prerequisites_policy ask
> install Mail::SpamAssassin
> quit

In alternativa può essere scaricato da http://spamassassin.org il file tar o zip contenente il software ed installato con i comandi:

[unzip/untar del file prelevato]
*# cd Mail-SpamAssassin-**
 # perl Makefile.PL

[opzioni: aggiungere -DSPAMC_SSL a $CFLAGS per avere spamc con SSL abilitato]
 # make
 # make install [come root]

Se invece si vuole utilizzare una interfaccia grafica semplice si può ricorrere a Webmin (http://www.webmin.com).
Webmin segue in sostanza il primo metodo scaricando il software da CPAN.

In questo caso i passi da seguire per l'installazione sono:

1. Dal menu principale di Webmin selezionare "Altro" ed apparirà la Figura 1:

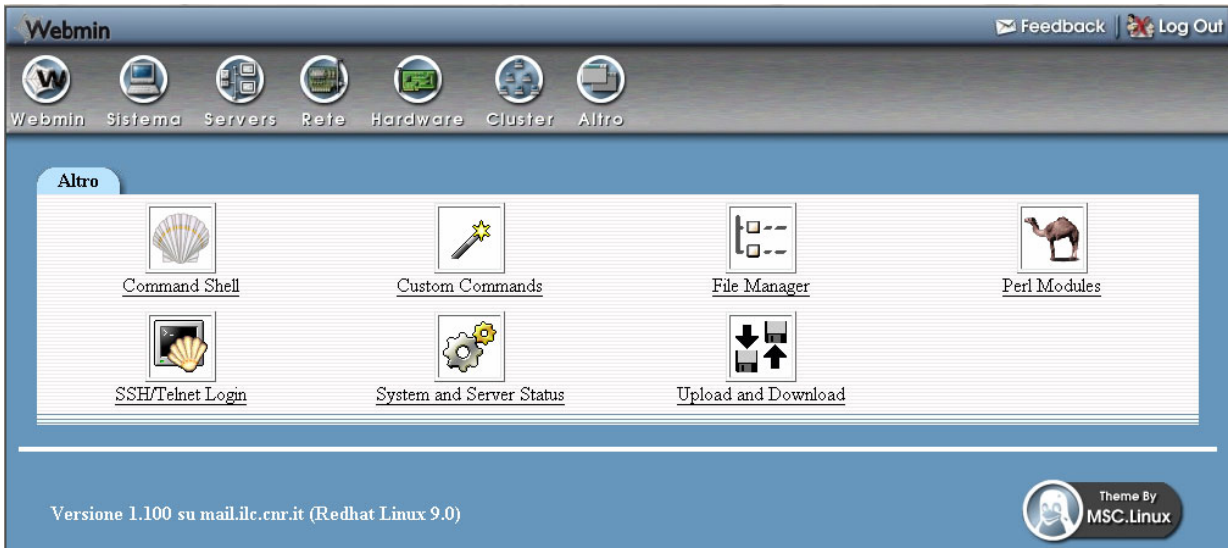


Figura 1

2. Selezionare "Perl Modules" e fare scorrere la finestra fino all'apparire del testo mostrato in Figura 2:

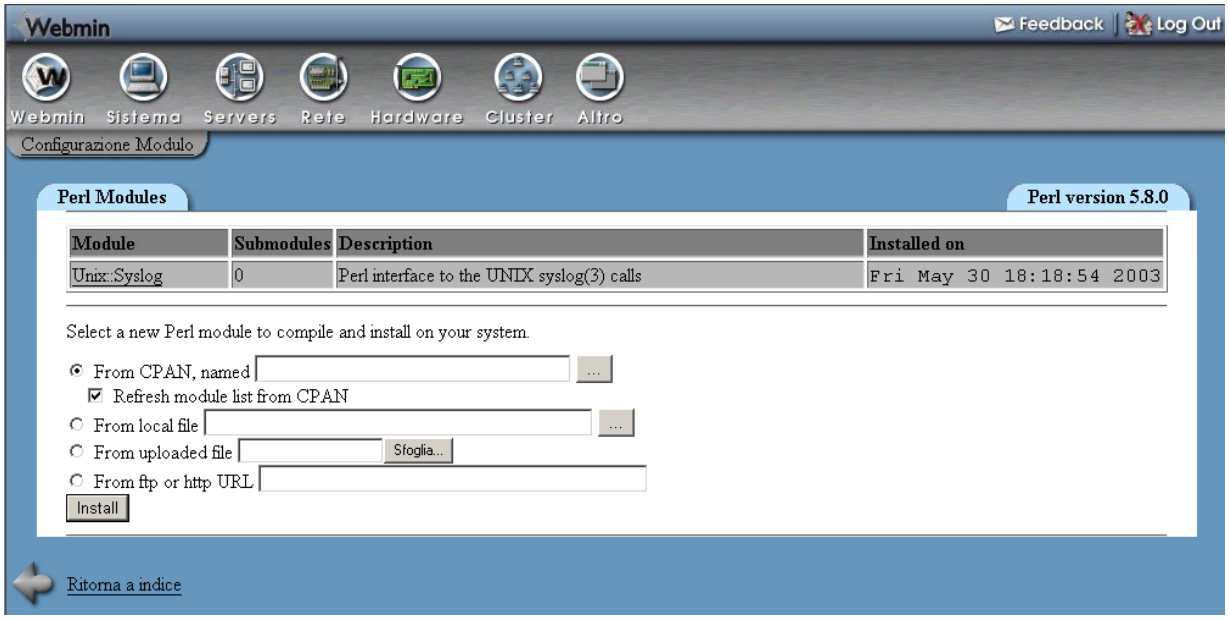


Figura 2

3. Nel primo dei campi disponibili per l'installazione di un nuovo modulo Perl (quello già selezionato recante il testo "From CPAN, named") inserire: Mail::SpamAssassin e cliccare su Install.

Questa procedura può essere seguita anche successivamente per effettuare periodici aggiornamenti del software.

3.2 Configurazione

Dopo aver installato SpamAssassin non é necessaria nessuna operazione di configurazione perché sarà Amavisd-new a mandare in esecuzione la versione standalone di SpamAssassin passando gli opportuni parametri.
E' possibile, comunque, effettuare delle personalizzazioni creando o modificando i file con estensione .cf inseriti nella directory /etc/mail/spamassassin.
Ecco, ad esempio, come abbiamo configurato l'uso del client Razor2 (si veda http://razor.sourceforge.net) come ulteriore filtro per i messaggi spam: modificando il file /etc/mail/spamassassin/local.cf abbiamo aggiunto in fondo la riga:

use_razor2 1

Per maggiori dettagli sulle opzioni di configurazione si può consultare la pagina:

http://useast.spamassassin.org/doc/Mail_SpamAssassin_Conf.html

4. McAfee VirusScan

4.1 Installazione

L'antivirus che abbiamo utilizzato é il McAfee VirusScan per Linux versione 4.24.0.
Il software viene distribuito all'interno del file vlnx424l.tar.Z che possiamo memorizzare in una qualunque directory all'interno del sistema.
Per scompattare questo file va eseguito il comando:

 # zcat vlnx424l.tar.Z | tar -xf –

Al termine del comando possiamo passare all'installazione utilizzando il comando:

 # ./install-uvscan <directory dove installare l'antirirus>

e fra le parentesi angolate occorre indicare la directory dove si vuole installare il software, che tipicamente è /usr/local/uvscan.
Se la directory di installazione non esiste, il programma chiederà se la si vuole creare.
Il programma di installazione termina chiedendo se creare dei link simbolici al file binario uvscan, alle librerie condivise e alle man pages. E' raccomandato crearli.
E' possibile verificare la corretta installazione seguendo questa breve procedura:

1. copiare la seguente linea di testo in un file e salvarlo con il nome EICAR.COM:
   X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
   la dimensione del file sarà di 68 o 70 bytes;

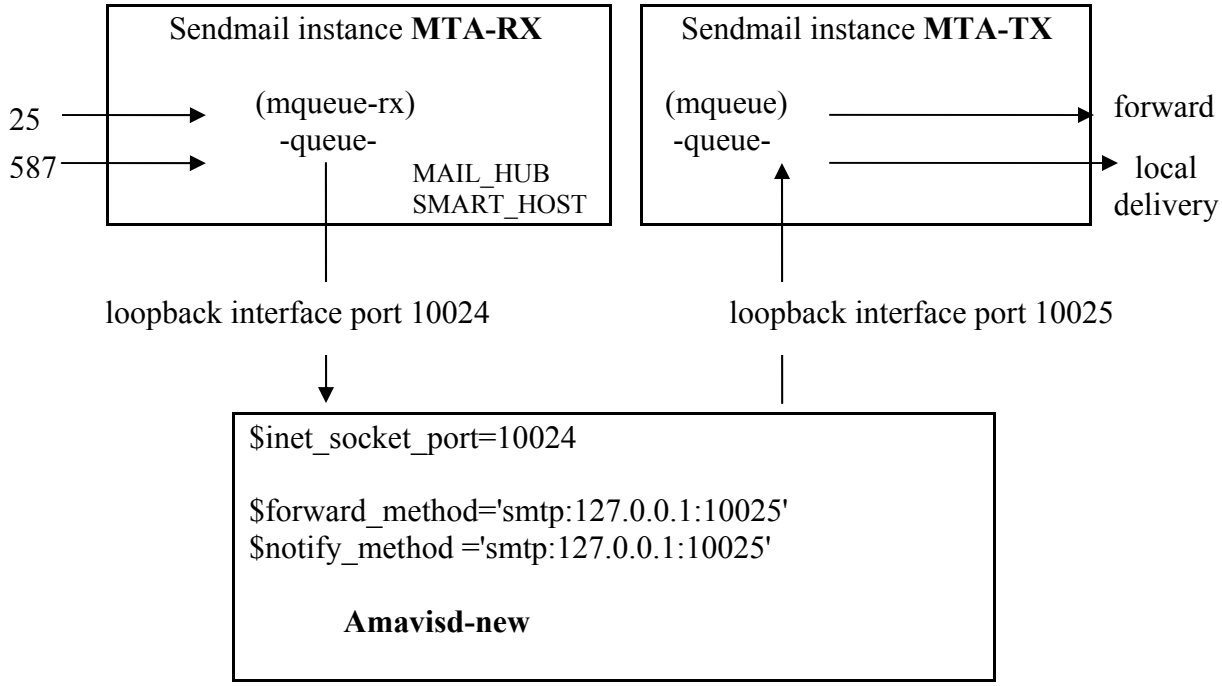
2. eseguire uvscan indicandogli di effettuare la scansione della directory che contiene il file EICAR.COM;
   se l'installazione è corretta dovrà indicare che il file contiene un virus;
3. cancellare il file EICAR.COM per evitare inutili allarmi futuri.

L'antivirus è così installato, ma bisogna costantemente aggiornare l'elenco dei virus che il software è in grado di riconoscere; questo può essere effettuato scaricando periodicamente l'aggiornamento del file DAT da questa pagina del sito McAfee:

http://www.networkassociates.com/us/downloads/updates/dat.asp

5. Flusso dei dati fra Sendmail e Amavisd-new

Descriviamo adesso il setup di questa configurazione:



La configurazione é basata sul sendmail recente (8.12.9) con il suo insieme m4 di macro di configurazione. Se si utilizza una versione precedente in cui una particolare macro o feature non è disponibile, generalmente è possibile crearla manualmente scrivendo una nuova specifica "mailer" all'interno del file .cf.

Vanno configurati due demoni di sendmail, che indicheremo nel seguito come **MTA-RX** (ricevente, accettante) a **MTA-TX** (trasmittente, delivering).

Per comodità teniamo il nome del file di configurazione e della coda (queue) con i nomi di default per ciascuna istanza del mailer.

MTA-RX (il mailer ricevente) avrà la responsabilità di accettare la posta in arrivo da Internet sulla porta 25 e di accettare le spedizioni locali sulla porta 587 (in opzione). Dovrà poi trasmettere tutta la posta (sia quella locale che quella esterna) utilizzando il protocollo SMTP a

127.0.0.1 (una interfaccia di lookback) sulla porta tcp 10024, dove il demone Amavisd sarà in ascolto.

Principali parametri di configurazione di Sendmail relativi a **MTA-RX**:
la sua coda: **/var/spool/mqueue-rx**
i suoi file di configurazione: **/etc/mail/sendmail-rx.cf**, **/etc/mail/submit.cf**
il sorgente (.mc) del file di configurazione: **thishost-rx.mc**

MTA-TX (il mailer trasmittente) avrà la responsabilità di accettare la posta controllata e le notifiche da Amavisd-new via SMTP (o LMTP) sull'interfaccia di loopback (127.0.0.1), porta tcp 10025, e trasmetterà tutta la posta alle rispettive destinazioni finali, sia locali che su Internet agli altri mailers interni.

Principali parametri di configurazione di Sendmail relativi a **MTA-TX**:
la sua coda: **/var/spool/mqueue**
il suo file di configurazione: **/etc/mail/sendmail-tx.cf**
il sorgente (.mc) del file di configurazione: **thishost-tx.mc**

In mezzo ai due MTA, un demone Amavisd accetterà i messaggi utilizzando il protocollo SMTP sulla porta 10024, li verificherà e trasmetterà la posta controllata e le notifiche via SMTP a **MTA-TX** sulla porta 10025.

Se é stata già effettuata una installazione di sendmail, sarà già presente una directory pronta ad ospitare la coda /var/spool/mqueue ed i file di configurazione (il sorgente .mc e il compilato .cf). Molte delle istruzioni esistenti nel precedente file .mc potranno essere riutilizzate, e quindi passate all'interno dei nuovi file di configurazione thishost-rx.mc o thishost-tx.mc o su entrambi. Le istruzioni pertinenti la ricezione della posta devono essere inserite in thishost-rx.mc, mentre le istruzioni pertinenti la consegna della posta (locale o verso altri mailers) devono essere inserite in thishost-tx.mc, e le istruzioni di carattere generale devono essere inserite in entrambi i files.

I nomi dei file thishost-rx.mc e thishost-tx.mc sono arbitrari, ed hanno il solo scopo di servire da sorgenti per produrre i file .cf, che saranno utilizzati come controllo delle istanze di sendmail.

MTA-TX utilizzerà per la sua coda la directory già creata dall'installazione di sendmail.

Per **MTA-RX** é necessario creare una nuova directory per la coda che ospiterà i messaggi in ricezione. Vanno utilizzati gli stessi diritti e la stessa protezione già impostati per /var/spool/mqueue, ossia:

```
# mkdir /var/spool/mqueue-rx
# chown root:wheel /var/spool/mqueue-rx
# chmod 700 /var/spool/mqueue-rx
```

NOTA DI SICUREZZA:
facendo partire sendmail 8.12 è possibile far partire il demone di sendmail come root e dargli dei privilegi (assegnandolo ad uno specifico utente con RunAsUser) dopo averlo associato alla porta 25. Questa modalità è normalmente utilizzata da MSP, e può essere benissimo utilizzata anche da MTA-RX, poiché non è necessario accedere a mailbox degli utenti o a file .forward. Per utilizzare questa feature occorre specificare utente e gruppo nella macro confRUN_AS_USER (dentro il file thishost-rx.mc), e settare il proprietario di mqueue-rx con questo utente:

```
# chown smmsp:smmsp /var/spool/mqueue-rx
```

Sono possibili anche configurazioni più complesse delle code, se necessario.
Per maggiori dettagli sui gruppi di code vedere la documentazione di sendmail.

Riportiamo in dettaglio il file **thishost-rx.mc**:

```
dnl To be used for MTA-RX, the first MTA instance (receiving mail)

dnl Insert here the usual .mc preamble, including OSTYPE and DOMAIN calls.

dnl Specify here also access controls, anti-spam measures,
dnl mail submission settings, client authentication, maximum mail size,
dnl and other settings needed for receiving mail

define(`confRUN_AS_USER',`smmsp:smmsp')dnl  Drop privileges (NOTA DI SICUREZZA)

define(`STATUS_FILE', `/etc/mail/stat-rx')dnl   Non-default stat file
define(`QUEUE_DIR', `/var/spool/mqueue-rx')dnl  Non-default queue area
define(`confQUEUE_SORT_ORDER',`Modification')dnl

dnl Match the number of queue runners (R=) to the number of amavisd-new
dnl child processes ($max_servers).  2 per CPU is OK, 10 is plenty
QUEUE_GROUP(`mqueue', `P=/var/spool/mqueue-rx, R=2, F=f')dnl

dnl Let's direct all mail to be forwarded to amavisd-new at 127.0.0.1:10024
FEATURE(stickyhost)dnl  Keep envelope addr "u@local.host" when fwd to MAIL_HUB
define(`MAIL_HUB',  `esmtp:[127.0.0.1]')dnl  Forward all local mail to amavisd
define(`SMART_HOST',`esmtp:[127.0.0.1]')dnl  Forward all other mail to amavisd

define(`confDELIVERY_MODE',`q')dnl     Delivery mode: queue only
define(`ESMTP_MAILER_ARGS',`TCP $h 10024')dnl  To tcp port 10024 instead of 25
MODIFY_MAILER_FLAGS(`ESMTP', `+z')dnl  Speak LMTP (this is optional)
define(`SMTP_MAILER_MAXMSGS',`10')dnl  Max no. of msgs in a single connection
define(`confTO_DATAFINAL',`20m')dnl    20 minute timeout for content checking
DAEMON_OPTIONS(`Name=MTA-RX')dnl       Daemon name used in logged messages

dnl Disable local delivery, as all local mail must go to MAIL_HUB
undefine(`ALIAS_FILE')dnl     No aliases file, all local mail goes to MAIL_HUB
define(`confFORWARD_PATH')dnl Empty search path for .forward files

MAILER(smtp)
```

Riportiamo in dettaglio il file **thishost-tx.mc**:

```
dnl To be used for MTA-TX, the second MTA instance
dnl (delivering outgoing and local mail)

dnl Insert here the usual .mc preamble, including OSTYPE and DOMAIN calls.

dnl Specify here also the required outgoing mail processing and
dnl local delivery settings such as mailertables, required mailers,
dnl aliases, local delivery mailer settings, deliverey mode, ...

FEATURE(`no_default_msa')dnl  No need for another msa, MTA-RX already has one
DAEMON_OPTIONS(`Addr=127.0.0.1, Port=10025, Name=MTA-TX')dnl Listen on lo:10025
```

define(`confSMTP_LOGIN_MSG', `$w.tx.$m Sendmail $v/$Z; $b')dnl
define(`confTO_IDENT', `0')dnl Disable IDENT

MAILER(smtp)
MAILER(local)

Elaboriamo i file .mc per ottenere i relativi file .cf:
(modificare se necessario il cammino dove risiedono i file cf/m4/cf.m4)

 # m4 /usr/share/sendmail/cf/m4/cf.m4 thishost-rx.mc >/etc/mail/sendmail-rx.cf
 # m4 /usr/share/sendmail/cf/m4/cf.m4 thishost-tx.mc >/etc/mail/sendmail-tx.cf

Facciamo eseguire i demoni MTA-RX e MTA-TX:

 # /usr/sbin/sendmail -C/etc/mail/sendmail-rx.cf -L sm-mta-rx -bd -qp
 # /usr/sbin/sendmail -C/etc/mail/sendmail-tx.cf -L sm-mta-tx -bd -q30m

Facciamo partire il demone per il MSP client queue se viene utilizzato:
 # /usr/sbin/sendmail -Ac -L sm-msp-queue -q30m

Facciamo partire amavisd-new:
 # amavisd

Verifichiamo se MTA-RX è in ascolto:
 # telnet localhost 25
QUIT

Verifichiamo se MTA-RX é in ascolto su MSA porta 587 (è una recente feature di sendmail)
 # telnet localhost 587
QUIT

Verifichiamo se MTA-TX è in ascolto:
 # telnet localhost 10025
QUIT

Verifichiamo se Amavisd è in ascolto:
 # telnet localhost 10024
QUIT

Per praticità é utile definire alcuni alias nella shell:

alias mailq-rx='mailq -C/etc/mail/thishost-rx.cf'
alias mailq-tx='mailq -C/etc/mail/thishost-tx.cf '
alias sendmail-rx='/usr/sbin/sendmail -C/etc/mail/thishost-rx.cf'
alias sendmail-tx='/usr/sbin/sendmail -C/etc/mail/thishost-tx.cf'

Finito!

NOTA

- Per utilizzare MTA-RX come content-check soltanto per alcuni messaggi e non per tutti, si possono utilizzare le mailertables invece di MAIL_HUB e SMART_HOST, impostando per esempio che alcuni domini in ricezione siano passati direttamente a MTA-TX su 127.0.0.1:10025 (ad esempio via mailer 'esmtp'), ed inviare tutti gli altri messaggi a amavisd su 127.0.0.1:10024. Per poter specificare il numero di porta deve essere definito un nuovo 'mailer', chiamandolo 'amavis', con impostazioni simili a quello già definito 'esmtp', ad eccezione della porta numero 10024.

NOTA SULLE PRESTAZIONI

La gestione della posta comporta un'intensa attività di I/O. Per ottimizzare le prestazioni si possono mettere le due aree di coda delle mail (/var/spool/mqueue e /var/spool/mqueue-rx) e la directory /var/amavis ($TEMPBASE) su tre dischi separati.

6. Amavisd-new

6.1 Installazione

La versione utilizzata nel nostro Istituto è la 20030314.
Vediamo, in generale, come scaricare il software:

1. prelevare il file compresso e scompattarlo con i comandi:

   wget http://www.ijs.si/software/amavisd/amavisd-new-<version>.tar.gz
   gzip -d -c amavisd-new-<version>.tar.gz | tar xvf -
   cd amavisd-new-<version>

2. Controllare nella pagina web http://www.ijs.si/software/amavisd se esistono e devono essere applicate patch al software; in questo caso scaricarle con i comandi:

   wget http://www.ijs.si/software/amavisd/amavisd-new-<version>-1.patch
   patch < amavisd-new-<version>-1.patch

   (o prelevare il file tar con le patch già applicate, se disponibile).

Amavisd-new utilizza diversi moduli Perl per il suo funzionamento. Ecco la lista completa dei moduli da installare prima della configurazione:

Archive::Tar	(Archive-Tar-x.xx)
Archive::Zip	(Archive-Zip-x.xx)
Compress::Zlib	(Compress-Zlib-x.xx)
Convert::TNEF	(Convert-TNEF-x.xx)
Convert::UUlib	(Convert-UUlib-x.xxx)
MIME::Base64	(MIME-Base64-x.xx)
MIME::Parser	(MIME-Tools-x.xxxx)
Mail::Internet	(MailTools-1.58 o successive che abbiano workarounds per bugs di Perl 5.8.0)
Net::Server	(Net-Server-x.xx)
Net::SMTP	(libnet-x.xx)
Digest::MD5	(Digest-MD5-x.xx)
IO::Stringy	(IO-stringy-x.xxx)

Time::HiRes (Time-HiRes-x.xx)
Unix::Syslog (Unix-Syslog-x.xxx)

Sono opzionali i moduli:

DBI con appropriati DBD::* se si vogliono utilizzare SQL lookups
Net::LDAP se si vogliono utilizzare LDAP lookups

6.2 Configurazione

Vediamo adesso i passi da seguire per la configurazione:

1. creare (o scegliere) un gruppo Unix da dedicare alla esecuzione del demone amavisd e possibilmente anche del virus scanner. Questo gruppo **NON** deve essere uno dei gruppi di sistema, e non deve essere condiviso con il mailer.
   E' consigliato usare come nome del gruppo 'amavis' (o anche 'sweep');
2. creare (o scegliere) un account Unix (username e relativo UID) da dedicare alla esecuzione del demone amavisd e possibilmente anche del virus scanner. Questo utente **NON** deve essere uno degli utenti di sistema, e non deve essere condiviso con il mailer (in generale non utilizzare "root", e "nobody" o account usati dal mailer come "smmsp"). E' consigliato usare come nome dell'utente 'amavis' o 'vscan'.

   Scegliere una home directory (es. /var/amavis o /var/lib/amavis) per questo utente.

   Creare la sua home directory, se la procedura di creazione dell'account non l'ha già fatto.

   mkdir /var/amavis

   Controllare o impostare la proprietà e la protezione della directory in modo tale che sia leggibile e scrivibile dall'UID scelto, e non scrivibile da qualsiasi altro utente non privilegiato;

   chown amavis:amavis /var/amavis
   chmod 750 /var/amavis

3. copiare il file amavisd dove desiderate sia residente, ad esempio in /usr/local/sbin, e verificare che sia eseguibile ma non sovrascrivibile da parte di utenti non privilegiati. Questo file è un sorgente Perl, dunque é leggibile con qualunque editore di testo fosse necessario.
   E' pesantemente commentato, nel caso fossero necessarie maggiori informazioni;

   cp amavisd /usr/local/sbin/
   chown root /usr/local/sbin/amavisd
   chmod 755 /usr/local/sbin/amavisd

4. copiare il file amavisd.conf dove desiderate sia residente, ad esempio in /etc/, e verificare che non sia scrivibile da parte di utenti non privilegiati.

   cp amavisd.conf /etc/
   chown root /etc/amavisd.conf
   chmod 644 /etc/amavisd.conf

Potrebbe essere preferibile copiarlo in /etc/amavis/amavisd.conf. Se si utilizza una posizione non di default, allora si deve utilizzare l'opzione –c sulla linea di comando quando si fa partire il demone per specificarne la posizione;

5. creare una directory (es. /var/virusmails) da far utilizzare ad amavisd-new come area di quarantena (se si vogliono mettere in quarantena i messaggi con virus o spam). Impostare la proprietà e la protezione della directory in modo che sia leggibile e scrivibile dall'UID scelto e non scrivibile da qualsiasi altro utente non privilegiato;

   mkdir /var/virusmails
   chown amavis:amavis /var/virusmails
   chmod 750 /var/virusmails

6. modificare il file /etc/amavisd.conf e sistemare le variabili $daemon_group e $daemon_user per il gruppo e lo username scelto in precedenza, sistemare le variabili $MYHOME, $TEMPBASE e $QUARANTINEDIR per le directories appena create, e infine controllare/sistemare le altre variabili, in particolare quelle della 'Section I', compresa $mydomain;
7. installare l'antivirus e il modulo Perl SpamAssassin, e modificare di conseguenza le variabili dell'ultima sezione 'Section VII'; il demone amavisd ha bisogno di diversi altri moduli Perl per funzionare correttamente (si veda il paragrafo 4.1). Se non sono stati ancora installati e si fa partire il demone amavisd sarà visualizzata una lista di moduli mancanti;
8. eseguire il programma 'amavisd', come root o con "su" con l'utente creato in precedenza. Il programma dovrebbe partire e (se root) cambiare il suo GID/UID al valore configurato. E' consigliato farlo eseguire per la prima volta con l'opzione 'debug':

   /usr/local/sbin/amavisd debug

9. dopo aver verificato che tutto funziona correttamente si può utilizzare lo script amavisd_init.sh per invocare il processo nei momenti di startup/shutdown del sistema.

6.3 Test

Per testare il corretto funzionamento di tutto il sistema, nella directory test-message sono presenti 4 messaggi di prova per le diverse situazioni.
Si possono utilizzare eseguendo i comandi:

 # sendmail -i your-address@example.com <sample-nonspam.txt
 # sendmail -i your-address@example.com <sample-virus-simple.txt
 # sendmail -i your-address@example.com <sample-virus-nested.txt
 # sendmail -i your-address@example.com <sample-spam.txt

7. Statistiche

Amavisd-new conserva tutti i messaggi filtrati come spam o come virus nella directory /var/virusmails.
E' interessante effettuare delle statistiche su questi file; a tale scopo può essere utilizzato il software open-source **Mail::Graph** (http://search.cpan.org/search?dist=Mail-Graph vers. 0.14).
E' un modulo Perl che può essere installato con l'interfaccia grafica Webmin similmente a come già descritto per SpamAssassin nel paragrafo 3.1.

Analizziamo come esempio le statistiche create con questo software utilizzando i file filtrati dal mail server del nostro dominio @ilc.cnr.it nei mesi di Settembre e Ottobre 2003.

7.1 Spam

Iniziamo con il grafico giornaliero dei messaggi spam ricevuti (Figura 3):

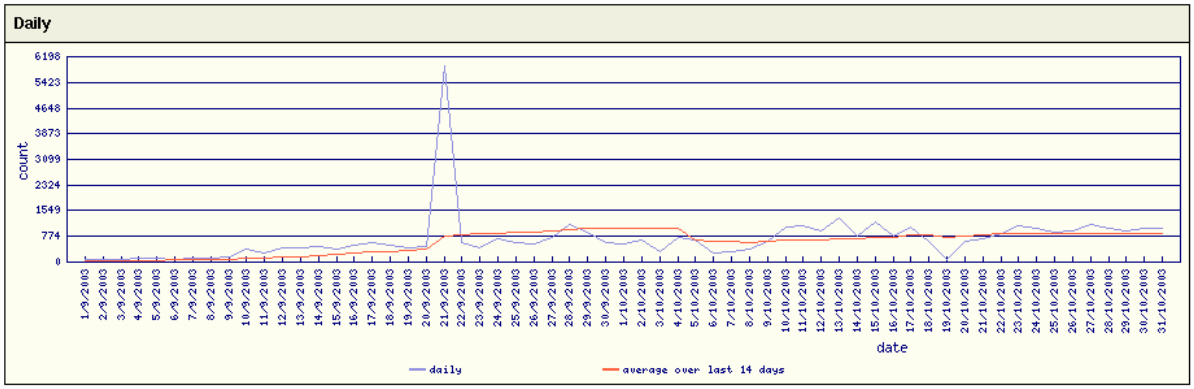


Figura 3

Il grafico mostra un evidente picco corrispondente al giorno 21 Settembre 2003, dove sono stati ricevuti 6198 messaggi spam, ma è più interessante notare la linea rossa che traccia il valore medio, da cui risulta che il fenomeno é in aumento.
Questo aumento è ben evidenziato nella successiva Figura 4, dove il calcolo è stato effettuato a partire dal 22 Settembre 2003 fino al 31 Ottobre 2003, questo per evitare il giorno 21 Settembre che disturba con la sua eccezionalità il calcolo del valore medio.

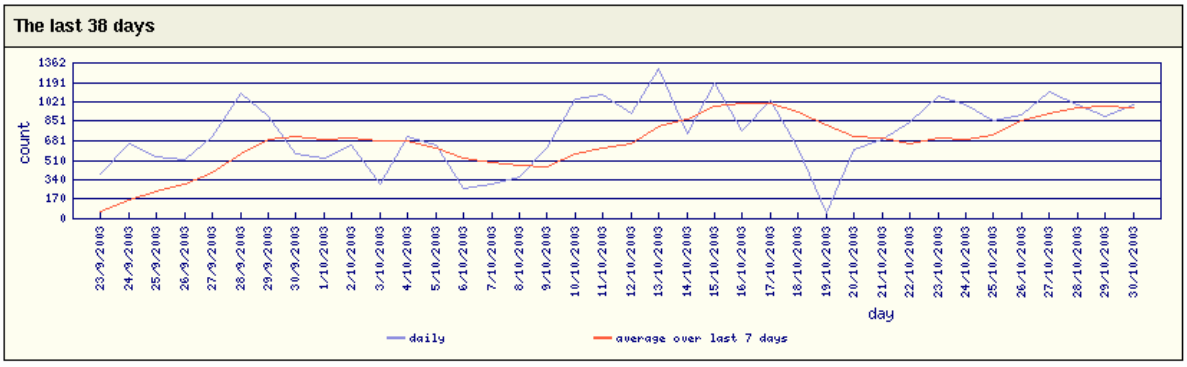


Figura 4

Nelle successive figure vediamo altri utili grafici generati dal programma. Nella Figura 5 sono confrontati i due totali mensili con una evidente crescita nel mese di Ottobre; nella Figura 6 vediamo la frequenza dei messaggi spam nell'arco dei sette giorni che compongono la settimana, ed il picco registrato nel giorno di domenica conferma l'analisi fatta da alcune riviste del settore informatico (si veda l'articolo "Rende fare spamming?" di Azzura Pici, pubblicato su ZeusNews del 16/4/2003 [http://www.zeusnews.it/index.php3?ar=stampa&cod=2040]) secondo la quale esistono degli "spammers" occasionali che si attivano solo nei giorni festivi (http://www.computerworld.com/softwaretopics/software/groupware/story/0,10801,75736,00.html).

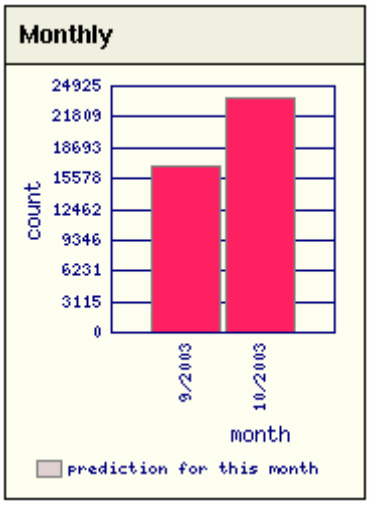


Figura 5

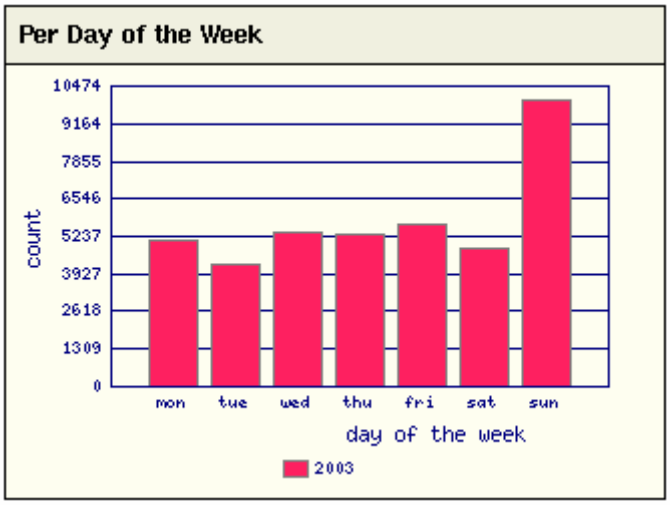


Figura 6

Completiamo i grafici commentando la Figura 7 che mostra la distribuzione del punteggio assegnato da SpamAssassin (per dettagli si veda http://www.spamassassin.org/tests.html) ai messaggi filtrati.
Il nostro mail server filtra i messaggi con punteggio superiore a 6,3 e proprio nella fascia tra 5 e 10 vengono registrati la maggioranza dei messaggi filtrati.

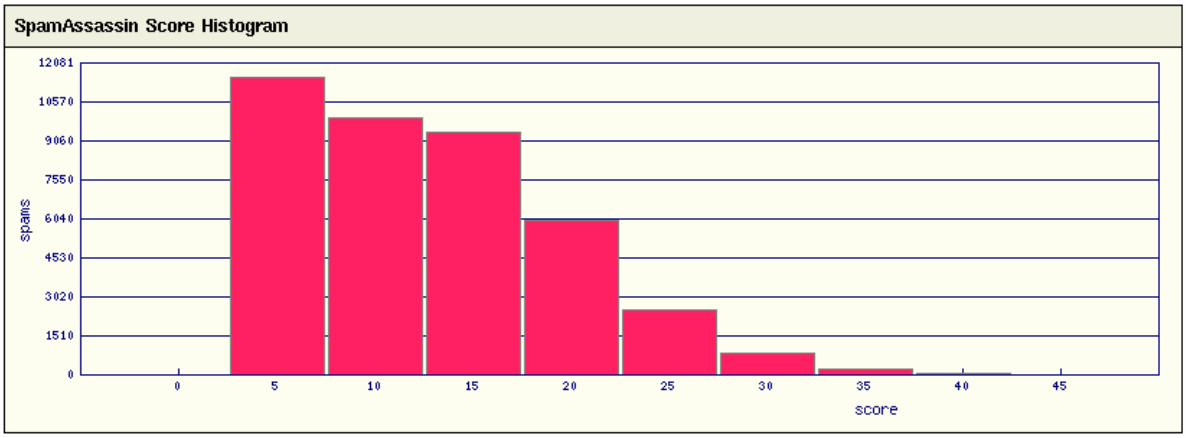


Figura 7

7.2 Virus

Vediamo adesso i grafici ottenuti in base al numero dei file con virus allegati a messaggi di posta elettronica filtrati dal mail server nello stesso intervallo di tempo Settembre-Ottobre 2003 considerato in precedenza per i messaggi spam.
Dal grafico di Figura 8 si nota immediatamente che il numero di messaggi filtrati è notevolmente più basso (siamo intorno ai 750 messaggi con virus contro i circa 38000 con spam), e che il fenomeno ha un andamento molto variabile: giorni con picchi più alti e giorni con valori molto bassi.

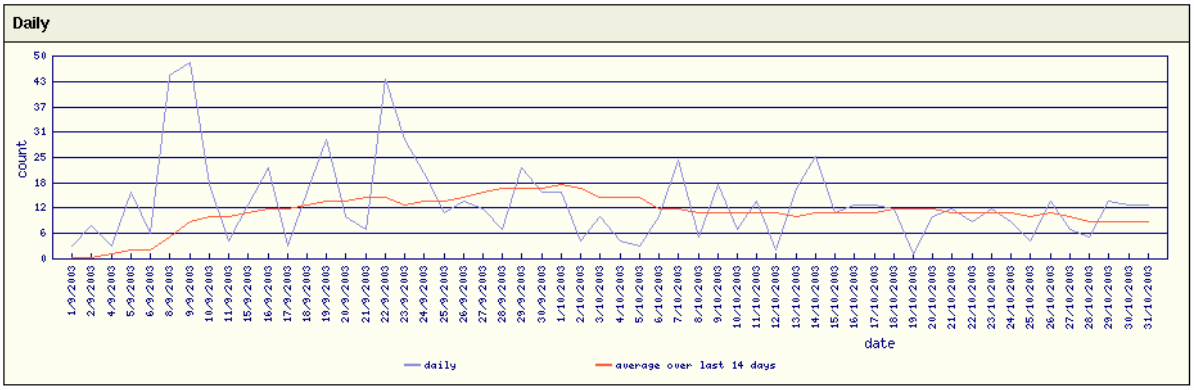


Figura 8

In Figura 9 il grafico mensile mostra una leggera diminuzione nel mese di Ottobre.

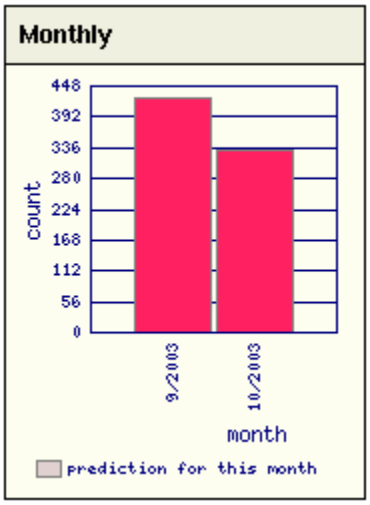


Figura 9

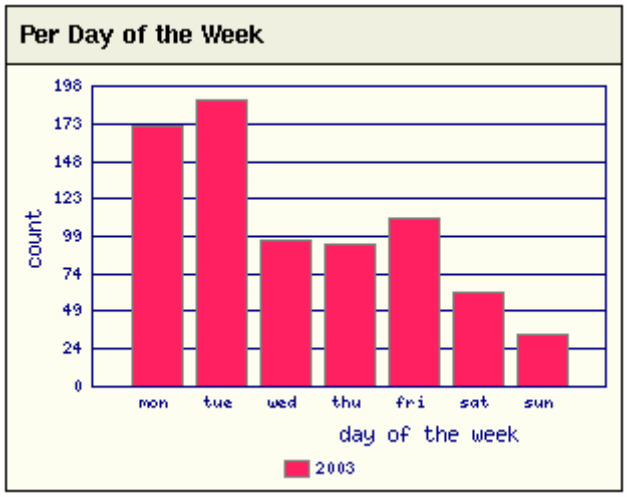


Figura 10

Il dato più interessante è quello presentato in Figura 10, il grafico settimanale, dove si può notare che i messaggi con virus si diffondono prevalentemente nei giorni feriali, mentre nei giorni festivi sono quasi assenti.
Una conferma di questo ci viene dal grafico di Figura 11:

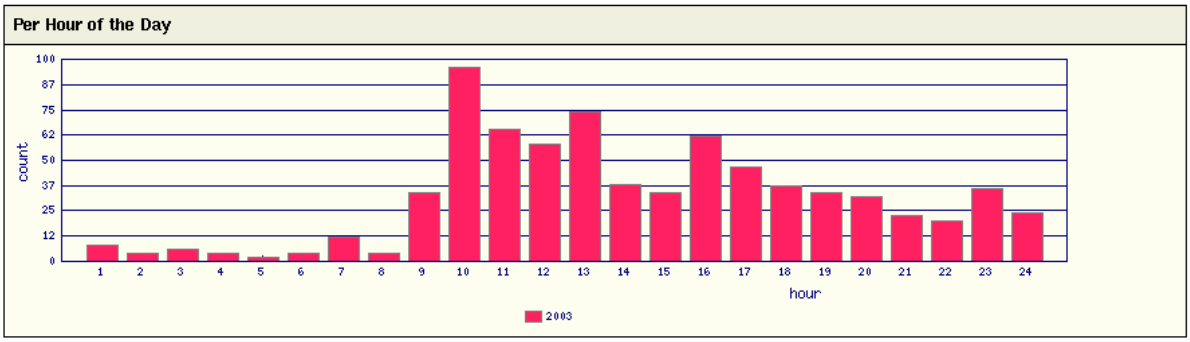


Figura 11

Qui si nota che le ore del giorno durante le quali arrivano sul mail server i messaggi con virus sono quelle d'ufficio.
Evidentemente i messaggi con virus che il nostro mail server riceve provengono per la maggior parte da altri server italiani; questo dato è confermato graficamente dalla Figura 12:

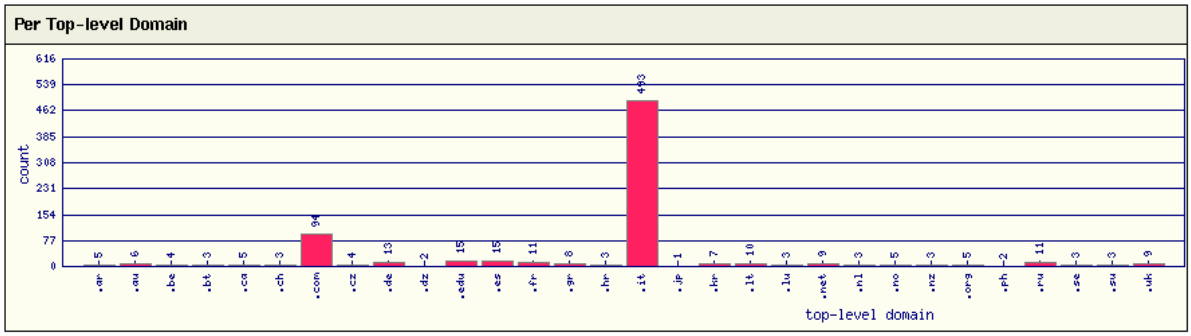


Figura 12

I messaggi provenienti dagli altri domini sono numericamente bassi rispetto a quelli provenienti dal dominio .it.

8. Conclusioni

Il fenomeno spam è preoccupante perché, non solo è in aumento il volume di traffico causato da spam, ma anche perché le tecniche utilizzate dagli "spammers" diventano sempre più sofisticate (così come avviene per i virus) e quindi diventa sempre più difficile bloccare i messaggi spam. Questo significa che l'utilizzo di un filtro, ormai necessario, non impedirà ad alcuni messaggi spam di arrivare al destinatario, ed inoltre alcuni messaggi "non spam" verranno bloccati.
La gestione di questi eventi, indicati nella letteratura tecnica come "falsi negativi" (messaggi spam arrivati al destinatario) e come "falsi positivi" (messaggi non spam bloccati), impongono un particolare studio del valore numerico del punteggio di soglia oltre il quale l'algoritmo antispam filtra il messaggio e suggeriscono l'adozione di interfacce web dette Webmail che consentono all'utente finale di controllare i messaggi filtrati ed eventualmente recuperare i falsi positivi.
Questa problematica del punteggio di soglia non è presente nell'antivirus, dove viene applicato un preciso criterio di confronto per l'individuazione del virus.

Riferimenti

AMaViS - A Mail Virus Scanner. http://www.amavis.org
Amavisd-new. http://www.ijs.si/software/amavisd
CPAN (Comprehensive Perl Archive Network). http://www.cpan.org
Creating a Spamfilter Relay Server, Scott L. Henderson, 2002/2003. http://www.geocities.com/scottlhenderson/spamfilter.html
Mail::Graph. http://search.cpan.org/search?dist=Mail-Graph
McAfee VirusScan. http://www.mcafee.com
Razor2. http://razor.sourceforge.net
Rende fare spamming?, Azzurra Pici, ZeusNews del 16/4/2003 http://www.zeusnews.it/index.php3?ar=stampa&cod=2040
SpamAssassin. www.spamassassin.org
The Other Side, Melissa Solomon, ComputerWorld del 11/11/2002 http://www.computerworld.com/softwaretopics/software/groupware/story/0,10801,75736,00.html
Webmin. http://www.webmin.com